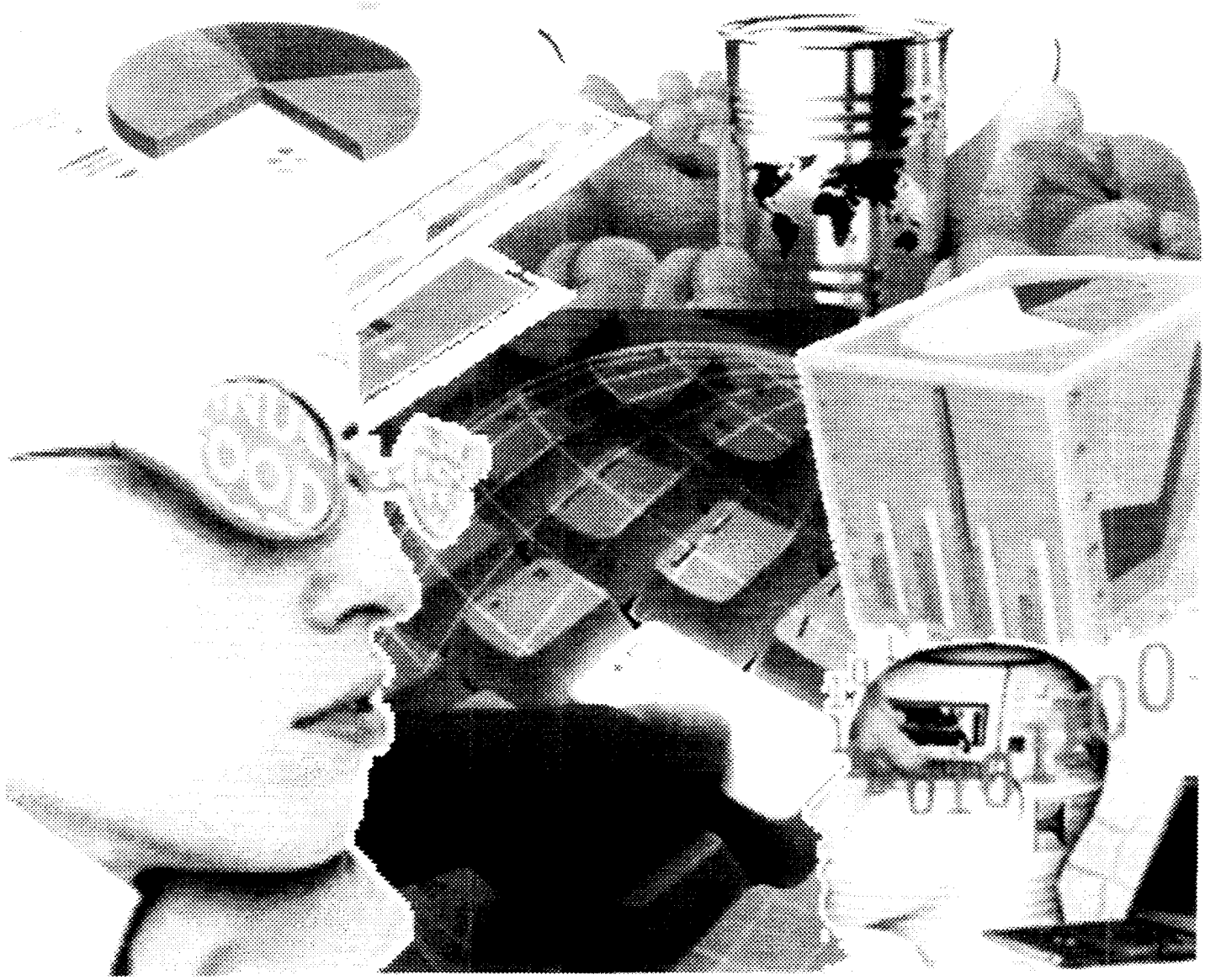
PN-ADC-824

Food Export Business Intelligence

Export Market Data Analysis

الذكاء التصديرى للأغذية

تحليل بيانات أسواق التصدير



Agriculture-Led Export Businesses

Supporting Egypt's Processed Foods Export Industry

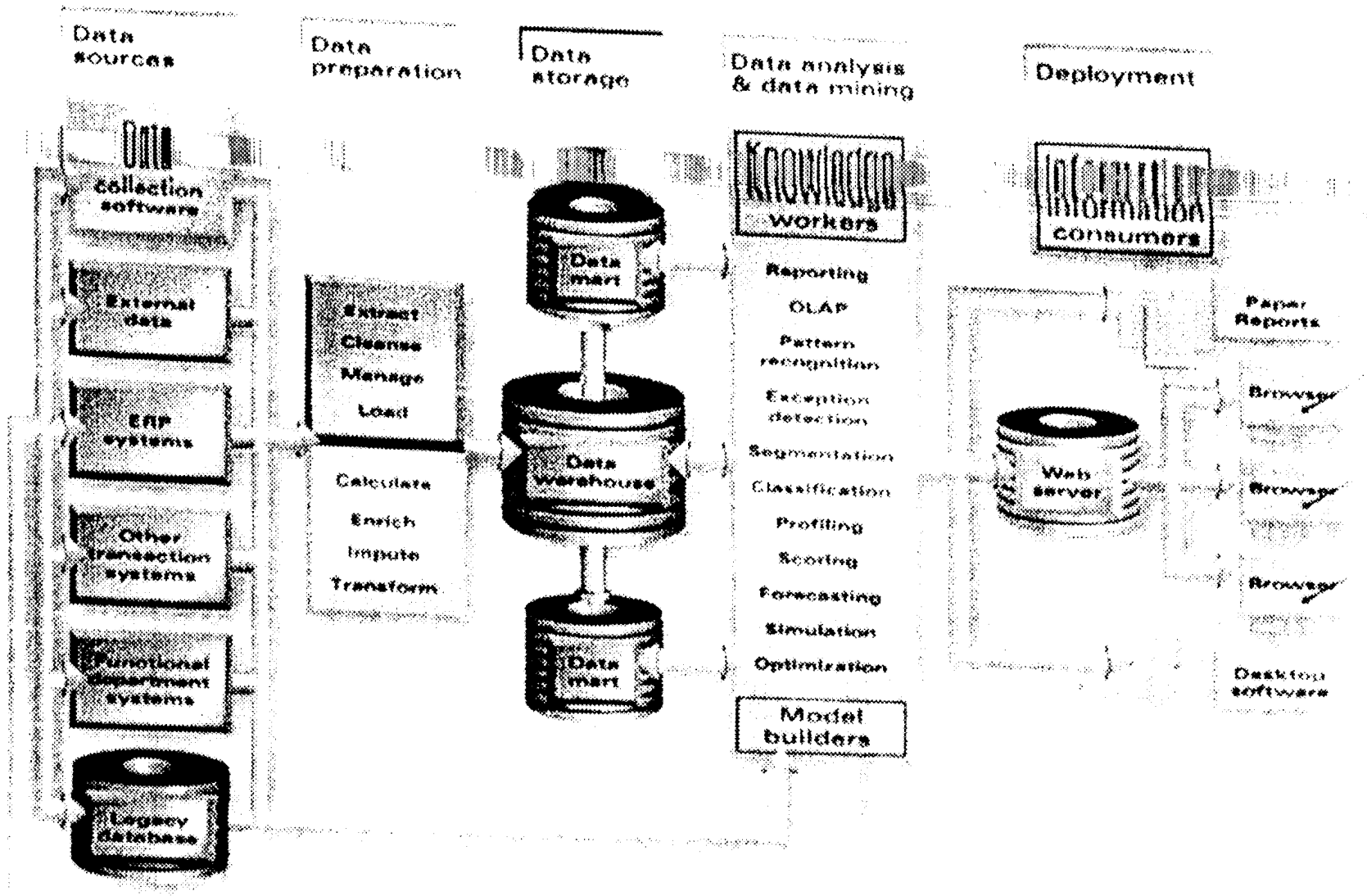
Project funded by USAID

Implemented by Abt Associat

In cooperation with

THE AMERICAN UNIVERSITY IN CAIRO

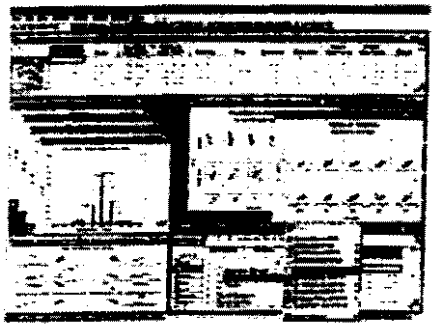
Business Intelligence Model



Analytical solutions

STATISTICA

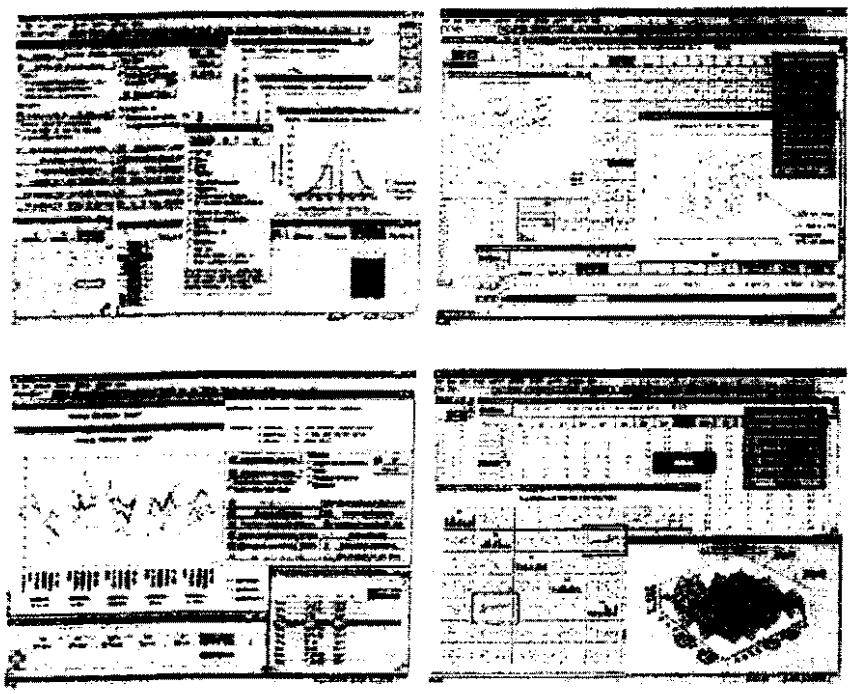
Basics
Data Mining
Multivariate
Control



DESCRIPTIVE STATISTICS, BREAKDOWNS, AND EXPLORATORY DATA ANALYSIS. *STATISTICA Base* offers a wide selection of methods for exploratory analyses:

Descriptive Statistics and Graphs. The program will compute practically all common, general-purpose descriptive statistics including medians, modes, quartiles, user-specified percentiles, average and standard deviations, quartile ranges, confidence limits for the mean, skewness and kurtosis (with their respective standard errors), harmonic means, geometric means, as well as many specialized descriptive statistics and diagnostics, either for all cases or broken down by one or more categorical (grouping) variables. As with all modules of *STATISTICA*, a wide variety of graphs will aid exploratory analyses, e.g., various types of box-and-whisker plots, histograms, bivariate distribution (3D or categorized) histograms, 2D and 3D scatterplots with marked subsets, normal, half-normal, detrended probability plots, *Q-Q* plots, *P-P* plots, etc. A selection of tests is available for fitting the normal distribution to the data (via the Kolmogorov-Smirnov, Lilliefors, and Shapiro-Wilks' tests; facilities for fitting a wide variety of other distributions are also available; see also ***STATISTICA Process Analysis***; and the section on fitting in the ***Graphics*** section).

By-Group Analyses (Breakdowns). Practically all descriptive statistics as well as summary graphs can be computed for data that are categorized (broken down) by one or more grouping variables. For example, with just a few mouse clicks the user can break down the data by *Gender* and *Age* and review categorized histograms, box-and-whisker plots, normal probability plots, scatterplots, etc. If more than two categorical variables are chosen, *cascades* of the respective graphs can be automatically produced. Options to categorize by continuous variables are provided, e.g., you can request that a variable be split into a requested number of intervals, or use the on-line recode facility to custom-define the way in which the variable will be recoded (categorization options of practically unlimited complexity can be specified at any point and they can reference relations involving all variables in the dataset). In addition, a specialized hierarchical breakdown procedure is provided that allows the user to categorize the data by up to six categorical variables, and compute a variety of categorized graphs, descriptive statistics, and correlation matrices for *subgroups* (the user can interactively request to ignore some factors in the complete breakdown table, and examine statistics for any marginal tables). Numerous formatting and labeling options allow the user to produce publication-quality tables and reports with long labels and descriptions of variables. Note that extremely large analysis designs can be specified in the breakdown procedure (e.g., 100,000 groups for a single categorization variable), and results include all relevant ANOVA statistics (including the complete ANOVA table, tests of assumptions such as the Levene and Brown-Forsythe tests for homogeneity of variance, a selection of seven *post-hoc* tests, etc.). As in all other modules of *STATISTICA*, extended precision calculations (the "quadruple" precision, where applicable) are used to provide an unmatched level of accuracy (see the section on ***Precision***). Because of the interactive nature of the program, exploration of data is very easy. For example, exploratory graphs can be produced directly from all results Spreadsheets by pointing with the mouse to specific cells or ranges of cells. *Cascades* of even complex (e.g., multiple categorized) graphs can be produced with a single-click of the mouse and reviewed in a slide-show manner. In addition to numerous predefined statistical graphs, countless graphical visualizations of raw data, summary statistics, relations between statistics, as well as all breakdowns and categorizations can be custom-defined by the user via straightforward *point-and-click* facilities designed to reduce the necessary number of mouse clicks. All exploratory graphical techniques (described in the section on ***Graphics***) are integrated with statistics to facilitate graphical data analyses (e.g., via interactive outlier removal, subset selections, smoothing, function fitting, extensive brushing options allowing the user to easily identify and/or extract the selected data, etc.). See also the section on ***Block Statistics***, below.



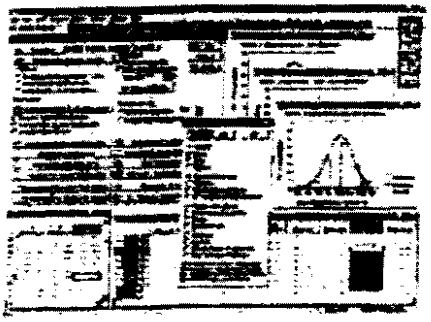
CORRELATIONS. A comprehensive set of options allows for the exploration of correlations and partial correlations between variables. First, practically all common measures of association can be computed, including Pearson *r*, Spearman rank order *R*, Kendall *tau (b, c)*, *Gamma*, tetrachoric *r*, *Phi*, Cramer *V*, contingency coefficient *C*, Sommer's *D*, uncertainty coefficients, part and partial correlations, autocorrelations, various distance measures, etc. (nonlinear regressions, regressions for censored data and other specialized measures of correlations are available in ***Nonlinear Estimation, Survival Analysis***, and other modules offered in ***STATISTICA Advanced Linear/Non-Linear Models***). Correlation matrices can be computed using casewise (listwise) or pairwise deletion of missing data, or mean substitution. As in all other modules of *STATISTICA*, extended precision calculations (the "quadruple" precision, where applicable) are used to yield an unmatched level of accuracy (see the section on ***Precision***). Like all other results in *STATISTICA*, correlation matrices are displayed in Spreadsheets offering various formatting options (see below) and extensive facilities to visualize numerical results; the user can "point to" a particular correlation in the Spreadsheet and choose to display a variety of "graphical summaries" of the coefficient (e.g., scatterplots with confidence intervals, various 3D bivariate distribution histograms, probability plots, etc.).

Brushing and outlier detection. The extensive *brushing* facilities in the scatterplots allow the user to select/deselect individual points in the plot and assess their effect on the regression line (or other fitted function lines).

Display formats of numbers. A variety of global display formats for correlations are supported; significant correlation coefficients can be automatically highlighted, each cell of the Spreadsheet can be expanded to display *n* and *p*, or detailed results may be requested that include all descriptive statistics (pairwise means and standard deviations, B weights, intercepts, etc.). Like all other numerical results, correlation matrices are displayed in Spreadsheets offering the zoom option and interactively-controlled display formats (e.g., from +.4 to +.4131089276410193); thus, large matrices can be compressed (via either the zoom or format-width control adjustable by dragging) to facilitate the visual search for coefficients which exceed a user-specified magnitude or significance level (e.g., the respective cells can be marked red in the Spreadsheet).

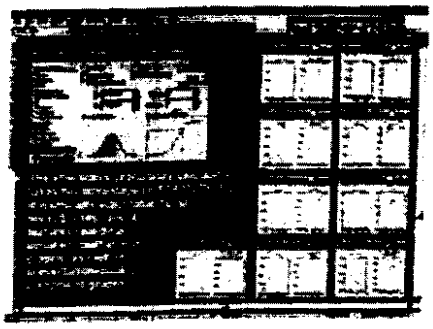
Scatterplot, scatterplot matrices, by-group analyses. As in all output selection dialogs, numerous global graphics options are available to further study patterns of relationships between variables, e.g., 2D and 3D scatterplots (with or without case labels) designed to identify patterns of relations across subsets of cases or series of variables. Correlation matrices can be computed as categorized by grouping variables and visualized via categorized scatterplots. Also "breakdowns of correlation matrices" can be generated (one matrix per subset of data), displayed in queues of Spreadsheets, and saved as stacked correlation matrices (which can later be used as input into the ***Structural Equations Modeling and Path Analysis [SEPATH]*** module offered in ***STATISTICA Advanced Linear/Non-Linear Models***). An entire correlation matrix can be summarized in a single graph via the *Matrix scatterplot* option (of practically unlimited density); large scatterplot matrices can then be reviewed interactively by "zooming in" on selected portions of the graph (or scrolling large graphs in the zoom mode) [see the illustration]. Also, categorized scatterplot matrix plots can be generated (one matrix plot for each subset of data). Alternatively, a *multiple-subset* scatterplot matrix plot can be created where specific subsets of data (e.g., defined by levels of a grouping variable or selection conditions of any complexity) are marked with distinctive point markers. Various other graphical methods can be used to

visualize matrices of correlations in search of global patterns (e.g., contour plots, non-smoothed surfaces, icons, etc.). All of these operations require only a few mouse clicks and various shortcuts are provided to simplify selections of analyses; any number of Spreadsheets and graphs can be displayed simultaneously on the screen, making interactive exploratory analyses and comparisons very easy.

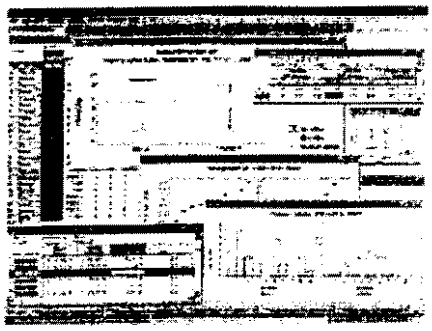


BASIC STATISTICS FROM RESULTS SPREADSHEETS (TABLES). *STATISTICA* is a single integrated analysis system that presents all numerical results in spreadsheet tables that are suitable (without any further modification) for input into subsequent analyses. Thus, basic statistics (or any other statistical analysis) can be computed for results tables from previous analyses; for example, you could very quickly compute a table of means for 2000 variables, and next use this table as an input data file to further analyze the distribution of those means across the variables. Thus, basic statistics are available at any time during your analyses, and can be applied to any results spreadsheet.

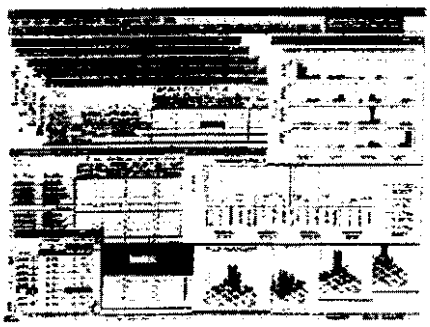
Block Statistics. In addition to the detailed descriptive statistics that can be computed for every spreadsheet, you can also highlight blocks of numbers in any spreadsheet, and produce basic descriptive statistics or graphs for the respective subset of numbers only. For example, suppose you computed a results spreadsheet with measures of central tendency for 2000 variables (e.g., with Means, Modes, and Medians, Geometric Means, and Harmonic Means); you could highlight a block of, for example, 200 variables and the Means and Medians, and then in a single operation produce a multiple line graph of those two measures across the subset of 200 variables. Statistical analysis by blocks can be performed by *row* or by *column*; for example, you could also compute a multiple line graph for a subset of variables across the different measures of central tendency. To summarize, the block statistics facilities allow you to produce statistics and statistical graphs from values in arbitrarily selected (highlighted) blocks of values in the current data spreadsheet or output Spreadsheet.



INTERACTIVE PROBABILITY CALCULATOR. A flexible, interactive *Probability Calculator* is accessible from all toolbars. It features a wide selection of distributions (including *Beta, Cauchy, Chi-square, Exponential, Extreme value, F, Gamma, Laplace, Lognormal, Logistic, Pareto, Rayleigh, t (Student), Weibull,* and *Z (Normal)*); interactively (in-place) updated graphs built into the dialog (a plot of the density and distribution functions) allow the user to visually explore distributions taking advantage of the flexible *STATISTICA Smart MicroScrolls* which allow the user to advance either the last significant digit (press the LEFT-mouse-button) or next to the last significant digit (press the RIGHT-mouse-button). Facilities are provided for generating customizable, compound graphs of distributions with requested cutoff areas. Thus, this calculator allows you to interactively explore the distributions (e.g., the respective probabilities depending on shape parameters).

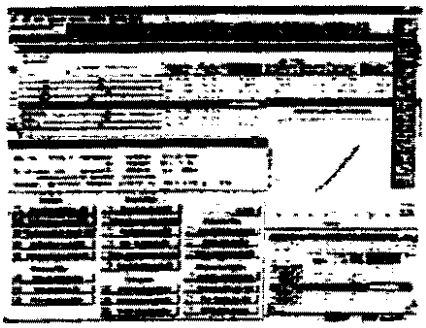


t-TESTS and Other Tests of Group Differences. *T*-tests for dependent and independent samples, as well as single samples (testing means against user-specified constants) can be computed, multivariate Hotelling's T^2 tests are also available (see also ***ANOVA/MANOVA***, and ***GLM (General Linear Models)*** offered in *STATISTICA Advanced Linear/Non-Linear Models*. Flexible options are provided to allow comparisons between variables (e.g., treating the data in each column of the input spreadsheet as a separate sample) and coded groups (e.g., if the data includes a categorical variable such as Gender to identify group membership for each case). As with all procedures, extensive diagnostics and graphics options are available from the results menus. For example, for the *t*-test for independent samples, options are provided to compute *t*-tests with separate variance estimates, Levene and Brown-Forsythe tests for homogeneity of variance, various box-and-whisker plots, categorized histograms and probability plots, categorized scatterplots, etc. Other (more specialized) tests of group differences are part of many modules (e.g., ***Nonparametrics*** (below), ***Survival Analysis*** (available in *STATISTICA Advanced Linear/Non-Linear Models*), ***Reliability/Item Analysis*** (available in *STATISTICA Multivariate Exploratory Techniques*).



FREQUENCY TABLES, CROSSTABULATION TABLES, STUB-AND-BANNER TABLES, MULTIPLE RESPONSE ANALYSIS, AND TABLES. Extensive facilities are provided to tabulate continuous, categorical, and multiple response variables, or multiple dichotomies. A wide variety of options are offered to control the layout and format of the tables. For example, for tables involving multiple response variables or multiple dichotomies, marginal counts and percentages can be based on the total number of respondents or responses, multiple response variables can be processed in pairs, and various options are available for counting (or ignoring) missing data. Frequency tables can also be computed based on user-defined logical selection conditions (of any complexity, referencing any relationships between variables in the dataset) that assign cases to categories in the table. All tables can be extensively customized to produce final (publication-quality) reports. For example, unique "multi-way summary" tables can be produced with breakdown-style, hierarchical arrangements of factors, crosstabulation tables may report row, column, and total percentages in each cell, long value labels can be used to describe the categories in the table, frequencies greater than a user-defined cutoff can be highlighted in the table, etc. The program can display cumulative and relative frequencies, Logit- and Probit-transformed frequencies, normal expected frequencies (and the Kolmogorov-Smirnov, Lilliefors, and Shapiro-Wilks' tests), expected and residual frequencies in crosstabulations, etc. Available statistical tests for crosstabulation tables include the Pearson, Maximum-Likelihood and Yates-corrected *Chi-squares*; McNemar's *Chi-square*, the Fisher exact test (one- and two-tailed), *Phi*, and the tetrachoric *r*; additional available statistics include Kendall's *tau (a, b)*, Gamma, Spearman *r*, Sommer's *D*, uncertainty coefficients, etc.

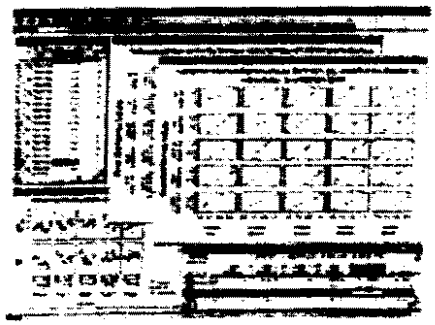
Graphs. Graphical options include simple, categorized (multiple), and 3D histograms, cross-section histograms (for any "slices" of the one-, two-, or multi-way tables), and many other graphs including a unique "interaction plot of frequencies" that summarizes the frequencies for complex crosstabulation tables (similar to plots of means in ANOVA). *Cascades* of even complex (e.g., multiple categorized, or interaction) graphs can be interactively reviewed. See also the section on ***Block Statistics***, above, and sections on ***Log-linear Analysis*** (available in *STATISTICA Advanced Linear/Non-Linear Models*) and ***Correspondence Analysis*** (available in *STATISTICA Multivariate Exploratory Techniques*).



MULTIPLE REGRESSION METHODS. The *Multiple Regression* module is a comprehensive implementation of linear regression techniques, including simple, multiple, stepwise (forward, backward, or in blocks), hierarchical, nonlinear (including polynomial, exponential, log, etc.), Ridge regression, with or without intercept (regression through the origin), and weighted least squares models; additional advanced methods are provided in the General Regression Models (GRM) module (e.g., best subset regression, multivariate stepwise regression for multiple dependent variables, for models that may include categorical factor effects; statistical summaries for validation and prediction samples, custom hypotheses, etc.). The Multiple Regression module will calculate a comprehensive set of statistics and extended diagnostics including the complete regression table (with standard errors for *B*, *Beta* and intercept, R-square and adjusted R-square for intercept and non-intercept models, and ANOVA table for the regression), part and partial correlation matrices, correlations and covariances for regression weights, the sweep matrix (matrix inverse), the Durbin-Watson *d* statistic, Mahalanobis and Cook's distances, deleted residuals, confidence intervals for predicted values, and many others.

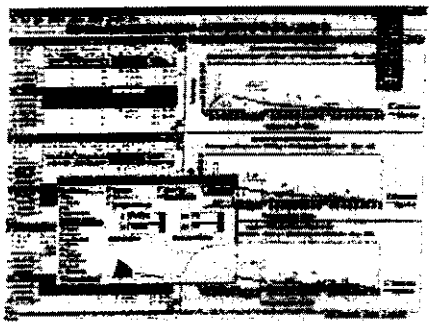
Predicted and residual values. The extensive residual and outlier analysis features a large selection of plots, including a variety of scatterplots, histograms, normal and half-normal probability plots, detrended plots, partial correlation plots, different casewise residual and outlier plots and diagrams, and others. The scores for individual cases can be visualized via exploratory icon plots and other multidimensional graphs integrated directly with the results Spreadsheets. Residual and predicted scores can be appended to the current data file. A forecasting routine allows the user to perform *what-if* analyses, and to interactively compute predicted scores based on user-defined values of predictors.

By-group analysis; related procedures. Extremely large regression designs can be analyzed. An option is also included to perform multiple regression analyses broken down by one or more categorical variable (multiple regression analysis by group); additional add-on procedures include a regression engine that supports models with thousands of variables, a *Two-stage Least Squares* regression, as well as *Box-Cox* and *Box-Tidwell* transformations with graphs. An add-on package, *STATISTICA Advanced Linear/Non-Linear Models*, also includes general nonlinear estimation modules (*Nonlinear Estimation*, *Generalized Linear Models (GLZ)*, *Generalized Additive Models (GAM)*, *Partial Least Squares models (PLS)*) that can estimate practically any user-defined nonlinear model, including Logit, Probit, and others. The add-on also includes *SEPATH*, the general *Structural Equation Modeling and Path Analysis* module, which allows the user to analyze extremely large correlations, covariances, and moment matrices (for intercept models).



NONPARAMETRIC STATISTICS. The Nonparametric Statistics module features a comprehensive selection of inferential and descriptive statistics including all common tests and some special application procedures. Available statistical procedures include the Wald-Wolfowitz runs test, Mann-Whitney *U* test (with exact probabilities [instead of the *Z* approximations] for small samples), Kolmogorov-Smirnov tests, Wilcoxon matched pairs test, Kruskal-Wallis ANOVA by ranks, Median test, Sign test, Friedman ANOVA by ranks, Cochran *Q* test, McNemar test, Kendall coefficient of concordance, Kendall *tau (b, c)*, Spearman rank order *R*, Fisher's exact test, *Chi-square* tests, *V*-square statistic, *Phi*, *Gamma*, Sommer's *d*, contingency coefficients, and others. (Specialized nonparametric tests and statistics are also part of many add-on modules, e.g., *Survival Analysis*, *STATISTICA Process Analysis*, and others.) All (rank order) tests can handle tied ranks and apply corrections for small *n* or tied ranks. The program can handle extremely large analysis designs. As in all other modules of *STATISTICA*, all tests are integrated with graphs (that include various scatterplots, specialized box-and-whisker plots, line plots, histograms and many other 2D and 3D displays).

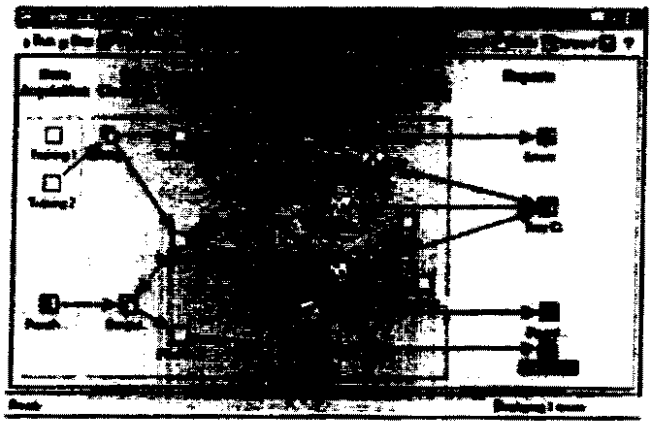
ANOVA/MANOVA. The ANOVA/MANOVA module includes a subset of the functionality of the ***General Linear Models*** module (part of the Advanced Linear/Non-Linear Models add-on), and can perform univariate and multivariate analysis of variance of factorial designs with or without one repeated measures variable. For more complicated linear models with categorical and continuous predictor variables, random effects, and multiple repeated measures factors you need the *General Linear Models* module (stepwise and best-subset options are available in the ***General Regression Models*** module). In the ANOVA/MANOVA module, you can specify all designs in the most straightforward, functional terms of actual variables and levels (not in technical terms, e.g., by specifying matrices of dummy codes), and even less-experienced ANOVA users can analyze very complex designs with *STATISTICA*. Like the *General Linear Models* module, ANOVA/MANOVA provides three alternative user interfaces for specifying designs: (1) A Design Wizard, that will take you step-by-step through the process of specifying a design, (2) a simple dialog-based user-interface that will allow you to specify designs by selecting variables, codes, levels, and any design options from well-organized dialogs, and (3) a Syntax Editor for specifying designs and design options using keywords and a common design syntax. Computational methods. The program will use, by default, the sigma restricted parameterization for factorial designs, and apply the effective hypothesis approach (see Hocking, 19810) when the design is unbalanced or incomplete. Type I, II, III, and IV hypotheses can also be computed, as can Type V and Type VI hypotheses that will perform tests consistent with the typical analyses of fractional factorial designs in industrial and quality-improvement applications (see also the description of the Experimental Design module). Results statistics. The ANOVA/MANOVA module is not limited in any of its computational routines for reporting results, so the full suite of detailed analytic tools available in the *General Linear Models* module is also available here (please see the detailed description of the *General Linear Models* module for details); results include summary ANOVA tables, univariate and multivariate results for repeated measures factors with more than 2 levels, the Greenhouse-Geisser and Huynh-Feldt adjustments, plots of interactions, detailed descriptive statistics, detailed residual statistics, planned and post-hoc comparisons, testing of custom hypotheses and custom error terms, detailed diagnostic statistics and plots (e.g., histogram of within-cell residuals, homogeneity of variance tests, plots of means versus standard deviations, etc.).



DISTRIBUTION FITTING. The *Distribution Fitting* options allow the user to compare the distribution of a variable with a wide variety of theoretical distributions. You may fit to the data the *Normal, Rectangular, Exponential, Gamma, Lognormal, Chi-square, Weibull, Gompertz, Binomial, Poisson, Geometric,* or *Bernoulli* distribution. The fit can be evaluated via the Chi-square test or the Kolmogorov-Smirnov one-sample test (the fitting parameters can be controlled); the Lilliefors and Shapiro-Wilks' tests are also supported (see above). In addition, the fit of a particular hypothesized distribution to the empirical distribution can be evaluated in customized histograms (standard or cumulative) with overlaid selected functions; line and bar graphs of expected and observed frequencies, discrepancies and other results can be produced from the output Spreadsheets. Other distribution fitting options are available in ***STATISTICA Process Analysis***, where the user can compute maximum-likelihood parameter estimates for the *Beta, Exponential, Extreme Value (Type I, Gumbel), Gamma, Log-Normal, Rayleigh,* and *Weibull* distributions. Also included in that module are options for automatically selecting and fitting the best distribution for the data, as well as options for general distribution fitting by moments (via Johnson and Pearson curves). User-defined 2- and 3-dimensional functions can also be plotted and overlaid on the graphs. The functions may reference a wide variety of distributions such as the *Beta, Binomial, Cauchy, Chi-square, Exponential, Extreme value, F, Gamma, Geometric, Laplace, Logistic, Normal, Log-Normal, Pareto, Poisson, Rayleigh, t (Student),* or *Weibull* distribution, as well as their integrals and inverses. Additional facilities to fit predefined or user-defined functions of practically unlimited complexity to the data are available in ***Nonlinear Estimation*** (available in *STATISTICA Advanced Linear/Non-Linear Models*).

STATISTICA
Data Miner

STATISTICA Data Miner is a truly unique application in terms of its sheer comprehensiveness, power, and technology, and flexibility of the available user interfaces:

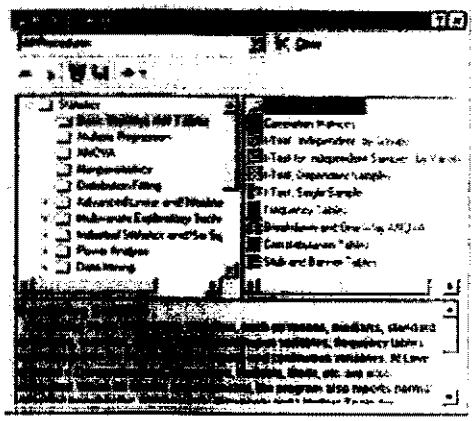


- Choose from the largest selection of algorithms on the market (based on the *STATISTICA* technology) for classification, prediction, clustering, and modeling;
- Access and process huge data sets in remote databases in-place; off-load time-consuming database queries to the server;
- Write predicted values, classifications, classification probabilities, etc., computed from trained models directly to an external database; score very large databases using one or more deployed models;
- Access huge data files on your local (desktop) Windows computer; as specialized queries into custom data warehouses are sometimes expensive (requiring the services of designated consultants), it can be more cost effective to download even huge databases to your local machine; such data files can then be processed with unmatched speed by *STATISTICA Data Miner* routines;
- Data mining project templates can be selected from menus; with only a few clicks of the mouse, you can apply even advanced methods such as meta-learning techniques (voting, bagging, etc) to your specific analysis problems;
- Integrate diverse methods and technologies into the data mining project, from quality control charting and process capability analysis, Weibull analysis, power analysis, or linear and nonlinear models, to advanced automated searches for neural network architectures; all *STATISTICA* procedures can be selected as nodes for data mining projects, and no programming or custom-development work is required to use these procedures;
- Graphical/visual data mining: All of *STATISTICA's* unique and unmatched graphical capabilities are available for data mining; choose from hundreds of graph types to visualize data after cleaning, slicing, or drilling down;
- Intuitive user interface and full integration with *STATISTICA's* award winning solutions: you will be up-and-running in minutes;
- Complete integration into StatSoft's desktop (*STATISTICA*) and Web (*WebSTATISTICA*) applications; interactively explore, drill down on, chart, etc., all intermediate results;
- Organize results in reports, spreadsheets, graphs, etc., or publish results on the Web;
- Access to *STATISTICA's* comprehensive library of analytic facilities;
- Update analyses and results automatically when the data change;
- Open architecture design. Fully integrate your own proprietary algorithms and methods or third-party algorithms;
- Fully programmable and customizable system (using the industry standard languages such as the built-in Visual Basic, C++, C#, Java, etc.). Develop highly customized data mining systems specifically tailored to your needs;
- Automatically deploy solutions in seconds using built-in tools, or add automatically generated computer code for deployment (e.g., in C++, PMML) to your own programs.

Click here for more information on the unique features of *STATISTICA Data Miner*

***Data Miner* in the *WebSTATISTICA* Client-Server installation.**
The desktop version of *STATISTICA Data Miner* is designed for the Windows environment. The Client-Server

version of *STATISTICA Data Miner* is platform independent on the Client side and features an Internet browser-based user interface; the Server side works with all major Web server operating systems (e.g., UNIX Apache) and Wintel server computers.



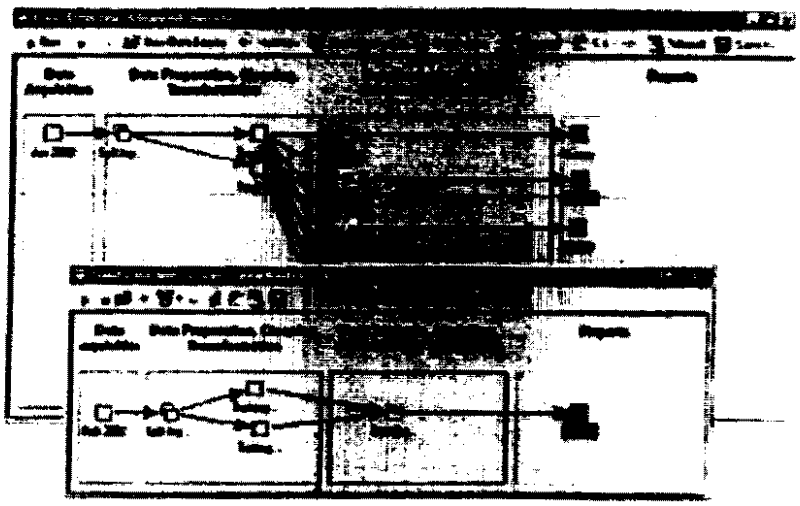
- Seamless integration of desktop and *WebSTATISTICA* data mining tools: design models on one platform (desktop or WebSTATISTICA server), execute on the other; train models on one platform (desktop or *WebSTATISTICA* server) and deploy to the other platform.
- Distributed processing and multi-threaded evaluation of projects: the program will automatically take advantage of multi-processor and/or multiple-server architectures, to evaluate models via multiple simultaneous processes (multithreading, distributed processing); hence the ability of *WebSTATISTICA Data Miner* installations to take full advantage of such architectures provides tremendous flexibility for scaling the system to mine even extremely large databases.
- Full flexibility of *WebSTATISTICA*: analyze data in batch mode, receive notification by email when the results are ready; share results in designated folders (repositories) with other stakeholders in the data mining projects; etc.
- Integrate input data, stakeholders, analysts, and users of results of data mining projects from any location around the world; *WebSTATISTICA* enables you to connect to data on one server (over the Internet), share analyses with other data mining professionals world wide, and deploy solutions and results to users in even the most remote locations (e.g., to branch managers in small rural areas, engineers on remote drilling platforms, ships en-route across oceans, etc.); as long as even slow Internet access is available, you can include individuals in those locations in your data mining project)
- Ideal for training (teaching) data mining: provide participants (students) with the option to analyze data from home or their office, wherever there is access to the Internet; allow professionals to complete assignments at the time and place that most conveniently fits their schedules. *WebSTATISTICA* allows all course or training participants hands-on experience with the most advanced data mining tools available today!

Advanced Software Technology = Efficient and Elegant User Interface

***STATISTICA* analysis "objects" and nodes.** At the heart of *STATISTICA Data Miner* is a set of over 300 highly optimized, efficient, and extremely fast *STATISTICA* procedures embedded in user-selectable nodes, which is used to specify the relations between the procedures (objects) and control the logic of the project (and the "flow" of data). This flexible, customizable architecture delivers the full functionality of all statistical and analytic procedures to the data mining environment as self-contained analysis objects. Behind each node, and accessible to advanced users of the *STATISTICA Data Miner* system, are simple scripts (analysis objects encoded in industry-standard Visual Basic) that serve as the "wrappers" or glue for defining the flow of data through the project, while the actual numerical analyses are performed via the extremely fast analytic procedures of *STATISTICA*. These objects, which can be used as the nodes for data cleaning and/or filtering, and for analyzing the data, are organized in the Node Browser. The nodes available in the node browser (and, hence, available to the data mining project) are:

- *Nodes for data input and data acquisition.* Here you can create and store the scripts necessary to connect to remote (protected) data sources on a server. Of course, you can also analyze *STATISTICA* data files or place holders for in-place processing of remote databases (**see IDP**), in which case no special nodes (scripts) have to be created.
- *Nodes for data filtering, cleaning, verification, feature selection, and sub-sampling.* These options are essential to data mining, to detect and correct erroneous information that may bias final conclusions.

The sub-sampling facilities are useful for analyzing very large data sets (see **Comparative performance benchmarks using large data sets**), to extract random or stratified random samples for further analyses. The feature selection options allow you to automatically select informative variables (predictors) from among, for example, hundreds of thousands of possible predictors (see also **Feature Selection and Variable Filtering**).

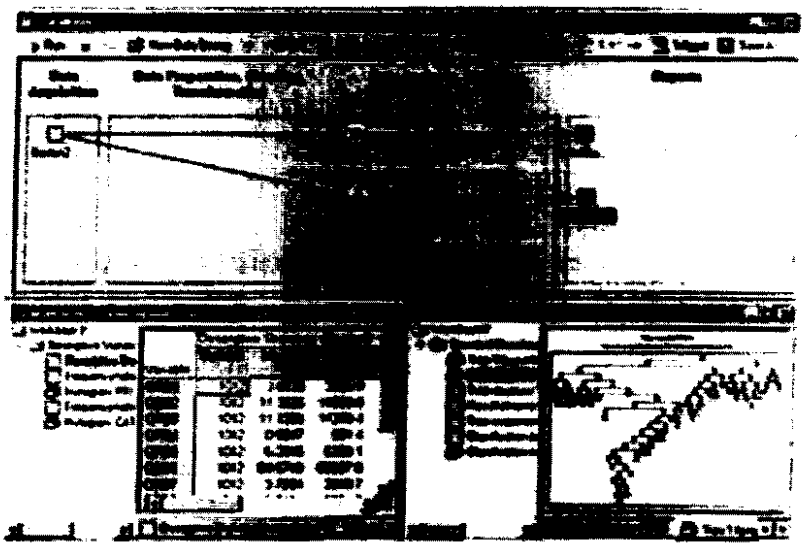


- *Nodes for data analyses.* These nodes contain the full functionality of all *STATISTICA* analyses and graphics capabilities; hundreds of procedures are available to address essentially all analytic needs that can possibly arise in your data mining project.

Creating the data mining project. These nodes can simply be connected in the data mining workspace.

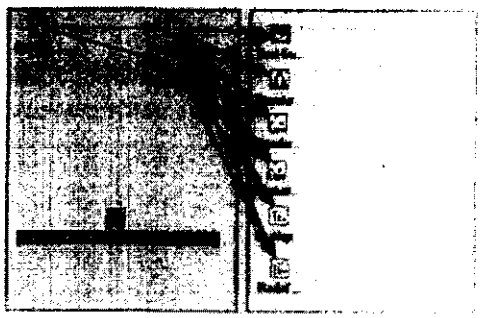
The data mining workspace is a structured, highly efficient, user-friendly data analysis environment, where you can move around and interconnect data, analyses, and results by simply dragging icons and connecting arrows. You can simultaneously open, modify, and run as many data mining workspaces as you like and drag nodes (objects) between workspaces and node browsers. The workspace area is pre-divided to make room for:

- *Data acquisition.* Here is where the data sources can be specified (e.g., *STATISTICA* data files, place-holders for in-place processing of data on remote servers, programs that generate data programmatically, for use in advanced modeling).
- *Data preparation, cleaning, transformation.* The nodes in this area will accept one or more data sources for input, and create one or more (filtered, cleaned, transformed) data sources for further "downstream" analyses.
- *Data analysis, modeling, classification, forecasting.* The nodes in this area will perform the numeric analyses.
- *Reports.* This area will show the results of the analyses.



Creating a Data Mining project is easy: first select a data source; second, apply any data preparation, cleaning, or transformation; third, connect the desired analyses to the cleaned data; and, fourth, review and/or publish the results. Many users of *STATISTICA Data Miner* will never need to go beyond this simple interactive, "point and click" user interface.

Specifying complex models. The simple user interface -- based on point-and-click selections from menus and browsers -- will allow you to apply even very advanced methods. Several comprehensive and flexible project "templates" can be selected to address common data mining tasks. For example, in order to find a good model for predicting credit risk of new clients based on historical data that includes various potentially useful predictors, you could simply select the template for the *Advanced Comprehensive Regression Models* project.



All you need to do next is connect your historical data, specify the variables of interest, and "train" the project; thus, in just a few seconds (select data file, select variables, select the arrow tool to connect the data), the program will automatically:

- Create two samples for training and for cross-validation, to avoid over-fitting;
- Apply best subset linear regression, standard regression trees algorithms, CHAID and exhaustive CHAID, a 3-layer multilayer perceptron neural network, and a radial basis function neural network to find a good model for predicting credit risk;
- Combine all responses into a meta-learner that picks the best model, or combines the predictions from multiple models.

After applying these cutting-edge techniques for modeling linear, nonlinear, or even chaotic relationships, you are ready for deployment: simply connect the data source for the new data (new customers) to the *Compute Best Prediction From All Models* node, and the program will automatically apply the fully trained models to derive the best prediction possible.

Speed. The analysis nodes (objects) contain the full functionality of *STATISTICA*, encapsulated into nodes that can further be customized using standard Visual Basic syntax. The actual analyses are performed via the highly optimized *STATISTICA* analysis modules, which have been refined for almost two decades to deliver maximum speed, capacity, and accuracy (see also **Accuracy Benchmarks**).

Large data sets. *STATISTICA Data Miner* uses a number of technologies specifically developed to optimize the processing of large data sets, and it is designed to handle even the largest scale computational problems and process very large databases. For example, data sets with over one million variables can be processed and screened automatically (using a wide selection of methods) to search for best predictors or most relevant variables (see also **Feature Selection and Variable Filtering** and **Comparative performance benchmarks using large data sets**).

Customizing analyses. The analyses or data cleaning/filtering operations implemented by the nodes of *STATISTICA Data Miner* can further be customized by simply double-clicking on the respective icons: Every icon contains the options to fully customize the respective operations; for example, clicking on a neural network node will bring up a dialog (and dialog help) for customizing the specific analysis (to change the number of iterations, number of layers in the network, the detail of reported results, etc.).

Saving the project. The entire project (workspace) can be saved, along with all customization, intermediate data sources, comments, etc. Routine analyses (e.g., for regular updating of a trained complex set of models for voted classification based on various methods) can be saved and later applied by clicking on a single button ("update").

Technical Note: *STATISTICA Data Miner* Node Scripts. *STATISTICA Data Miner's* computational routines are extremely fast and highly optimized. For example, in the *WebSTATISTICA* Client-Server environment, the program will automatically take advantage of multi-processor and/or multiple-server architectures (with proper hardware support), to evaluate models via multiple simultaneous processes (multithreading, distributed processing). Moreover, the highly optimized routines for processing data will outperform other software in head-to-head comparisons (see the benchmarks at www.statsoft.com for details). Yet, advanced users will find

it very easy to customize the system: each node in *STATISTICA Data Miner* consists of a standardized *STATISTICA Visual Basic* script (that calls the respective *STATISTICA* procedures), with access to additional functions to provide the user interface to further customize analyses. It may never be necessary to modify or customize these scripts; however, if your in-house IT department or consultants want to insert proprietary algorithms into *STATISTICA Data Miner*, this can very easily be accomplished. Any number of proprietary or highly customized numeric operations could be performed inside the script, to change practically all aspects of the data, or to apply any of the thousands of analytic functions available in form of simple function calls that can be made from C++ or *STATISTICA Visual Basic*. This general open architecture of *STATISTICA Data Miner* provides numerous unique (to data mining software) advantages (also further elaborated in the section on **Unique Features**).

- Each node can handle multiple data sources on input, and multiple data sources on output; identical operations can be applied to multiple data sources via a single node.
- A data source can be mapped into a database that does not need to actually (physically) reside on the machine running *STATISTICA Data Miner*, nor does it have to be copied; this is extremely important for the processing of large data sets, as they commonly occur in data mining (see IDP technology).
- You can perform operations within and between data sources; for example, you could merge data in different remote databases into a single data file, for further processing with *STATISTICA Data Miner* analytic nodes.
- Visual Basic itself is a simple, object-oriented language, available for most industry-standard application programs; there is a virtually limitless supply of programming resources, talented and experienced programmers, and ready-to-use third-party applications that can be integrated with *STATISTICA Data Miner*. Likewise, *STATISTICA Data Miner* can be integrated with other applications; for example, to automatically deliver results to the Web or email, or to export results into other applications. Also, a fully Web-based version of *STATISTICA Data Miner*, powered by *WebSTATISTICA* is available.
- *STATISTICA's* macro recording facilities will automatically record interactive analyses; these recordings can easily be converted into scripts for custom nodes.
- Where applicable, *STATISTICA's* analyses contain options for generating *STATISTICA Visual Basic* code for deployment (e.g., of trained neural networks); those scripts can be directly used in scripts for custom deployment nodes.

Deploying solutions. The results of analyses via *STATISTICA Data Miner* can be deployed (applied to new data or inside other automated data processing systems) in several ways.

- *Automatic deployment of models.* Data mining templates with deployment for standard types of analyses can be chosen as options from pull-down menus: select a template, connect training data to estimate models, and you are ready to apply the best solution (average solution, voted solution, etc.) to new data; the end user only needs to connect new data to the deployment node to compute predictions, classifications, forecasts, etc.
- *PMML-based rapid deployment of predictive models.* The *Rapid Deployment of Predictive Models* options provide the fastest, most efficient methods for computing predictions from fully trained models; in fact, it is very difficult to "beat" the performance (speed of computations) of this tool, even if you were to write your own compiled C++ code, based on the (C, C++, or C#) deployment code generated by the respective models. The *Rapid Deployment of Predictive Models* options allows you to load one or more PMML files with deployment information, and to compute very quickly (in a single pass through the data) predictions for large numbers of observations (for one or more models). PMML (Predictive Models Markup Language) files can be generated from practically all analytic procedures for predictive data mining (as well as the *Generalized EM & k-Means Cluster Analysis* options). PMML is an XML-based (Extensible Markup Language) industry standard set of syntax convention that is particularly well suited to allow sharing of deployment information in a Client-Server architecture (e.g., via *WebSTATISTICA*).
- *C, C++, C#, Visual Basic code generator options.* Code-generator options are also available for regression (prediction of continuous variables), classification (prediction of categorical variables), and clustering types of problems; for example, you can save C++ code or Visual Basic code that implements the prediction from tree-classification algorithms, linear discriminant function analysis, generalized linear models, neural networks, *MARSplines* (multivariate adaptive regression splines), k-means or EM clustering solutions (unsupervised learning), etc. The code generated by these options can quickly be integrated into custom programs for deployment. For example, the Visual

Basic code generated from *STATISTICA* analysis modules will seamlessly integrate into the *STATISTICA Data Miner* architecture; based on the Visual Basic code generated by *STATISTICA*, custom deployment nodes can be programmed in minutes, even by inexperienced programmers.

Using *STATISTICA Data Miner* with Extremely Large Data Sets

The entire *STATISTICA* family of products and *STATISTICA Data Miner* in particular are specifically optimized to efficiently process extremely large data sets (see **Comparative performance benchmarks using large data sets**), with millions of observations (records) and millions of variables (fields).

Processing databases that are larger than the local storage device. *STATISTICA Data Miner* (and optionally other *STATISTICA* products) can process data in (remote) databases "in-place" via its highly optimized **In-place Database Processing (IDP) technology**, which combines the processing resources of the database server and the local computer to (a) perform the queries (using the database server CPU) while simultaneously (b) processing the fetched records "on-the-fly" on the local machine (using the local computer (client) CPU). This way, databases that are larger than what could fit on the local machine can be processed, and significant performance gains can be achieved by saving the time that would normally be required to first import the data to the local device and only then process them locally. Practically all common database formats are supported, and powerful tools are provided for defining the database connection (query).

Processing databases with extremely large numbers of variables (fields): The unique feature selection and variable screening facilities. When the number of variables in the input data file is extremely large, *STATISTICA Data Miner* can automatically select subsets of variables from among even over a million of variables (candidates) for predictive data mining. The extremely fast and efficient algorithm will select variables (features) that are likely to be the most relevant predictors in the current data set, without introducing biases into subsequent model building for predictive data mining.

Processing data files with extremely large numbers of cases (records): Flexible and efficient random sampling. *STATISTICA* products (including *STATISTICA Data Miner*) can process data files with practically unlimited numbers of cases (records), and *STATISTICA's* data access procedures are highly optimized. However, including all records in the analyses when the number of records is extremely large is (a) entirely unnecessary, (b) time consuming, and (c) often impractical or impossible (in extreme cases it could take hours merely to read all records). In order to speed up the analytic process, *STATISTICA Data Miner* includes sophisticated tools for drawing random or stratified random samples from huge data sets (databases). The user can quickly extract simple or systematic random samples of appropriate sizes, with or without replacement, from huge data sets (e.g., with many millions of records) for further analyses with sophisticated modeling tools that may require multiple passes through the data (e.g., neural networks, generalized linear models, etc.). The random sub-sampling is based on *STATISTICA's* validated random number generator. Note that *STATISTICA* is one of only few commercially available software products that have passed the most advanced and most recognized tests for randomness (the **DIEHARD suite of tests**).

Distributed processing and multi-threaded evaluation of projects in the Client-Server environment. The *WebSTATISTICA* Client-Server installation of *STATISTICA Data Miner* offers additional advantages for processing very large datasets. The program will automatically take advantage of multi-processor and/or multiple-server architectures (with proper hardware support), to evaluate models via multiple simultaneous processes (multithreading, distributed processing). Hence, considering the decreasing costs for advanced server hardware (with multiple processors, or for multiple-server installations), the ability of *WebSTATISTICA Data Miner* installations to take full advantage of such architectures provides tremendous flexibility for scaling the system to mine even extremely large databases.

Data Mining Tools

STATISTICA Data Miner offers the most comprehensive selection of statistical, exploratory, and visualization techniques available on the market, including leading edge and highly efficient neural network/machine learning and classification procedures. Also, the complete analytic functionality of *STATISTICA* is available for data mining, encapsulated in over 300 nodes that can be selected in a structured and customizable Node Browser and dragged into the data mining workspace.

The specialized tools for data mining are optimized for speed and efficiency and can be classified into the following five general "areas" (each comprising of a set of *STATISTICA* modules, some of them offered only in the *STATISTICA Data Miner* environment):

General Slicer/Dicer and Drill-Down Explorer. A large number of analysis nodes are available for creating exploratory graphs, to compute descriptive statistics, tabulations, etc. These nodes can be connected to input data sources, or to all intermediate results. A specialized *STATISTICA* application module is available (***STATISTICA Drill-Down Explorer***) for interactively exploring your data by drilling down on selected variables, and categories or ranges of values in those variables. For example, you can drill-down on Gender, to display the distribution for a variable Income for females only; next you could drill down on a specific income group, to explore (e.g., create graphical summaries for) selected variables, for females in the selected income group only. A unique feature of *STATISTICA Drill-Down Explorer* is the ability to select and deselect drill-down variables and categories in any order; so you could next deselect variable Gender and thus display selected graphs and statistics for the selected Income group, but now for both males and females. Another unique feature of the *Drill-Down Explorer* is its variety of categorization ("slicing") methods. Hence, the *STATISTICA Drill-Down Explorer* offers tremendous flexibility for "slicing-and-dicing" the data. The *STATISTICA Drill-Down Explorer* can be applied to raw data, database connections for in-place processing of data in remote databases, or to any intermediate result computed in a *STATISTICA Data Miner* project. (A fully integrated OLAP application is also available (as an optional add-on module for enterprise installations), please contact StatSoft for details.)

General Classifier. *STATISTICA Data Miner* offers the widest selection of tools to perform data mining classification techniques (and build related deployable models) available on the market, including **generalized linear models** (for binomial and multinomial responses), **classification trees**, **general classification and regression tree modeling** (*GTrees*), **general *CHAID* models**, **cluster analysis techniques** (including "large capacity" implementations of tree-clustering as well as k-means and EM clustering methods with v-fold crossvalidation options to determine automatically the best number of clusters), and **general discriminant analysis models** (including best-subset selection of predictors). Also, the numerous advanced neural network classifiers available in ***STATISTICA Neural Networks*** are available in *STATISTICA Data Miner*, and can be used in conjunction or competition with other classification techniques.

- *Deployment.* Where applicable, the program includes options for generating C, C++, C#, *STATISTICA Visual Basic*, or (XML-syntax) PMML code for deployment of final solutions in your custom programs. Models are also automatically available for deployment after training, so all you need to do is connect new data to the special deployment node to compute predicted classifications.

General Modeler/Multivariate Explorer. *STATISTICA Data Miner* offers the widest selection of tools to build deployable data mining models, based on linear, nonlinear, or neural network techniques and tools to explore data; the user can also build predictive models based on general multivariate techniques. In summary, *STATISTICA* offers the full range of techniques, from **linear** and **nonlinear regression models**, **advanced generalized linear** and **generalized additive models**, regression trees and *CHAID*, to advanced neural network methods and multivariate adaptive regression splines (*MARSplines*). *STATISTICA Data Miner* also includes techniques that are not usually found in data mining software, such as **partial least squares methods** (for feature selection, to reduce large numbers of variables), **survival analysis** (for analyzing data containing censored observations; e.g. for medical research data and data from industrial reliability and quality control studies), **structural equation modeling techniques** (to build and evaluate confirmatory linear models), **correspondence analysis** (for analyzing the structure of complex tables), **factor analysis** and **multidimensional scaling** (for exploring structure in large numbers of variables), and many others.

- *Deployment.* Where applicable, the program includes options for generating C, C++, C#, *STATISTICA Visual Basic*, or (XML-syntax) PMML code for deployment of final solutions in your custom

programs; models are also automatically available for deployment after training, so all you need to do is connect new data to the special deployment node, to compute predicted values.

General Forecaster. *STATISTICA Data Miner* includes a broad selection of traditional (i.e., non-neural networks-based) forecasting techniques (including **ARIMA**, **exponential smoothing** with seasonal components, Fourier spectral decomposition, **seasonal decomposition**, regression- and **polynomial lags analysis**, etc.), as well as neural network methods for time series data.

- *Deployment.* Forecasts can automatically be computed for multiple models in data mining projects, and plotted in a single graph for comparative evaluation. For example, you can compute and compare predictions from multiple ARIMA models, different methods for seasonal and non-seasonal exponential smoothing, and the best time-series neural network architectures (after searching over 100 different architectures).

General Neural Networks Explorer. This tool contains the most comprehensive selection of neural network methods available on the market. This powerful component of *STATISTICA Data Miner* offers tools to approach virtually any data mining problem (including classification, hidden structure detection, and powerful forecasting). One of the unique features of the *NN Explorer* is the selection of intelligent problem solvers and automatic wizards that use Artificial Intelligence methods to help you solve the most demanding problems involved in advanced NN analysis (such as selecting the best network architecture and the best subset of variables). The Explorer offers the widest selection of cutting-edge NN architectures and procedures and highly optimized algorithms that include: multilayer perceptrons, radial basis function networks, probabilistic neural networks, generalized regression neural networks, self-organizing feature maps, linear models, principal components network, and cluster networks. Network ensembles of these architectures can also be evaluated. Estimation methods include back propagation, conjugate gradient decent, quasi-Newton, Levenberg-Marquardt, quick propagation, delta-bar-delta, LVQ, pruning algorithms, and more; options are available for cross validation, bootstrapping, subsampling, sensitivity analysis, etc.

- *Deployment.* ***STATISTICA Neural Networks*** includes code generator options to produce C, C++, C#, and *STATISTICA Visual Basic* code for one or more trained networks as well as ensembles of networks. This code can be quickly incorporated into your own custom deployment programs. In addition, fully trained neural networks and ensembles of neural networks can be saved, to be applied later for computing predicted responses or classifications for new data. A deployment node can be dragged into the data miner workspace to perform prediction and predictive classification based on trained neural networks automatically; all you have to do (after the participating network architectures are trained) is connect the data for deployment.

Specialized Data Mining Modules

A large portion of analytic functionality used by *STATISTICA Data Miner* is driven by the computational engines of modules that are included in various other *STATISTICA* products (refer to ***STATISTICA* Products** for detailed information about those modules):

- Neural Networks techniques (the largest selection of architectures available, automatic problem solver tools, advanced feature selection techniques).
- All *STATISTICA* Graphics Tools and interactive exploration/visualization tools; Descriptive statistics, breakdowns, and exploratory data analysis; Frequency Tables, Crosstabulations, Tables and Stub-and-Banner Tables, Multiple Response Analysis; Nonparametric Statistics; Distribution Fitting; Power Analysis Techniques.
- General Linear Models (GLM); General Regression Models (GRM); Generalized Linear Models (GLZ); General Partial Least Squares Models (PLS); Variance Components and Mixed Model ANOVA/ANCOVA; Survival/Failure Time Analysis; General Nonlinear Estimation with Logit and Probit Regression; Log-Linear Analysis of Frequency Tables; Time Series Analysis/Forecasting; Structural Equation Modeling/Path Analysis (SEPATH).

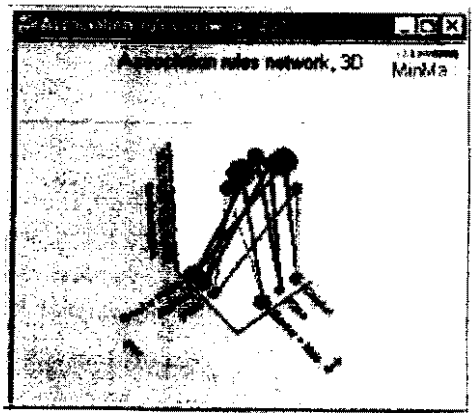
- Cluster Analysis Techniques; Factor Analysis; Principal Components & Classification Analysis; Canonical Correlation Analysis; Reliability/Item Analysis; Classification Trees; Correspondence Analysis; Multidimensional Scaling; Discriminant Analysis; General Discriminant Analysis Models (GDA).
- Quality Control Charts techniques, Process Analysis, and Experimental Design (DOE) procedures.

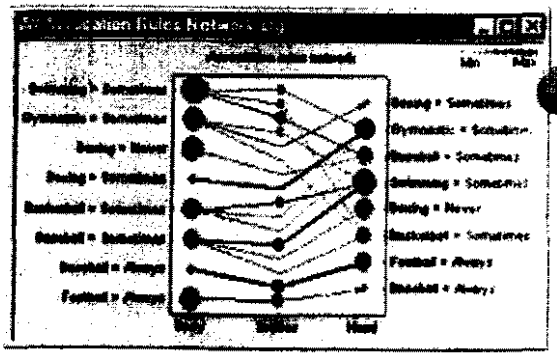
However, several modules include selections of highly specialized data mining and data mining modeling techniques that are offered only as part of *STATISTICA Data Miner*. The following sections include technical information about these modules.

- **Feature Selection and Variable Filtering (for very large data sets)**
- **Association Rules**
- **Interactive Drill-Down Explorer**
- **Generalized EM & k-Means Cluster Analysis**
- **Generalized Additive Models (GAM)**
- **General Classification and Regression Trees (GTrees)**
- **General CHAID (Chi-square Automatic Interaction Detection) Models**
- **Interactive Classification and Regression Trees**
- **Boosted Trees**
- **Multivariate Adaptive Regression Splines (Mar Splines)**
- **Goodness of Fit Computations**
- **Rapid Deployment of Predictive Models**

FEATURE SELECTION AND VARIABLE FILTERING. This module will automatically select subsets of variables from extremely large data files or databases connected for in-place processing (IDP). The module can handle a practically unlimited number of variables: over a million (!) input variables can be scanned to select predictors for regression or classification. Specifically, the program includes several options for selecting variables ("features") that are likely to be useful or informative in specific subsequent analyses. The unique algorithms implemented in the *Feature Selection and Variable Filtering* module will select continuous and categorical predictor variables which show a relationship to the continuous or categorical dependent variables of interest, regardless of whether that relationship is simple (e.g., linear) or complex (nonlinear, non-monotone). Hence, the program does not bias the selection in favor of any particular model that you may use to find a final best rule, equation, etc. for prediction or classification. Various advanced feature selection options are also available. This module is particularly useful in conjunction with the **in-place processing of databases** (without the need to copy or import the input data to the local machine), when it can be used to scan huge lists of input variables, select likely candidates that contain information relevant to the analyses of interest, and automatically select those variables for further analyses with other nodes in the data miner project. Subsets of variables based on an initial scan via this module can be submitted to further (post-) feature selection methods based on neural networks, *MAR Splines*, linear regression or classifiers, or *CHAID*. These options allow *STATISTICA Data Miner* to handle data sets in the multiple giga- and terabyte range (see **Comparative performance benchmarks using large data sets**).

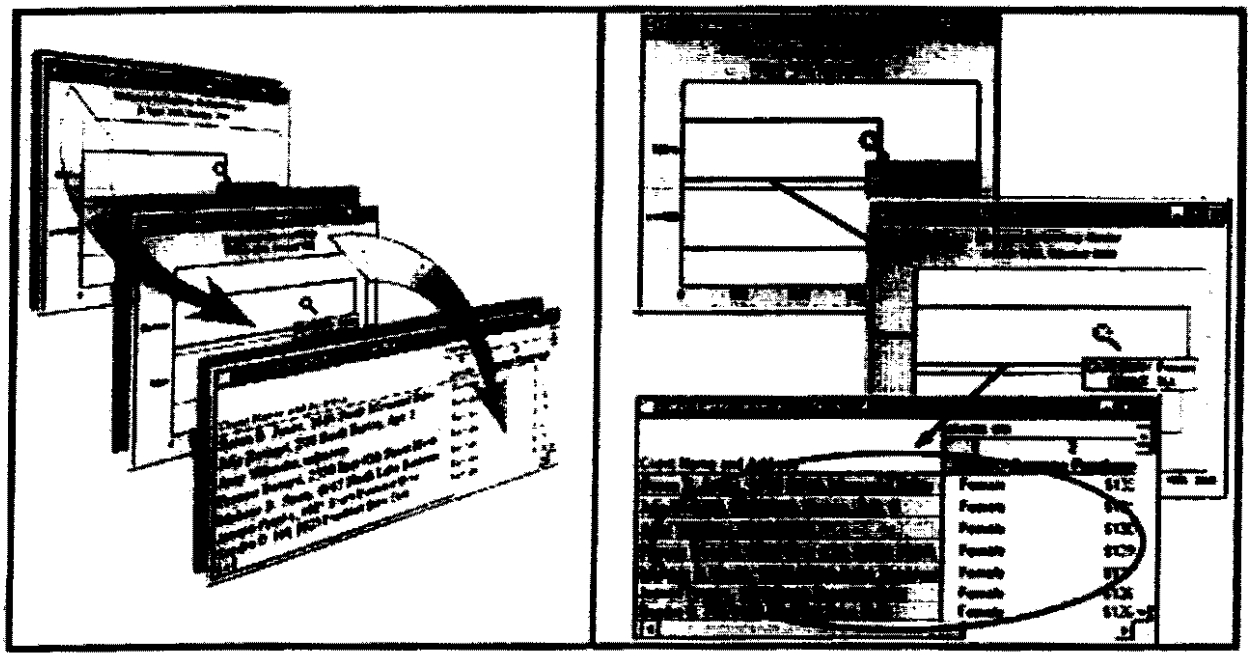
ASSOCIATION RULES. This module contains a complete implementation of the so-called A-priori algorithm for detecting ("mining for") association rules such as "customers who order product A, often also order product B or C" or "employees who said positive things about initiative X, also frequently complain about issue Y but are happy with issue Z" (see Agrawal and Swami, 1993; Agrawal and Srikant, 1994; Han and Lakshmanan, 2001; see also Witten and Frank, 2000). The *Association Rules* module allows you to process rapidly huge data sets for associations (relationships), based on pre-defined "threshold" values for detection. Specifically, the program will detect relationships or associations between specific values of categorical variables in large data sets. This is a common task in many data mining projects applied to databases containing records of customer transactions (e.g., items purchased by each customer), and also in the area of **text mining**. Like all modules of *STATISTICA*, data in external databases can be processed by the *STATISTICA Association Rules* module in-place (see **IDP technology**), so the program is prepared to handle efficiently extremely large analysis tasks.





The results can be displayed in tables, and also in unique 2D and 3D graphs where strong associations are highlighted by thick lines connecting the respective items.

INTERACTIVE DRILL-DOWN EXPLORER. A first step of many data mining projects is to explore the data interactively, to gain a first "impression" of the types of variables in the analyses, and their possible relationships. The purpose of the *Interactive Drill-Down Explorer* is to provide a combined graphical, exploratory data analysis, and tabulation tool that will allow you to quickly review the distributions of variables in the analyses, their relationships to other variables, and to identify the actual observations belonging to specific subgroups in the data.



How the *Drill-Down Explorer* Works. The "drill-down" metaphor within the data mining context summarizes the basic operation of this analytic process quite well: The program allows you to select observations from larger data sets by selecting subgroups based on specific values or ranges of values of particular variables of interest (e.g., *Gender* and *Average Purchase* in the example above); in a sense you can expose the "deeper layers" or "strata" in the data by reviewing smaller and smaller subsets of observations selected by increasingly complex logical selection conditions.

Drilling "up." The interactive nature of the *Drill Down Explorer* allows you not only to drill down into the data or database (select groups of observations with increasingly specific logical selection conditions), but also to "drill up": At any time, you can select one of the previously specified variable (*category*) groups and de-select it from the list of drill-down conditions; while processing the data the program will then only select those observations that fit the remaining logical (*case*) selection conditions, and update the results accordingly.

Applications of the *Interactive Drill-Down Explorer*. The example shown earlier is very simple, exposing only the basic functionality of the program. The real power of the *STATISTICA Interactive Drill-Down Explorer* lies in the various auxiliary results which can automatically be updated during the interactive drill-down/up exploration: you can select a list of variables for review, and compute for the selected cases:

- Descriptive statistics and frequency tables;
- Box-and-whiskers plots summarizing the distributions of continuous variables;
- Scatterplot matrices summarizing the relationships between continuous variables;
- All of the other statistical and graphical analyses available in *STATISTICA* by extracting the observations belonging to the current subset;

For example, you could review the types of purchases that customers made with different demographic characteristics, study the effectiveness of certain drugs within different treatment groups, ages, etc., or extract likely customers for a new product from a database of previous customers based on careful study of apparent (market) segments exposed by the drill-down analysis.

***Interactive Drill-Down Explorer* and OLAP (On-Line Analytic Processing).** On the surface, the operation of the simplest aspect of the *Interactive Drill-Down Explorer* (exploration of multidimensional tables) is very similar to the functionality offered by designated OLAP tools (such as those offered in the optional OLAP add-

on module for *STATISTICA Data Miner*). OLAP tools allow users to quickly query a database to extract observations and summary information about those observations taking advantage of the optimized OLAP Server facilities offered for a specific database platform (e.g., Oracle, or MS SQL Server), and often providing significant performance advantages over tools based on traditional (non-OLAP driven) query tools. However, the main advantages *STATISTICA Interactive Drill-Down Explorer* over OLAP are:

(a) its tight integration with *STATISTICA's* flexible categorization tools and exploratory environment (the analytic capabilities provided in the *STATISTICA Interactive Drill-Down Explorer* are much more comprehensive and also general than typical OLAP tools, supporting flexible "drill up" operations, and allowing you to quickly review custom, complex summary graphs, detailed descriptive statistics, etc.), and

(b) the fact that the *STATISTICA Interactive Drill-Down Explorer* is not limited to any particular database platform and does not require a designated OLAP Server to be present (e.g., it can operate directly on *STATISTICA* data files). At the same time, by connecting to the *STATISTICA* application a (remote) database for **in-place processing**, you can efficiently perform drill-down operations on any data source, regardless of whether or not designated OLAP tools are available on the server.

GENERALIZED EM & K-MEANS CLUSTER ANALYSIS. The *STATISTICA Generalized EM (Expectation Maximization) and k-Means Clustering* module is an extension of the techniques available in the general *STATISTICA Cluster Analysis* options, specifically designed to handle large data sets and to allow clustering of continuous and/or categorical variables, and to provide the functionality for complete unsupervised learning (clustering) for pattern recognition, with all deployment options for predictive clustering. Various cross-validation options are provided (including modified v-fold cross-validation options) that will automatically choose and evaluate a best final solution for the clustering problem; you do not need to specify the number of clusters before an analysis; instead the program will use automatic (cross-validation based) methods to choose a best cluster solution (number of clusters) for you! The advanced EM clustering technique available in this module is sometimes referred to as probability-based clustering or statistical clustering. The program will cluster observations based on continuous and categorical variables, assuming different distributions for the variables in the analyses (as specified by the user). Various cross-validation options are provided to allow you to choose and evaluate a best final solution for the clustering problemDetailed output summaries and graphs (e.g., distribution plots for EM clustering), and detailed classification statistics are computed for each observation. These methods are optimized to handle very large data sets, and various results are provided to facilitate subsequent analyses using the assignment of observations to clusters. Options for deploying cluster solutions (in C, C++, C#, Visual Basic, or XML syntax based PMML), for classifying new observations, are also included.

GENERALIZED ADDITIVE MODELS (GAM). The *STATISTICA Generalized Additive Models* facilities are an implementation of methods developed and popularized by Hastie and Tibshirani (1990); additional detailed discussion of these methods can also be found in Schimek (2000). The program will handle continuous and categorical predictor variables. Note that *STATISTICA* includes a comprehensive selection of methods for fitting non-linear models to data, such as the ***Nonlinear Estimation module***, ***Generalized Linear Models***, ***General Classification and Regression Trees***, etc.

Distributions and link functions. The program allows the user to choose from a wide variety of distributions for the dependent variable, and link functions for the effects of the predictor variables on the dependent variable:

Normal, Gamma, and Poisson distributions:

Log link:	$f(z) = \log(z)$
Inverse link:	$f(z) = 1/z$

Identity link: $f(z) = z$

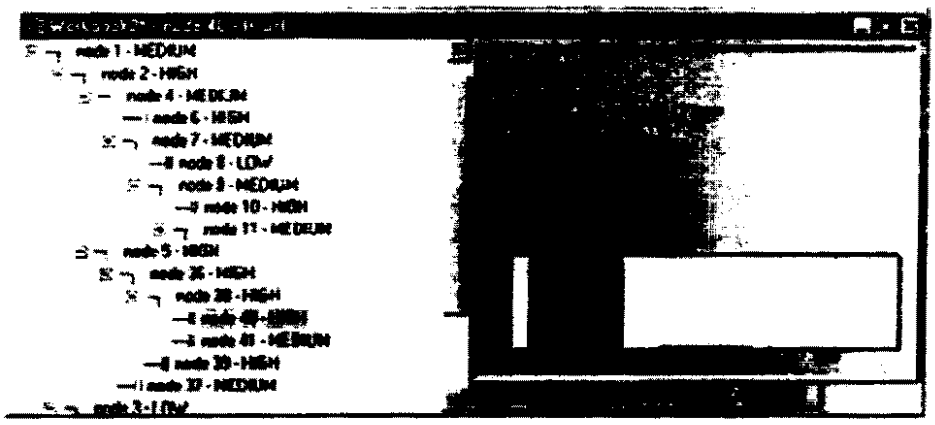
Binomial distribution:

Logit link: $f(z)=\log(z/(1-z))$

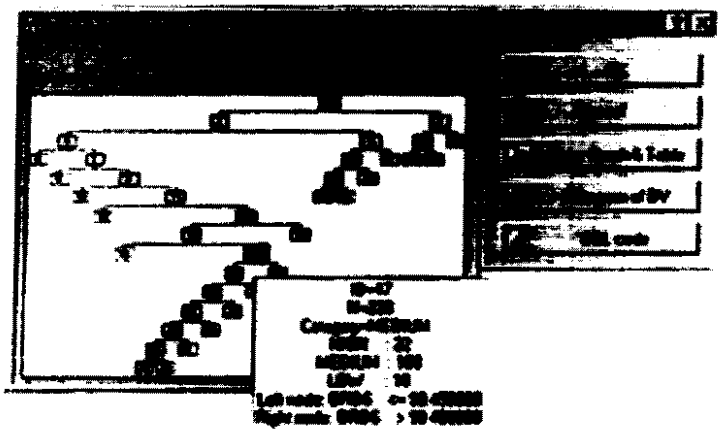
Scatterplot smoother. The program uses the cubic spline smoother with user-defined degrees of freedom to find an optimum transformation (function) of the predictor variables.

Results statistics. The program will report a comprehensive set of results statistics to aid in the evaluation of the model-adequacy, model fit, and interpretation of results; specifically, results include: the iteration history for the model fitting computations, summary statistics including the overall R-square value (computed from the deviance statistic) model degrees of freedom, and detailed observational statistics pertaining to the predicted response, residuals, and the smoothing of the predictor variables. Results graphs include plots of observed responses vs. residual responses, predicted values vs. residuals, histograms of observed and residual values, normal probability plots of residual values, and partial residual plots for each predictor, indicating the cubic spline smoothing fit for the final solution; for binary responses (e.g., logit-models) lift charts can also be computed.

GENERAL CLASSIFICATION AND REGRESSION TREES (GTrees). This module is a comprehensive implementation of the methods described as CART® by Breiman, Friedman, Olshen, and Stone (1984). However, the *GTrees* module contains various extensions and options that are typically not found in implementations of this algorithm, and that are particularly useful for data mining applications.



User interface; specifying "models." In addition to standard analyses (as described by Breiman, et al.), the implementation of these methods in *STATISTICA* allow you to specify ANOVA/ANCOVA-like designs with continuous and/or categorical predictor variables, and their interactions. Three alternative user interfaces are provided to allow you to specify such designs; these are analogous to the methods provided in ***GLM (General Linear Models)***, ***GLZ (Generalized Linear Models)***, ***GRM (General Regression Models)***, ***GDA (General Discriminant Analysis Models)***, and ***PLS (General Partial Least Squares Models)***, and are described in detail in the respective sections. In short, ANOVA/ANCOVA-like predictor designs can be specified via dialogs, Wizards, or (design) command syntax; moreover, the command syntax is compatible across modules, so you can quickly apply identical designs to very different analyses (e.g., compare the quality of classification using *GDA* vs. *GTrees*).



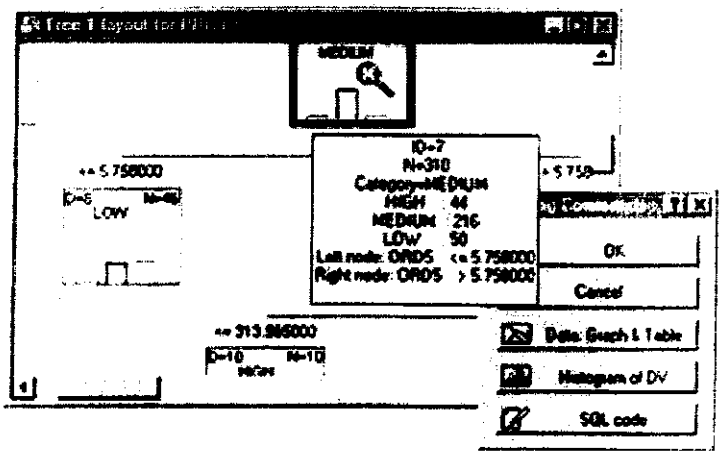
Tree pruning, selection, validation. The program provides a large number of options for controlling the building of the tree(s), the pruning of the tree(s), and the

selection of the best-fitting solution. For continuous dependent (criterion) variables, pruning of the tree can be based on the variance, or on FACT-style pruning. For categorical dependent (criterion) variables, pruning of the tree can be based on misclassification errors, variance, or FACT-style pruning. You can specify the maximum number of nodes for the tree or the minimum *n* per node. Options are provided for validating the best decision tree, using V-fold cross validation, or by applying the decision tree to new observations in a validation sample. For categorical dependent (criterion) variables, i.e., for classification problems, various measures can be chosen to modify the algorithm and to evaluate the quality of the final classification tree: Options are provided to specify user-defined prior classification probabilities and misclassification costs; goodness-of-fit measures include the Gini measure, Chi-square, and G-Square.

Missing data and surrogate splits. Missing data values in the predictors can be handled by allowing the program to determine splits for surrogate variables, i.e., variables that are similar to the respective variable used for a particular split (node).

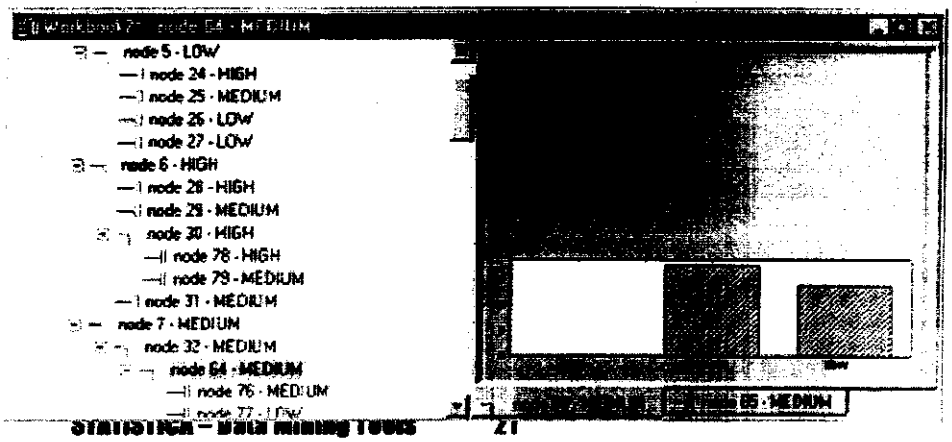
ANOVA/ANCOVA-like designs. In addition to the traditional CART®-style analysis, you can combine categorical and continuous predictor variables into ANOVA/ANCOVA-like designs and perform the analysis using a design matrix for the predictors. This allows you to evaluate and compare complex predictor models, and their efficacy for prediction and classification using various analytic techniques (e.g., ***General Linear Models***, ***Generalized Linear Models***, ***General Discriminant Analysis Models***, etc.).

Tree browser. In addition to simple summary tree graphs, you can display the results trees in intuitive interactive tree-browsers that allow you to collapse or expand the nodes of the tree, and to quickly review the most salient information regarding the respective tree node or classification. For example, you can highlight (click on) a particular node in the browser-panel and immediately see the classification and misclassification rates for that particular node. The tree-browser provides a very efficient and intuitive facility for reviewing complex tree-structures, using methods that are commonly used in windows-based computer application to review hierarchically structured information. Multiple tree-browser can be displayed simultaneously, containing the final tree, and different sub-trees pruned from the larger tree, and by placing multiple browsers side-by-side it is easy to compare different tree structures and sub-trees. The *STATISTICA Tree Browser* is an important innovation to aid with the interpretation of complex decision trees.



Interactive trees. Options are also provided to review trees interactively, either by using *STATISTICA* Graphics brushing tools or by placing large tree graphs into scrollable graphics windows where large graphs can be inspected "behind" a smaller (scrollable) window.

Results statistics. The *STATISTICA GTrees* module provides a very large number of results options. Summary results for each node are accessible, detailed statistics are computed pertaining to classification, classification costs, gain, and so on. Unique graphical summaries are also available, including histograms (for

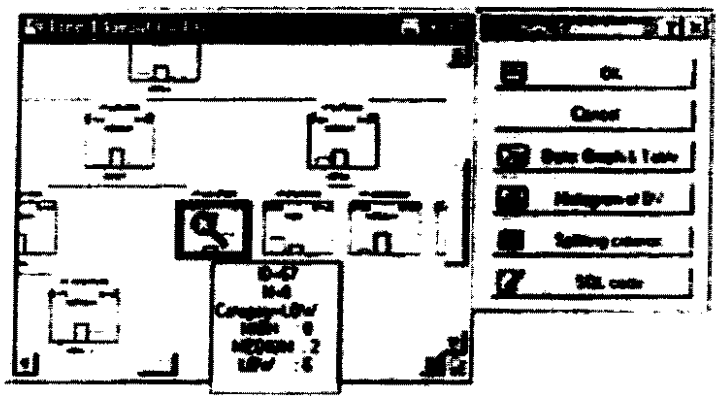


classification problems) for each node, detailed summary plots for continuous dependent variables (e.g., normal probability plots, scatterplots), and parallel coordinate plots for each node, providing an efficient summary of patterns of responses for large classification problems. As in all statistical procedures of *STATISTICA*, all numerical results can be used as input for further analyses, allowing you to quickly explore and further analyze observations classified into particular nodes (e.g., you could use the *GTrees* module to produce an initial classification of cases, and then use best-subset selection of variables in *GDA* to find additional variables that may aid in the further classification).

C, C++, *STATISTICA* Visual Basic, SQL Code generators. The information contained in the final tree can be quickly incorporated into your own custom programs or database queries via the optional C, C++, *STATISTICA* Visual Basic, or SQL query code generator options. The *STATISTICA* Visual Basic will be generated in form that is particularly well suited for inclusion in custom nodes for *STATISTICA Data Miner*.

GENERAL CHAID (Chi-square Automatic Interaction Detection) MODELS. Like the implementation of ***General Classification and Regression Trees GTrees*** in *STATISTICA*, the *General Chi-square Automatic Interaction Detection* module provides not only a comprehensive implementation of the original technique, but extends these methods to the analysis of ANOVA/ANCOVA - like designs.

Standard CHAID. The *CHAID* analysis can be performed for both continuous and categorical dependent (criterion) variables. Numerous options are available to control the construction of hierarchical trees: the user has control over the minimum *n* per node, maximum number of nodes, and probabilities for splitting and for merging categories; the user can also request exhaustive searches for the best solution (Exhaustive *CHAID*); V-fold validation statistics can be computed to evaluate the stability of the final solution; for classification problems, user-defined misclassification costs can be specified.



ANOVA/ANCOVA-like designs. In addition to the traditional *CHAID* analysis, you can combine categorical and continuous predictor variables into ANOVA/ANCOVA-like designs and perform the analysis using a design matrix for the predictors. This allows you to evaluate and compare complex predictor models, and their efficacy for prediction and classification using various analytic techniques (e.g., ***General Linear Models, Generalized Linear Models, General Discriminant Analysis Models, General Classification and Regression Tree Models***, etc.). Refer also to the description of ***GLM (General Linear Models)*** and ***General Classification and Regression Trees (GTrees)***, above, for details.

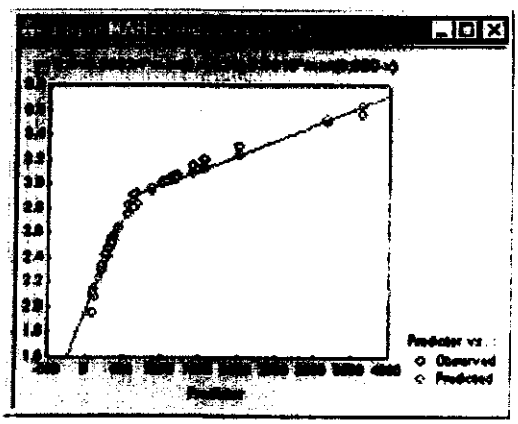
Tree browser. Like the binary results tree used to summarize binary classification and regression trees (see ***GTrees***), the results of the *CHAID* analysis can be reviewed in the ***STATISTICA Tree Browser***. This unique tree browser provides a very efficient and intuitive facility for reviewing complex tree-structures and for comparing multiple tree-solutions side-by-side (in multiple tree-browsers), using methods that are commonly used in windows-based computer applications to review hierarchically structured information. The *STATSTICA Tree Browser* is an important innovation to aid with the interpretation of complex decision trees. For additional details, see also the description of the tree browser in the context of the ***General Classification and Regression Trees (GTrees)***.

Results statistics. The *STATISTICA General CHAID Models* module provides a very large number of results options. Summary results for each node are accessible, detailed statistics are computed pertaining to classification, classification costs, and so on. Unique graphical summaries are also available, including histograms (for classification problems) for each node, detailed summary plots for continuous dependent variables (e.g., normal probability plots, scatterplots), and parallel coordinate plots for each node, providing an

efficient summary of patterns of responses for large classification problems. As in all statistical procedures of *STATISTICA*, all numerical results can be used as input for further analyses, allowing you to quickly explore and further analyze observations classified into particular nodes (e.g., you could use the ***GTrees*** module to produce an initial classification of cases, and then use best-subset selection of variables in ***GDA*** to find additional variables that may aid in the further classification).

INTERACTIVE CLASSIFICATION AND REGRESSION TREES. In addition to the modules for automatic tree building (e.g., *General Classification and Regression Trees, General CHAID models*), *STATISTICA Data Miner* also includes designated tools for building such trees interactively. You can choose either the (binary) *General Classification and Regression Trees* method or the *CHAID* method for building the (decision) tree, and at each step grow the tree either interactively (by choosing the splitting variable and splitting criterion) or automatically. When growing trees interactively, you have full control over all aspects of how to select and evaluate candidates for each split, how to categorize the range of values in predictors, etc. The highly interactive tools available for this module allow you to grow and prune back trees to quickly evaluate the quality of the tree for classification or regression prediction and to compute all auxiliary statistics at each stage to fully explore the nature of each solution. This tool is extremely useful for predictive data mining as well as for exploratory data analysis (EDA), and includes the complete set of options for automatic deployment, for the prediction or predicted classification of new observations (see also the description of these options in the context of *CHAID* and the *General Classification and Regression Trees* modules).

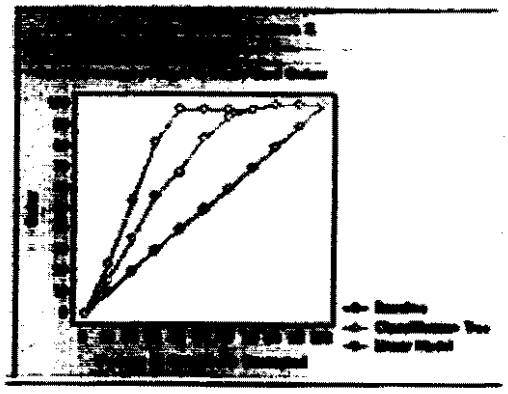
BOOSTED TREES. The most recent research on statistical and machine learning algorithms suggests that for some "difficult" estimation and prediction (predicted classification) tasks, using successively boosted simple trees can yield more accurate predictions than neural network architectures or complex single trees alone. *STATISTICA Data Miner* includes an advanced *Boosted Trees* module for applying this technique to predictive data mining tasks. You have control over all aspects of the estimation procedure and detailed summaries of each stage of the estimation procedures are provided so that the progress over successive steps can be monitored and evaluated. The results include most of the standard summary statistics for classification and regression computed by the *General Classification and Regression Trees* module. Automatic methods for deployment of the final boosted tree solution for classification or regression prediction are also provided.



MULTIVARIATE ADAPTIVE REGRESSION SPLINES (MAR Splines). The *STATISTICA MAR Splines (Multivariate Adaptive Regression Splines)* module is based on a complete implementation of this technique, as originally proposed by Friedman (1991; Multivariate Adaptive Regression Splines, Annals of Statistics, 19, 1-141); in *STATISTICA Data Miner*, the *MARSplines* options have further been enhanced to accommodate regression and classification problems, with continuous and categorical predictors.

The program, which in terms of its functionality can be considered a generalization and modification of stepwise *Multiple Regression and Classification and Regression Trees (GC&RT)*, is specifically designed (optimized) for processing very large data sets. A large number of results options and extended diagnostics are available to allow you to evaluate numerically and graphically the quality of the *MAR Splines* solution.

C/C++, C#, *STATISTICA Visual Basic*, XML syntax based PMML code generators. The information contained in the model can be quickly incorporated into your own custom programs via the optional C/C++/C#, *STATISTICA Visual Basic*, or (XML-syntax based) PMML code generator options. *STATISTICA Visual Basic* will be generated in a form that is particularly well suited for inclusion in custom nodes for *STATISTICA Data Miner*. PMML (Predictive Models Markup Language) files with deployment information can be used with the *Rapid Deployment of Predictive Models* options to compute predictions for large numbers of cases very efficiently; PMML files are fully portable, and deployment information generated via the desktop version of *STATISTICA Data Miner* can be used in *WebSTATISTICA Data Miner* (i.e., on the server side of Client-Server installations), and vice versa.

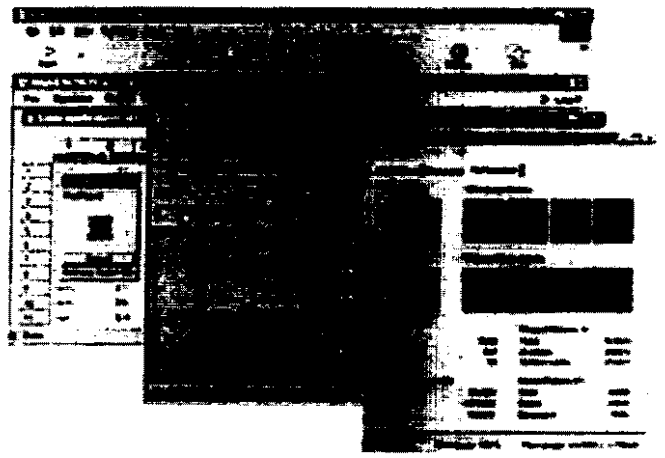


GOODNESS OF FIT COMPUTATIONS. The *STATISTICA Goodness of Fit* module will compute various goodness of fit statistics for continuous and categorical response variables (for regression and classification problems). This module is specifically designed for data mining applications to be included in "competitive evaluation of models" projects as a tool to choose the best solution. The program uses as input the predicted values or classifications as computed from any of the *STATISTICA* modules for regression and classification, and computes a wide selection of fit statistics as well as graphical summaries for each fitted response or classification. Goodness of fit statistics for continuous responses include least squares deviation (LSD), average deviation, relative squared error, relative absolute error, and the correlation coefficient. For classification problems (for categorical response variables), the program will compute Chi-square, G-square (maximum likelihood chisquare), percent disagreement (misclassification rate), quadratic loss, and information loss statistics.

RAPID DEPLOYMENT OF PREDICTIVE MODELS. The *Rapid Deployment of Predictive Models* module allows you to load one or more PMML (Predictive Models Markup Language) files with deployment information, and to compute very quickly (in a single pass through the data) predictions for large numbers of observations (for one or more models). PMML files can be generated from practically all modules for predictive data mining (as well as the *Generalized EM & k-Means Cluster Analysis* options). PMML is a XML-based (Extensible Markup Language) industry standard set of syntax convention that is particularly well suited to allow sharing of deployment information in a Client-Server architecture (e.g., via *WebSTATISTICA*).

The *Rapid Deployment of Predictive Models* options provide the fastest, most efficient methods for computing predictions from fully trained models. All models are pre-programmed in generic form in a highly optimized compiled program; the PMML code only supplies the parameter estimates etc. for the fully trained models, to allow the *Rapid Deployment of Predictive Models* program to compute predictions or predicted classifications (or cluster assignments) in a single pass through the data.

In fact, it is very difficult to "beat" the performance (speed of computations) of this tool, even if you were to write your own compiled C++ code, based on the (C, C++, or C#) deployment code generated by the respective models.



Note that the *Rapid Deployment of Predictive Models* module will also automatically compute summary statistics for each model, and if observed values or classifications are available, the program will automatically compute goodness-of-fit indices for participating models, including Gains and Lift charts for one or more models (overlaid lift and gain charts), for binary or multinomial (multi-category) classification problems.

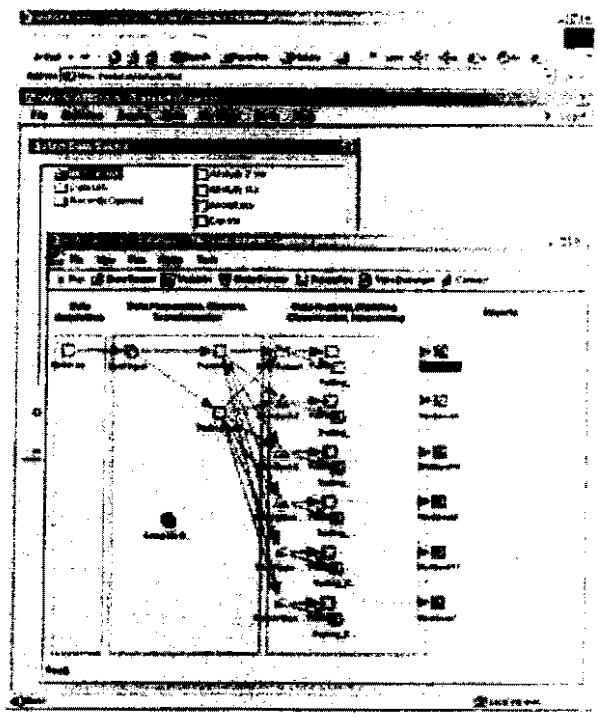
The Client-Server Version of *STATISTICA Data Miner* and Data Mining Via *WebSTATISTICA*

In the desktop version of *STATISTICA Data Miner*, all computations are performed on the local computer, and resources of other computers are used only in the case when the **In-Place Database Processing (IDP)** interface to external databases is established. IDP is a technology that reads data asynchronously directly from remote database servers (using distributed processing if supported by the server), and bypassing the need to "import"

data and create a local copy of the data set. Records of data are retrieved and sent to the *STATISTICA* computer asynchronously by the CPU of the database server, while *STATISTICA* simultaneously processes them using the CPU of the local computer.

The Client-Server Architecture. When a Client-Server version of *STATISTICA Data Miner* is used, the local computer drives only the user interface of *Data Miner*, and all calculations are performed on the server. The Client-Server architecture which uses advanced multithreading and distributed processing technology (see below) and optionally scales to multiple servers which can work in parallel, offers obvious advantages when your data mining projects are large (e.g., computationally intensive or involving processing of extremely large data sets), and thus when they can be offloaded to the servers, freeing your local computer to perform other jobs.

Multithreading, Distributed Processing Technology. Many additional advantages are offered by the specific implementation of the Client-Server architecture in *STATISTICA Data Miner*, which is based on the *WebSTATISTICA* technology. The *WebSTATISTICA* platform is built on advanced distributed processing and multithreading technology to support optimal management of large computational loads. This technology enables rapid processing of even very large and computationally intensive projects, taking full advantage of the multiple CPUs on the server, or even multiple servers working in parallel. The illustration below shows a project running on a quad processor server, along with the server performance monitor demonstrating the full utilization of the resources of all four CPUs executing in the multithreading mode a single, computationally intensive *STATISTICA Data Miner* project.



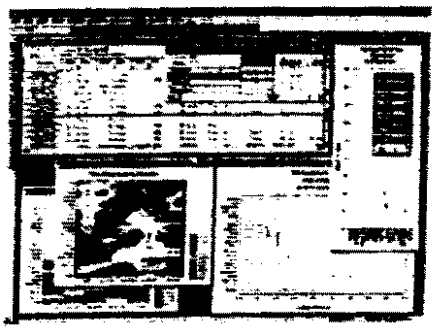
In addition, the *WebSTATISTICA* architecture delivers a platform-independent, Web browser-based user interface, and provides an ultimate, large enterprise-level ability to manage projects or groups of users "across the hall or across continents".

***WebSTATISTICA Data Miner* User Interface.** The *WebSTATISTICA* implementation of *STATISTICA Data Miner* allows users to design, modify, and edit data mining projects on a client machine in a Web browser interface that is essentially identical to that available for the desktop installation.

Therefore, the client side of the application (the "front end") can be run on any computer (even a laptop) as long as it is connected to the Internet. However, the actual computations and other operations performed on the data will remain on the (remote) server with its usually more powerful processors and storage resources (and they will be managed using the optimized, multithreading and distributed processing architecture of the system for maximum performance).

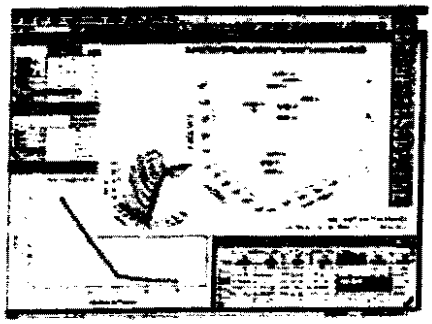
In essence, the user interface aspects of *STATISTICA Data Miner* can be run by one or multiple users from any computer in the world (as long as they are connected to the Internet, even by a slow connection), while the server performs all computations and data operations, enforcing the proper security and access privileges applicable to the respective projects and classes of users, as designed by the network administrator.

Multivariate Techniques in Statistica



CLUSTER ANALYSIS. This module includes a comprehensive implementation of clustering methods (*k-means*, hierarchical clustering, two-way joining). The program can process data from either raw data files or matrices of distance measures. The user can cluster cases, variables, or both based on a wide variety of distance measures (including Euclidean, squared Euclidean, City-block (Manhattan), Chebychev, Power distances, Percent disagreement, and *1-r*) and amalgamation/linkage rules (including single, complete, weighted and unweighted group average or centroid, Ward's method, and others). Matrices of distances can be saved for further analysis with other modules of the *STATISTICA* system. In k-means clustering, the user has full control over the initial cluster centers. Extremely large analysis designs can be processed; for example, hierarchical (tree) joining can analyze matrices with over 1,000 variables, or with over 1 million distances. In addition to the standard cluster analysis output, a comprehensive set of descriptive statistics and extended diagnostics (e.g., the complete amalgamation schedule with cohesion levels in hierarchical clustering, the ANOVA table in *k-means* clustering) is available. Cluster membership data can be appended to the current data file for further processing. Graphics options in the *Cluster Analysis* module include customizable tree diagrams, discrete contour-style two-way joining matrix plots, plots of amalgamation schedules, plots of means in *k-means* clustering, and many others.

Back to Top



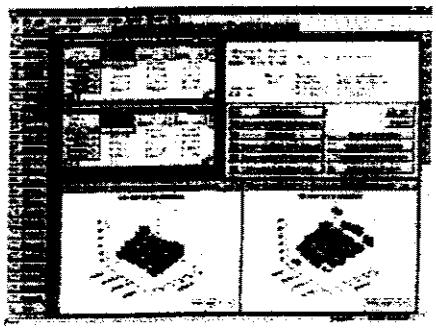
FACTOR ANALYSIS. The *Factor Analysis* module contains a wide range of statistics and options, and provides a comprehensive implementation of factor (and hierarchical factor) analytic techniques with extended diagnostics and a wide variety of analytic and exploratory graphs. It will perform principal components, common, and hierarchical (oblique) factor analysis, and can handle extremely large analysis problems (e.g., with thousands of variables). Confirmatory factor analysis (as well as path analysis) can also be performed via the ***Structural Equation Modeling and Path Analysis (SEPATH)*** module found in the add-on *STATISTICA Advanced Linear/Non-Linear Models*.

Back to Top

PRINCIPAL COMPONENTS & CLASSIFICATION ANALYSIS. *STATISTICA* also includes a designated program for principal components and classification analysis. The output includes eigenvalues (regular, cumulative, relative), factor loadings, factor scores (which can be appended to the input data file, reviewed graphically as icons, and interactively recoded), and a number of more technical statistics and diagnostics. Available rotations include Varimax, Equimax, Quartimax, Biquartimax (either normalized or raw), and Oblique rotations. The factorial space can be plotted and reviewed "slice by slice" in either 2D or 3D scatterplots with labeled variable-points; other integrated graphs include Scree plots, various scatterplots, bar and line graphs, and others. After a factor solution is determined, the user can recalculate (i.e., reconstruct) the correlation matrix from the respective number of factors to evaluate the fit of the factor model. Both raw data files and matrices of correlations can be used as input. Confirmatory factor analysis and other related analyses can be performed with the ***Structural Equation Modeling and Path Analysis (SEPATH)*** module available in *STATISTICA Advanced Linear/Non-Linear Models*, where a designated *Confirmatory Factor Analysis Wizard* will guide you step by step through the process of specifying the model.

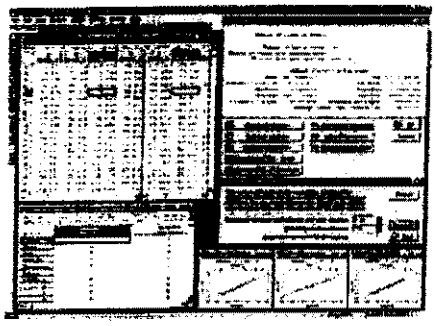
Click here to read a real-life application story using *STATISTICA*'s Principal Components Analysis tools.

Back to Top



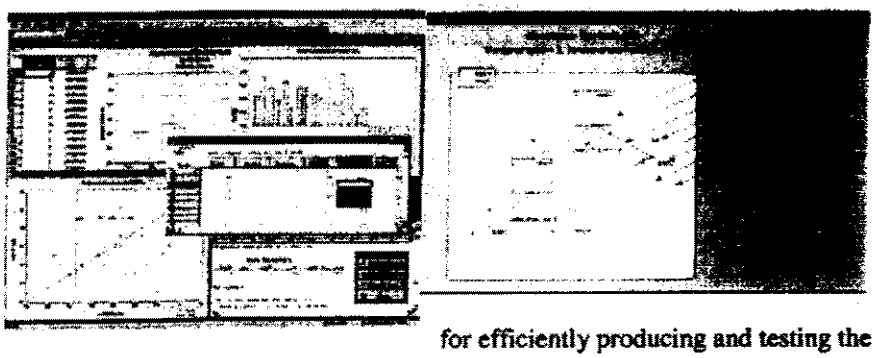
CANONICAL CORRELATION ANALYSIS. This module offers a comprehensive implementation of canonical analysis procedures; it can process raw data files or correlation matrices and it computes all of the standard canonical correlation statistics (including eigenvectors, eigenvalues, redundancy coefficients, canonical weights, loadings, extracted variances, significance tests for each root, etc.) and a number of extended diagnostics. The scores of canonical variates can be computed for each case, appended to the data file, and visualized via integrated icon plots. The *Canonical Analysis* module also includes a variety of integrated graphs (including plots of eigenvalues, canonical correlations, scatterplots of canonical variates, and many others). Note that confirmatory analyses of structural relationships between latent variables can also be performed via the ***SEPATH (Structural Equation Modeling and Path Analysis)*** module in *STATISTICA Advanced Linear/Non-Linear Models*; advanced stepwise and best-subset selection of predictor variables for MANOVA/MANCOVA designs (with multiple dependent variables) is available in the ***General Regression Models (GRM)*** module in *STATISTICA Advanced Linear/Non-Linear Models*.

Back to Top



RELIABILITY/ITEM ANALYSIS. This module includes a comprehensive selection of procedures for the development and evaluation of surveys and questionnaires. As in all other modules of *STATISTICA*, extremely large designs can be analyzed. The user can calculate reliability statistics for all items in a scale, interactively select subsets, or obtain comparisons between subsets of items via the "split-half" (or split-part) method. In a single run, the user can evaluate the reliability of a sum-scale as well as subscales. When interactively deleting items, the new reliability is computed instantly without processing the data file again. The output includes correlation matrices and descriptive statistics for items, Cronbach *alpha*, the standardized *alpha*, the average inter-item correlation, the complete ANOVA table for the scale, the complete set of item-total statistics (including multiple item-total *R*'s), the split-half reliability, and the correlation between the two halves corrected for attenuation. A selection of graphs (including various integrated scatterplots, histograms, line plots and other plots) and a set of interactive *what-if* procedures are provided to aid in the development of scales. For example, the user can calculate the expected reliability after adding a particular number of items to the scale, and can estimate the number of items that would have to be added to the scale in order to achieve a particular reliability. Also, the user can estimate the correlation corrected for attenuation between the current scale and another measure (given the reliability of the current scale).

Back to Top



CLASSIFICATION TREES. *STATISTICA's Classification Trees* module provides a comprehensive implementation of the most recently developed algorithms for efficiently producing and testing the robustness of classification trees (a classification tree is a rule for predicting the class of an object from the values of

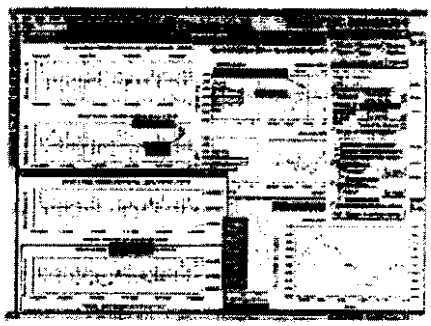
its predictor variables). Advanced methods for tree classifications, including flexible options for model building and interactive tools to explore the trees are also available in the *General Classification and Regression Tree Models (GTrees)* and *General CHAID (Chi-square Automatic Interaction Detection)* models facilities. Classification trees can be produced using categorical predictor variables, ordered predictor variables, or both, and using univariate splits or linear combination splits. Analysis options include performing exhaustive splits (as in *THAID* and *C&RT*) or discriminant-based splits; unbiased variable selection (as in *QUEST*); direct stopping rules (as in *FACT*) or bottom-up pruning (as in *C&RT*); pruning based on misclassification rates or on the deviance function; generalized *Chi-square*, *G-square*, or *Gini*-index goodness of fit measures. Priors and misclassification costs can be specified as equal, estimated from the data, or user-specified. The user can also specify the v value for v-fold cross-validation during tree building, *v* value for *v*-fold cross-validation for error estimation, size of the SE rule, minimum node size before pruning, seeds for random number generation, and *alpha* value for variable selection. Integrated graphics options are provided to explore the input and output data.

See Also: **General Classification and Regression Trees (GTrees) General CHAID (Chi-square Automatic Interaction Detection) Models**

Back to Top

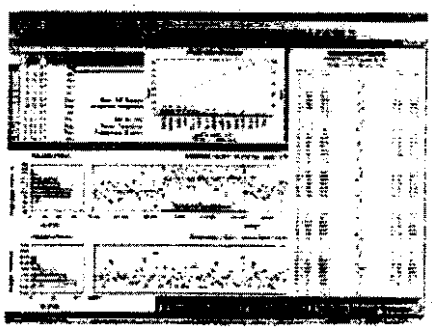
Quality Control

STATISTICA Quality Control Charts features a wide selection of quality control analysis techniques with presentation-quality charts of unmatched versatility and comprehensiveness. It is uniquely ideal for both automated shop-floor quality control systems of all types and levels of complexity (see also ***STATISTICA Enterprise-wide Systems***, as well as sophisticated analytic and quality improvement research. A selection of automation options and user-interface shortcuts simplify routine work and practically all of the numerous graph layout options and specifications can be permanently modified (saved as system default settings or as reusable templates). Finally, *STATISTICA Quality Control Charts* includes powerful and easy to use facilities to custom design entirely new analytic procedures and add them permanently to the application, and those options are particularly useful when quality control analyses need to be integrated into existing data collection/monitoring systems.



STATISTICA Quality Control Charts is compatible with Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Me.

Standard charts. The program offers flexible implementations of Pareto charts, *X-bar* charts, *R* charts, *S* charts, *S-squared* (variance) charts, *C* charts, N_p charts (binomial counts), *P* charts (binomial proportions), *U* charts, *CuSum* (cumulative sum) charts with *V*-masks (the *V*-mask is automatically moved to detect outliers), moving range charts, runs charts (for individual observations), regression control charts, multivariate control charts (Hotelling T^2 charts), *MA* charts (moving average), and *EWMA* charts (exponentially-weighted moving average). These charts may be based on user-specified values or on parameters (e.g., means, ranges, proportions, etc.) computed from the data. Most of the variable control charts can be constructed from single observations (e.g., moving range chart) as well as from samples of multiple observations. Control limits can be specified in terms of multiples of sigma (e.g., 3 * sigma), in terms of normal or non-normal (Johnson-curves) probabilities (e.g., p=.01, .99), or as constant values. For unequal sample sizes, control charts can be computed with variable control limits or based on standardized values. For most charts, multiple sets of specifications can be used in the same chart (e.g., control limits for all new samples can be computed based on a subset of previous samples, etc.). As with all *STATISTICA* graphs, QC charts in *STATISTICA Quality Control Charts* are highly customizable; you can add titles, comments, draw lines or mark regions dynamically anchored to specific scale values, or label the samples with dates, ID codes, etc.

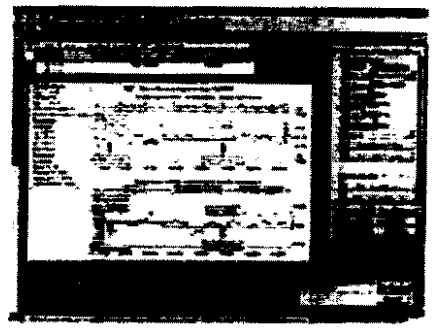


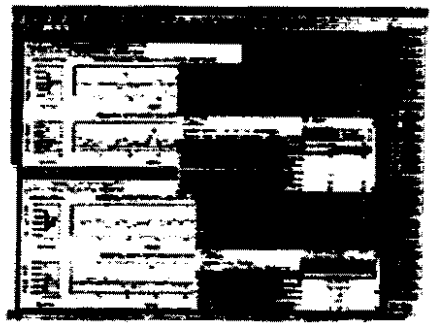
Interactive, analytic brushing and labeling of points. General "intelligent" and comprehensive analytic brushing facilities are available for interactive removal or labeling of outliers (or what-if analyses) in individual charts or sets of charts. The user can select individual samples or groups of samples based on currently specified chart criteria (control limits, runs rules), and exclude them from the computations for the chart (but still show them in the chart), or drop them from the chart altogether. Multiple charts can be set up to use the same sample inclusion/exclusion criteria; in this manner several charts can be simultaneously brushed (e.g., a point excluded from the *X-bar* and *R* chart will simultaneously be excluded from all histograms). The user can also request to plot all individual observations for selected or for all samples.

Assigning causes and actions. The user can assign causes, actions, and/or comments to outliers or any other points in most charts. Labels for causes and actions can be assigned via interactive brushing, or the program can detect and select out-of-control samples.

Flexible, customizable alarm notification system. A comprehensive selection of options are provided

for specifying user-defined criteria that define an out-of-control condition or "noteworthy event" (e.g., runs test violation, individual observation outside specification limits, etc.). The alarm notification system can be customized to trigger various types of "responses" to a particular event. For example, you can set up a system to respond to an out-of-control sample. *STATISTICA Quality Control Charts* will automatically prompt the operator to enter a cause, then launch a *STATISTICA* Visual Basic program to compute various other statistics or invoke an external program, and then run another external program to (for example) call a particular pager number or send an e-mail to the supervising engineers. The alarm notifications setup can be saved in a configuration file (that can be applied to future charts), or used as the default for all future charts.





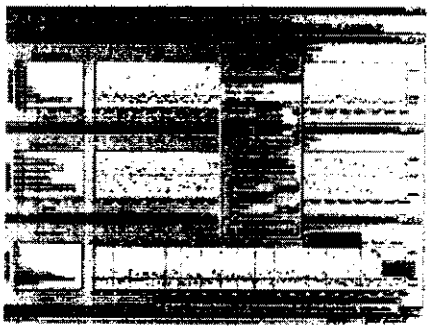
Supervisor and operator mode; password protection. The chart-editing features for shop-floor control charts (including the assignment of causes, actions, brushing, alarm notification, etc.), chart specifications, as well as the input data file can be password-protected, to create a customized operator mode with only limited access to the charts or data. The charts can be saved (e.g., by the supervising engineer), and loaded by the operator in this limited-access operator mode.

Organization of data. For most charts, the data can be organized to accommodate practically all formats in which data are gathered for quality control applications. Samples can be identified by sample identifiers or code numbers, or you can specify a fixed number of measurements per sample (and part, see below).

Short run charts. Most standard variable control charts (*X-bar, R, S, S-squared, MA, EWMA*) and attribute control charts (*C, U, P, Np*) can be used for short production runs (*short run charts* for multiple parts or machines). For short run variable control charts, you can specify nominal target values only (*nominal chart* or *target chart*), or target values and variability values for standardized *short run charts*. Options are provided for sorting sample points in the respective charts, and for plotting them by sample number, by part, or in the order in which the respective samples were taken. Detailed statistics are computed by parts and samples. The respective sample and part identifiers for each measurement can be specified in the data file, and/or you can choose to assign a fixed number of consecutive cases to consecutive samples and/or parts. Note that all chart options and statistics (e.g., process capability and performance indices, runs rules, etc.) commonly reported for standard charts are also available for short run charts.

Chart options and statistics. A wide variety of additional quality control statistics are included. The user can compute the process capability and performance indices (e.g., normal distribution C_{pk}, P_{pk}, etc., non-normal distribution C_{pk}, P_{pk}, etc.), include histograms of the respective quality characteristics, or automatically perform any or all of seven different runs tests (runs rules). The standard variable control charts can be produced as compound multigraphic displays; for example, the *X*-bar and the *R* (or *S*, or *S-squared*) chart will be displayed together with optional corresponding histograms for the respective means, ranges, proportions, etc. also shown in the same chart. Outliers (samples outside the

control limits) or sections of data identified via runs tests are automatically highlighted (marked) in the plots. The user can also add to the plot warning lines, moving average or exponentially-weighted moving average lines, or lines indicating specification ranges.



Non-normal control limits and process capability and performance indices. For variable control charts, in addition to the customary normal distribution based charts and statistics, the program will also compute charts for measurements that are not normally distributed (e.g., are highly skewed). These options are particularly important for situations where the sample sizes are small and where, as a consequence, deviations from normality may lead to greatly inflated or deflated error rates if the customary normal distribution based statistics were used. The program will compute control limits based on the Johnson curves fit to the first four moments of the observed data; user-specified values for the moments can also be supplied. Process capability indices can be computed based on the fitting of Johnson curves as well as Pearson curves. Note that capability indices based on specific distributions can also be computed in *STATISTICA Process Analysis* (an add-on product available from StatSoft, Inc.).

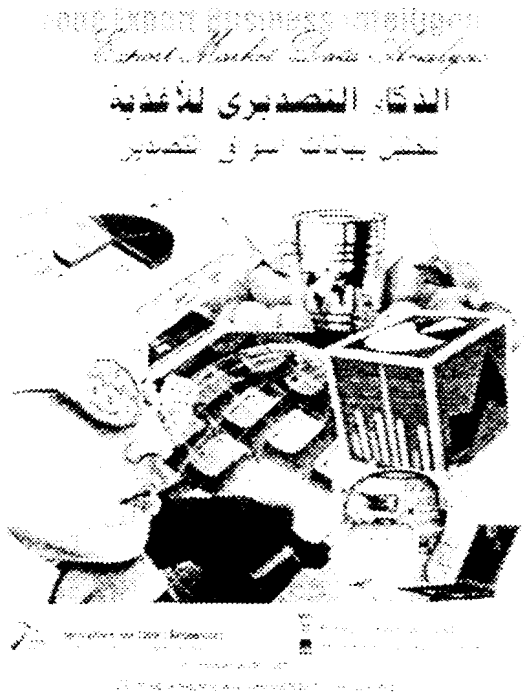
Other plots and Spreadsheets. For most charts (including the *R-chart*), the user may compute and plot the respective operating characteristic curve (OC curve). In addition to the charts, the respective values (plotted in the charts) can also be reviewed via Spreadsheets, allowing the user to examine the precise values of plotted lines and points. Customized (blank) charts can be printed that can later be "filled in" by hand by the quality control engineer. Note that as with all other graphs in *STATISTICA*, the graphs produced by *STATISTICA Quality Control Charts* can be extensively customized and saved for further analysis and/or customization.

Real-time QC systems; external data sources. Most graphs and charts in *STATISTICA Quality Control Charts* can be automatically linked to the data, and updated when the data are updated. To facilitate data transfers powerful (optional) *STATISTICA* applications are available (*SEWSS* and *SEDAS*).

STATISTICA Enterprise-wide Data Analysis System (SEDAS). *SEDAS* is a groupware version of *STATISTICA* fully integrated with a powerful central data warehouse that provides an efficient general interface to enterprise-wide repositories of data and a means for collaborative work (extensive groupware functionality).

STATISTICA Enterprise-wide SPC System (SEWSS). *SEWSS* is an integrated multi-user software package that provides complete statistical process control (SPC) functionality for enterprise installations. *SEWSS* includes a central database, provides all tools necessary to process and manage data from multiple channels, and coordinate the work of multiple operators, QC engineers and supervisors.

SEWSS and *SEDAS* provide very flexible facilities to integrate the procedures in *STATISTICA Quality Control Charts* into your enterprise-wide database, and to design elaborate company-wide quality monitoring systems.



The Art of Market Signals Statistical Analysis Using MicroSoft Excel

CONTENTS

CHAPTER 1 WRITING FORMULAS 25

1.1 The Basics Of Writing Formulae 26

1.2 Tool for using this chapter effectively: Viewing the formula instead of the end result 26

1.2.a The "A1" vs. the "R1C1" style of cell references 28

1.2.b Writing a simple formula that references cells 29

1.3 Types Of References Allowed In A Formula 30

1.3.a Referencing cells from another worksheet 30

1.3.b Referencing a block of cells 30

1.3.c Referencing non-adjacent cells 31

1.3.d Referencing entire rows 32

1.3.e Referencing entire columns 32

1.3.f Referencing corresponding blocks of cells/rows/columns from a set of worksheets 33

CHAPTER 2 COPYING/CUTTING AND PASTING FORMULAE 35

2.1 Copying And Pasting A Formula To Other Cells In The Same Column 36

2.2 Copying And Pasting A Formula To Other Cells In The Same Row 37

2.3 Copying And Pasting A Formula To Other Cells In A Different Row And Column 38

2.4 Controlling Cell Reference Behavior When Copying And Pasting Formulae (Use Of The "$" Key) 39

2.4.a Using the "$" sign in different permutations and computations in a formula 41

2.5 Copying And Pasting Formulas From One Worksheet To Another 42

2.6 Pasting One Formula To Many Cells, Columns, Rows 43

2.7 Pasting Several Formulas To A Symmetric But Larger Range 43

2.8 Defining And Referencing A "Named Range" 43

Adding several named ranges in one step 46

Using a named range 47

2.9 Selecting All Cells With Formulas That Evaluate To A Similar Number Type 48

2.10 Special Paste Options 48

2.10.a Pasting only the formula (but not the formatting and comments) 48

2.10.b Pasting the result of a formula, but not the formula itself 48

2.11 Cutting And Pasting Formulae 49

2.11.a The difference between "copying and pasting" formulas and "cutting and pasting" formulas 49

2.12 Creating A Table Of Formulas Using Data/Table 50

2.13 Saving Time By Writing, Copying And Pasting Formulas On Several Worksheets Simultaneously 50

CHAPTER 3 PASTE SPECIAL 52

3.1 Pasting The Result Of A Formula, But Not The Formula 53

3.2 Other Selective Pasting Options 56

3.2.a Pasting only the formula (but not the formatting and comments) 56

3.2.b Pasting only formats 56

3.2.c Pasting data validation schemes 57

3.2.d Pasting all but the borders 57

3.2.e Pasting comments only 57

3.3 Performing An Algebraic "Operation" When Pasting One Column/Row/Range On To Another 58

3.3.a Multiplying/dividing/subtracting/adding all cells in a range by a number 58

3.3.b Multiplying/dividing the cell values in cells in several "pasted on" columns with the values of the copied range 59

3.4 Switching Rows To Columns 59

CHAPTER 4 INSERTING FUNCTIONS 61

4.1 Basics 61

4.2 A Simple Function 64

4.3 Functions That Need Multiple Range References 67

4.4 Writing A "Function Within A Function" 69

4.5 New Function-Related Features In The XP Version Of Excel 73

Searching for a function 73

4.5.a Enhanced Formula Bar 73

4.5.b Error Checking and Debugging 74

CHAPTER 5 TRACING CELL REFERENCES & DEBUGGING FORMULA ERRORS 76

5.1 Tracing the cell references used in a formula 76

5.2 Tracing the formulas in which a particular cell is referenced 78

5.3 The Auditing Toolbar 79

5.4 Watch window (only available in the XP version of Excel) 80

5.5 Error checking and Formula Evaluator (only available in the XP version of Excel) 81

5.6 Formula Auditing Mode (only available in the XP version of Excel) 84

5.7 Cell-specific Error Checking and Debugging 85

5.8 Error Checking Options 86

CHAPTER 6 FUNCTIONS FOR BASIC STATISTICS 89

6.1 "Averaged" Measures Of Central Tendency 90
- 6.1.a AVERAGE 90
- 6.1.b TRIMMEAN ("Trimmed mean") 91
- 6.1.c HARMEAN ("Harmonic mean") 92
- 6.1.d GEOMEAN ("Geometric mean") 93

6.2 Location Measures Of Central Tendency (Mode, Median) 94
- 6.2.a MEDIAN 95
- 6.2.b MODE 95

6.3 Other Location Parameters (Maximum, Percentiles, Quartiles, Other) 95
- 6.3.a QUARTILE 96
- 6.3.b PERCENTILE 96
- 6.3.c Maximum, Minimum and "Kth Largest" 97
  - MAX ("Maximum value") 97
  - MIN ("Minimum value") 98
  - LARGE 98
  - SMALL 99
- 6.3.d Rank or relative standing of each cell within the range of a series 99
  - PERCENTRANK 99
  - RANK 100

6.4 Measures Of Dispersion (Standard Deviation & Variance) 100
- Sample dispersion: STDEV, VAR 100
- Population dispersion: STDEVP, VARP 101

6.5 Shape Attributes Of The Density Function (Skewness, Kurtosis) 102
- 6.5.a Skewness 102
- 6.5.b Kurtosis 104

6.6 Functions Ending With An "A" Suffix 105

CHAPTER 7 PROBABILITY DENSITY FUNCTIONS AND CONFIDENCE INTERVALS 109

7.1 Probability Density Functions (PDF), Cumulative Density Functions (CDF), and Inverse functions 110
- 7.1.a Probability Density Function (PDF) 110
- 7.1.b Cumulative Density Function (CDF) 111
  - The CDF and Confidence Intervals 112
- 7.1.c Inverse mapping functions 114

7.2 Normal Density Function 115
Symmetry 116
Convenience of using the Normal Density Function 117
Are all large-sample series Normally Distributed? 117
Statistics & Econometrics: Dependence of Methodologies on the assumption of Normality 118
The Standard Normal and its power 119
7.2.a The Probability Density Function (PDF) and Cumulative Density Function (CDF) 119
7.2.b Inverse function 121
7.2.c Confidence Intervals 121
95% Confidence Interval 121
90% Confidence Interval 122

7.3 Standard Normal or Z–Density Function 123
Inverse function 124
Confidence Intervals 124

7.4 T–Density Function 125
Inverse function 126
Confidence Intervals 126
7.4.a One–tailed Confidence Intervals 127
95% Confidence Interval 127
90% Confidence Interval 127

7.5 F–Density Function 129
Inverse function 129
One–tailed Confidence Intervals 130

7.6 Chi-Square Density Function 130
Inverse function 131
One–tailed Confidence Intervals 131

7.7 Other Continuous Density Functions: Beta, Gamma, Exponential, Poisson, Weibull & Fisher 132
7.7.a Beta Density Function 132
Inverse Function 133
Confidence Intervals 134
7.7.b Gamma Density Function 134
Inverse Function 135
Confidence Intervals 136
7.7.c Exponential Density Function 136
7.7.d Fisher Density Function 138
7.7.e Poisson Density Function 138
7.7.f Weibull Density Function 138
7.7.g Discrete probabilities— Binomial, Hypergeometric & Negative Binomial 139
Binomial Density Function 139
Hypergeometric Density Function 139
Negative Binomial 139

7.8 List of Density Function 140

7.9 Some Inverse Function 141

CHAPTER 8 OTHER MATHEMATICS & STATISTICS FUNCTIONS 144

8.1 Counting and summing 145
COUNT function 145
COUNTA function also counts cells with logical or text values 147
COUNTBLANK function counts the number of empty cells in the range reference 148
SUM function 148
PRODUCT function 149
SUMPRODUCT function 149

8.2 The "If" counting and summing functions: Statistical functions with logical conditions 150
SUMIF function 150
COUNTIF function 151

8.3 Transformations (log, exponential, absolute, sum, etc) 153
Standardizing a series that follows a Normal Density Function 155

8.4 Deviations from the Mean 156
DEVSQ 156
AVEDEV 156

8.5 Cross series relations 157
8.5.a Covariance and correlation functions 157
8.5.b Sum of Squares 157
SUMXMY2 function 158
SUMX2MY2 function 158

CHAPTER 9 ADD-INS: ENHANCING EXCEL 161

9.1 Add-Ins: Introduction 161
9.1.a What can an Add-In do? 162
9.1.b Why use an Add-In? 162

9.2 Add–ins installed with Excel 162

9.3 Other Add-Ins 163

9.4 The Statistics Add-In 163
9.4.a Choosing the Add-Ins 163

CHAPTER 10 STATISTICS TOOLS 169

10.1 Descriptive statistics 170

10.2 Rank and Percentile 175
Interpreting the output: 177

10.3 Bivariate relations— correlation, covariance 178
Correlation analysis 178
Interpreting the output 179
10.3.a Covariance tool and formula 180

CHAPTER 11 HYPOTHESIS TESTING 183

11.1 Z-testing for population means when population variances are known 184
Interpreting the output 189

11.2 T-testing means when the two samples are from distinct groups 189
11.2.a The pretest— F-testing for equality in variances 189
Interpreting the output 191
11.2.b T-test: Two–Sample Assuming Unequal Variances 193
Interpreting the output 196
11.2.c T-test: Two–Sample Assuming Equal Variances 199

11.3 Paired Sample T-tests 199

11.4 ANOVA 205
Interpreting the output 207

CHAPTER 12 REGRESSION 211

12.1 Assumptions Underlying Regression Models 211
12.1.a Assumption 1: The relationship between any one independent series and the dependent series can be captured by a straight line in a 2–axis graph 213
12.1.b Assumption 2: The independent variables do not change if the sampling is replicated 213
12.1.c Assumption 3: The sample size must be greater than the number of independent variables (N should be greater than K–1) 214
12.1.d Assumption 4: Not all the values of any one independent series can be the same 215
12.1.e Assumption 5: The residual or disturbance error terms follow several rules 216
Assumption 5a: The mean/average or expected value of the disturbance equals zero 216
Assumption 5b: The disturbance terms all have the same variance 216
Assumption 5c: A disturbance term for one observation should have no relation with the disturbance terms for other observations or with any of the independent variables 217
Assumption 5d: There is no specification bias 217
Assumption 5e: The disturbance terms have a Normal Density Function 218
12.1.f Assumption 6: There are no strong linear relationships among the independent variables 218

12.2 Conducting the Regression 219

12.3 Brief guideline for interpreting regression output 222

12.4 Breakdown of classical assumptions: validation and correction 226

CHAPTER 13 OTHER TOOLS FOR STATISTICS 229

13.1 Sampling analysis 229

13.2 Random Number Generation 231

13.3 Time series 234
Exponential Smoothing 234
Moving Average analysis 235

CHAPTER 14 THE SOLVER TOOL FOR CONSTRAINED LINEAR OPTIMIZATION 239

14.1 Defining the objective function (Choosing the optimization criterion) 239

14.2 Adding constraints 243

14.3 Choosing Algorithm Options 244
Running the Solver 245

INDEX 245

STATISTICS PROCEDURES

Three chapters teach statistics functions including the use of Excel functions for building Confidence Intervals and conducting Hypothesis Testing for several types of distributions. The design of hypothesis tests and the intermediate step of demarcating critical regions are taught lucidly.

It seems that Microsoft has taken pains to "hide" some of the most powerful tools in Excel. These "hidden" tools are called "Add-Ins." These tools work on top of Excel, extending the power and abilities of Excel. Many Add-Ins are available for specific types of analysis like Risk Analysis. I show how to use three Add-Ins that install with Excel.

BASICS

The fundamental operations in Excel are taught in *Volume 1: Excel For Beginners*, *Volume 2: Charting in Excel*, and *Volume 3: Excel– Beyond The Basics*

FUNCTIONS

I teach the writing of formulas and associated topics in *Volume 3: Excel– Beyond The Basics*. I show, in a step-by-step exposition, the proper way for writing cell references in a formula. The book describe tricks for copying/cutting and pasting in several examples. In addition, I discuss special pasting options.

Finally, different types of functions are classified under logical categories and discussed within the optimal category. The categories include financial, Statistical, Text, Information, Logical, and "Smart" Logical.

MANAGING & TABULATING DATA

Excel has extremely powerful data entry, data management, and tabulation tools. The combination of tools provide almost database like power to Excel. Unfortunately, the poor quality of the menu layout and the help preclude the possibility of the user self-learning these features. These features are taught in *Volume 4: Managing & Tabulating Data in Excel*

CHARTING

Please refer to book two in this series. The book title is *Charting in Excel.*

Sample data

Most of the tutorials use publicly available data from the International labor Organization (ILO). I used a simple data set with only a few columns and observations. All the sample data files are included in the zipped file.

The samples for functions use several small data sets that are more suited to illustrating the power and usefulness of the functions.

I have not included the data set for conducting statistical procedures. This is intentional; often, readers fail to internalize the few key concepts of hypothesis testing because they do not subject themselves to a "sink-or-swim" inference-drawing thinking and imbibing process when interpreting the results of statistical procedures.

CHAPTER 1

WRITING FORMULAS

This chapter discusses the following topics:

— THE BASICS OF WRITING FORMULAE

— TOOL FOR USING THIS CHAPTER EFFECTIVELY: VIEWING THE FORMULA INSTEAD OF THE END RESULT

— The A1 VS THE R1C1 STYLE OF CELL REFERENCES

— TYPES OF REFERENCES ALLOWED IN A FORMULA

— REFERENCING CELLS FROM ANOTHER WORKSHEET

— REFERENCING A BLOCK OF CELLS

— REFERENCING NON–ADJACENT CELLS

— REFERENCING ENTIRE ROWS

— REFERENCING ENTIRE COLUMNS

— REFERENCING CORRESPONDING BLOCKS OF CELLS/ROWS/COLUMNS FROM A SET OF WORKSHEETS

The most important functionality offered by a spreadsheet application is the ease and flexibility of writing formulae. In this chapter, I start by showing how to write simple formula and then build up the level of complexity of the formulae.

Within the sections of this chapter, you will find tips and notes on commonly encountered problems or issues in formula writing.

1.1 THE BASICS OF WRITING FORMULAE

This section teaches the basics of writing functions.

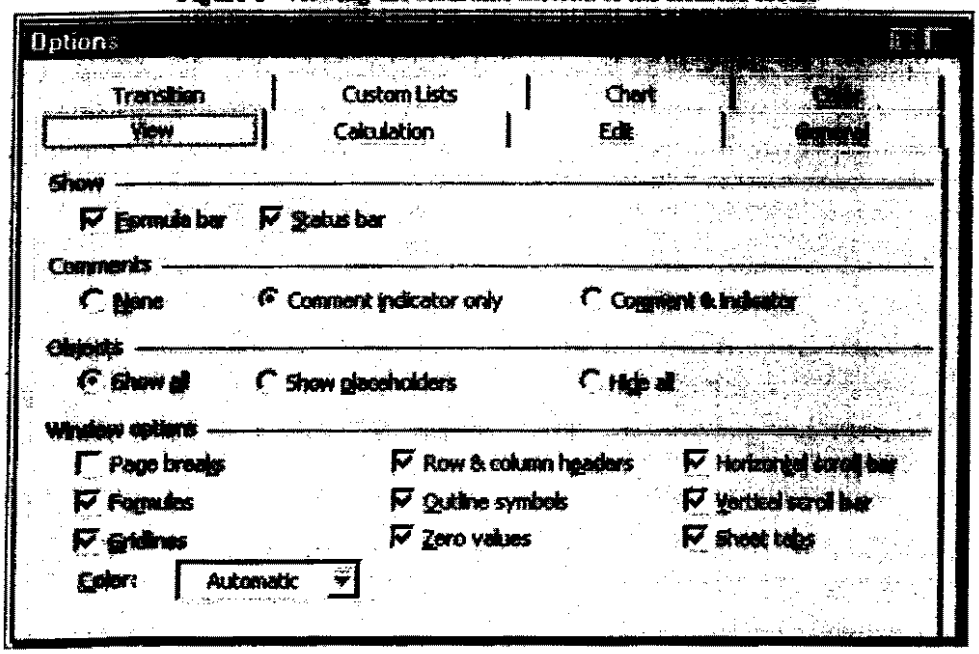
1.2 TOOL FOR USING THIS CHAPTER EFFECTIVELY: VIEWING THE FORMULA INSTEAD OF THE END RESULT

For ease of understanding this chapter, I suggest you use a viewing option that shows, in each cell on a worksheet, the formula instead of the result. Follow the menu path TOOLS/OPTIONS/VIEW. In the area "Window Options" select the option "Formulas" as shown in Figure 1.

Execute the dialog by clicking on the button OK. Go back to the worksheet. The formula will be shown instead of the calculated value.

Eventually you will want to return to the default of seeing the results instead of the formula. Deselect "formula" in the area "Windows Options" in TOOLS/OPTIONS/VIEW.

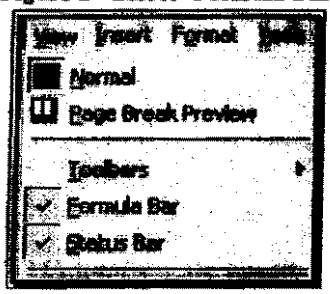
Figure 1: Viewing the formulas instead of the formula result



The effect is only cosmetic; the results will not change. As you shall see later, what you have just done will facilitate the understanding of functions.

In addition, leave the option VIEW/ FORMULA BAR selected as shown in Figure 2.

Figure 2: Select "Formula Bar"



1.2.A THE "A1" VS. THE "R1C1" STYLE OF CELL REFERENCES

The next figure shows a simple formula. The formula is written into cell G15. The formula multiplies the values inside cells F8 and F6.

Figure 3: A1-style cell referencing

=F8*F6

This style of referencing is called the "A1" style or "absolute" referencing. The exact location of the referenced cells is written. (The cells are those in the 6th and 8th rows of column F.) One typically works with this style.

However, there is another style for referencing the cells in a formula. This style is called the "R1C1" style or "relative" referencing. The same formula as in the previous figure but in R1C1 style is shown in the next figure.

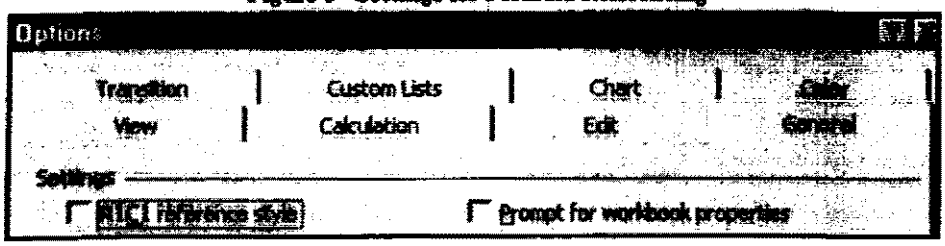
Figure 4: The same formula as in the previous figure, but in R1C1 (Offset) style cell referencing while the previous figure showed A1 (Absolute) style cell referencing

=R[-7]C[-1]*R[-9]C[-1]

Does not this formula look different? This style uses relative referencing. So, the first cell (F8) is referenced relative to its position in reference to the cell that contains the formula (cell G15). Row 8 is 7 rows below row 15 and column F is 1 column before column G. Therefore, the cell reference is "minus seven rows, minus 1 column" or "R[— 7]C[— 1]."

If you see a file or worksheet with such relative referencing, you can switch all the formulas back to absolute "A1" style referencing by going to TOOLS/OPTIONS/GENERAL and deselecting the option "R1C1 reference style."

Figure 5: Settings for Formula Referencing



1.2.B WRITING A SIMPLE FORMULA THAT REFERENCES CELLS

Open the sample file "File3.xls" and choose the worksheet "main." Assume you want to write add the values in cells C223[1] and D223 (that is, to calculate "C223 + D223") and place the result into cell F223.

Click on cell F223. Key-in "="and then write the formula by clicking on the cell C223, typing in "+" then clicking on cell "D223."

Figure 6: Writing a formula

=C223+D223

After writing in the formula, press the key ENTER. The cell F223 will contain the result for the formula contained in it.

Figure 7: The result is shown in the cell on which you wrote the formula

20,167,000

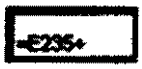
[1] Cell C223 is the cell in column C and row 223.

1.3 TYPES OF REFERENCES ALLOWED IN A FORMULA

1.3.A REFERENCING CELLS FROM ANOTHER WORKSHEET

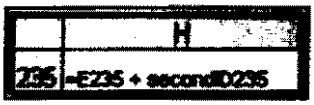
You can reference cells from another worksheet. Choose cell H235 on the worksheet "main." In the chosen cell, type the text shown in the next figure. (Do not press the ENTER key; the formula is incomplete and you will get an error message if you press ENTER.)

Figure 8: Writing or choosing the reference to the first referenced range



Then select the worksheet "second" and click on cell D235. Now press the ENTER key. The formula in cell H235 of worksheet "main" references the cell D235 from the worksheet "second". The next figure illustrates this.

Figure 9: Writing or choosing the reference to the second referenced range which is not on the worksheet on which you are writing the formula



In this formula, the part "second!" informs Excel that the range referenced is from the sheet "second.

1.3.B REFERENCING A BLOCK OF CELLS

Select the worksheet "main." Choose cell H236. In the chosen cell, type the text shown in the next figure.

Figure 10: This formula requires a block of cells as a reference

	H
236	=SUM(

Use the mouse to highlight the block of cells "E223 to E235." Type in a closing parenthesis and press the ENTER key. The resulting function is shown in the next figure.

Figure 11: Formula with a block of cells as the reference

	H
236	=SUM(E223:E235)

1.3.C REFERENCING NON-ADJACENT CELLS

Choose cell H237. Click in the cell and type the text shown in the next figure.

Figure 12: The core function is typed first

	H
237	=SUM(

As in the previous example, choose cells E223 to E235 by highlighting them— the formula should like the one shown in the next figure.

Figure 13: The first block of cells is referenced

	H
237	=SUM(E223:E235

Type a comma. The resulting formula should look like that shown in the next figure.

Figure 14: Getting the formula ready for the second block of cells

	H
237	=SUM(E223:E235,

Highlight the block of cells "E210 to E222." Key-in a closing parenthesis and press the ENTER key.

Figure 15: The formula with references to two non-adjacent blocks of cells

	H
237	=SUM(E223:E235,E210:E222)

1.3.D REFERENCING ENTIRE ROWS

Choose cell H238. In this cell, type the text shown in the next figure.

Using the mouse, highlight the rows 197 to 209. Type in a closing parenthesis and press the ENTER key. The resulting formula is shown in the next figure.

Figure 16: Referencing entire rows

	H
238	=SUM(197:209)

1.3.E REFERENCING ENTIRE COLUMNS

Choose cell H239. In this cell, type the text shown in the next figure. Using the mouse, highlight the columns C and D. Key-in a closing parenthesis and press the ENTER key.

Figure 17: Referencing entire columns

	H
239	=SUM(C:D)

1.3.F REFERENCING CORRESPONDING BLOCKS OF CELLS/ROWS/COLUMNS FROM A SET OF WORKSHEETS

Assume you have a workbook with six worksheets on similar data from six clients. You want to sum cells "C4 to F56" across all six worksheets.

One way to do this would be to create a formula in each worksheet to sum for that worksheet's data and then a formula to add the results of the other six formulae.

Another way is using "3–D references." The row and column make the first two dimensions; the worksheet set is the third dimension. You can use only one formula that references all six worksheets that the relevant cells within them.

While typing the formula,

- Type the "="sign,
- Write the formula (for example, "Sum"),
- Place an opening parenthesis "(," then
- Select the six worksheets by clicking at the name tab of the first one and then pressing down SHIFT and clicking on the name tab of the sixth worksheet, and then
- Highlight the relevant cell range on any one of them,
- Type in the closing parenthesis ")"
- And press the ENTER key to get the formula

=SUM(Sheet1:Sheet6!"C4:F56")

CHAPTER 2

COPYING/CUTTING AND PASTING FORMULAE

This chapter teaches the following topics:

— COPYING AND PASTING A FORMULA TO OTHER CELLS IN THE SAME COLUMN

— COPYING AND PASTING A FORMULA TO OTHER CELLS IN THE SAME ROW

— COPYING AND PASTING A FORMULA TO OTHER CELLS IN A DIFFERENT ROW AND COLUMN

— CONTROLLING CELL REFERENCE BEHAVIOR WHEN COPYING AND PASTING FORMULAE (USE OF THE "$" KEY)

— USING THE "$" SIGN IN DIFFERENT PERMUTATIONS AND COMPUTATIONS IN A FORMULA.

— COPYING AND PASTING FORMULAS FROM ONE WORKSHEET TO ANOTHER

— SPECIAL PASTE OPTIONS

— PASTING ONLY THE FORMULA (BUT NOT THE FORMATTING AND COMMENTS)

— PASTING THE RESULT OF A FORMULA, BUT NOT THE FORMULA ITSELF

— CUTTING AND PASTING FORMULAE

— THE DIFFERENCE BETWEEN "COPYING AND PASTING" FORMULAS AND "CUTTING AND PASTING" FORMULAS

— SAVING TIME BY WRITING, COPYING AND PASTING FORMULAS ON SEVERAL WORKSHEETS SIMULTANEOUSLY

2.1 COPYING AND PASTING A FORMULA TO OTHER CELLS IN THE SAME COLUMN

Often one wants to write analogous formulae for several cases. For example, assume you want to write a formula analogous to the formula in F223 into each of the cells F224 to F235[2]. The quick way to do this is to:

— Click on the "copied from" cell F223.

— Select the option EDIT/COPY. (The menu can also be accessed by right-clicking on the mouse or by clicking on the COPY icon.)

— Highlight the "pasted on" cells F224 to F235 and

— Choose the menu option EDIT/PASTE. (The menu can also be accessed by right-clicking on the mouse or by clicking on the PASTE icon.)

— Press the ENTER key.

— The formula is pasted onto the cells F224 to F235 and the cell

[2] The formula in F223 adds the values in cells that are 3 and 2 columns to the left (that is, cells in columns in C and D.)

52

references within each formula are adjusted[3] for the location difference between the "pasted on" cells and the "copied from" cell.

Figure 18: Pasting a formula

	C	D	E	F
223	9133000	11034000	15223000	=C223+D223
224	1626000	1852000	2618000	=C224+D224
225	1417000	1600000	2255000	=C225+D225
226	1202000	1369000	1802000	=C226+D226
227	976000	1176000	1550000	=C227+D227
228	607000	951000	1339000	=C228+D228
229	464000	589000	1124000	=C229+D229
230	398000	447000	897000	=C230+D230
231	331000	375000	544000	=C231+D231
232	279000	307000	400000	=C232+D232
233	221000	250000	319000	=C233+D233

2.2 COPYING AND PASTING A FORMULA TO OTHER CELLS IN THE SAME ROW

Select the range F223— F235 (which you just created in the previous sub-section). Select the option EDIT/COPY. Choose the range G223— G235 (that is, one column to the right) and choose the menu option EDIT/PASTE. Now click on any cell in the range G223— G235 and see how the column reference has adjusted automatically. The formula in

[3] The formula in the "copied cell" F223 is "C223 + D223" while the formula in the "pasted on" cell F225 is "C225 + D225." (Click on cell F225 to confirm this.) The cell F225 is two rows below the cell F223, and the copying-and-pasting process accounts for that.

53

G223 is "D223 + E223" while the formula in F223 was "C223 + D223".

The next figure illustrates this. Because you pasted one column to the right, the cell references automatically shifted one column to the right. So:

— The reference "C" became "D," and

— The reference "D" became "E."

Figure 19: Cell reference changes when a formula is copied and pasted

	F	G
223	=C223+D223	=D223+E223
224	=C224+D224	=D224+E224
225	=C225+D225	=D225+E225
226	=C226+D226	=D226+E226
227	=C227+D227	=D227+E227
228	=C228+D228	=D228+E228
229	=C229+D229	=D229+E229
230	=C230+D230	=D230+E230

The examples in 2.1 on page 36 and 2.2 on page 37 show the use of "Copy and Paste" to quickly replicate formula in a manner that maintains referential parallelism.

2.3 COPYING AND PASTING A FORMULA TO OTHER CELLS IN A DIFFERENT ROW AND COLUMN

Select the cell F223. Select the option EDIT/COPY. Choose the range H224 (that is, two columns to the right and one row down from the copied cell) and choose the menu option EDIT/PASTE. Observe how the column and row references have changed automatically— the formula in H224 is

57

"E224 + F224" while the formula in F223 was "C223 + D223".

The next figure illustrates this. Because you pasted two columns to the right and one row down, the cell references automatically shifted two columns to the right and one row down. So:

— The reference "C" became "E" (that is, two columns to the right)

— The reference "D" became "F" (that is, two columns to the right)

— The references "223" became "224" (that is, one row down)

Figure 20: Copying and pasting a formula

F	G	H
=C223+D223	=D223+E223	
=C224+D224	=D224+E224	=E224+F224

2.4 CONTROLLING CELL REFERENCE BEHAVIOR WHEN COPYING AND PASTING FORMULAE (USE OF THE "$" KEY)

The use of the dollar key "$" (typed by holding down SHIFT and choosing the key "4") allows you to have control over the change of cell references in the "Copy and Paste" process. The use of this feature is best shown with some examples.

— The steps in copy and pasting a formula from one range to another:

— Click on the "copied from" cell F223.

— Select the option EDIT/COPY. (The menu can also be accessed by right-clicking on the mouse or by clicking on the COPY icon.)

55

— Choose the "pasted on" cell F219 by clicking on it, and

— Select the menu option EDIT/PASTE. (The menu can also be accessed by right-clicking on the mouse or by clicking on the PASTE icon.)

— Press the ENTER key.

— The formula "C219 + D219" will be pasted onto cell F219. (For a pictorial reproduction of this, see Figure 21.)

Figure 21: The "pasted-on" cell

	F
219	=C219+D219

Change the formula by typing the dollar signs as shown Figure 22.

Figure 22: Inserting dollar signs in order to influence cell referencing

	F
219	=$C$219+D219

Copy cell F219. Paste into G220 (that is, one column to the right and one row down). The dollar signs will ensure that the cell reference is not adjusted for the row or column differential for the parts of the formula that have the dollar sign before them[4]— see the formula in cell F220 (reproduced in Figure 23).

[4] In this example, the parts are the "C" reference and "219" reference in "$C$219" part of the formula.

56

Figure 23: The "copied-from" and "pasted-on" cells with the use of the dollar sign

	F	G
219	=$C$219+D219	
220		=$C$219+E220

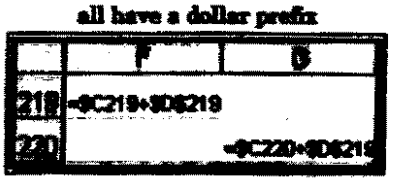
For the parts of the cell that do not have the dollar sign before them, the cell references adjust to maintain referential integrity[5].

2.4.A USING THE "$" SIGN IN DIFFERENT PERMUTATIONS AND COMPUTATIONS IN A FORMULA

The dollar sign in the "copied from" cell	The copy & paste action	The cell references in the "pasted on" cell depend on the location of the dollar signs in the formula in the original, "copied from" cell
Reference behavior with a dollar sign before one of the column references Original cell: F219 = $C219 + D219	Copy F219 and paste into G220.	Figure: 24: Only the reference to "C" does not adjust because only "C" has a dollar prefix F: =$C219+D219 G220: =$C220+E220
Reference behavior with a dollar sign before one of the row references Original cell: F219 = C$219 + D219	Copy F219 and paste into G220.	Figure 25: Only the reference to "219" (in the formula part "C$219") does not adjust because only that "219" has a dollar prefix F: =C$219+D219 G220: =C$219+E220

[5] The part "D219" adjusts to "E220" to adjust for the fact that the "pasted on" cell is one column to the right (so "D→E") and one row below (so "219→220".)

51

The dollar sign in the "copied from" cell	The copy & paste action	The cell references in the "pasted on" cell depend on the location of the dollar signs in the formula in the original, "copied from" cell
Reference behavior with a dollar sign before all but one of the row/column references Original cell: F219 = $C219 + $D$219	Copy F219 and paste into G220.	Figure 26: the references to "C," "D" and to "219" (in the formula part "$D$219") do not adjust because they all have a dollar prefix 
Original cell: F219 = $C$219 + $D$219	Copy F219 and paste into G220.	Try it... G220 = $C$219 + $D$219
Original cell: F219 = $C219 + $D219	Copy F219 and paste into G220.	Try it... G220 = $C220 + $D220
Original cell: F219 = C219 + $D$219	Copy F219 and paste into G220.	Try it... G220 = D220 + $D$219

2.5 COPYING AND PASTING FORMULAS FROM ONE WORKSHEET TO ANOTHER

The worksheet "second" in the sample data file has the same data as the worksheet you are currently on ("main.") In the worksheet main, select the cell F219 and choose the menu option EDIT/COPY. Select the worksheet "second" and paste the formula into cell F219. Notice that the formula is duplicated.

2.6 PASTING ONE FORMULA TO MANY CELLS, COLUMNS, ROWS

Copy the formula. Select the range for pasting and paste or "Paste Special" the formula.

2.7 PASTING SEVERAL FORMULAS TO A SYMMETRIC BUT LARGER RANGE

Assume you have different formulas in cells G2, H2, and I2. You want to paste the formula:

— In G2 to G3:G289

— In H2 to H3:H289

— In I2 to I3:I289

Select the range G2:I2. Pick the menu option EDIT/COPY. Highlight the range G3:I289. (Shortcut: select G3. Scroll down to I289 without touching the sheet. Depress the SHIFT key and click on cell I289.) Pick the menu option EDIT/PASTE.

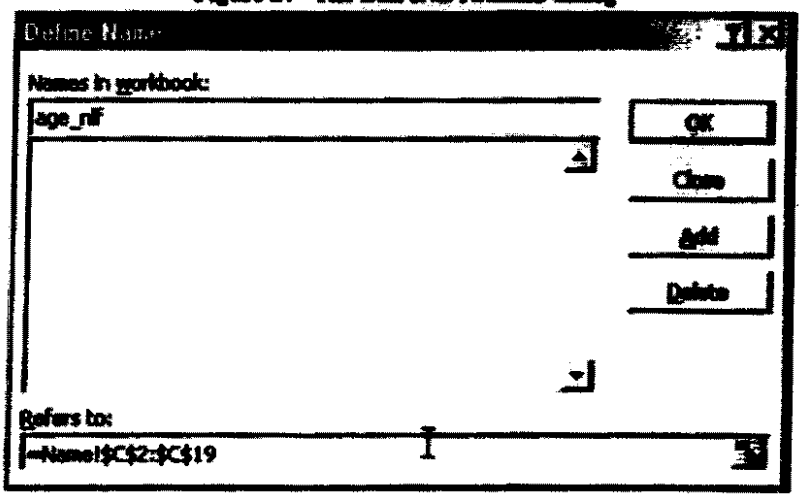
2.8 DEFINING AND REFERENCING A "NAMED RANGE"

You can use range names as references instead of exact cell references. Named ranges are easier to use if the names chosen are explanatory.

First, you have to define named ranges. This process involves informing Excel that the name, for example, "age_nlf," refers to the range "C2:C19."

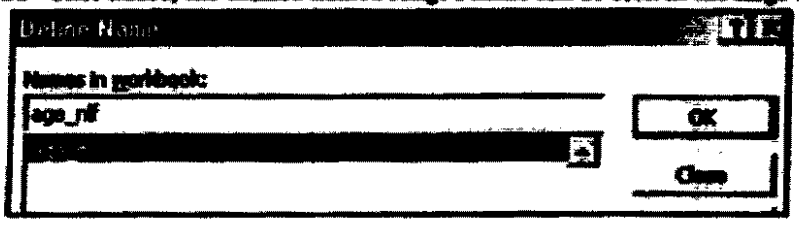
Pick the menu option "INSERT/NAME/DEFINE." The dialog (user-input form) that opens is shown in the next figure. Type the name of the range into the text-box "Names in workbook" and the "Cell References" in the box "Refers to:" See the next figure for an example.

Figure 27: The DEFINE NAMES dialog



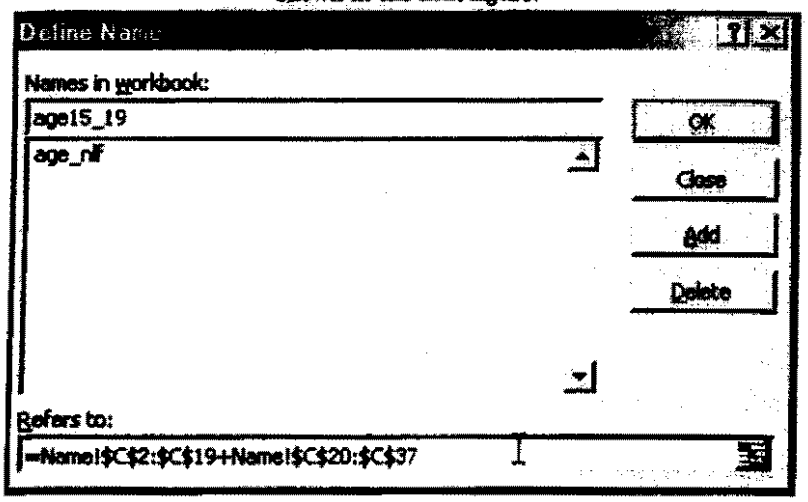
Click on the button "Add." The named range is defined. The name of a defined range is displayed in the large text-box in the dialog. The next figure illustrates this text.

Figure 28: Once added, the defined named range's name can be seen in the large text-box



Several named ranges can be defined. A named range can represent multiple blocks of cells.

Figure 29: Defining a second named range. On clicking "Add," the named range is defined, as shown in the next figure.



You can view the ranges represent by any name. Just click on the name in the central text-box and the range represented by the name will be displayed in the bottom box.

Figure 30: Two named ranges are defined

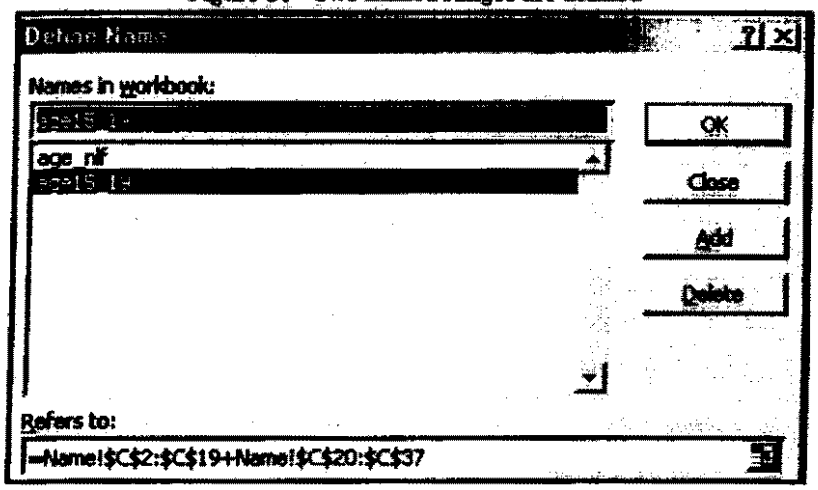
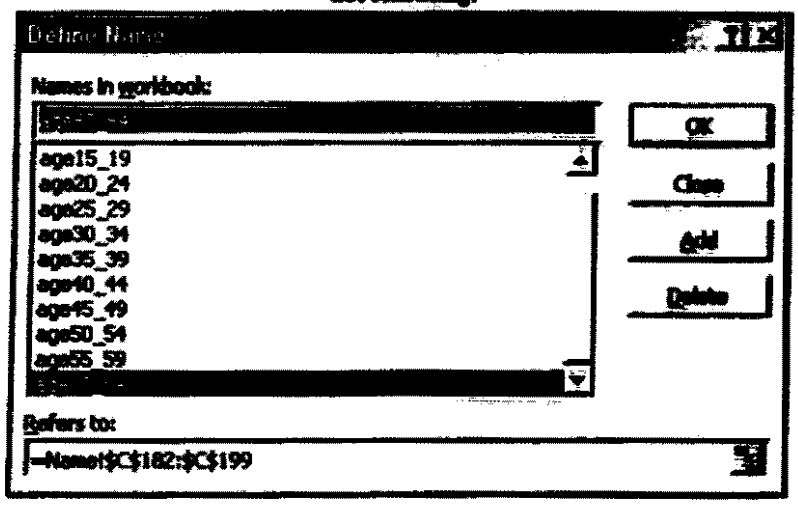


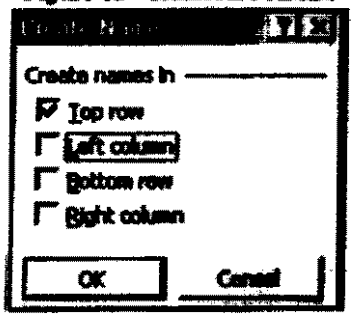
Figure 31: You can define many ranges. Just make sure that the names are explanatory and not confusing.



Adding several named ranges in one step

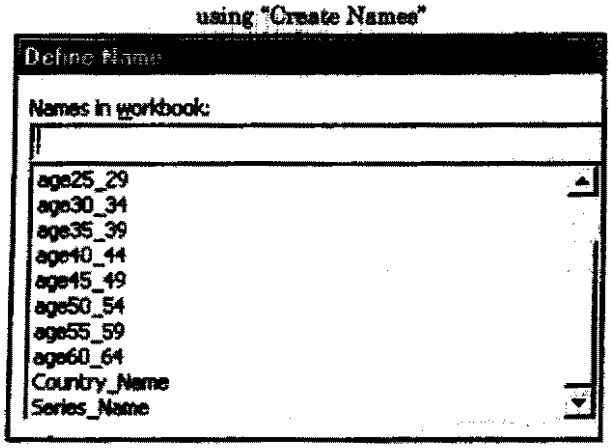
If the first/last row/column in your ranges has the labels for the range, then you can define names for all the ranges using the menu option INSERT/NAMES/CREATE. The dialog is reproduced in the next figure.

Figure 32: CREATE NAMES



In our sample data set, I selected columns "A" and "B" and created the names from the labels in the first row.

Figure 33: The named ranges "Country_Name," and "Series_Name" were defined in one step using "Create Names"



Using a named range

Named ranges are typically used to make formulas easier to read. The named ranges could also be used in other procedures

Assume you want to sum several of the ranges defined above. One way to sum them would be to select them one-by-one from the worksheet.

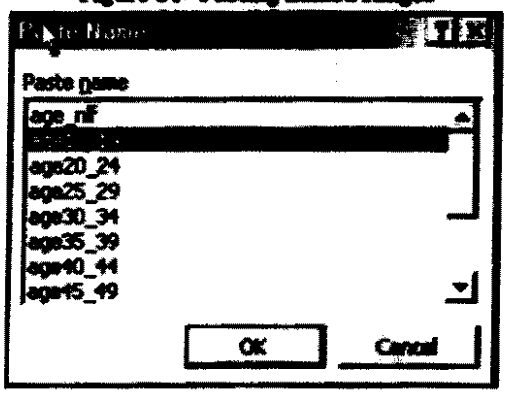
=SUM(

Another way is to use the menu option INSERT/NAME/PASTE to select and paste the names of the ranges. The names are explanatory and reduce the chances of errors in cell referencing.

A reference to the named range is pasted onto the formula as shown below.

=SUM(age15_19)

Figure 34: Pasting named ranges



2.9 SELECTING ALL CELLS WITH FORMULAS THAT EVALUATE TO A SIMILAR NUMBER TYPE

Volume 3: Excel– Beyond The Basics.

2.10 SPECIAL PASTE OPTIONS

2.10.A PASTING ONLY THE FORMULA (BUT NOT THE FORMATTING AND COMMENTS)

Refer to page 56 in chapter 3.

2.10.B PASTING THE RESULT OF A FORMULA, BUT NOT THE FORMULA ITSELF

Refer to page 53 in chapter 3.

2.11 CUTTING AND PASTING FORMULAE

2.11.A THE DIFFERENCE BETWEEN "COPYING AND PASTING" FORMULAS AND "CUTTING AND PASTING" FORMULAS

Click on cell F223, select the option EDIT/CUT, click on cell H224 and choose the menu option EDIT/PASTE. The formula in the "pasted on" cell is the same as was in the "cut from" cell. (The formula "=C223 + D223.") Therefore, there is no change in the cell references after cutting–and–pasting. While copy–and–paste automatically adjusts for cell reference differentials, cut–and–paste does not.

If you had used copy and paste, the formula in H224 would be "=D224 + E224."

Figure 35: Cut from cell F223

	F	G	H
223	=C223+D223	=D223+E223	
224	=C224+D224	=D224+E224	

Figure 36: Paste into cell H223. Note that the cell references do not adjust.

	F	G	H
223		=D223+E223	
224	=C224+D224	=D224+E224	=C223+D223

After doing this, select the option EDIT/UNDO because I want to maintain the formulas in F223— F235 (and not because it is required for a cut and paste operation).

65

2.12 CREATING A TABLE OF FORMULAS USING DATA/TABLE

The menu option DATA/TABLE supposedly offers a tool for creating an X-Y table of formula results. However, the method needs so much data arrangement that it is no better than using a simple copy and paste operation on cells!

2.13 SAVING TIME BY WRITING, COPYING AND PASTING FORMULAS ON SEVERAL WORKSHEETS SIMULTANEOUSLY

Refer to *Volume 3: Excel– Beyond The Basics* to learn how to work with multiple worksheets. The section will request you to follow our example of writing a formula for several worksheets together.

CHAPTER 3

PASTE SPECIAL

This chapter teaches the following topics:

— PASTING THE RESULT OF A FORMULA, BUT NOT THE FORMULA

— OTHER SELECTIVE PASTING OPTIONS

— PASTING ONLY THE FORMULA (BUT NOT THE FORMATTING AND COMMENTS)

— PASTING ONLY FORMATS

— PASTING DATA VALIDATION SCHEMES

— PASTING ALL BUT THE BORDERS

— PASTING COMMENTS ONLY

— PERFORMING AN ALGEBRAIC "OPERATION" WHEN PASTING ONE COLUMN/ROW/RANGE ON TO ANOTHER

— MULTIPLYING/DIVIDING/SUBTRACTING/ADDING ALL CELLS IN A RANGE BY A NUMBER

— MULTIPLYING/DIVIDING THE CELL VALUES IN CELLS IN SEVERAL "PASTED ON" COLUMNS WITH THE VALUES OF THE COPIED RANGE

— SWITCHING ROWS TO COLUMNS

This less known feature of Excel has some great options that save time and reduce annoyances in copying and pasting.

3.1 PASTING THE RESULT OF A FORMULA, BUT NOT THE FORMULA

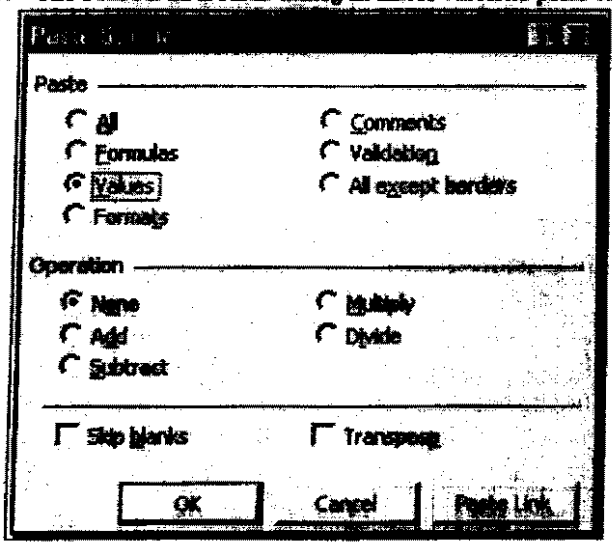
Sometimes one wants the ability to copy a formula (for example, "=C223 + D223)") but paste only the resulting value. (The example that follows will make this clear.)

Select the range "F223:F235" on worksheet ""main."

Choose the menu option FILE/NEW and open a new file. Go to any cell in this new file and choose the menu option EDIT/PASTE SPECIAL.

In the area "Paste," choose the option "Values" as shown in Figure 37.

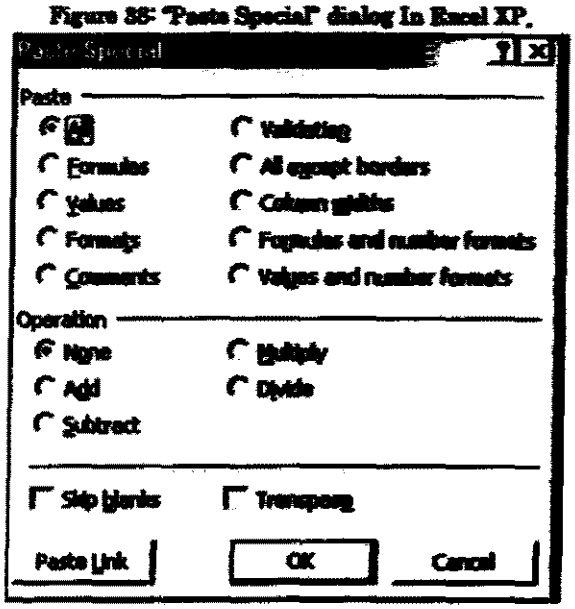
Figure 37: The PASTE SPECIAL dialog in Excel versions prior to Excel XP



In Excel XP, the "Paste Special" dialog has three additional options:

- Paste Formulas and number formats (and not other cell formatting like font, background color, borders, etc)
- Paste Values and number formats (and not other cell formatting like font, background color, borders, etc)
- Paste only "Column widths."

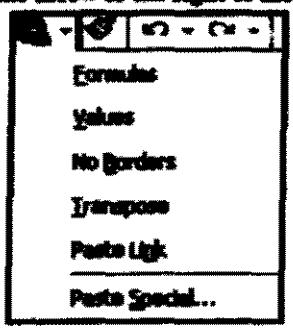
Figure 38: "Paste Special" dialog in Excel XP.



In Excel XP, the "Paste" icon provides quick access to some types of "Paste Special." The options are shown in the next figure.

The calculated values in the "copied" cells are pasted. The formula is not pasted. Try the same experiment using EDIT/PASTE instead of EDIT/PASTE SPECIAL. The usefulness of the former will

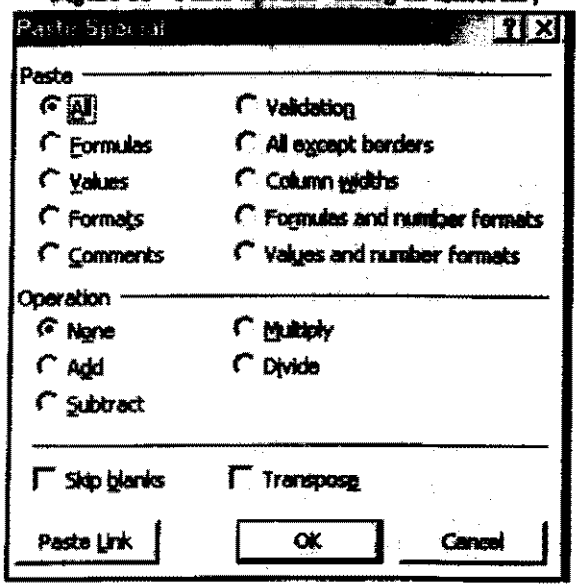
Figure 39: The pasting options can be accessed by clicking on the arrow to the right of the "Paste" icon



In Excel XP, the "Paste Special" dialog has three additional options:

- Paste Formulas and number formats (and not other cell formatting like font, background color, borders, etc)
- Paste Values and number formats (and not other cell formatting like font, background color, borders, etc)
- Paste only "Column widths."

Figure 38: "Paste Special" dialog In Excel XP,



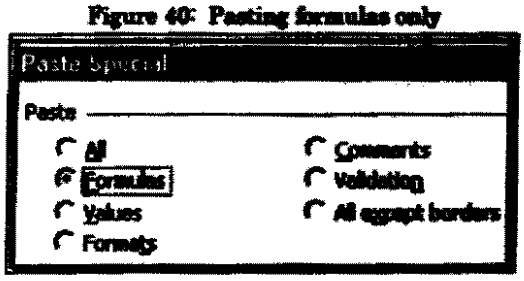
be apparent.

3.2 OTHER SELECTIVE PASTING OPTIONS

3.2.A PASTING ONLY THE FORMULA (BUT NOT THE FORMATTING AND COMMENTS)

Choose the option "Formulas" in the area "Paste" of the dialog (user-input form) associated with the menu "EDIT/PASTE SPECIAL." This feature makes the pasted values free from all cell references. The "pasted on" range will only contain pure numbers. The biggest advantage of this option is that it enables the collating of formula results in different ranges/sheets/workbooks onto one worksheet without the bother of maintaining all the referenced cells in the same workbook/sheet as the collated results.

Figure 40: Pasting formulas only



3.2.B PASTING ONLY FORMATS

Choose the option "Formats" in the area "Paste" of the dialog associated with the menu "EDIT/PASTE SPECIAL use the "Format Painter" icon. I prefer using the icon.

Refer to *Volume 1: Excel For Beginners* for a discussion on the format painter.

21

3.2.C PASTING DATA VALIDATION SCHEMES

Pick the option "Validation" in the area "Paste" of the dialog associated with the menu "EDIT/PASTE SPECIAL." Data validation schemes are discussed in *Volume 4: Managing & Tabulating Data in Excel.* This option can be very useful in standardizing data entry standards and rules across an institution.

3.2.D PASTING ALL BUT THE BORDERS

Choose the option "All except borders" in the area "Paste" of the dialog associated with the menu "EDIT/PASTE SPECIAL." All other formatting features, formulae, and data are pasted. This option is rarely used.

3.2.E PASTING COMMENTS ONLY

Pick the option "Comments" in the area "Paste" of the dialog associated with the menu "EDIT/PASTE SPECIAL." Only the comments are pasted. The comments are pasted onto the equivalently located cell. For example, a comment on the cell that is in the third row and second column that is copied will be pasted onto the cell that is in the third row and second column of the "pasted on" range. This option is rarely used.

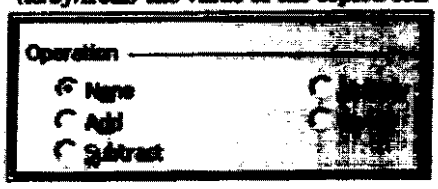
3.3 PERFORMING AN ALGEBRAIC "OPERATION" WHEN PASTING ONE COLUMN/ROW/RANGE ON TO ANOTHER

3.3.A MULTIPLYING/DIVIDING/SUBTRACTING/ADDING ALL CELLS IN A RANGE BY A NUMBER

Assume your data is expressed in millions. You need to change the units to billions— that is, divide all values in the range by 1000. The complex way to do this would be to create a new range with each cell in the new range containing the formula "cell in old range/1000." A much simpler way is to use PASTE SPECIAL. On any cell in the worksheet, write the number 1000. Click on that cell and copy the number. Choose the range whose cells need a rescaling of units. Go to the menu option EDIT/PASTE SPECIAL and choose "Divide" in the area *Options*. The range will be replaced with a number obtained by dividing each cell by the copied cells value!

The same method can be used to multiply, subtract or add a number to all cells in a range

Figure 41: You can multiply (or add/subtract/divide) all cells in the "pasted on" range by (to/by/from) the value of the copied cell



3.3.B MULTIPLYING/DIVIDING THE CELL VALUES IN CELLS IN SEVERAL "PASTED ON" COLUMNS WITH THE VALUES OF THE COPIED RANGE

You can use the same method to add/subtract/multiply/divide one column's (or row's) values to the corresponding cells in one or several "pasted on" columns (or rows).

Try this Copy the cells in column E and paste special onto the cells in columns C and D choosing the option "Add" in the area "Operation" of the paste special dialog. (You can use EDIT/UNDO to restore the file to its old state.)

3.4 SWITCHING ROWS TO COLUMNS

Choose any option in the "Paste" and "Operations" areas and choose the option "Transpose." If pasting a range with many columns and rows you may prefer to paste onto one cell to avoid getting the error "Copy and Paste areas are in different shapes."

CHAPTER 4

INSERTING FUNCTIONS

This chapter teaches the following topics:

— A SIMPLE FUNCTION

— FUNCTIONS THAT NEED MULTIPLE RANGE REFERENCES

— WRITING A "FUNCTION WITHIN A FUNCTION"

— NEW IN EXCEL XP

— RECOMMENDED FUNCTIONS IN THE FUNCTION WIZARD

— EXPANDED AUTOSUM FUNCTIONALITY

— FORMULA EVALUATOR

— FORMULA ERROR CHECKING

4.1 BASICS

Excel has many in–built functions. The functions may be inserted into a formula.

Accessing the functions dialog/wizard

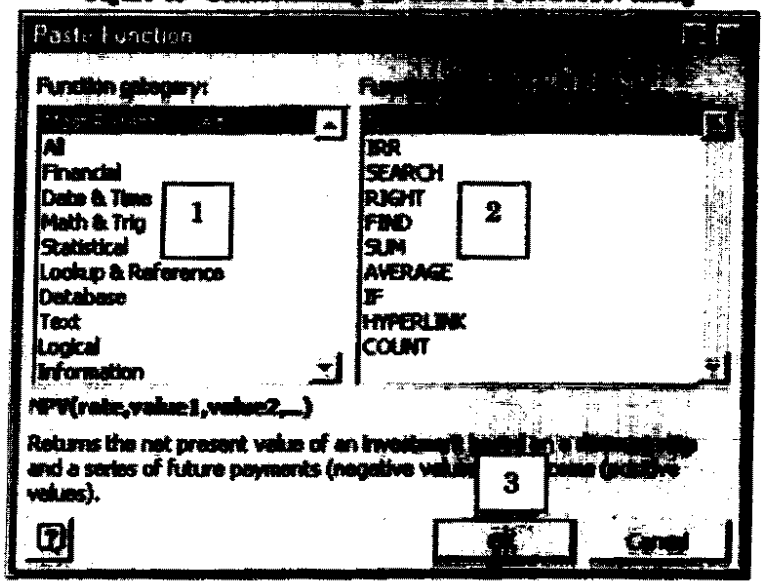
(a) select the menu path INSERT/FUNCTION, or

(b) click on the function icon (see Figure 42)

Figure 42: The Function icon

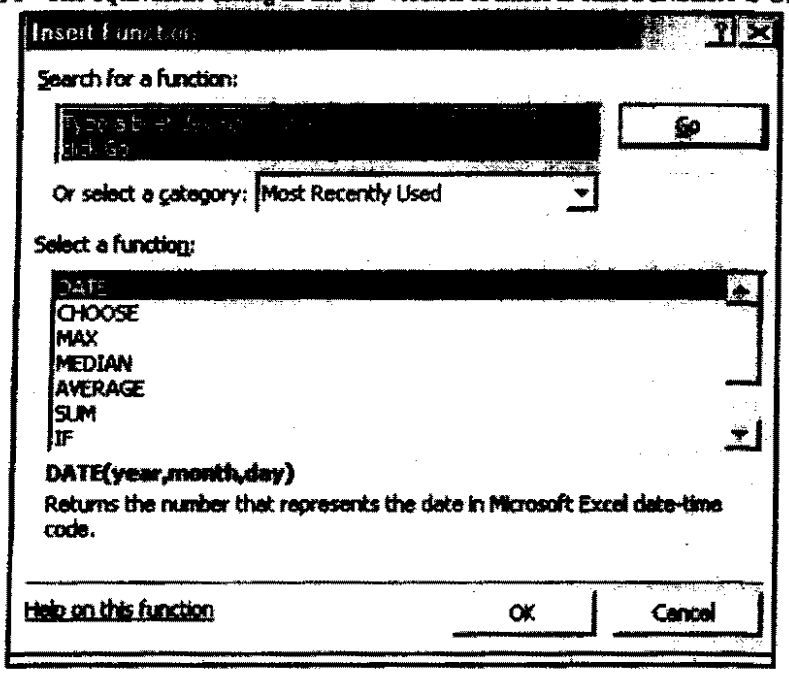
The "Paste Function" dialog (or wizard, because it is a series of dialogs) opens. The dialog is shown in Figure 43.

Figure 43: Understanding the PASTE FUNCTION dialog



The equivalent dialog in the XP version of Excel is called INSERT FUNCTION. (It is reproduced in the next figure below.) The dialog has one new feature—a "Search for a function" utility. The "Function category" is now available by clicking on the list box next to the label "Or select a category."

Figure 44: The equivalent dialog in the XP version of Excel is called INSERT FUNCTION



This dialog has three parts:

(1) The area "Function category" on the left half shows the labels of each group of functions. The group "Statistical" contains statistical functions like "Average" and "Variance." The group "Math & Trig" contains algebra and trigonometry functions like "Cosine." When you click on a category name, all the functions within the group are listed in the area "Function name."

(2) The area "Function name" lists all the functions within the category selected in the area "Function category." When you click on the name of a function, its formula, and description is shown in the gray area at the bottom of the dialog.

(3) The area with a description of the function

Step 2 for using a function in a formula

Click on the "Function category" (in area 1 or the left half of the dialog)

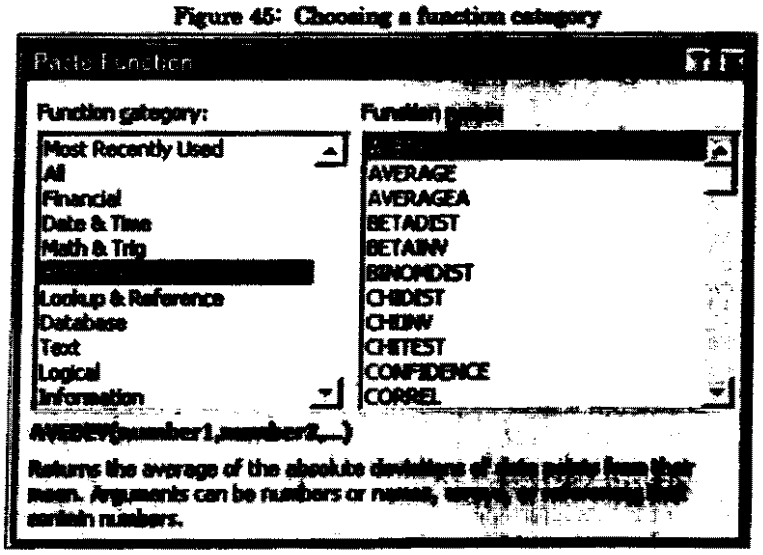
that contains the function, then click on the function name in the area "Function name" (in area 2 or the left half of the dialog) and then execute the dialog by clicking on the button OK.

4.2 A SIMPLE FUNCTION

In my first example, I show how to select and use the function "Average" which is under the category "Statistical."

Choose the category "Statistical" as shown in Figure 45.

Figure 45: Choosing a function category

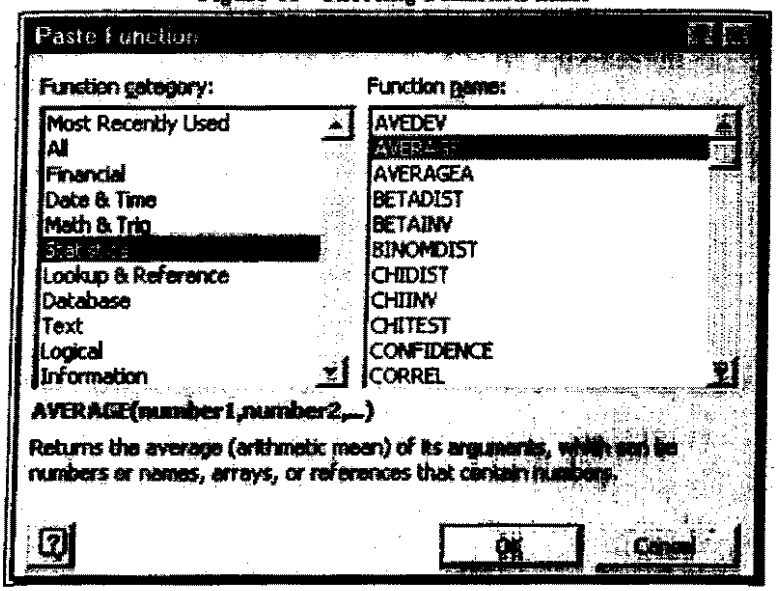


Choose the formula "Average" in the area "Function name."

This is shown in Figure 46.

Execute the dialog by clicking on the button OK.

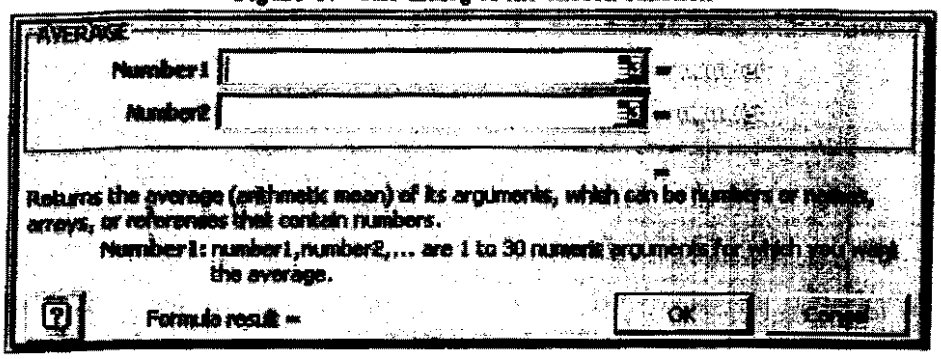
Figure 46: Choosing a function name



The dialog (user-input form) for the "Average" function opens.

For a pictorial reproduction of this, see Figure 47.

Figure 47: The dialog of the chosen function



<u>Step 3 for inserting a function —</u> defining the data arguments/requirements for the function

Figure 48: Selecting the cell references whose values will be the inputs into the function

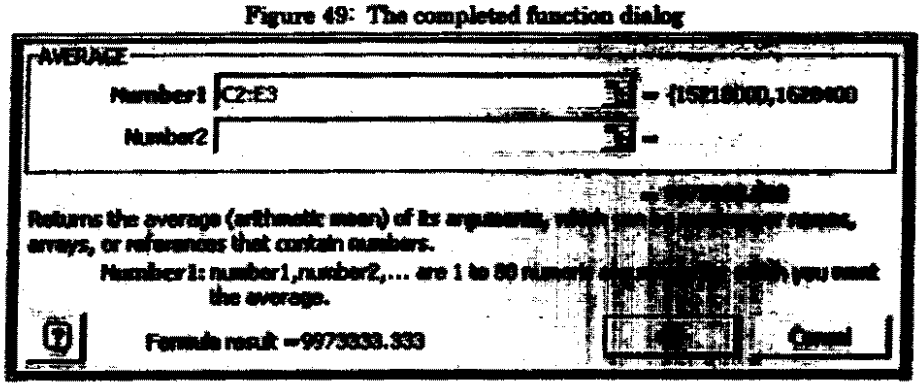
You have to tell Excel which cells contain the data to which you want to apply the function "AVERAGE." Click on the right edge of the text-box "Number1"[6]. (That is, on the red–blue–and–white corner of the cell.) Go to the worksheet that has the data you want to use and highlight the range "C2 to E3." Click on the edge of the text-box. (For a pictorial reproduction of this, see Figure 48.)

You will be taken back to the "Average" dialog. Notice that — as shown in Figure 49 — the cell reference "C2:E3" has been added.

Furthermore, note that the answer is provided at the bottom (see the line "Formula result = 9973333.333").

Execute the dialog by clicking on the button OK.

Figure 49: The completed function dialog



[6] If you want to use non-adjacent ranges in the formula, then use the text-box "Number 2" for the second range. Excel will add more text-boxes once you fill all the available ones. If the label for a text-box is not in bold then it is not essential to fill that text-box. In the AVERAGE dialog shown in Figure 402, the label for the first text-box ("Number 1") is in bold—so it has to be filled. The label for the second text-box ("Number 2") is not in bold — so, it can be left empty.

The formula is written into the cell and is shown in Figure 50.

Figure 50: The function is written into the cell

=AVERAGE(C2:E3)

Press the ENTER key and the formula will be calculated.

You can work with this formula in a similar manner as a simple formula — copying and pasting, cutting and pasting, writing on multiple worksheets, etc.

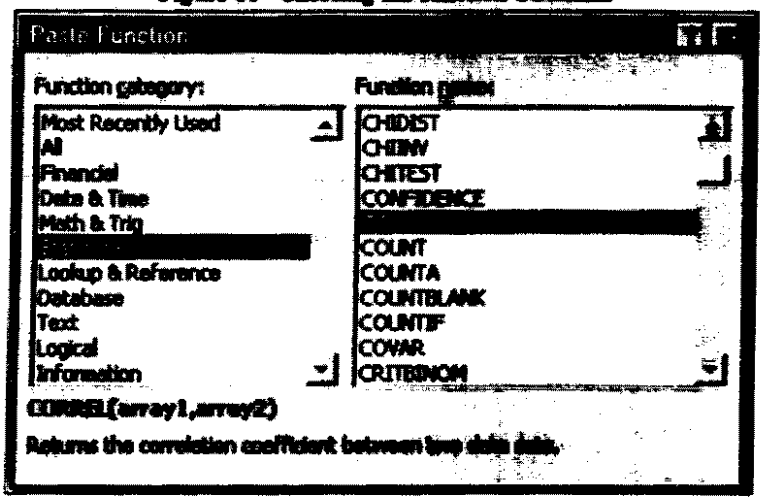
If you remember the function name, you do not have to use INSERT/FUNCTION. Instead, you can simple type in the formulas using the keyboard. This method is faster but requires that you know the function.

4.3 FUNCTIONS THAT NEED MULTIPLE RANGE REFERENCES

Some formulas need a multiple range reference. One example is the correlation formula ("CORREL"). Assume, in cell J1, you want to calculate the correlation between the data in the two ranges: "D2 to D14" and "E2 to E14."

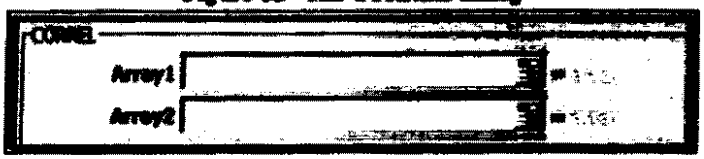
Activate cell J1. Select the option INSERT/FUNCTION. Choose the function category "Statistical." In the list of functions that opens in the right half of the dialog, choose the function "CORREL" and execute the dialog by clicking on the button OK.

Figure 51: Choosing the function CORREL



The CORREL dialog (shown in the next figure) opens. The function needs two arrays (or series) of cells references. (Because the labels to both the text-box labels are bold, both text-boxes have to be filled for the function to be completely defined.) Therefore, the pointing to the cell references has to be done twice as shown in Figure 53 and the next two figures.

Figure 52: The CORREL dialog



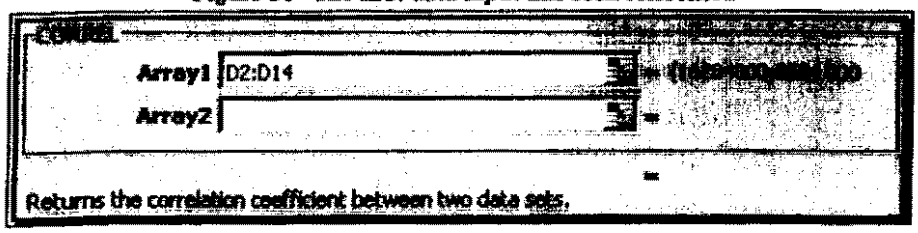
Choosing the first array/series

Click on the box edge of "Array1" (as shown in Figure 52.) Then go to the relevant data range (D2 to D14 in this example) and select it.

Figure 53: Selecting the first data input for the function

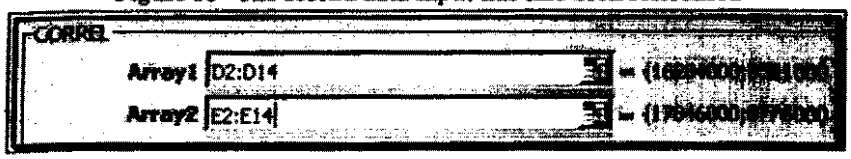
Repeat the same for "Array 2," selecting the range "E2:E14" this time.

Figure 54: The first data input has been referenced



The formula is complete. The result is shown in the dialog in the area at the bottom "Formula result." Execute the dialog by clicking on the button OK.

Figure 55: The second data input has also been referenced



Once the dialog closes, depress the ENTER key, and the function will be written into the cell and its result evaluated/calculated.

Figure 56: The function as written into the cell.

=CORREL(D2:D14,E2:E14)

4.4 WRITING A "FUNCTION WITHIN A FUNCTION"

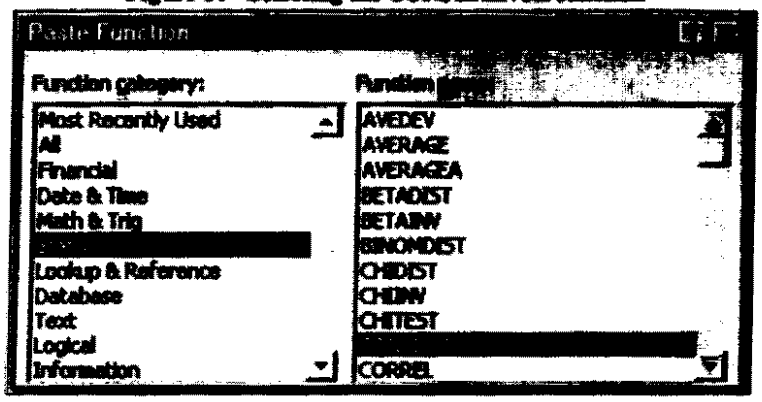
I use the example of the CONFIDENCE function from the category "Statistical."

Choose the menu option INSERT/FUNCTION.

Choose the function category "Statistical."

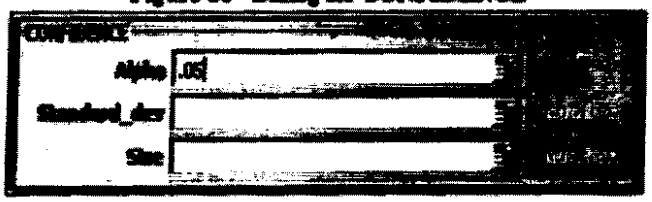
In the list of functions that opens in the right half of the dialog, choose the function CONFIDENCE and execute the dialog by clicking on the button OK.

Figure 57: Selecting the CONFIDENCE function



The Confidence dialog (user-input form) requires[7] three parameters: the alpha, standard deviation, and sample size. First type in the alpha desired as shown in Figure 58. (An alpha of ".05" corresponds to a 95% confidence level while an alpha value of ":.1" corresponds to a confidence interval of 90 %.)

Figure 58: Dialog for CONFIDENCE



[7] We know that all three are necessary because their labels are in bold.

84

Press the OK button.

Figure 59: The first part of the function

=CONFIDENCE(.05

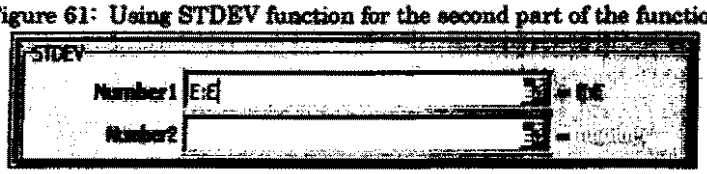
Type a comma after the ".05" (see Figure 60) and then go to INSERT/FUNCTION and choose the formula STDEV as shown in Figure 61.

Figure 60: Placing a comma before entering the second part

(.05,)

Choose the range for which you want to calculate the STDEV (for example, the range "E:E") and execute the dialog by clicking on the button OK.

Figure 61: Using STDEV function for the second part of the function



The formula now becomes:

Figure 62: A function within a function

=CONFIDENCE(.05, STDEV(E:E))

The main formula is still CONFIDENCE. The formula STDEV provides one of the parameters for this main formula. The STDEV function is nested within the CONFIDENCE function.

85

Type a comma, and then go to INSERT/FUNCTION and choose the function "Count" from the function category "Statistical" to get the final formula.

Figure 63: The completed formula

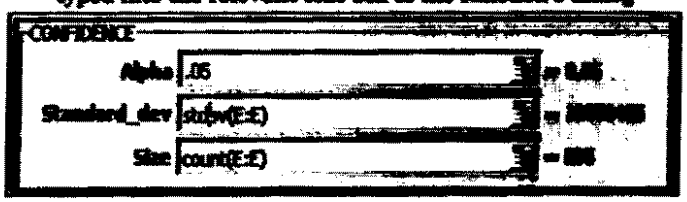
=CONFIDENCE(0.05,STDEV(E:E),COUNT(E:E))

<u>There are two other ways to write this formula.</u>

Select the option INSERT/FUNCTION, choose the function CONFIDENCE from the category "Statistical" and type in the formulae "STDEV(E:E)" and "COUNT(E:E)" as shown in Figure 64.

This method is much faster but requires that you know the function names STDEV and COUNT.

Figure 64: If sub-functions are required in the formula of a function, the sub-functions may be typed into the relevant text-box of the function's dialog



The third way to write the formula is to type it in. This is the fastest method.

Figure 65: The result is the same

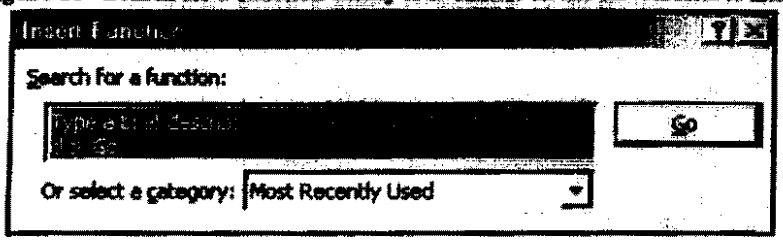
=CONFIDENCE(0.05,STDEV(E:E),COUNT(E:E))

4.5 NEW FUNCTION-RELATED FEATURES IN THE XP VERSION OF EXCEL

Searching for a function

Type a question (like "estimate maximum value") into the box "Search for a function" utility and click on the button "Go." Excel will display a list of functions related to your query.

Figure 66: Search for a function utility is available in the XP version of Excel



4.5.A ENHANCED FORMULA BAR

After you enter a number or cell reference for the first function "argument" (or first "requirement") and type in a comma, Excel automatically converts to bold format the next argument/requirement. In the example shown in the next figure, Excel makes bold the font for the argument placeholder *pmt* after you have entered a value for *nper* and a comma.

Figure 67: The Formula Bar Assistant is visible below the Formula Bar

Similarly, the argument/requirement after *pmt* has a bold font after you have entered a value or reference for the argument *pmt*

87

Figure 68: The next "expected" argument/requirement if highlighted using a bold font

=RATE(24,C3,
RATE(nper, pmt, pv,

The square brackets around the argument/requirement "fv" indicate that the argument is optional. You need not enter a value or reference for the argument.

Figure 69: An optional argument/requirement

=RATE(24,C3,C2,
RATE(nper, pmt, pv, [fv],

4.5.8 ERROR CHECKING AND DEBUGGING

The basics of this topic are taught in the next chapter. Advanced features are in *Volume 3: Excel—Beyond the basics.*

CHAPTER 5

TRACING CELL REFERENCES & DEBUGGING FORMULA ERRORS

This short chapter demonstrates the following topics:

— TRACING THE CELL REFERENCES USED IN A FORMULA

— TRACING THE FORMULAS IN WHICH A PARTICULAR CELL IS REFERENCED

— WATCH WINDOW

— ERROR CHECKING

— FORMULA EVALUATION

5.1 TRACING THE CELL REFERENCES USED IN A FORMULA

Click on the cell that contains the formula whose references need to be visually traced. Pick the menu option TOOLS/AUDITING/TRACE PRECEDENTS. (For a pictorial reproduction of this, see Figure 70.)

Figure 70: Tracing precedents. These options are from Excel versions prior to Excel XP.

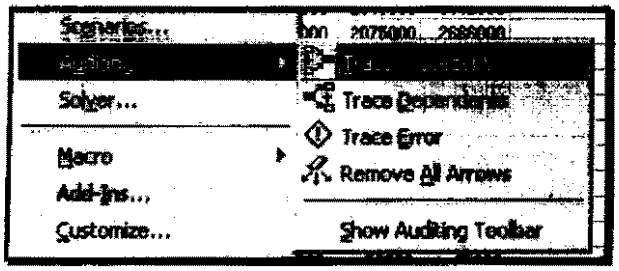
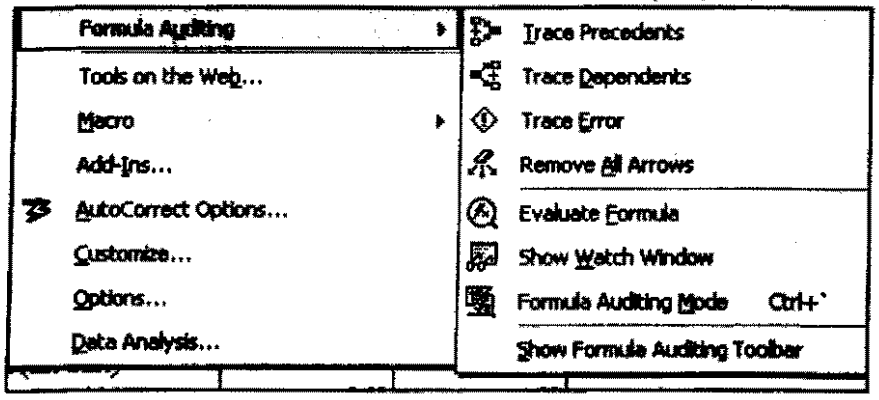


Figure 71: Excel XP offers several error-checking and debugging tools.

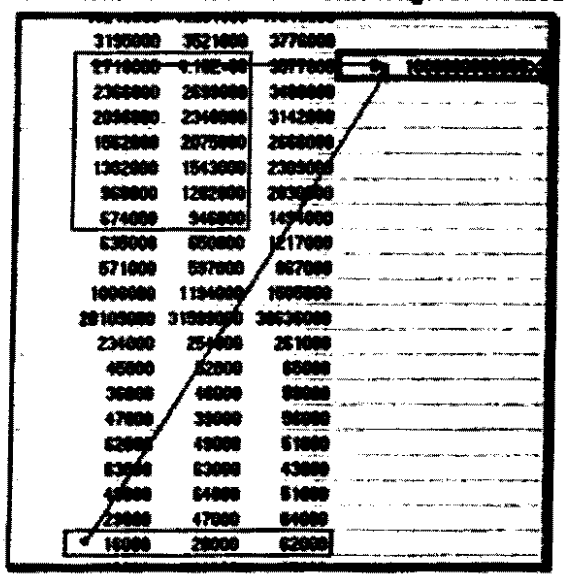


As shown in Figure 72, blue arrows will trace the references.

If a group of cells is referenced, then the group will be marked by a blue rectangle. The two rectangular areas are referenced in the formula.

> In *Volume 3: Excel- Beyond The Basics*, you are taught the simple process through which you can select all the cells whose formulas are *precedents* of the active cell.

Figure 72: The arrows define and trace all the cells/ranges referenced in the active cell

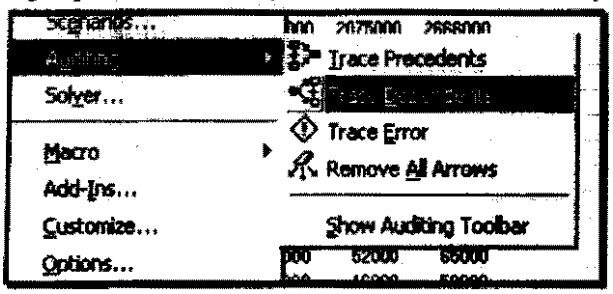


5.2 TRACING THE FORMULAS IN WHICH A PARTICULAR CELL IS REFERENCED

You may want to do the opposite— see which formulas reference a particular cell.

- First, click on the cell of interest.
- Then, pick the menu option TOOLS/AUDITING/TRACE DEPENDENTS as shown in Figure 73. Now the arrows will go from the active cell to all the cells that have formulas that use the active cell.

Figure 73: Tracing Dependents. These options are from Excel versions prior to Excel XP.



Remove all the auditing arrows by following the menu path TOOLS/AUDITING/REMOVE ALL ARROWS.

> In Volume 3: Excel- Beyond The Basics you learn the simple process through which you can select all the cells whose formulas are dependents of the active cell.

5.3 THE AUDITING TOOLBAR

The "Auditing" toolbar opens automatically when you are using the auditing option (TOOLS/AUDITING) to review formula references.

> Refer to *Volume 3: Excel- Beyond The Basics* for details on using toolbars.

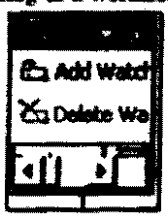
In the XP version of Excel, you can launch the toolbar through the menu option TOOLS/AUDITING/SHOW FORMULA AUDITING TOOLBAR.

Figure 74: The "Auditing" toolbar

5.4 WATCH WINDOW (ONLY AVAILABLE IN THE XP VERSION OF EXCEL)

The window is accessed through the menu path TOOLS/ AUDITING/ SHOW WATCH WINDOW, or VIEW/ TOOLBARS/ WATCH WINDOW.

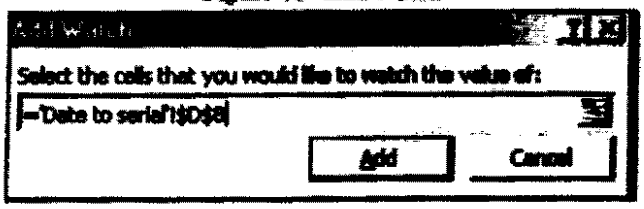
Figure 75: The Watch Window may not display correctly. Use the mouse to drag the walls of the dialog to a workable size.



Add one cell on whose values you want to keep tabs.

The value will be shown in the Watch Window so that you can see the value even if you are working on cells or sheets that are far from the cell whose value is being "watched."

Figure 76: Add Watch



You can add many cells to the Watch Window. Note that the Watch Window provides precise information on the location of the cell being watched and the formula in the cell. For example, the first watched cell is on cell D8 in sheet "Date to serial" in the file "Date and Time.xls." The formula in the cell is "=DATE(F7, E7, D7)".

Figure 77: You can add many cells to the Watch Window

Watch Window

Add Watch... Delete Watch

Book	Sheet	Name	Cell	Value	Formula
Date and Time.xls	Date to serial		D8	36988.00	=DATE(F7,E7,D7)
Date and Time.xls	Date to serial		D12	0.141932870	=TIME(D11,E11,F...
Date and Time.xls	Date to serial		B2	37444.284525	=NOW()
Date and Time.xls	Date to serial		B3	37444.00	=TODAY()

5.5 ERROR CHECKING AND FORMULA EVALUATOR (ONLY AVAILABLE IN THE XP VERSION OF EXCEL)

The tools are accessed through TOOLS/ERROR CHECKING and TOOLS/FORMULA AUDITING/EVALUATE FORMULA.

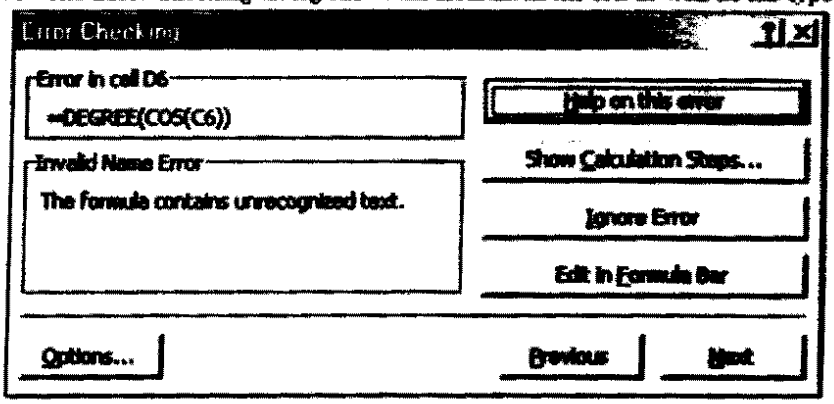
The Error Checking dialog shows the formula in the cell as well as the type of error. In this example, these are "=DEGREE(COS(C6))" and "Invalid Name Error," respectively.

The button ("Help on this error") links to a help file containing assistance on understanding and debugging the error.

The button "Show Calculation Steps" links to a step-by-step debugger that assists in catching the calculation step at which the error occurred.

This debugger has the same functionality as the Formula Auditor (accessed through TOOLS/FORMULA AUDITING/EVALUATE FORMULA).

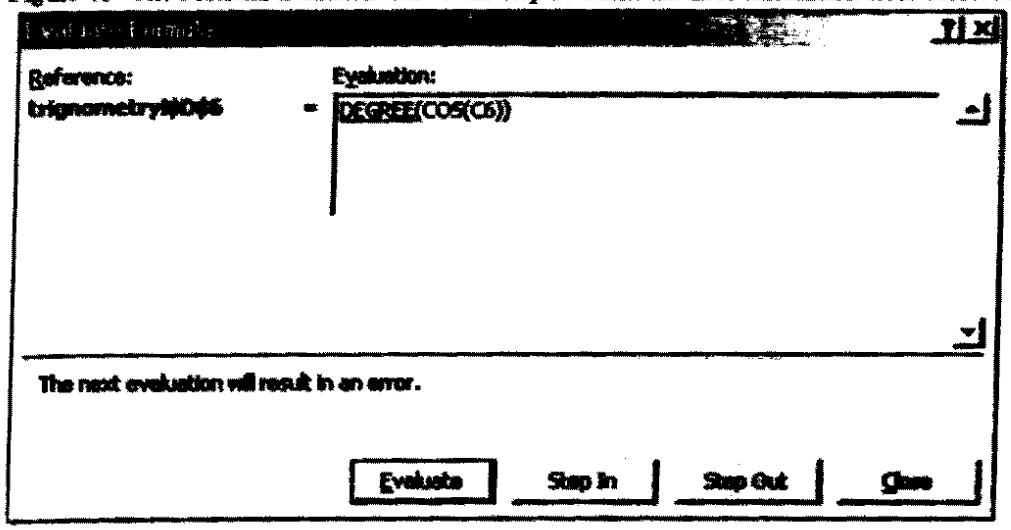
Figure 78: The Error Checking dialog shows the formula in the cell as well as the type of error



The button "Ignore Error" keeps the error "as is." The button Options opens the dialog for setting error-checking options. The choices within the dialog are listed in section 5.8.

The Formula Evaluator shows the step at which the first calculation error occurred. This helps in identifying the primary problem. In this example, no error has occurred in the formula part "COS(C6))". The dialog informs you that "The next evaluation (that is, calculation step), will result in an error."

Figure 79: The Formula Evaluator shows the step at which the first calculation error occurred

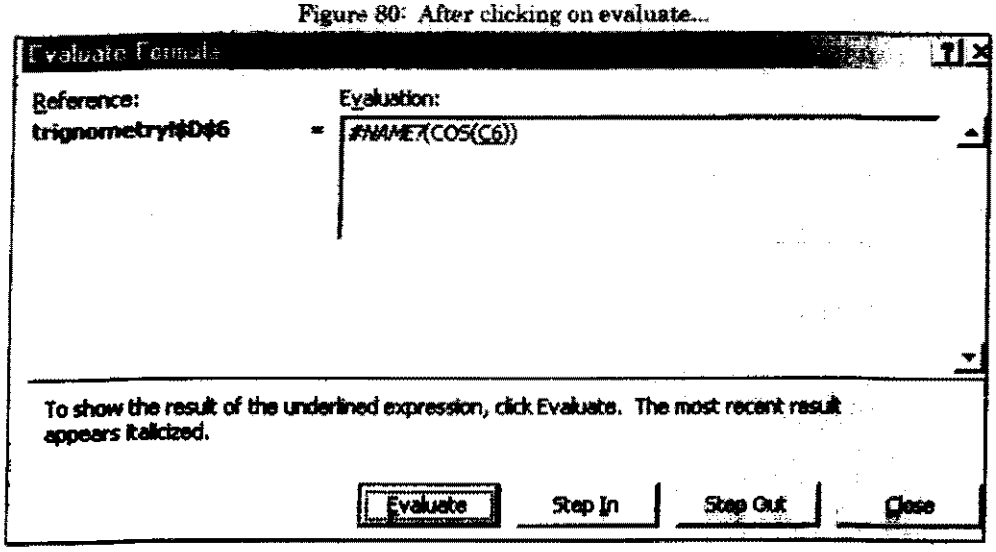


95

After clicking on evaluate, you see that the error is in the formula part "DEGREE." Excel also informs you of the type of error— "#NAME?" suggests that "DEGREE" does not match the name of any Excel function. (The correct function is "DEGREES.")

The "COS" function is nested within the DEGREE function. Clicking on "Step In" will evaluate the nested function only.

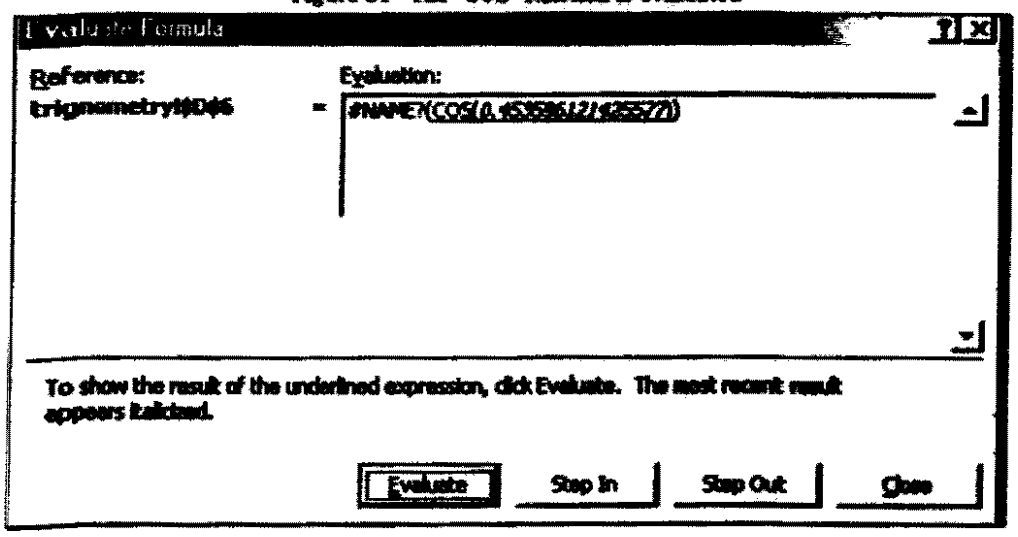
Figure 80: After clicking on evaluate...



The "COS" function is evaluated. The function has no error.

> If a function has more than two levels of nesting, then you can use the "Step Out" button to evaluate the function at the higher level of nesting.

Figure 81: The "COS" function is evaluated

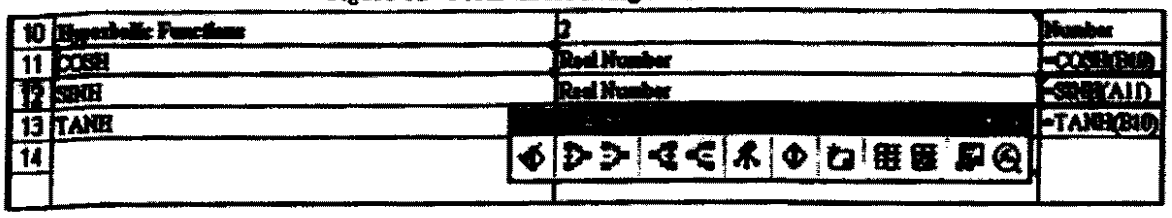


5.6 FORMULA AUDITING MODE (ONLY AVAILABLE IN THE XP VERSION OF EXCEL)

This feature is accessed through TOOLS/FORMULA AUDITING/FORMULA AUDITING MODE. After this mode is selected, when you select a cell that has or is referenced by a formula, Excel highlights the other referenced/referencing cells.

In addition, you have quick access (via the "Formula Auditing" toolbar) to all the Auditing tools discussed earlier in this chapter.

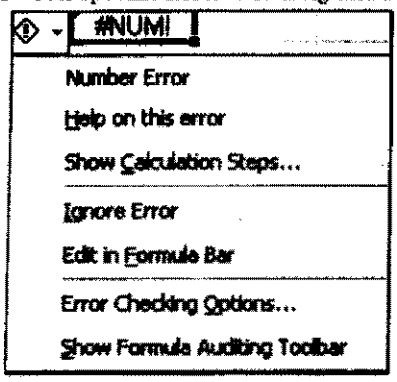
Figure 82: Formula Auditing Mode



5.7 CELL-SPECIFIC ERROR CHECKING AND DEBUGGING

On every cell whose value evaluates to an error value, you will see a small icon with a "!" image and a downward arrow. Click on the arrow to obtain assistance for debugging the error.

Figure 83: Cell-specific Error Checking and Debugging



In the example shown in the figure, the options show:

— the error type ("Number Error"),

— a link to assistance on understanding and debugging the error ("Help on this error"),

— a step-by-step debugger to catch the calculation step at which the error occurred ("Show Calculation Steps"),

— the option to ignore and thereby keep the error as is ("Ignore Error"),

— a link to directly edit the formula in the cell ("Edit in Formula Bar"),

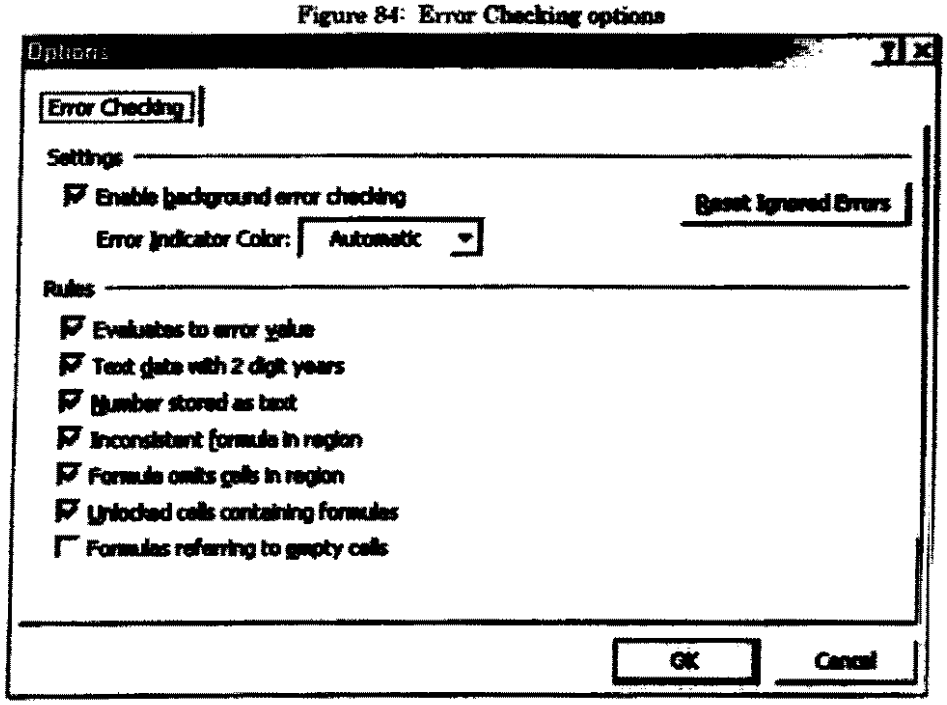
— the overall error-checking options ("Error Checking Options"), and

— direct access to the Formula Auditing Toolbar ("Show Formula Auditing Toolbar") and, thereby, to all the features of Auditing (these features are taught in this chapter)

5.8 ERROR CHECKING OPTIONS

The Error Checking options can be assessed through TOOLS/OPTIONS/ERROR CHECKING or through TOOLS/ERROR CHECKING/OPTIONS. The dialog is reproduced in the next figure.

Figure 84: Error Checking options



You can inform Excel to show as an error any cell: that contains:

- A formula that evaluates to an error value
- A formula that refers to an empty cell
- A formula that is not consistent with the other formulas and cell references in neighboring cells
- A two-digit year (like "02") instead of a four-digit year (like "2002")
- A number stored as text

The other options are beyond the scope of this book. I recommend sticking with the default settings reproduced in the next figure.

CHAPTER 6

FUNCTIONS FOR BASIC STATISTICS

This chapter discusses the following topics:

— "AVERAGED" MEASURES OF CENTRAL TENDENCY

— AVERAGE, TRIMMED MEAN, HARMONIC MEAN, GEOMETRIC MEAN

— LOCATION MEASURES OF CENTRAL TENDENCY

— MEDIAN, MODE

— OTHER LOCATION PARAMETERS

— QUARTILE, PERCENTILE

— MAXIMUM VALUE, MINIMUM VALUE, LARGE, SMALL

— RANK OR RELATIVE STANDING OF EACH CELL WITHIN THE RANGE OF A SERIES

— MEASURES OF DISPERSION (STANDARD DEVIATION & VARIANCE)

— STDEV, VAR, STDEVA, VARA, STDEVP, VARP, STDEVPA, VARPA

— SHAPE ATTRIBUTES OF THE DENSITY FUNCTION

— SKEWNESS, KURTOSIS

— FUNCTIONS ENDING WITH AN "A" SUFFIX

I am presuming that the reader is familiar with basic statistical functions and/or has access to a basic statistics reference for learning more about

each function.

6.1 "AVERAGED" MEASURES OF CENTRAL TENDENCY

These set of functions perform some type of averaging to measure a "mean" value. You may want to use the Trimmed Mean function to estimate an average that excludes the extreme values of the data series. The Harmonic Mean estimates the averages of the reciprocals of the numbers in the series. The Geometric Mean is used to average rates of change.

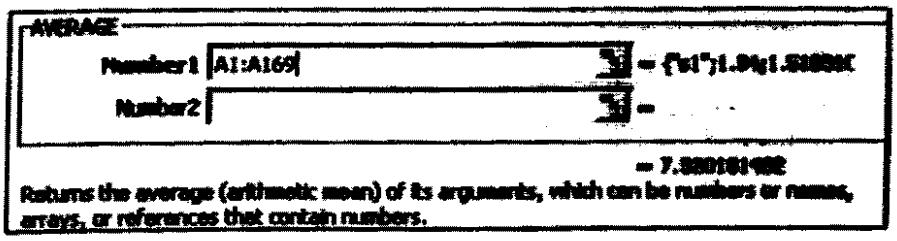
Samples will be available at http://www.vjbooks.net/excel/samples.htm.

6.1.A AVERAGE

The function calculates the simple arithmetic average of all cells in the choseen range.

Menu path to function: Go to the menu option INSERT/FUNCTION and choose the formula "AVERAGE the function category STATISTICAL.

Figure 86: AVERAGE function



102

Data requirements: The X values can be input as references to one or more ranges that may be non–adjacent. The second range can be referenced in the first text-box "Number1" after placing a comma after the first range, or it could be referenced in the second text-box "Number2." If you use the second text-box, then a third text-box "Number3" will automatically open. (As you fill the last visible box, another box opens until the maximum number of boxes — 30 — is reached.)

The function does not count invalid cell values when counting the number of X values. The X values can take any real number value.

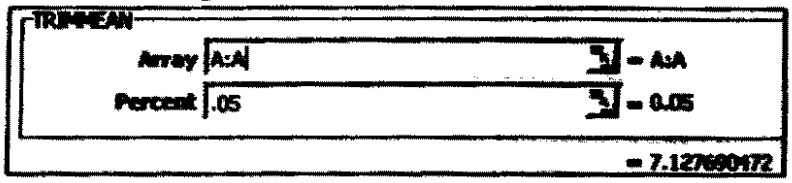
6.1.B TRIMMEAN ("TRIMMED MEAN")

This function is a variation of the average or mean. This function calculates the average for a set of X values after removing "extreme values" from the set. The excluded cells are chosen by the user based on the extremity (from mean/median) of the values in the range. TRIMMEAN calculates the mean taken by excluding a percentage of data points from the top and bottom tails of a data set. The user decides on the percentage of extreme values to drop. For symmetry, TRIMMEAN excludes a set of values from the top and bottom of the data set before moving on to the next exclusion.

Menu path to function: INSERT/FUNCTION/STATISTICAL/TRIMMEAN.

Data requirements: The X values can be input as references to one or more ranges that may be non–adjacent. The function does not count invalid cell values when counting the number of X values. The X values can take any real number value.

Figure 86: TRIMMEAN (Trimmed Mean)



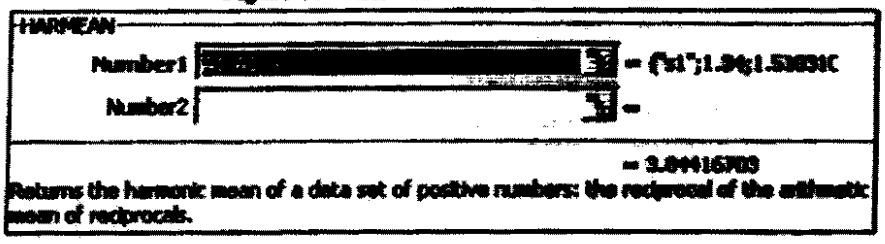
In the dialog (shown above), *Percent* is the fractional number of data points to exclude from the calculation. Percent must be greater than zero and less than one.

6.1.C HARMEAN ("HARMONIC MEAN")

The function calculates the harmonic mean of all cells in the chosen range(s). The harmonic mean is the reciprocal of the arithmetic mean of reciprocals. In the formula below, H is the harmonic mean, n the sample/range size and the Y's are individual data values.

Menu path to function: INSERT/FUNCTION/STATISTICAL/HARMEAN.

Figure 87: HARMEAN (Harmonic Mean)



Data requirements: The X values can take any real number value except zero.

Table 10: Comparing the results of the functions Average, Trimmed Mean and Harmonic Mean

Function	s1	s2	x1	x2	x3	x4
Average/mean	7.32	7.23	1173.00	14.55	0.17	1158.45
Trimmed Mean	7.13	7.00	1173.00	14.42	0.02	1158.71
Harmonic Mean	3.84	3.18	120.17	13.52	0.01	#NUM!

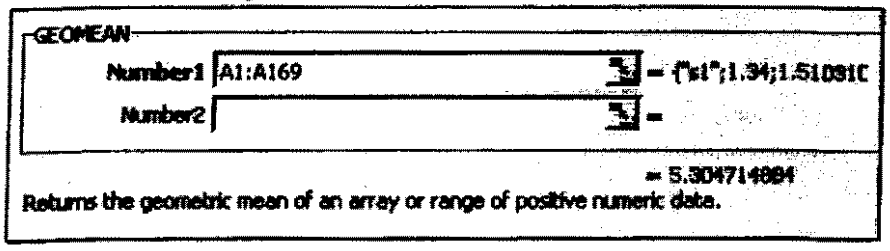
Harmonic mean for x4 is zero because one value of x4 is not positive.

6.1.D GEOMEAN ("GEOMETRIC MEAN")

This function is typically used to calculate average growth rate given compound interest with series rates. In general, the function is good for estimating average growth or interest rates.

Menu path to function: INSERT/ FUNCTION/ STATISTICAL/ GEOMEAN. *Data requirements*: All values should be positive.

Figure 88: GEOMEAN (Geometric Mean)



105

6.2 LOCATION MEASURES OF CENTRAL TENDENCY (MODE, MEDIAN)

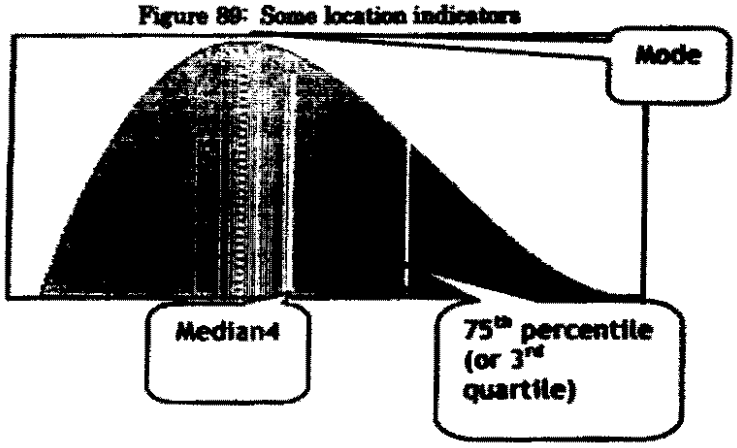
The Median and — less often — the Mode are also used for estimating the central tendency of a series. The Median is much better in situations where, either:

(a) A few extreme highs or lows are influencing the Mean (note that the TRIMMEAN or Trimmed Mean function shown in the previous section can reduce the chance of extreme values over-influencing a Mean estimate), or

(b) The central tendency is required to obtain the mid-point of observed values of the data series as in the "Median Voter" models, which are used to know if the "Median Voter" threshold is crossed in support of a point on the nominee's agenda. (In a two-person face-off, any more than the Median vote will result in a greater than 50% majority).

Samples will be available at http://www.vjbooks.net/excel/samples.htm.

Figure 89: Some location indicators



6.2.A MEDIAN

The Median is the number in the middle of a set of numbers. It is the 50th percentile.

Menu path to function: INSERT/FUNCTION/STATISTICAL/MEDIAN.
Data requirements: Any array/range with real numbers.

6.2.B MODE

This function returns the most frequently occurring value in a range.

Menu path to function: INSERT/FUNCTION/STATISTICAL/MODE.
Data requirements: Any array/range with real numbers. The range has to contain duplicate data values.

6.3 OTHER LOCATION PARAMETERS (MAXIMUM, PERCENTILES, QUARTILES, OTHER)

Other useful location indicators for key points in a series are the quartiles, percentiles, maximum value, minimum value, the Kth largest value, and the rank.

Samples will be available at http://www.vjbooks.net/excel/samples.htm.

6.3.A QUARTILE

This function calculates a quartile of a data series.

QUARTILE (Data, Quartile)

Choose the quartile you desire to obtain. The five quartiles are shown in the next table.

Table 11: Choosing the Quartile

Quartile value of...	*Calculates the...*
0	0.0....1% ile
1	First quartile (25th percentile)
2	Median value (50th percentile)
3	Third quartile (75th percentile)
4	Fourth quartile (99.9x%ile)

Menu path to function: INSERT/FUNCTION/STATISTICAL/QUARTILE.

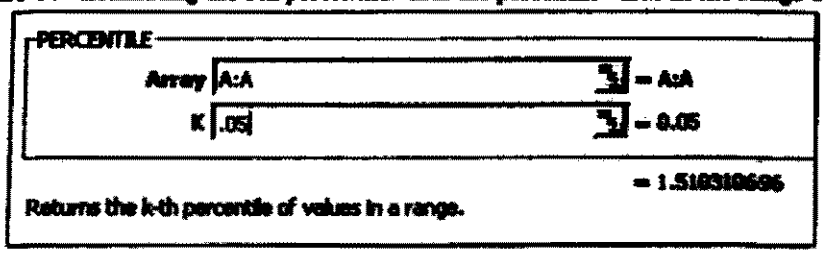
Data requirements: Any array/range with real numbers. Note: the data series has to contain between 1 and 8,191 data points

6.3.B PERCENTILE

This function returns the Pth percentile of values in a data series. You can use this function to establish a threshold of acceptance. For example, you can prefer to examine candidates who score above the 95th percentile will qualify for a scholarship.

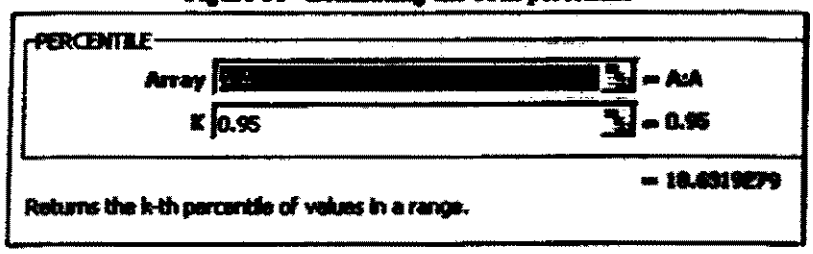
Menu path to function:
INSERT/FUNCTION/STATISTICAL/PERCENTILE.

Figure 90: Estimating the 5th percentile. K is the percentile value in the range 0 to 1.



Data requirements: Any array/range with real numbers. If the data array is empty or contains more than 8,191 data points, PERCENTILE returns the" #NUM!" error value. If K is not a multiple of (1/(n — 1)), then Excel interpolates the value at the Kth percentile.

Figure 91: Estimating the 95th percentile



6.3.C MAXIMUM, MINIMUM AND "KTH LARGEST"

MAX ("Maximum value")

MAX and MAXA: The functions calculate the largest value in a series.

Menu path to function: STATISTICAL/MAX , & STATISTICAL/MAXA.

Data Requirements: Any array/range with real numbers. In addition, MAXA may include "True," "False," or numbers in text format

MIN ("Minimum value")

MIN and MINA: The functions calculate the smallest value in a series

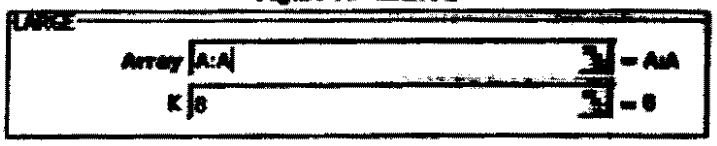
Menu path to function: STATISTICAL/MIN, & STATISTICAL/MINA

Data Requirements: Any array/range with real numbers. In addition, MINA include "True," "False," or numbers in text format

LARGE

This function calculates the Kth largest value in a range.

Figure 98: LARGE



Menu path to function: STATISTICAL/LARGE

Data Requirements: Any real number.

SMALL

This function calculates the Kth smallest value in a range.

Menu path to function: STATISTICAL/SMALL

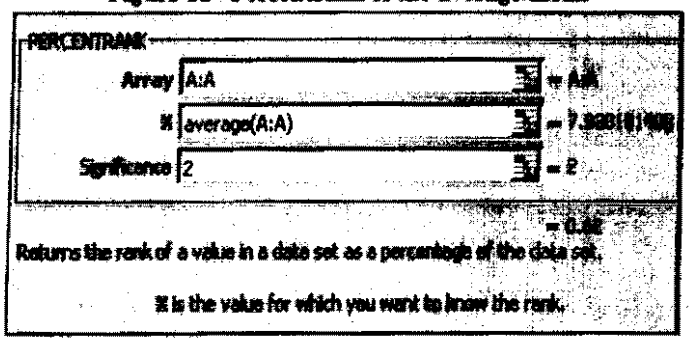
Data Requirements: Any real number.

6.3.D RANK OR RELATIVE STANDING OF EACH CELL WITHIN THE RANGE OF A SERIES

PERCENTRANK

The PERCENTRANK function returns the rank of a value in a data set as a percentage of the data set. The function can be used to evaluate the relative standing of a value within a data set. For example, you can use PERCENTRANK to evaluate the standing of a test score among all scores for the test.

Figure 93: Percentrank of the average/mean



Menu path to function: INSERT/FUNCTION / STATISTICAL / PERCENTRANK.

Data requirements: Any array/range with real numbers.

RANK

The function RANK calculates the relative rank of a value within a series of numbers data. You can choose to obtain the ranks on the basis of ascending or descending values. X is the data point whose rank is desired within the range. Order sets the sorting direction— 1 for ascending ranking, 0 or blank for descending ranking. Cells with the same value cells are given the same rank.

Menu path to function: INSERT / FUNCTION / STATISTICAL / RANK.

Data requirements: Any array/range with real numbers.

6.4 MEASURES OF DISPERSION (STANDARD DEVIATION & VARIANCE)

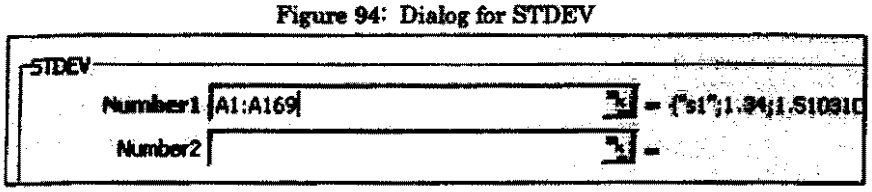
Table 12: Standard Deviation & Variance.

Function	*Description*	*Location within INSERT / FUNCTION*	*Data Requirements*
Sample dispersion: STDEV, VAR	The functions STDEV and VAR estimate the sample standard deviation and variance, respectively. VAR is the square of STDEV.	STATISTICAL / STDEVA & STATISTICAL / VARA	Any range with sufficient number of numeric data points. Text and logical values are excluded.
STDEVA, VARA	These are variants of the functions above but	STATISTICAL /	Text and logical values such as TRUE and

112

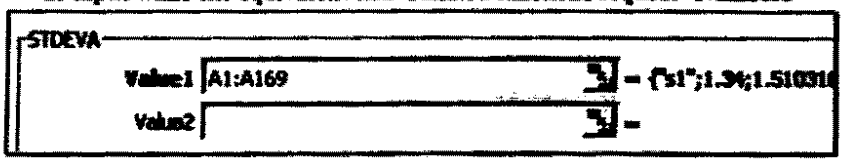
Function	*Description*	*Location within INSERT / FUNCTION*	*Data Requirements*
	with a wider range of acceptable data types as input data.	STDEVA & STATISTICAL / VARA	FALSE are included in the calculation. TRUE is valued as 1; text or FALSE is valued as 0.
Population dispersion: STDEVP, VARP	The less often used population dispersion functions are sometimes also used for large sample sizes. STDEVP assumes that its data are the entire population. Typically, you use the sample formulae. For large sample sizes, STDEV and STDEVP return approximately equal values. VARP is square of STDEVP	STATISTICAL / STDEVA & STATISTICAL / VARA	A large number of observations. Text and logical values are excluded.
STDEVPA, VARPA	These are variants of the functions above but with a wider range of acceptable data types as input data	STATISTICAL / STDEVA & STATISTICAL / VARA	Text and logical values such as TRUE and FALSE are included in the calculation. TRUE is valued as 1; text or FALSE is valued as 0. Text and logical values such as TRUE and FALSE are included in the calculation. TRUE is valued as 1; text or FALSE is valued as 0.

Figure 94: Dialog for STDEV



113

Figure 95: Dialog for STDEVA. Note that the functions with the "A" suffix request "Values" as input while the equivalent non-suffixed functions request "Numbers"



6.5 SHAPE ATTRIBUTES OF THE DENSITY FUNCTION (SKEWNESS, KURTOSIS)

6.5.A SKEWNESS

Skewness measures asymmetry around the mean. The parameter is best interpreted as relative to the Normal Density Function (whose Skewness equals zero). The interpretation of the Skewness for a series (relative to the Normal Density Function) is:

— Skewness > 0 → asymmetric tail with more values above the mean.
— Skewness < 0 → asymmetric tail with more values below the mean.

The next three figures shown Density Functions that have a Skewness > 0, = 0, and < 0, respectively, for three variables Y1, Y2 and Y3. (Y2 is distributed Normally).

114

Figure 96: Distribution of series Y1.
Skewness > 0

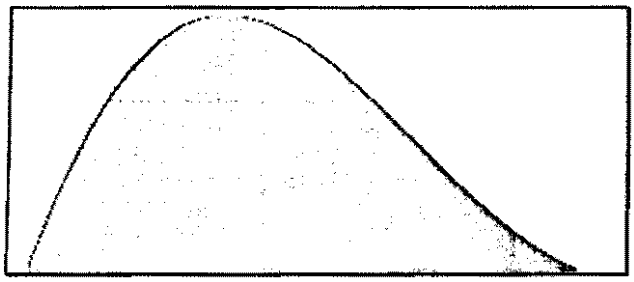


Figure 97: Distribution of series Y2.
Skewness = 0.

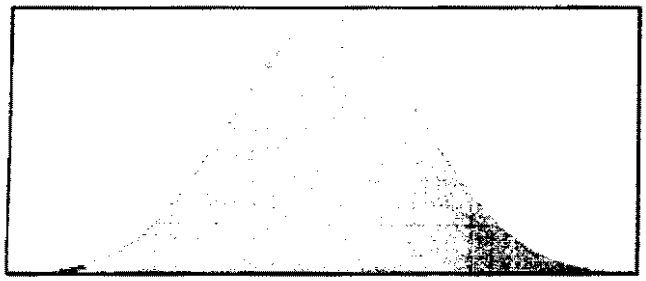
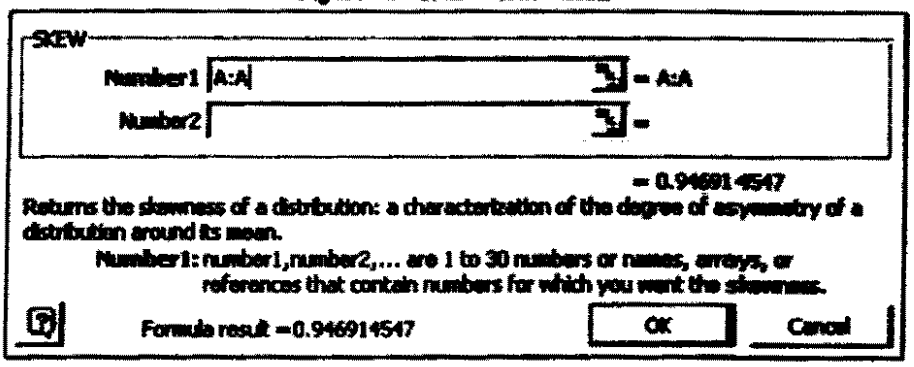


Figure 98: Distribution of series Y3
Skewness < 0

Samples will be available at http://www.vjbooks.net/excel/samples.htm.

Figure 99: SKEW (Skewness)



Menu path to function: INSERT / FUNCTION / STATISTICAL / SKEW

6.5.B KURTOSIS

Compared with the Normal Density Function (which has a Kurtosis of zero), the interpretation of the kurtosis for a series is:

— Kurtosis > 0→ peaked relative to the Normal Density Function
— Kurtosis < 0→ flat relative to the Normal Density Function

The next figure shows three Density Functions. The Density Functions lie around the same Mean and Median, but note the difference in the relative flatness of the Density Functions:

Distribution of series X1 is the flattest with a Kurtosis < 0, that of X2 is less flat with a Kurtosis = 0 (a Normal Density Function) and that of series X3 is the least flat with a Kurtosis > 0.

Figure 100: Example of Density Functions with different Skewness

Samples will be available at http://www.vjbooks.net/excel/samples.htm.

Menu path to function: INSERT / FUNCTION / STATISTICAL / KURT

6.6 FUNCTIONS ENDING WITH AN "A" SUFFIX

These functions calculates the same statistic as their "twin" formula (the one without the prefix "A") but include a wider range of valid cell values in the relevant formula. The "A" –suffixed functions include the following types of cell values:

— Logical (and not numeric) like "True" and "False" (valued as 1 and 0, respectively),

— Blank cells (valued as 0), and

— Text (valued as 0).

A text string or a blank cell is valued as zero. The next table lists these twin functions:

117

Table 13: Functions ending with the "A" suffix.

The non-prefixed function	*The "A" prefixed "twin" formula*	*Comment*
AVERAGE	AVERAGEA	Simple average/mean
COUNT	COUNTA	Count of valid cells. The prefixed function is very useful in counting.
STDEV	STDEVA	Standard deviation
STDEVP	STDEVPA	Standard deviation from a population or a very large sample (relative to population)
VAR	VARA	Variance
VARP	VARPA	Variance from population (and not sample) data, or from a very large sample (relative to population)
MIN	MINA	Minimum value
MAX	MAXA	Maximum value

CHAPTER 7

PROBABILITY DENSITY FUNCTIONS AND CONFIDENCE INTERVALS

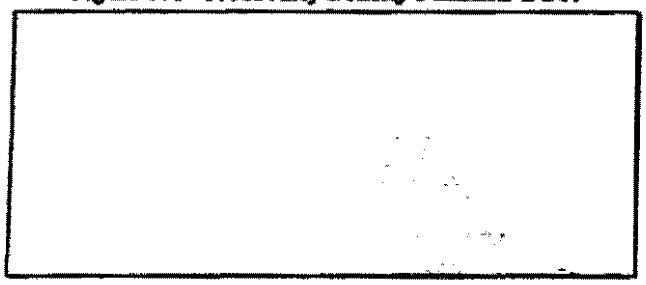
This chapter teaches the following topics

- **PROBABILITY DENSITY FUNCTION (PDF)**
- **CUMULATIVE DENSITY FUNCTION (CDF)**
- **THE CDF AND CONFIDENCE INTERVALS**
- **INVERSE MAPPING FUNCTIONS**
- **NORMAL DENSITY FUNCTION**
- **STANDARD NORMAL OR Z–DENSITY FUNCTION**
- **T–DENSITY FUNCTION**
- **F–DENSITY FUNCTION**
- **CHI-SQUARE DENSITY FUNCTION**
- **OTHER CONTINUOUS DENSITY FUNCTIONS: BETA, GAMMA, EXPONENTIAL, POISSON, WEIBULL & FISHER**
- **DISCRETE PROBABILITIES— BINOMIAL, HYPERGEOMETRIC & NEGATIVE BINOMIAL**
- **LIST OF DENSITY FUNCTION FUNCTIONS — PROBABILITY DENSITY FUNCTION (PDF), CUMULATIVE DENSITY FUNCTION (CDF)**
- **LIST OF SELECT INVERSE FUNCTION**

7.1 PROBABILITY DENSITY FUNCTIONS (PDF), CUMULATIVE DENSITY FUNCTIONS (CDF), AND INVERSE FUNCTIONS

7.1.A PROBABILITY DENSITY FUNCTION (PDF)

Figure 101: Probability Density Function (PDF)



The horizontal axis contains the values of the series/series. The vertical height of the curve at a point on the X–axis shows the probability associated (or frequency) with that point. (The total area under the curve equals 1; so, all the "heights" add up to 1 or 100 %.) The higher the frequency with which that point is observed in a series/series, the higher is its frequency.

An often–used probability Density Function — the "Normal" probability Density Function — is shown in the previous figure. This Density Function has some convenient properties:

— its Mean, Mode and Median are the same
— it does not have a left or right skew, and
— the left half is a mirror image of the right half.

All these "symmetrical" properties allow one to draw inferences from tests run on series that are distributed "Normally."

Based on several theorems, postulate, "most data series start behaving more and more like a series that follows a Normal Density Function as the sample size (or number of data points) increases." (This presumption follows from the "Central Limit Theorem.") This has made the Normal Density Function the bedrock of most statistics and econometrics.

7.1.B CUMULATIVE DENSITY FUNCTION (CDF)

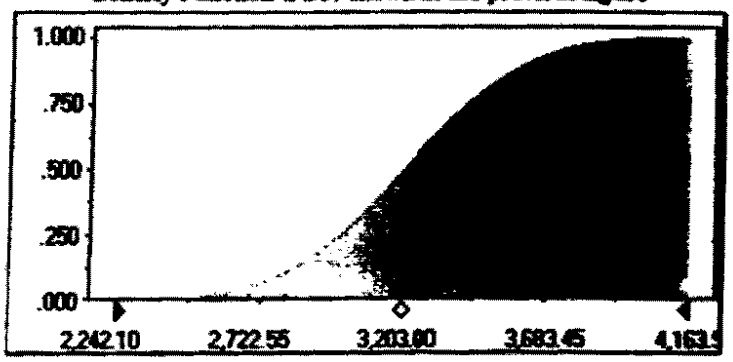
We are typically interested in measuring the area under the curve (a) to the left of an X value (b) to the right of an X value, or (c) between two X values. The height of the curve at any X value is not so useful by itself because it does not answer any of these questions directly.

A better graphical tool to measure the "area under the curve" is the Cumulative Density Function (CDF). A CDF plots the X categories against the "probability of a value taking a value below the chosen X value."

The CDF for the Normal Density Function is reproduced in the next figure. The curve increases from left to right (from 0 to 1[8]). The height at any X-value tells us "the probability of a value having a value below this X-value equals the Y-axis value of the CDF at this X."

[8] The area under any Density Function curve always equals 1. The relative frequency equals (frequency that X takes on this particular value) divided by (the total sample size). Therefore, in a sense, the height gives the frequency weight for each X value. If you sum all the relative frequencies, their sum is "sample size divided by sample size" equals 1. This is the area under the curve. It can also be expressed in percentage terms; the total percentage area then becomes 100%.

Figure 102: The "Cumulative Density Function" (CDF) associated with the Probability Density Function (PDF) shown in the previous figure



The CDF is a better tool for answering the typical questions about the properties of a data series. CDF is of great importance for building Confidence Intervals and implementing hypothesis tests.

In fact, for some Density Functions, Excel only measures the CDF only (and not the CDF & PDF).

The CDF and Confidence Intervals

The concept of a Confidence Interval for a measured parameter (typically for a mean) is based on the concept of probability depicted by a Density Function curve. A Confidence Interval of 95% is a range of X values within whose range the sum of the relative frequencies is 0.95 or 95%.

I will use this property to show how to create Confidence Intervals for various distributions using the Inverse of the CDF. (You will learn more on the Inverse in the next sub-section.)

Table 14: Probability Density Function (PDF) and Cumulative Density Function (CDF)

Function	Is there an option to request the Cumulative Density Function (CDF)?	Cumulative Density Function (CDF) & Probability Density Function (PDF): Information requirements for parameterization			
		Mean	*Std Dev*	*Degrees of freedom*	*Other*
TDIST				✓	Tails #
LOGNORMDIST	✓	✓	✓		
FDIST				✓	2nd degree of freedom
BETADIST				alpha, beta, upper and lower bound	
CHIDIST				✓	
NORMDIST	✓	✓	✓		
NORMSDIST	✓				
WEIBULL					Alpha and beta
NEGBINOMDIST		(Probability)			# of successes
BINOMDIST	✓	(Probability)			
EXPONDIST	✓				Lambda
GAMMADIST	✓				Alpha and beta
HYPGEOMDIST		Sample size, population size, # of successes in population			
POISSON	✓	✓			

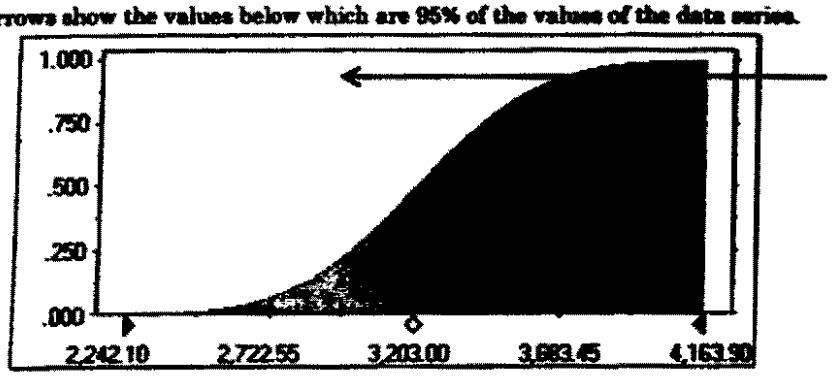
7.1.C INVERSE MAPPING FUNCTIONS

The Cumulative Density Function (CDF) tells us "For any X series, the probability of the value of X falling below a specific x value can be calculated from the height of the Cumulative Density Function (CDF) at that x value."

An inverse function does the reverse mapping: "For a probability *P*, the *X* to who's left the probability of the data lying can be obtained by a reverse reading of the Cumulative Density Function (CDF). That is, from "Desired Cumulative Probability → unknown *X* that will give this desired cumulative probability *P*."

Alternatively, "Inverse" functions find the X value that corresponds to a certain "probability of values below the X equaling a known cumulative probability."

Figure 103: Reading inverse mapping from a Cumulative Density Function (CDF). The arrows show the values below which are 95% of the values of the data series.



Inverse functions permit easy construction of Confidence intervals. This will be shown several times in further sections whenever I discuss the construction of Confidence intervals.

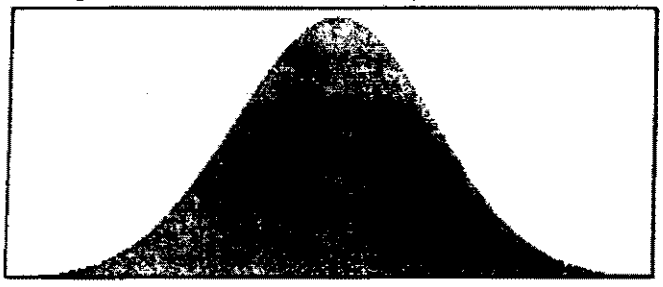
Table 15: Inverse functions (also used to create Confidence intervals). Samples will be available at http://www.vjbooks.net/excel/samples.htm.

Function	Inverse Function ("probability to value") of this Cumulative Density Function (CDF)?	Information required by all inverse functions	Other information requirements			
		Probability for which the corresponding value is sought	Mean	Std Dev	Degrees of freedom	Other
TINV	TDIST	✓			✓	
LOGINV	LOGNORMDIST	✓	✓	✓		
FINV	FDIST	✓			✓	Second degree of freedom
BETAINV	BETADIST	✓				alpha, beta, upper and lower bound
CHIINV	CHIDIST	✓			✓	
NORMINV	NORMDIST	✓	✓	✓		
NORMSINV	NORMSDIST	✓				

7.2 NORMAL DENSITY FUNCTION

The Normal Density Function has several properties that make it easy to make generalized inferences for the attributes of a series whose Density Function can be said to be "Normal."

Figure 104: A Normal Probability Density Function



125

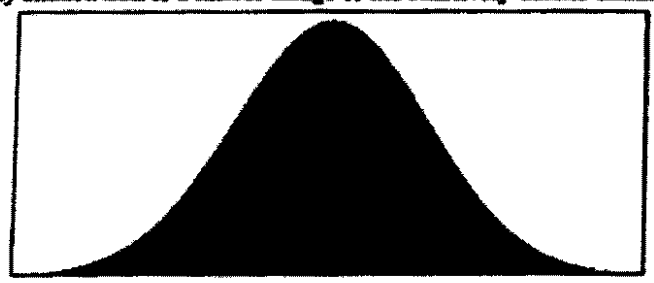
Symmetry

The major measures of central tendency — the mean, median, and mode — all lie at the same point right at the place where the bell shaped curve is at its greatest height.

The Density Function is perfectly symmetrical around this "confluence" of central tendencies. Therefore, the left half of the Density Function (measured as all points to the left of the mode/median/mean) is a mirror image of the right half of the Density Function.

This is shown in the next figure — the lighter shaded half is a mirror image of the darker shaded half. So, the frequency of the values of the variables becomes lower (that is, the height of the curve lowers) as you move away from the mode/mean/median towards either extreme. This change is gradual and occurs at the same rate for negative and positive deviations from the mean.

Figure 105: An idealized "symmetrical" Normal Density Function. Note that the relatively lightly shaded half is a mirror image of the relatively darker shaded half



The symmetry also implies that:

(a) The Density Function is not "skewed" to the left or right of the mode/median/mean (and, thus, the Skewness measure = 0)

(b) The Density Function is not "too" peaked (which would imply that the

change in probability is very rapid when moving from the mode/median/mean towards an extreme) nor "too" flat (which would imply that the change in probability is very slow when moving from the mode/median/mean towards an extreme). The first property implies that Skewness = 0, and the second implies that Kurtosis = 0.

Convenience of using the Normal Density Function

If a series is Normally distributed, then you just need two parameters for defining the Density Function for any series X— the mean and standard deviation of the variables values! This is because, once you know the mean, you also know the mode and median (as these two statistics equal the mean for a Normal Density Function).

Once you know the standard deviation, you know the spread of values around the mean/mode/median. (A series that follows a Normal Density Function is not skewed to the left or right, nor is "too" peaked or "too" flat.)

Are all large-sample series Normally Distributed?

Some formal mathematical theorems and proofs support the theory that "as the sample size gets larger most Density Functions become more like the Normal Density Function." Therefore, for example, if a series has a left skewed Density Function when a sample of 20 observations is used, it may also behave more like a symmetrical (that is, a zero–skewed) Normal Density Function if the sample size is, for example, 1000 observations.

(Even if the Density Function does not have the classic bell–shape of a

normal curve, it can behave like a Normal Density Function if it satisfies — to a sufficient extent — the conditions that imply normality –

- The fact that the mode, mean and median are very close to each other
- An additional feature is that the Density Function is roughly symmetrical around the mode/median/mean.

Statistics & Econometrics: Dependence of Methodologies on the assumption of Normality

Assuming that variables are distributed Normally is a practice that underlies— and even permits— most hypothesis testing in econometrics and statistics. Without this assumption, statistics, as we know it, would lose much of its power to estimate coefficients and establish relationships amongst variables.

Assume you have three variables — X1, X2, and X3. X1 is measured in dollars with a mean of $2.30, X2 also in dollars with a mean of $30,000 and X3 in tons. You assume that all the variables are distributed Normally. This permits you to make inferences about the series. Once you know the mean and standard deviation for X1, you can make statements like "60% of the values of X1 lie below $2.62," "Between the values $24,000 and $28,000, we will find that 18% of the values of X2 will lie," or, "Over 40% of the values of X3 lie below 24 tons." (Note: the figures are chosen arbitrarily). This is fine. But the problem is that the relation between the "mean, standard deviation, X values and probability" must be calculated anew for each of the variables because they are measured in different units (dollars versus tons) or/and on different scales and ranges (X1 versus X2 in our example).

This limits the usefulness of using the Normal Density Function to assess the relation between series values and the probability of values occurring less than, equal to or above them. In practical terms, you would need a statistics textbook that lays out the relationship between an X value and probability for all possible combinations of mean and standard deviation!

The Standard Normal and its power

Luckily, a method removes the need for such exhaustive table listings. This method involves rescaling all series that follows a Normal Density Functions to a common scale such that, on the new scale, the variables have a mean/mode/median of zero and a standard deviation of one. The process is called "standardization" and this standardized Density Function is called the standard Normal Density Function or the Z – Density Function.

The Z –scores are also used to standardize the Density Functions of the means of variables or the estimates of statistical coefficients. If the standard error of mean for the population from which the sample is unknown (as is typically the case), then the T Density Function is used instead of the Z Density Function.

7.2.A THE PROBABILITY DENSITY FUNCTION (PDF) AND CUMULATIVE DENSITY FUNCTION (CDF)

<u>PDF</u>:

NORMDIST (x, mean, standard deviation, false) → probability of values taking the value X

<u>CDF</u>:

NORMDIST (x, mean, standard deviation, true) → probability of values lying to the left of X

Figure 106: The dialog for estimating the probability associated with a value of a point in a series that follows a Normal Density Function

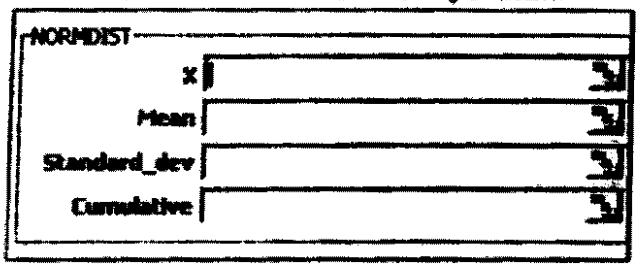
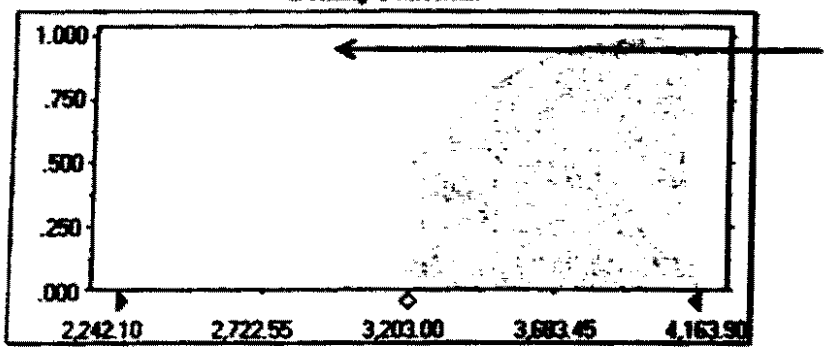


Figure 107: The Cumulative Density Function (CDF) for a series that follows a Normal Density Function. The arrows show the value to the left of which lie 95% of the values in the Density Function.



The Cumulative Density Function (CDF) is the integral of the function on the right hand side in the above equation. The range of integration is negative infinity (or the population minimum) to the X value being studied.

Menu path to function: INSERT / FUNCTION / STATISTICAL / NORMDIST.

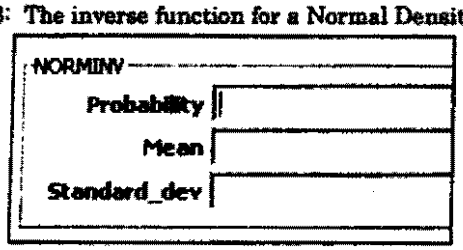
Data requirements: The data series should follow the assumed Density Function type (Normal).

7.2.B INVERSE FUNCTION

This function calculates the inverse of the normal cumulative Density Function for a user–specified mean and standard deviation.

NORMINV (probability below the X, MEAN, STANDARD DEVIATION) → X

Figure 108: The inverse function for a Normal Density Function



7.2.C CONFIDENCE INTERVALS

Menu path to function: INSERT / FUNCTION / STATISTICAL / NORMINV.

Data requirements: The data series should follow the assumed Density Function type (Normal).

95% Confidence Interval

The Confidence Interval contains all but 2.5% of the extreme values on each of the tails of the Density Function (Probability Density Function (PDF)) or is the value that corresponds to 0.025 and 0.975 on the Cumulative Density Function (CDF). The 95% Confidence Interval for a series that follows a Normal Density Function with mean = μ and standard deviation = σ is defined by the results of the two inverse

functions at these two probabilities:

— NORMINV (0.025, mean, standard deviation)
— NORMINV (0.975, mean, standard deviation) for the lower and upper limit

90% Confidence Interval

The 90% Confidence Interval for a series that follows a Normal Density Function with mean = μ and standard deviation = σ is defined by the results of the two inverse functions at these two probabilities:

— NORMINV (0.05, mean, standard deviation)
— NORMINV (0.95, mean, standard deviation) for the lower and upper limit

Table 16: Normal Density Function— Formulas for 90%, 95%, and 99% Confidence limits

Confidence level	Formula for lower bound	Formula for upper bound
90%	NORMINV (0.05, mean, standard deviation)[9]	NORMINV (0.95, mean, standard deviation)
95%	NORMINV (0.025, mean, standard deviation)	NORMINV (0.975, mean, standard deviation)

[9] Note that many books use the following symbols or phrases for the mean and standard deviation"

- μ or "mu" for mean
- σ or "sigma standard deviation/error
- σ^2 or "sigma square variance

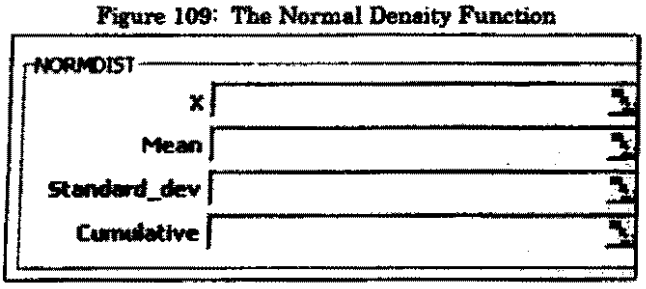
Confidence level	Formula for lower bound	Formula for upper bound
99%	NORMINV (0.005, mean, standard deviation)	NORMINV (0.995, mean, standard deviation)

7.3 STANDARD NORMAL OR Z–DENSITY FUNCTION

The Cumulative Density Function (CDF) is the integral of the function on the right hand side in the above equation. The range of integration is negative infinity to the Z value being studied.

CDF:
NORMSDIST (z) → probability of values lying to the left of Z

Figure 109: The Normal Density Function



Menu path to function: INSERT / FUNCTION / STATISTICAL /NORMSDIST.

Data requirements: The data series 'z' should follow the assumed Density Function type (Standard Normal).

Inverse function

This function calculates the inverse of the Standard Normal CDF. The inverse function for a Standard Normal Density Function requires only one parameter.

NORMSINV (probability below the X) → X

Figure 110: NORMSINV

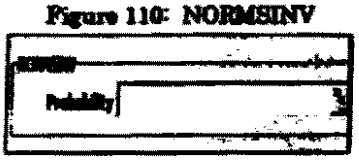
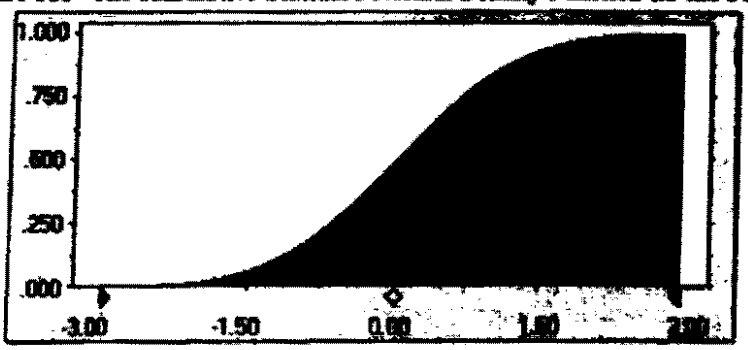


Figure 111: The cumulative Standard Normal Density Function (or the Probit)



Menu path to function: INSERT / FUNCTION / STATISTICAL / NORMSINV. *Data requirements*: The data series 'z' should follow the assumed Density Function type (Standard Normal).

Confidence Intervals

Table 17: Standard Normal Density Function: Formulae for 90%, 95% and 99% Confidence limits.

Confidence level	Formula for lower bound	Formula for upper bound

134

Confidence level	Formula for lower bound	Formula for upper bound
90%	NORMSINV (0.05)	NORMSINV (0.95)
95%	NORMSINV (0.025)	NORMSINV (0.975)
99%	NORMSINV (0.005)	NORMSINV (0.995)

7.4 T–DENSITY FUNCTION

CDF:

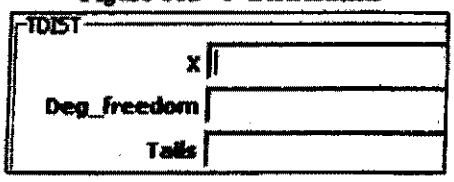
TDIST (x, degrees of freedom, tails) → probability of values lying to the left of X

In the box Tails, specify the number of tails to return.

- If tails = 1, TDIST returns the one–tailed Density Function.
- If tails = 2, TDIST returns the two–tailed Density Function.

For example, TDIST (1.96, 60, 2) equals 0.054645, or 5.46 percent

Figure 112: T–Distribution



Menu path to function: INSERT /FUNCTION /STATISTICAL /TDIST.

Data Requirements: The data series should follow the T Density Function.

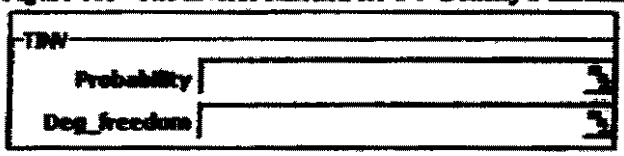
135

Inverse function

This function calculates the t–value of the Student's t–Density Function as a function of the probability and the degrees of freedom.

TINV (probability below the X, degrees of freedom) → X

Figure 113: The inverse function for a T–Density Function



A one–tailed t–value can be returned by replacing probability with 2*probability. For a probability of 0.05 and degrees of freedom of 10, the two–tailed value is calculated with T (0.05, 10), which returns 2.28139. The one–tailed value for the same probability and degrees of freedom can be calculated with T (2*0.05, 10), which returns 1.812462.

TINV (0.054645, 60) equals 1.96

Menu path to function: INSERT /FUNCTION /STATISTICAL /TINV.

Data requirements: The data series should follow the assumed Density Function type (T).

Confidence Intervals

Table 18: T Density Function— Formulae for 90%, 95%, and 99% Confidence limits (2 tails).

Confidence level	Formula for lower bound	Formula for upper bound
90%	TINV (0.05, degrees of freedom)	TINV (0.95, degrees of freedom)

136

Confidence level	Formula for lower bound	Formula for upper bound
95%	TINV (0.025, degrees of freedom)	TINV (0.975, degrees of freedom)
99%	TINV (0.005, degrees of freedom)	TINV (0.995, degrees of freedom)

7.4.A ONE-TAILED CONFIDENCE INTERVALS

95% Confidence Interval

A 95 % Confidence Interval contains all but 5% of the extreme values on one–tail of the Density Function (Probability Density Function (PDF)) or is the value that corresponds to 0.05 or 0.95 on the Cumulative Density Function (CDF) (the former for the left tail of 5% and the latter for a right tail of 5%).

The 95% Confidence Interval for a T–distributed series is defined by the results of the two inverse functions at this probability:

Left tail: Negative infinity to TINV (0.05, degrees of freedom).

Right tail: TINV(0.95, degrees of freedom) to positive infinity.

Note:
TINV(0.05, degrees of freedom) = –TINV(0.95, degrees of freedom)

90% Confidence Interval

A 90 % Confidence Interval contains all but 10% of the extreme values on one–tail of the Density Function (Probability Density Function (PDF)) or

is the value that corresponds to 0.1 or 0.9 on the Cumulative Density Function (CDF) (the former for the left tail of 10% and the latter for a right tail of 10%).

The 90% Confidence Interval for a T–distributed series is defined by the results of the two inverse functions at this probability:

- Left tail: Negative infinity to TINV (0.1, degrees of freedom).
- Right tail: TINV (0.9, degrees of freedom) to positive infinity.

Note: TINV(0.1, degrees of freedom) = –TINV(0.9, degrees of freedom)

Table 19: T Density Function— Formulae for 90%, 95%, and 99% Confidence limits (right tail only).

Confidence level	Formula for lower left-tail Confidence upper limit (the lower limit equals negative infinity)
90%	TINV (0.9, degrees of freedom)
95%	TINV (0.95, degrees of freedom)
99%	TINV (0.99, degrees of freedom)

Table 20: T Density Function— Formulae for 90%, 95%, and 99% Confidence limits (left tail only)

Confidence level	Formula for right-tail Confidence lower limit (the upper limit equals positive infinity)
90%	–TINV (0.1, degrees of freedom)
95%	–TINV (0.05, degrees of freedom)
99%	–TINV (0.01, degrees of freedom)

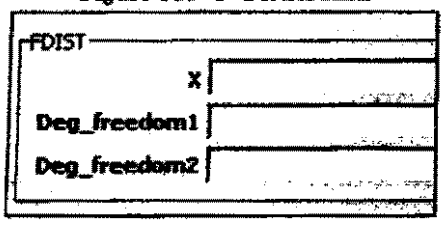
7.5 F–DENSITY FUNCTION

The F test is used for testing model significance and other joint hypothesis in ANOVA, Regression Analysis, etc.

CDF:
FDIST (x, numerator degrees of freedom, denominator degrees of freedom)

Menu path to function: INSERT / FUNCTION / STATISTICAL / FDIST.

Figure 114: F–Distribution



Data requirements: The data series should follow the assumed Density Function type (F).

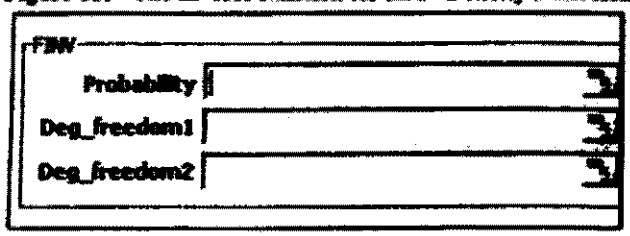
Inverse function

FINV (probability below the X, numerator degrees of freedom, denominator degrees of freedom) → X

Menu path to function: INSERT / FUNCTION / STATISTICAL /FINV.

Data requirements: The data series should follow the F Density Function.

Figure 115: The inverse function for an F-Density Function



One-tailed Confidence Intervals

Table 21: F Density Function— Formulas for 90%, 95%, and 99% Confidence Intervals (right tail only).

Confidence level	Formula for upper One-tail Confidence lower limit (the upper limit equals positive infinity)
90%	FINV (0.9, numerator degrees of freedom, denominator degrees of freedom)
95%	FINV (0.95, numerator degrees of freedom, denominator degrees of freedom)
99%	FINV (0.99, numerator degrees of freedom, denominator degrees of freedom)

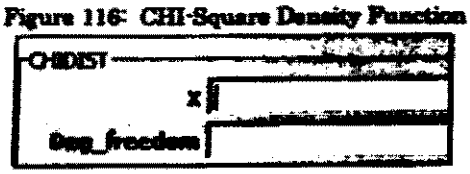
7.6 CHI-SQUARE DENSITY FUNCTION

The Chi-square test is used for testing model significance and other joint hypothesis in Maximum Likelihood estimation, Logit, Probit, etc.

<u>The one-tailed probability of the Chi-Square Density Function: CDF:</u>

CHIDIST (x, degrees of freedom) → probability of values lying to the left of X

Figure 116: CHI-Square Density Function



Menu path to function: INSERT /FUNCTION /STATISTICAL /CHIDIST.

Data requirements: The data series should follow the assumed Density Function type (Chi-Square).

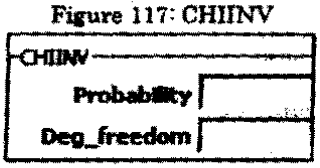
Inverse function

CHIINV (probability, degrees of freedom) → X

Menu path to function: INSERT /FUNCTION /STATISTICAL /CHIINV.

Data requirements: The data series should follow the Chi-Square Density Function.

Figure 117: CHIINV



One-tailed Confidence Intervals

Table 22: Chi-Square Density Function: Formulae for 90%, 95%, and 99% Confidence limits (right tail only). Samples will be available at http://www.vjbooks.net/excel/samples.htm.

Confidence level	Formula for upper One-tail Confidence lower limit (the upper limit equals positive infinity)
90%	CHIINV (0.9, degrees of freedom)
95%	CHIINV (0.95, degrees of freedom)
99%	CHIINV (0.99, degrees of freedom)

7.7 OTHER CONTINUOUS DENSITY FUNCTIONS: BETA, GAMMA, EXPONENTIAL, POISSON, WEIBULL & FISHER

7.7.A BETA DENSITY FUNCTION

CDF:

BETADIST (x, alpha, beta, lower bound A, upper bound B) → probability of values lying to the left of X

Menu path to function: INSERT / FUNCTION / STATISTICAL / BETADIST.

Data requirements: **The data series should follow the Beta Density Function.**

Figure 118: BETA Density Function

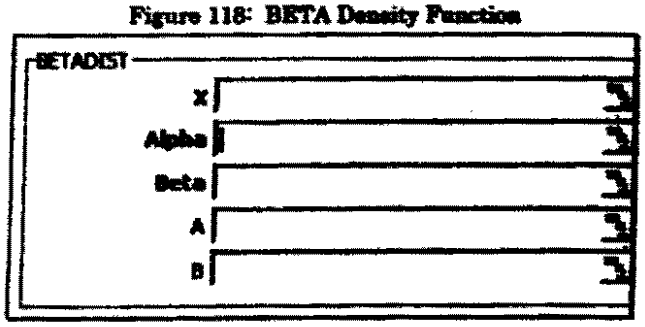


Figure 119: Note how the Density Function Probability Density Function (PDF) is skewed to one side and has a less sharp "hill" at top — compared to a Normal Probability Density Function (PDF)

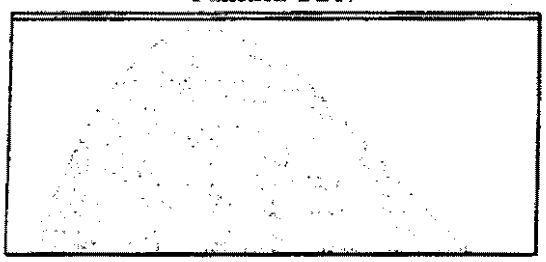


Figure 120: The Cumulative Density Function (CDF) shows (on the Y –Axis) the proportion of values that lie below a certain X value of the series

Inverse Function

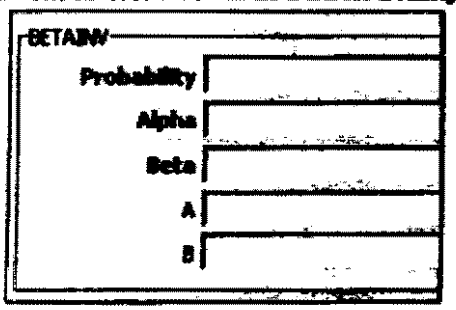
BETAINV (probability, alpha, beta, lower bound A, upper bound B) → X

Menu path to function: INSERT / FUNCTION / STATISTICAL / BETAINV.

Data requirements: The data series should follow the assumed Density Function type (Beta).

143

Figure 121: The inverse function for a BETA Density Function



Confidence Intervals

Table 23: BETA Density Function— Formulae for 90%, 95%, and 99% Confidence limits.

Confidence level	Formula for lower bound	Formula for upper bound
90%	BETAINV (0.05, alpha, beta, A, B)	BETAINV (0.95, alpha, beta, A, B)
95%	BETAINV (0.025, alpha, beta, A, B)	BETAINV (0.975, alpha, beta, A, B)
99%	BETAINV (0.005, alpha, beta, A, B)	BETAINV (0.995, alpha, beta, A, B)

7.7.8 GAMMA DENSITY FUNCTION

The Gamma Density Function is commonly used in queuing analysis.

<u>CDF</u>:

GAMMADIST (x, Alpha, Beta, true) → probability of values lying to the left of X)

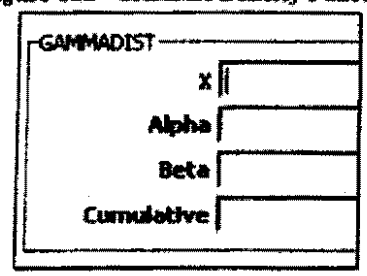
<u>PDF</u>:

GAMMADIST (x, Alpha, Beta, false) → probability of values taking the value X)

Menu path to function: INSERT /FUNCTION /STATISTICAL

/GAMMADIST.

Figure 122: GAMMA Density Function



Data requirements: The data series should follow the assumed Density Function type (Gamma).

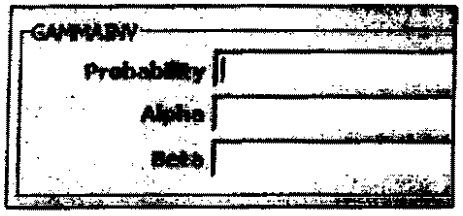
Inverse Function

GAMMAINV (probability below the X, alpha, beta) → X

Menu path to function: INSERT /FUNCTION /STATISTICAL /GAMMAINV.

Data requirements: The data series should follow the assumed Density Function type (Gamma).

Figure 123: The inverse function for a GAMMA Density Function



145

Confidence Intervals

Table 24: Gamma Density Function: Formulae for 90%, 95% and 99% Confidence limits. Samples will be available at http://www.vjbooks.net/excel/samples.htm.

Confidence level	Formula for lower bound	Formula for upper bound
90%	GAMMAINV (0.05, alpha, beta)	GAMMAINV (0.95, alpha, beta)
95%	GAMMAINV (0.025, alpha, beta)	GAMMAINV (0.975, alpha, beta)
99	GAMMAINV 0005, alpha, beta	GAMMAINV 0995, alpha, beta

If an inverse function does not converge after 100 iterations, the function returns the #N/A error value.

7.7.C EXPONENTIAL DENSITY FUNCTION

PDF:

EXPONDIST (x, lambda, False) → probability of values taking the value X

CDF:

EXPONDIST (x, lambda, True) → probability of values lying to the left of X

Menu path to function: INSERT /FUNCTION /STATISTICAL /EXPONDIST.

Data requirements: The data series should follow the Exponential Density Function.

Figure 124: Dialog for the Exponential Distribution

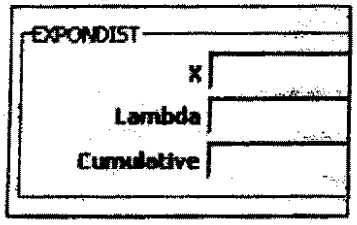


Figure 125: Exponential Probability Density Function (PDF)

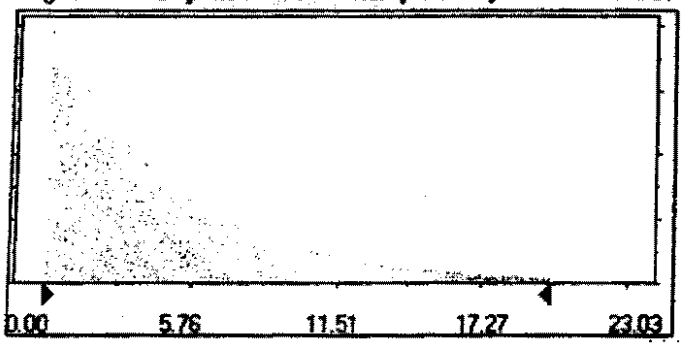
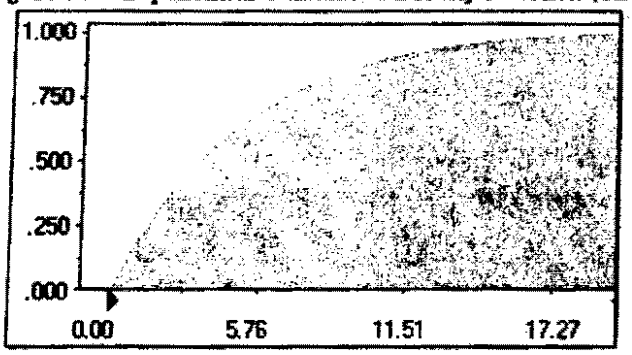


Figure 126: Exponential Cumulative Density Function (CDF)



EXPONDIST (0.2, 10, TRUE) equals 0.864665 while EXPONDIST (0.2, 10, FALSE) equals 1.353353

Further detail is beyond the scope of this book.

7.7.D FISHER DENSITY FUNCTION

This topic is beyond the scope of this book.

7.7.E POISSON DENSITY FUNCTION

This Density Function is used for predicting the number of events 'X' occurring over a specific time.

PDF:
POISSON (x, expected value, false) → probability of values taking the value X

CDF:
POISSON (x, expected value, true) → probability of values lying to the left of X

Further detail is beyond the scope of this book.

7.7.F WEIBULL DENSITY FUNCTION

PDF:
WEIBULL (x, a, b, false) → probability of values taking the value X

CDF:
WEIBULL (x, a, b, true) → probability of values lying to the left of X

Further detail is beyond the scope of this book.

7.7.G DISCRETE PROBABILITIES— BINOMIAL, HYPER GEOMETRIC & NEGATIVE BINOMIAL

This topic is beyond the scope and aim of this book.

Binomial Density Function

This function is used to ascertain the probability of obtaining a "head" in a coin toss. X can take only two discrete values. Further detail is beyond the scope of this book.

Hypergeometric Density Function

The Density Function captures event probabilities in problems of sampling without replacement. The sample is taken from a discrete finite population like a deck of cards. Further detail is beyond the scope of this book.

Negative Binomial

This function measures the probability of "number of coin tosses before first or K^{th} heads (in a coin toss)."

7.8 LIST OF DENSITY FUNCTION

Table 25: PDF and CDF functions

Function	Is there a function that does the converse of this mapping and, if so, the name of the function?	Is there an option to request the cumulative probability?	Information required by all functions	Other information requirements			
			Value (s) for which the probability is being sought	Mean	Std Dev	Degrees of freedom	Other
TDIST	TINV		✓			✓	
LOGNORMDIST	LOGINV	✓	✓	✓	✓		
FDIST	FINV		✓			✓	2nd degree of freedom
BETADIST	BETAINV		✓	alpha, beta, upper and lower bound			
CHIDIST	CHIINV		✓			✓	
NORMDIST	NORMINV	✓	✓	✓	✓		
NORMSDIST	NORMSINV	✓	✓				
WEIBULL			✓				alpha, beta
NEGBINOMDIST	—		# of failures	(Probability)			# of successes
BINOMDIST	—	✓	# of successes	(Probability)			
EXPONDIST	—	✓	✓				Lambda
GAMMADIST	GAMMAINV	✓	✓				alpha, beta
HYPGEOMDIST	—		# of successes in sample	Sample & population size, # of successes in population			
POISSON	—	✓	✓	✓			

7.9 SOME INVERSE FUNCTION

Table 26: Inverse Functions

Function	Inverse mapping ("probability to value") of which cumulative probability function?	Information required by all inverse functions	Other information requirements			
		Probability for which the corresponding value is sought	Mean	Std Dev	Degrees of freedom	Other
TINV	TDIST	✓			✓	
LOGINV	LOGNORMDIST	✓	✓	✓		
FINV	FDIST	✓			✓	2nd degree of freedom
BETAINV	BETADIST	✓		alpha, beta, Upper and lower bound		
CHIINV	CHIDIST	✓			✓	
NORMINV	NORMDIST	✓	✓	✓		
NORMSINV	NORMSDIST	✓				

CHAPTER 8

OTHER MATHEMATICS & STATISTICS FUNCTIONS

This chapter briefly displays some other functions available in Excel. The topics in this chapter are:

- **COUNTING AND SUMMING**
- **COUNT, COUNTA**
- **COUNTBLANK**
- **COMPARING COUNT, COUNTA AND COUNTBLANK**
- **SUM**
- **PRODUCT**
- **SUMPRODUCT**
- **THE "IF "COUNTING AND SUMMING FUNCTIONS**
- **SUMIF**
- **COUNTIF**
- **TRANSFORMATIONS (LIKE LOG, EXPONENTIAL, ABSOLUTE, ETC)**
- **STANDARDIZING A SERIES THAT FOLLOWS A NORMAL DENSITY FUNCTION**
- **DEVIATIONS FROM THE MEAN**
- **CROSS SERIES RELATIONS**
- **COVARIANCE AND CORRELATION FUNCTIONS**
- **SUM OF THE SUM OF THE SQUARES OF TWO VARIABLES**

— SUM OF THE SQUARES OF DIFFERENCES ACROSS TWO VARIABLES

— SUM OF THE DIFFERENCE OF THE SQUARES OF TWO VARIABLES

8.1 COUNTING AND SUMMING

COUNT function

This function counts the number of valid cells in a range. Cells are valid only if there value is numeric or a date.

Menu path to function: INSERT /□FUNCTION /□STATISTICAL /□COUNT.

Data requirements: Numbers and dates are included in the count. Not counted cells include those that contain error values, text, blank cells, and logical values (like TRUE and FALSE). The X values can be input as references to one or more ranges that may be non–adjacent.

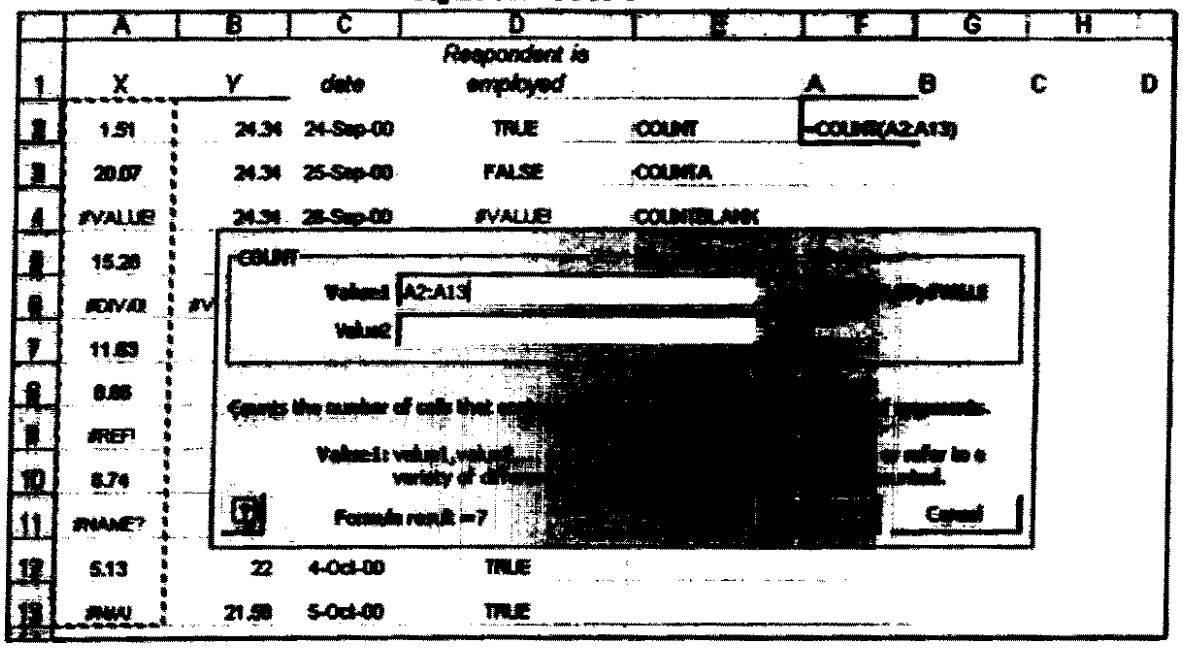
The second range can be referenced in the first text-box "Value1" after placing a comma after the first range, or it could be referenced in the second text-box "Value2."

If you use the second text-box, then a third text-box "Value3" will automatically open. (As you fill the last visible box, another box opens until the maximum number of boxes — 30 — is reached.)

Table 27: Sample data for the "Count" functions.
The example is in the sample file "Count.xls."

A	B	C	D
	Y	Date	Respondent is employed
.51	24.34	24— Sep— 2000	TRUE
20.07	24.34	25— Sep— 2000	FALSE
VALUE!	24.34	26— Sep— 2000	#VALUE!
15.28	24.34	27— Sep— 2000	FALSE
DIV/0!	#VALUE!	28— Sep— 2000	TRUE
11.63	24.34	29— Sep— 2000	#N/A!
.86		30— Sep— 2000	TRUE
REF!	22.00	1— Oct— 2000	FALSE
.74	22.00		TRUE
NAME?	22.00	3— Oct— 2000	
.13	22.00	4— Oct— 2000	TRUE
N/A!	21.58	5— Oct— 2000	TRUE

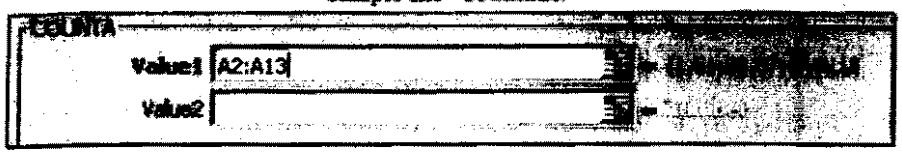
Figure 127: COUNT



COUNTA function also counts cells with logical or text values

This function counts the number of valid cells in a range. Valid values include cells with numeric, date, text, logical, or error value. COUNTA only excludes empty cells, but text and logical values are only counted if you type them directly into the list of arguments are counted. If an argument is a data array or range reference, only numbers in that data array or range reference.

Figure 128: The function COUNTA is a variant of the COUNT function. The example is in the sample file "Count.xls."



Menu path to function: INSERT / FUNCTION / STATISTICAL / COUNTA.

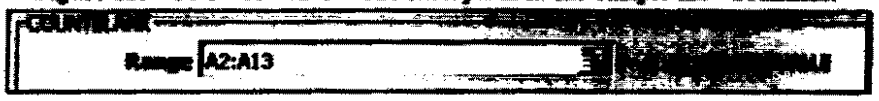
Data requirements: Unlike the COUNT function, COUNTA will include the label row in the count. (So, if you have one label in the referenced range, you may want to use "= COUNTA (A:A) — 1".) The X values can be input as references to one or more ranges that may be non–adjacent. The second range can be referenced in the first text-box "Value1" after placing a comma after the first range, or it could be referenced in the second text-box "Value2." If you use the second text-box, then a third text-box "Value3" will automatically open. (As you fill the last visible box, another box opens until the maximum number of boxes — 30 — is reached.) The function does not count invalid cell values when counting the number of X values.

COUNTBLANK function counts the number of empty cells in the range reference

This function counts the number of blank cells in a range.

Menu path to function: INSERT /FUNCTION /INFORMATION/COUNTBLANK.

Figure 129: COUNTBLANK. The example is in the sample file "Count.xls."



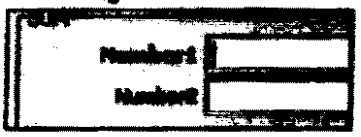
SUM function

This function sums the values in the data array.

$$SUM = X_1 + X_2 + ... + X_n$$

Menu path to function: INSERT / FUNCTION / MATH / SUM.

Figure 130: SUM



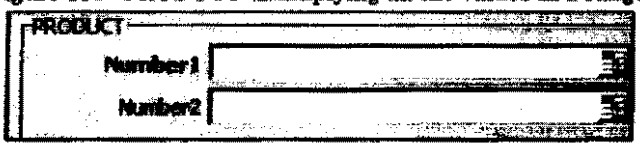
Data requirements: This function does not include blank cells or cells with values that are of the following formats: text, and logical values (that is, TRUE and FALSE.)

PRODUCT function

This function multiplies all the values referenced.

$$PRODUCT = X_1 * X_2 * \ldots * X_n$$

Figure 131: PRODUCT (multiplying all the values in a range)



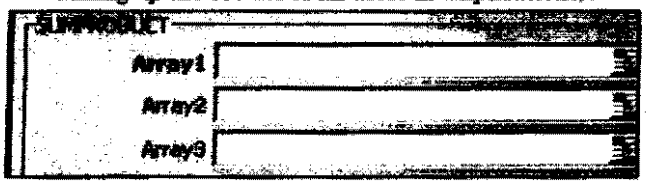
Menu path to function: INSERT / FUNCTION / MATH / PRODUCT.

SUMPRODUCT function

This function multiplies corresponding components in two or more data arrays/ranges, and then sums the results of these multiplications. The data arrays/ranges must have the same number of data points.

Menu path to function: INSERT /FUNCTION /MATH /SUMPRODUCT

Figure 132: SUMPRODUCT (multiplying individual data points across data series and then adding up the results of all these multiplications).



Data Array1, data Array2, data Array3 ... are 2 to 30 data arrays/ranges whose components you desire to multiply and then add. The minimum number of arrays is two. The data arrays must have the same number of data points. Non-numeric cell values are assigned the value of zero.

The X values can be input as references to two or more ranges that may be non–adjacent. The second range should be referenced in the second text-box "Array2." If you use the third text-box, then a fourth text-box "Array4" will automatically open. (As you fill the last visible box, another box opens until the maximum number of boxes — 30 — is reached.)

Example

The following formula multiplies all the components of the two data arrays on the preceding worksheet and then adds the products— that is, 3*2 + 4*7 + 8*6 + 6*7 + 1*5 + 9*3.

Note:
Samples will be available at http://www.vjbooks.net/excel/samples.htm.

8.2 THE "IF" COUNTING AND SUMMING FUNCTIONS: STATISTICAL FUNCTIONS WITH LOGICAL CONDITIONS

I display two "if-then" two-step functions in this section. The functions first evaluate a criterion. If a cell in the referenced range satisfies the criteria then the second part of the function includes this cell.

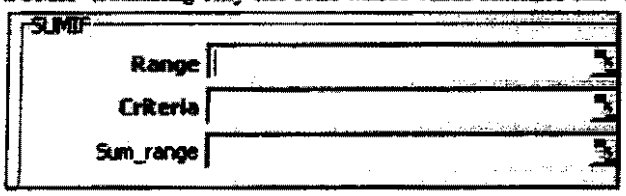
SUMIF function

This function adds the values in a range if the cell with the value satisfies a user-defined criterion.

- In the box *Range*, enter a reference to the range of cells you want

evaluated.

Figure 133: SUMIF (summing only the cells whose value satisfies one "if" condition)



- In the box *Criteria*, enter the condition (a number, expression, or text) that defines which cells values will be summed. For example, *Criteria* can be expressed as 32, "32," ">32".

- In the box *Sum_range*, you may reference the actual cells to sum. The cells in sum range are summed only if their corresponding cells in the entire *Range* match the criteria. If sum range is omitted, all the "criterion-satisfying" cells in the *Range* are summed.

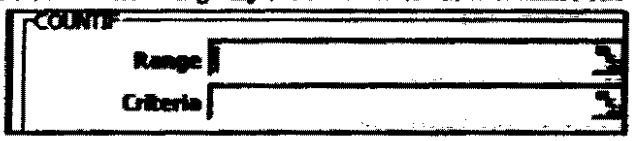
Menu path to function: INSERT /□FUNCTION /□MATH /□SUMIF. The Criteria should be relevant to the type of data/text in the queried range.

COUNTIF function

This function counts the number of cells in a range that satisfy a user-defined criterion.

The dialog for "COUNTIF" requires two inputs from the user. The "Range" is similar to the functions shown previously. The "Criteria" is a logical condition set by you.

Figure 134: COUNTIF (counting only the cells whose value satisfies one "if" condition)



- In the box *Range*, enter a reference to the range of cells you seek to evaluate.

- In the box *Criteria*, enter the condition (a number, expression, or text) that defines which cells will be counted. For example, *Criteria* can be expressed as 32, "32," ">32," "tea."

Menu path to function: INSERT /FUNCTION /STATISTICAL /COUNTIF.

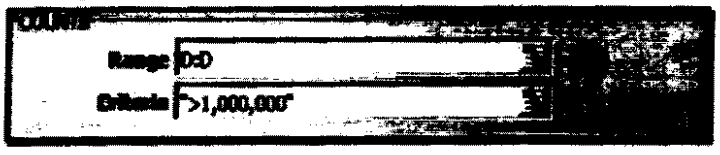
Data requirements: The range can take any values. The Criteria should be relevant to the type of data/text in the queried range.

Example

Choose the range "D:D" and the condition ">1,000,000". The function is "Count the number of cases in the range D:D, but only if the value of the cell is greater than 1 million."

For a pictorial reproduction of this, see the next figure.

Figure 135: Entering the data input and logical criterion



Execute the dialog by clicking on the button OK. The formula is written

onto the cell. The next figure illustrates this. Depress the ENTER key.

Figure 136: The function as written into the cell

=COUNTIF(D:D,">1,000,000")

8.3 TRANSFORMATIONS (LOG, EXPONENTIAL, ABSOLUTE, SUM, ETC)

Table 28: Common transformation functions

Function	*Description*	*Location within INSERT /FUNCTION*	*Data Requirements*
Sign	This function outputs the sign of a number. Returns 1 if the number is positive, zero (0) if the number is 0, and –1 if the number is negative. Useful for red–flagging data, or using in functions like IF, COUNTIF, SUMIF and CHOOSE.	MATH /SIGN	Any real value.
Absolute number	ABS = | X |	MATH /ABS	One real number.
Square root	The square root of a number.	MATH/SQRT	One positive real number.

160

Function	*Description*	*Location within INSERT /FUNCTION*	*Data Requirements*
	$Y = X^{1/2}$		
Log natural	LN (X) This function calculates the natural logarithm of a number. Natural logarithms are based on the constant *e* (2.718). LN (85) = 4.454347. This mean: "If you raise the base e to the power of 4.45 you will get 85. → LN (85) = 4.45. Conversely, exp (4.45) = e^ (4.45) = 2.718^ (4.45) = 85.	MATH /LN	One positive real number.
Exponential	This function calculates the exponential to a number.	MATH /EXP	One positive real number.
Log to the base 10	LOG10 (X) This function calculates the base 10 logarithm of a number. LOG10 (85) = 1.934 because the base of 10 needs to be raised 1.934 times to get 85: $10^{1.934} = 85$ LOG10 (10) = 1 because $10^1 = 10$ LOG10 (1000) = 3 because $10^3 = 1000$	MATH /LOG10	One positive real number.

Function	*Description*	*Location within INSERT /FUNCTION*	*Data Requirements*
Log to a user defined base	This function calculates the logarithm of a number to the base you specify. The default base is 10. For natural log use base e = 2.718. LOG (X, base) LOG (100) = 2 → base 10. (Since $10^2 = 100$). LOG (27, 3) = 3 → base 3. (Since $3^3 = 27$). LOG (86, 2.7182818) = 4.45 → same as natural log. Because— (exp (4.45) = 85).	MATH/LOG.	A positive real number *X* and the (optional) *base* of the logarithm. If base is omitted, it is assumed = 10.

Standardizing a series that follows a Normal Density Function

Converts a value in a series X to its equivalent standard normal transformation.

STANDARDIZE (x, AVERAGE (X), STDEV (X)) where X is all the numbers in the X data series.

Menu path to function:
INSERT/FUNCTION/STATISTICAL/STANDARDIZE.

Data requirement: The function requires three input numbers: x, mean of the X series, and the standard deviation of the X series. The mean and standard deviation can be written as a "function within a function."

8.4 DEVIATIONS FROM THE MEAN

The formulas in this and the next section provide estimates of functions used in formulas for parameters obtained in advanced analysis like ANOVA, Correlation, Regression, etc.

DEVSQ

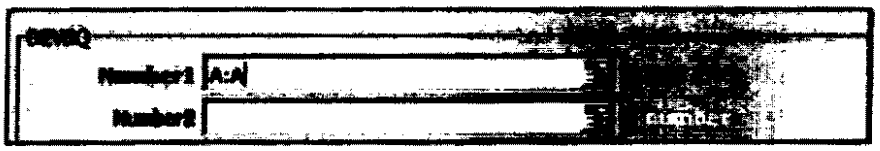
This function calculates the sum of squares of deviations of data points from their sample mean

$$\Sigma\,((x - \text{mean}\,(x))^2$$

Menu path to function: MATH/DEVSQ

Data Requirements: A range(s) of real numbers, inclusive of zero.

Figure 137: Summation of the squares of the "differences of individual points from the mean of the series"



AVEDEV

This function calculates the average of the absolute deviations of data points from their mean. AVEDEV is a measure of the variability in a data set.

$$\frac{1}{n}\sum|x-\bar{x}|$$

Menu path to function: STATISTICAL/AVEDEV

Menu path to function: A range(s) of real numbers, inclusive of zero.

8.5 CROSS SERIES RELATIONS

8.5.A COVARIANCE AND CORRELATION FUNCTIONS

The functions are CORREL, COVAR, PEARSON, & RSQ. I recommend using the Analysis ToolPak Add-In — refer to 10.3.

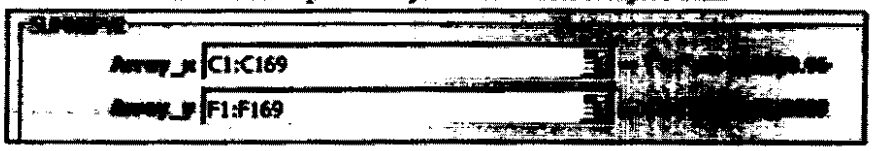
8.5.B SUM OF SQUARES

SUMX2PY2 function evaluates the "Sum of the sum of the squares of each case in two variables"

This function estimates the summation of the squares of individual points in two series.

$$\Sigma\,(x^2 + y^2)$$

Figure 138: Summation of the squares of individual points in two series. Samples will be available at http://www.vjbooks.net/excel/samples.htm.



Menu path to function: INSERT/FUNCTION/MATH/SUMX2PY2.

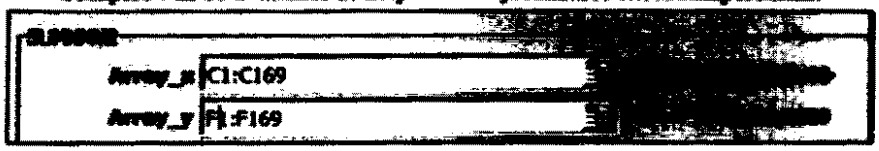
Data requirements: This function needs two data series.

SUMXMY2 function

This function estimates Sum of the squares of differences of each case in two across two variables.

$$\Sigma\ ((x - y)^2)$$

Figure 139: Summation of the squares of the "differences in individual points in two series." Samples will be available at http://www.vjbooks.net/excel/samples.htm.



Menu path to function: INSERT/FUNCTION/MATH/SUMXMY2. *Data requirements*: This function needs two data series.

SUMX2MY2 function

This function estimates the Sum of the difference of the squares of each case in two variables.

165

$\Sigma\ (x^2 - y^2)$

Menu path to function: INSERT/FUNCTION/MATH/SUMX2MY2.

Data requirements: This function needs two data series.

CHAPTER 9

ADD-INS: ENHANCING EXCEL

This chapter discusses the following topics:

— WHAT CAN AN ADD-IN DO?

— WHY USE AN ADD-IN (AND NOT JUST EXCEL MACROS/PROGRAMS)?

— ADD–INS INSTALLED WITH EXCEL

— OTHER ADD-INS

— THE STATISTICS ADD-IN

— CHOOSING THE ADD-INS

9.1 ADD-INS: INTRODUCTION

An "Add-In" is a software application that adds new functionality to Excel. The Add-In typically seamlessly fits into the Excel interface, providing accessibility to its functionality through

— new menus

— new options in existing menus

— new functions

— new toolbars and specific toolbar icons

9.1.A WHAT CAN AN ADD-IN DO?

Almost anything an imaginative software developer could create. Usually, an Add-In provides functionality that is useful for a particular type of analysis/industry — statistics, finance, real estate, etc.

9.1.B WHY USE AN ADD-IN?

The Add-In could have its base code written in software languages like C, C++, FORTRAN, Pascal, etc. This is important because some algorithms and operations (like simulations) operate best when written in a specific language. Therefore, the developer uses the best language/tool to create the functionality and then packages this inside an Add-In.

9.2 ADD–INS INSTALLED WITH EXCEL

Some Add–Ins are available in the Microsoft Office CD–ROM and are installed (but not activated[10]) along with Excel. I show the use of three Add–ins.

[10] Figure 540 and Figure 542 show how to activate the Add-ins

9.3 OTHER ADD-INS

Many commercially sold Add-Ins can be almost like separate software just needing Excel as the "host." Two examples:

— Crystal Ball™ risk analysis software

— *UNISTAT*™ software for conducting advanced statistics and econometrics from inside Excel

Hundreds of software companies construct Add-Ins. The greatest contribution of this book, if I succeed in doing so, would be the opening of this massive potential functionality to Excel users.

9.4 THE STATISTICS ADD-IN

The Analysis ToolPak Add-In that ships with Excel can conduct several procedures including descriptives, regression, ANOVA, F-test, correlation, T-tests, moving average, and histogram. Let us learn how to use this "Add-In."

9.4.A CHOOSING THE ADD-INS

Choose the menu option TOOLS/ADD-INS. You will see several Add-Ins as shown in Figure 140. (You may not see all the Add–Ins shown in the next two figures.)

Figure 140: Selecting an Add-In

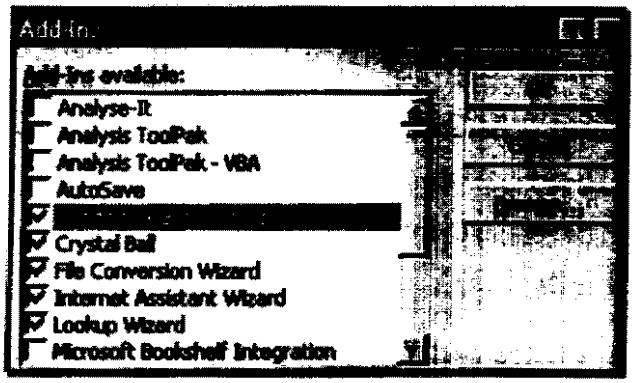
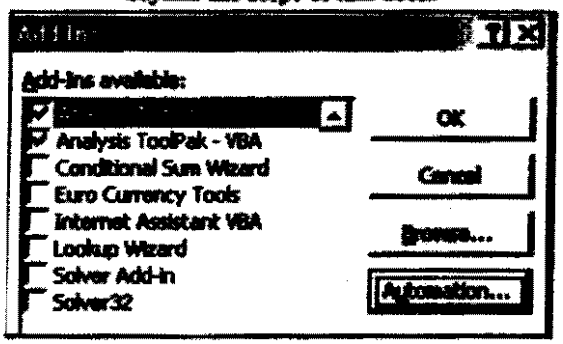


Figure 141: In Excel XP, the Add-Ins dialog provides access to "Automation." This topic is beyond the scope of this book.



You need the "Analysis ToolPak Add-Ins." Select — by clicking on it — the box to the left of these Add-Ins (shown in Figure 142). Execute the dialog by clicking on the button OK and wait for some time while the Add-Ins are "loaded" or "registered" with Excel. An Add-In has to be loaded/registered before it is available for use. The Add-In remains loaded across sessions. It is only "unloaded" when you select the option

TOOLS/ADD-INS and deselect the Add-In[11].

Figure 142: The Add-In pair for data analysis

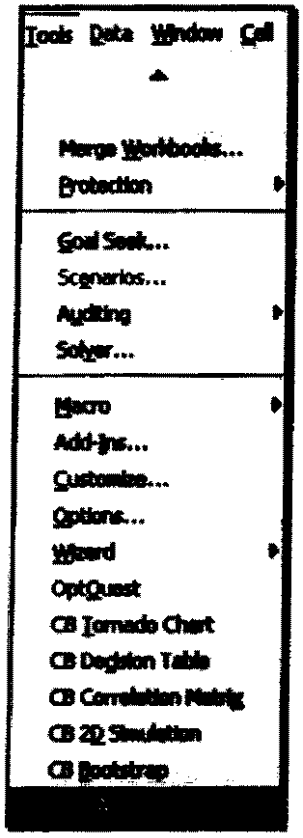
You have activated the "Analysis ToolPak." At the bottom of the menu TOOLS, you will see the option "DATA ANALYSIS the bottom— this option was not there before you accessed the Add-In. (This is illustrated in Figure 143.)

The statistical procedures are accessed through this new option.

> Note:
> Usually Add-Ins expose their functionality by creating new menu options or even new menus. The menu option "Data analysis" provides the statistics functionality available in "Analysis ToolPak" and "Analysis ToolPak VB." The menu options "Optquest" down till CB Bootstrap" are linked to the Add-in "Crystal Ball" (not shipped in the Office CD-ROM).

[11] If too many Add-Ins are loaded, Excel may work too slowly, or even freeze. If you find this problem occurring, then just load the Add-in when you are going to use it and unload it before quitting Excel.

Figure 143: The "Data Analysis" menu option



CHAPTER 10

STATISTICS TOOLS

This chapter discusses the following topics:

— DESCRIPTIVE STATISTICS

— RANK AND PERCENTILE

— BIVARIATE RELATIONS— CORRELATION, COVARIANCE

A proper analysis of data must begin with an analysis of the statistical attributes of each series in isolation — univariate analysis. From such an analysis, you can learn:

— How the values of a series are distributed — normal, binomial, etc.

— The central tendency of the values of a series (mean, median, and mode)

— Dispersion of the values (standard deviation, variance, range, and quartiles)

— Presence of outliers (extreme values)

The answer to these questions illuminates and motivates further, more complex, analysis. Moreover, failure to conduct univariate analysis may restrict the usefulness of further procedures (like correlation and regression). Reason: even if improper/incomplete univariate analysis may not directly hinder the conducting of more complex procedures, the interpretation of output from the latter will become difficult (because you will not have an adequate understanding of how each series behaves).

Note: I do not go into the details of each statistics procedure. For such details, refer to your statistics textbook or to "SPSS for Beginners" (available at http://www.vjbooks.net and amazon.com).

This chapter requires the Analysis ToolPak Add-Ins; chapter 9 shows how to learn how to launch the Add-Ins.

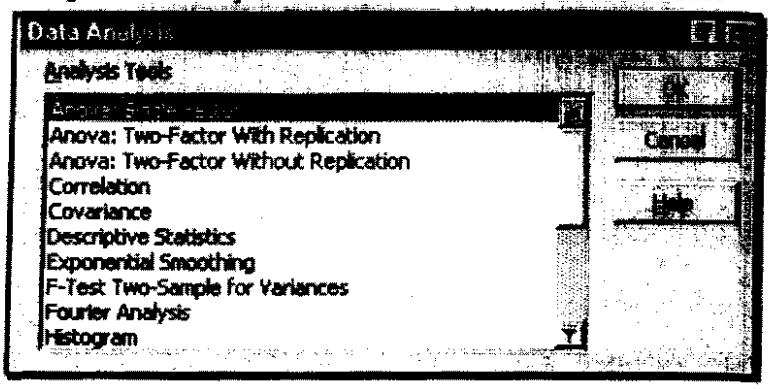
10.1 DESCRIPTIVE STATISTICS

I do not supply the sample data for most of the examples in chapters 36-40. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

Choose the menu option TOOLS/DATA ANALYSIS[12]. The dialog shown in Figure 144 opens.

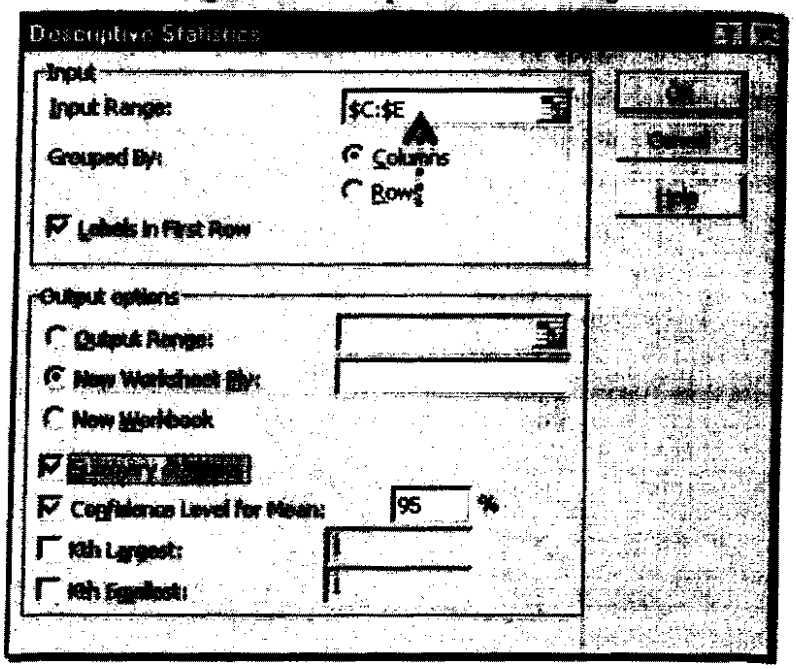
[12] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

Figure 144: The options for the menu TOOLS/DATA ANALYSIS



Choose the statistical procedure "Descriptive Statistics." The dialog for "Descriptive Statistics" opens. Figure 145 shows this dialog (user-input form).

Figure 145: Descriptive Statistics dialog



Input (or, "Source") data

Choose the data series whose descriptives you desire. Click on the edge of the box next to "Input Range" (at the point where the dotted arrow points in Figure 145).

Options

Choose other options shown in Figure 145. Select the option "Labels in first row" because the names of the three series are in the first row of the range you selected (the labels are in cells C1, D1, and E1)— this way Excel picks up the names of the variables and uses these names in the output[13]. Execute the dialog by clicking on the button OK.

Output

Excel produces the descriptive statistics and places the results in a new worksheet. (This is illustrated in Figure 146.)

[13] Note that in the output of this procedure (shown in Figure 546) the first row has the labels for the three variables— 1995, 2000, and 2010.

Figure 146: Output of Descriptive Statistics procedure

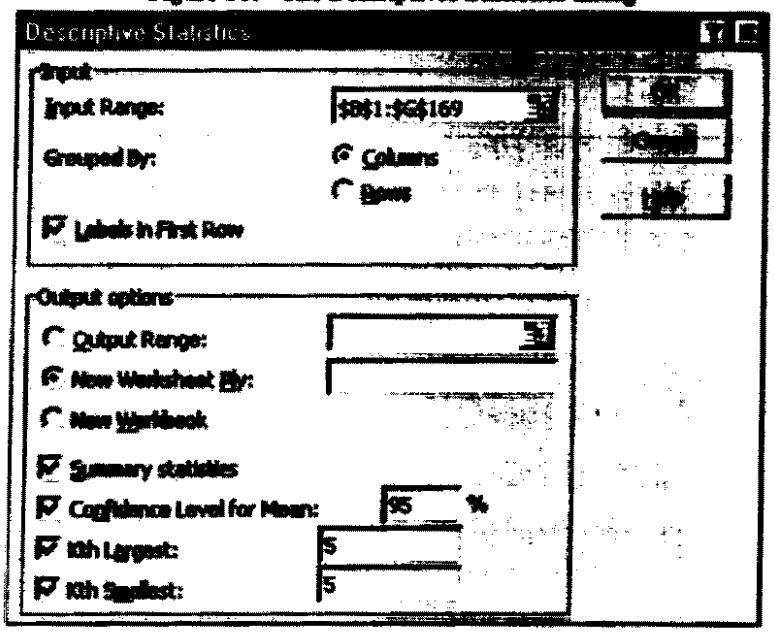
A	B	C	D	E	F
1995		2000		2010	
Mean	5257914.9	Mean	9440351.4	Mean	7406944.7
Standard Err	2685017.3	Standard Err	5047964.7	Standard Err	3746949.2
Median	492000	Median	569000	Median	799000
Mode	47000	Mode	254000	Mode	51000
Standard De	40853942	Standard De	77383634	Standard De	57470302
Sample Varia	1.669E+15	Sample Varia	5.988E+15	Sample Varia	3.303E+15
Kurtosis	218.59277	Kurtosis	179.18282	Kurtosis	219.84937
Skewness	14.560356	Skewness	13.001357	Skewness	14.619033
Range	617798000	Range	1.109E+09	Range	870291000
Minimum	7000	Minimum	12000	Minimum	25000
Maximum	617805000	Maximum	1.109E+09	Maximum	870316000
Sum	1.236E+09	Sum	2.218E+09	Sum	1.741E+09
Count	235	Count	235	Count	235
Confidence L	5250489.2	Confidence L	8945257.7	Confidence L	7385899.6

This tool generates a report of univariate statistics for data in the input range, providing information about the central tendency and variability of your data

<u>Example 2: Adding additional parameters to the descriptives table</u>

Go to the menu option TOOLS/DATA ANALYSIS. Select the option "Descriptive Statistics." In addition to the statistics requested in the previous example, I request Excel to report on the fifth largest and fifth smallest values for each column/series.

Figure 147: The Descriptives Statistics dialog



Output

The output for the procedure is reproduced in the next table. In one simple step, you have created a table that captures the basic statistical attributes of several data series and the fifth highest and lowest values of each data series.

Table 29: Output of the Descriptive Statistics tool including the Kth largest and smallest values. The names of the three variable are: s1, s2, and x1.

s1		s2		x1	
Mean	7.32	Mean	7.23	Mean	1173.00
Standard Error	0.44	Standard Error	0.49	Standard Error	52.67
Median	5.31	Median	4.81	Median	1173.00
Mode	1.34	Mode	23.00	Mode	#N/A
Standard Deviation	5.72	Standard Deviation	6.33	Standard Deviation	682.73
Sample	32.68	Sample	40.13	Sample	466119.22

s1		s2		x1	
Variance		Variance		Variance	
Kurtosis	–0.22	Kurtosis	0.04	Kurtosis	–1.20
Skewness	0.95	Skewness	1.06	Skewness	0.00
Range	19.66	Range	22.00	Range	2344.00
Minimum	1.34	Minimum	1	Minimum	1
Maximum	21	Maximum	23	Maximum	2345
Sum	1229.79	Sum	1215.395	Sum	197064
Count	168	Count	168	Count	168
Largest (5)	21	Largest (5)	23	Largest (5)	2288.86
Smallest (5)	1.34	Smallest (5)	1	Smallest (5)	57.14
Confidence Level (95.0%)	0.87	Confidence Level (95.0%)	0.96	Confidence Level (95.0%)	103.99

Interpretation of the statistical parameters is discussed in chapter 6, and of Confidence levels is discussed in 7.1.

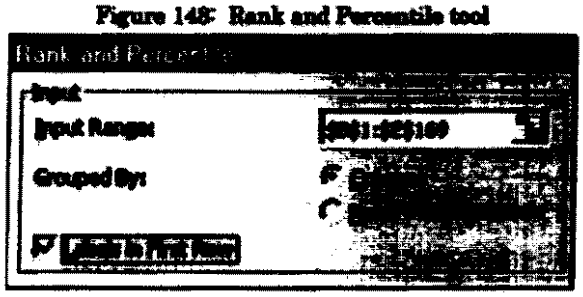
10.2 RANK AND PERCENTILE

This tool produces a table that contains the ordinal and percentage rank of each value in a data set. You can analyze the relative standing of values in a data set. The Percentile values can assist in learning about the spread of the series across its range. For a series provides information on the ranges for the lowest 25%, the next 25%, the next 25%, and the

highest 25%.

Go to[14] the menu option TOOLS/DATA ANALYSIS[15]. Select the option "Rank and Percentile." The dialog is shown in the next figure.

Figure 148: Rank and Percentile tool



The result is reproduced in the next table. Each output table contains four columns:

— The place of the data point in the data series,

— The value of the data (with the label for the series as the label on the output column),

— The rank of the data point within the range, and

[14] I do not supply the sample data for most of the examples in chapter 42 to chapter 46. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

[15] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

— The percentage rank of the data point. The columns are sorted in order of ascending rank.

Table 30: Output of the Rank and Percentile tool

Point	*s1*	*Rank*	*Percent*	*Point*	*s2*	*Rank*	*Percent*
24	21.00	1	96.40%	1	23.00	1	96.40%
48	21.00	1	96.40%	25	23.00	1	96.40%
72	21.00	1	96.40%	49	23.00	1	96.40%
96	21.00	1	96.40%	73	23.00	1	96.40%
120	21.00	1	96.40%	97	23.00	1	96.40%
144	21.00	1	96.40%	121	23.00	1	96.40%
168	21.00	1	96.40%	145	23.00	1	96.40%
23	18.63	8	92.20%	2	20.07	8	92.20%
47	18.63	8	92.20%	26	20.07	8	92.20%
71	18.63	8	92.20%	50	20.07	8	92.20%
95	18.63	8	92.20%	74	20.07	8	92.20%
119	18.63	8	92.20%	98	20.07	8	92.20%
143	18.63	8	92.20%	122	20.07	8	92.20%
167	18.63	8	92.20%	146	20.07	8	92.20%
22	16.53	15	88.00%	3	17.51	15	88.00%
46	16.53	15	88.00%	27	17.51	15	88.00%
70	16.53	15	88.00%	51	17.51	15	88.00%
94	16.53	15	88.00%	75	17.51	15	88.00%
118	16.53	15	88.00%	99	17.51	15	88.00%
142	16.53	15	88.00%	123	17.51	15	88.00%
166	16.53	15	88.00%	147	17.51	15	88.00%

Interpreting the output:

The last row's last four columns can be interpreted as—

The 147th data point in the selected range has a value of 17.51, which gives it rank 15 in the selected range, with 88% of the cells in the range having a value less than or equal to this data point.

10.3 BIVARIATE RELATIONS– CORRELATION, COVARIANCE

Correlation analysis

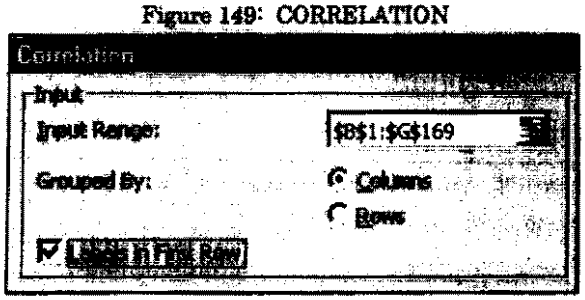
This tool and its formulas measure the relationship between two data sets that are scaled to be independent of the unit of measurement. The correlation coefficient depicts the basic relationship across two variables: "Do two variables have a tendency to increase together or to change in opposite directions and, if so, by how much?" Bivariate correlations measure the correlation coefficients between two variables at a time, ignoring the effect of all other variables.

Go to the menu option TOOLS/DATA ANALYSIS[16]. Select the option "Correlation."

Select the "Input Range" — it must have more than one data series.

[16] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

Figure 149: CORRELATION



The output is reproduced in the next table.

Table 31: Output from Correlation Analysis tool

	s1	*s2*	*x1*	*x2*	*x3*	*x4*
s1	1.00000					
s2	–0.75973	1.00000				
x1	–0.13434	0.13226	1.00000			
x2	0.21423	0.47238	0.01658	1.00000		
x3	0.20122	–0.08459	–0.15748	0.14568	1.00000	
x4	–0.13567	0.12935	0.99998	0.01040	–0.15839	1.00000

Interpreting the output

— A high level of correlation is implied by a correlation coefficient that is greater than 0.5 in absolute terms (that is, greater than 0.5 or less than –0.5).

— A mid level of correlation is implied if the absolute value of the coefficient is greater than 0.2 but less that 0.5.

— A low level of correlation is implied if the absolute value of the coefficient is less than 0.2.

10.3.A COVARIANCE TOOL AND FORMULA

The options are same as for the CORRELATION TOOL. The covariance is dependent on the scale of measurement of the data series. Therefore, there is no standard scale from which to infer if a covariance value is "high" or "low." Thus, use the correlation tool that provides a uniform scale of "–1 to 1."

The coefficient of determination can be roughly interpreted as the proportion of variance in a series that can be explained by the values of the other series. The coefficient is calculated by squaring the correlation coefficient.

CHAPTER 11

HYPOTHESIS TESTING

This chapter teaches:

— Z-TESTING FOR POPULATION MEANS WHEN POPULATION VARIANCES ARE KNOWN

— PAIRED SAMPLE T-TESTS

— T-TESTING MEANS WHEN THE TWO SAMPLES ARE FROM DISTINCT GROUPS

— THE PRETEST— F-TESTING FOR EQUALITY IN VARIANCES

— T-TEST: TWO–SAMPLE ASSUMING UNEQUAL VARIANCES

— T-TEST: TWO–SAMPLE ASSUMING EQUAL VARIANCES

— ANOVA

The statistics Add-In provides some procedures for hypothesis testing. The "Inverse Functions" in Excel Cross-reference (see 7.1) and other statistics software can be used to build Confidence Interval's that provide the values for the "Critical Regions" for conducting hypothesis tests. The use of the functions opens up a much wider range of possible hypothesis tests limited only by the Inverse functions available in Excel.

I include a set of "testing rules" in several of the examples. These rules will blow your mind — it will make hypothesis testing a readily comprehensible step–by–step process. The rules will assist you in all hypothesis tests— in Excel or otherwise.

This chapter requires the Analysis ToolPak Add-Ins; chapter 9 shows how to learn how to launch the Add-Ins.

11.1 Z-TESTING FOR POPULATION MEANS WHEN POPULATION VARIANCES ARE KNOWN

This tool performs a two–sample Z-test for means with known variances. This tool is used to test hypotheses about the difference between two population means.

Possible hypothesis for testing

u1 is the mean of sample one. *u2* is the mean for sample two. The critical regions are based on a 5% significance level (or, equivalently, a 95% Confidence Interval)

(a) Two–tailed

The hypothesis

— H_0 (Null Hypothesis): *u1*— *u2* = 1

— Ha (Alternate hypothesis): *u1*— *u2* <> 1

Critical region:

— "Fail to accept" the null hypothesis if the absolute value of the calculated Z is higher than 1.96. Examples of such Z values are: "+2.12" and "–2.12."

— "Fail to reject" the null hypothesis if the absolute value of the calculated Z is lower than 1.96. Examples of such Z values are: "+1.78," "0.00" and "–1.78."

In short, if the absolute value of the Z is higher than 1.96, then one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

(b) One–tailed (left-tail)

The hypothesis:

— H_0 (Null Hypothesis): $u1 — u2 >= 1$

— Ha (Alternate hypothesis): $u1 — u2 < 1$ (one–tailed)

Critical region:

— "Fail to accept" the null hypothesis if the value of the calculated Z is lower than "–1.64." Examples of such Z values are: "–2.12" and "–1.78."

— "Fail to reject" the null hypothesis if left-tail)

The value of the calculated Z is greater than "–1.64." Examples of such Z values are: "+1.78" and "0.00."

In short, if the Z is lower than "–1.64," then one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

(c) One–tailed (right-tail)

The hypothesis:

— H_0 (Null Hypothesis): $u1 - u2 <= 1$

— Ha (Alternate hypothesis): $u1 - u2 > 1$ (one–tailed)

Critical region:

— "Fail to accept" the null hypothesis if the value of the calculated Z is greater than "+1.64." Examples of such Z values are: "+2.12" and "+1.78."

— "Fail to reject" the null hypothesis if the absolute value of the calculated Z is less than "+1.64." Examples of such Z values are: "–1.78" and "0.00."

In short, if the Z is greater than "+1.64," then one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

Excel calculates the P or Significance value for each test you run.

— If P is less than 0.10, then the test is significant at 90% Confidence (equivalently, the hypothesis that the means are equal can be rejected at the 90% level of Confidence). This criterion is considered too "loose" by some.

— If P is less than 0.05, then the test is significant at 95% Confidence (equivalently, the hypothesis that the means are equal can be rejected at the 95% level of Confidence). This is the standard criterion used.

— If P is less than 0.01, then the test is significant at 99% Confidence (equivalently, the hypothesis that the means are equal can be rejected at the 99% level of Confidence). This is the strictest criterion used.

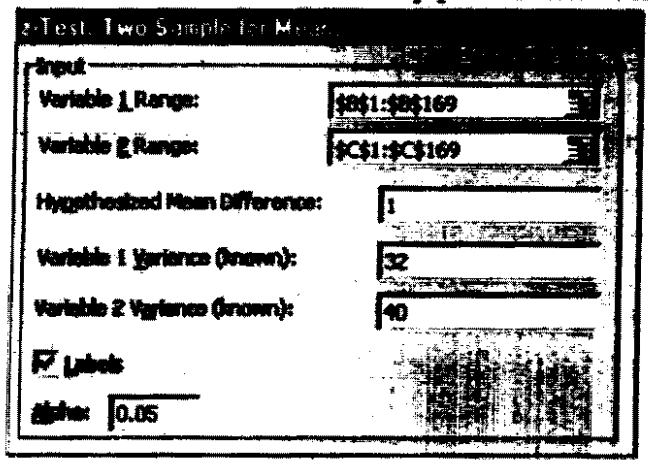
You should memorize these criteria, as nothing is more helpful in interpreting the output from hypothesis tests (including all the tests intrinsic to every regression, ANOVA and other analysis).

Go to TOOLS/DATA ANALYSIS[17]. Select the option "Z-test." The dialog (user-input form) that opens is shown in the next figure.

Enter the hypothesized mean difference (that is, the Null Hypothesis) into the text-box "Hypothesized Mean Difference." Enter the variances for the two populations.

[17] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

Figure 150: Z-test for mean differences when population variance is known



The next table shows the result of a Z-test[18].

Table 32: Output for Z-test for mean differences when population variance is known

Z-test: Two Sample for Means		
	s1[19]	s2
Mean	7.3202	7.2345
Known Variance	32	40
Observations	168	168
Hypothesized Mean Difference	1.0	
	–1.397	
P (Z< = z) one–tail	0.081	
Z Critical one–tail	1.645	

[18] I do not supply the sample data for most of the examples in chapter 42 to chapter 46. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

[19] s1 and s2 are the labels, picked up from the first row in the range b1:b25 and c1:c25.

Z-test: Two Sample for Means		
P (Z< = z) two–tail	0.163	
Z Critical two–tail	1.960	

Interpreting the output

The P value (that is "P (Z<= or >= z) two–tail") of 0.081 implies that we fail to reject the null for the two one–tail hypothesis. Moreover, Z= –1.397 implies that we "fail to reject" the null hypothesis because the Z is in the acceptance region ("1.96," "–1.96") for the two–tail hypothesis.

The P value (that is "P (Z<>z) two–tail") of 0.163 implies that we fail to reject the null for the two–tail hypothesis. In addition, if we use a one–tailed (left tail) test, we again fail to reject the null hypothesis because the Z is in the acceptance region ("> –1.645") for the left–tail hypothesis. If we use a one–tail (right tail) test, we fail to reject the Null because the Z is in the acceptance region ("< +1.645") for the right–tail hypothesis.

11.2 T-TESTING MEANS WHEN THE TWO SAMPLES ARE FROM DISTINCT GROUPS

11.2.A THE PRETEST— F-TESTING FOR EQUALITY IN VARIANCES

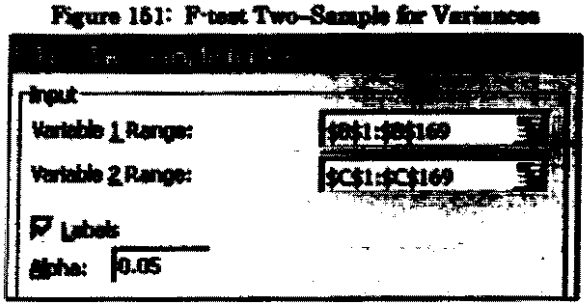
The T-test is used most often to test for differences in means across samples from distinct groups. The respondents in the two samples differ. An example is a pair of samples from two surveys on earnings, one survey in country A and the other in country B. The formula used in estimating the T statistic depends on the equality of variance for the data series

acr[illegible]ss the two samples. In particular, if the variances of the two samples are [illegible] unequal the formula takes into account this difference across the sam[illegible]ples. An F-test is used to test for unequal variances.

Th[illegible] "F-test Two-sample for Variances" performs a test to compare the var[illegible]iances across two groups of data. Launch the procedure by accessing the [illegible] menu option TOOLS/DATA ANALYSIS[20] and selecting the "F-test Tw[illegible]-sample for Variances."

Th[illegible] relevant dialog is reproduced in the next figure.

Figure 151: F-test Two-Sample for Variances



Ch[illegible]oose the "alpha" for level of significance. A 0.05 level sets up a 95% con[illegible]fidence test.

<u>Th[illegible] hypothesis:</u>

— H_0 (Null Hypothesis): $\sigma_1^2 - \sigma_2^2 = 0$

[20] If y[illegible]u do not see this option, then use TOOLS / ADD-INS to activate the Add-In for dat[illegible]a analysis. Refer to section 41.4.

— H_a (Alternate hypothesis): $\sigma_1^2 - \sigma_2^2 <> 0$, Where σ_1^2 is the variance of sample one, and σ_2^2 is the variance for sample two.

The F has a one–tail test only.

The next table shows the output of a typical F-test[21].

Table 33: Output for F-test tool for equality of variances

	s1	s2
Mean	7.3202	7.2345
Variance	32.6754	40.1309
Observations	168	168
Df	167	167
	0.8142	
P (F< = f) one–tail	0.0926	
F Critical one–tail	0.8747	

Interpreting the output

— The row "Variance" shows the estimated variance parameters.

— Inferences from the P value of "0.0926":

[21] I do not supply the sample data for most of the examples in chapter 42 to chapter 46. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

— If P is less than 0.10, then the test is significant at 90% Confidence (equivalently, the hypothesis that the variances are equal can be rejected at the 90% level of Confidence). The P of 0.0926 implies the test is significant at the 90% Confidence level. Being "significant" implies that the estimated F statistic lies in the critical region and the "null hypothesis cannot be accepted." You are in the area represented by the alternate hypothesis — the variances are unequal.

— If P is less than 0.05, then the test is significant at 95% Confidence (equivalently, the hypothesis that the variances are equal can be rejected at the 95% level of Confidence). The hypothesis cannot be rejected at the 0.05 level of significance.

— If P is less than 0.01, then the test is significant at 99% Confidence (equivalently, the hypothesis that the variances are equal can be rejected at the 99% level of Confidence). The hypothesis of equal variances cannot be rejected at the 0.01 level of significance.

The test is significant only at the 0.10 level of significance. The critical estimated F of 0.81 is higher than the critical F of 0.8747 implying that the "null hypothesis of equal variances" cannot be accepted at a 0.05 level of Confidence.

Once you know if the null hypothesis of equal variances can be accepted, you can resolve whether to use the "Two–Sample T-test Assuming Equal Variances" or "Two–Sample T-test Assuming Unequal Variances."

11.2.B T-TEST: TWO-SAMPLE ASSUMING UNEQUAL VARIANCES

This T-test form assumes that the variances of both ranges of data are unequal. Use this test when the groups under study are distinct. Use a paired test (discussed in the next section) when there is one group before and after a treatment.

Possible hypothesis for testing

u1 is the mean of sample one. *u2* is the mean for sample two. The critical regions are based on a 5% significance level (or, equivalently, a 95% Confidence Interval)

(a) Two-tailed

The hypothesis

— H_0 (Null Hypothesis): $u1 - u2 = 0$ (or any non-zero value)

— Ha (Alternate hypothesis): $u1 - u2 <> 0$

Critical region:

— "Fail to accept" the null hypothesis if the absolute value of the calculated T is higher than 1.96. Examples of such Z values are: "+2.12" and "–2.12."

— "Fail to reject" the null hypothesis if the absolute value of the calculated T is lower than 1.96. Examples of such T values are: "+1.78," "0.00" and "–1.78."

In short, if the absolute value of the T is higher than 1.96, then one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

(b) One–tailed (left-tail)

The hypothesis:

— H_0 (Null Hypothesis): $u1 - u2 >= 0$

— Ha (Alternate hypothesis): $u1 - u2 < 0$ (one–tailed)

Critical region:

— "Fail to accept" the null hypothesis if the value of the calculated T is lower than "–1.64." Examples of such T values are: "–2.12" and "–1.78."

— "Fail to reject" the null hypothesis if the absolute value of the calculated T is greater than "–1.64." Examples of such T values are: "+1.78" and "0.00."

In short, if the T is lower than "–1.64," one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

(c) One–tailed (right-tail)

The hypothesis:

— H_0 (Null Hypothesis): $u1 - u2 <= 0$

— Ha (Alternate hypothesis): $u1 - u2 > 0$ (one–tailed)

Critical region:

— "Fail to accept" the null hypothesis if the value of the calculated T is greater than "+1.64." Examples of such T values are: "+2.12" and "+1.78."

— "Fail to reject" the null hypothesis if the absolute value of the calculated T is less than "+1.64." Examples of such T values are: "–1.78" and "0.00."

In short, if the T is greater than "+1.64," then one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

Go to the menu option TOOLS/DATA ANALYSIS[22]. Select the option "T-test: Two–Sample Assuming Unequal Variances." The next table shows a sample output[23] for a T-test assuming unequal variances.

Table 34: Output of Two Sample T-test (assuming unequal variances)

	s1	*s2*

[22] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

[23] I do not supply the sample data for most of the examples in chapter 42 to chapter 46. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

	s1	*s2*
Mean	7.32	7.23
Variance	32.68	40.13
Observations	168	168
Hypothesized Mean Difference	5	
Df	331	
T Stat	–7.465	
P (T< = t) one–tail	3.72E–13	
T Critical one–tail	1.649	
P (T< = t) two–tail	7.43E–13	
T Critical two tail	.967	

Interpreting the output

The row "Mean" shows the estimated means for the two samples *s1* and *s2*. The next column "Variance" displays the calculated variance for these sample mean values. "Df" shows the "Degree of Freedom." The degrees of freedom equal the total sample points (the sum of the sample sizes of the two samples) minus the one degree of freedom to account for the one equation (the "hypothesized mean difference" which here is "u1 — u2 = 5") . So, degrees of freedom equals "168 + 168 -1 = 331".

(a) Two–tailed

The hypothesis was:

— H_0 (Null Hypothesis): $u1 - u2 = 5$

— Ha (Alternate hypothesis): $u1 - u2 <> 5$, where *u1* is the mean of sample *s1* and *u2* the mean of sample *s2*.

The calculated T statistic is "–7.465." The P value for the two–tailed test is "3.72 multiplied by the 13th point after the decimal" or "0.000000000000372." As the P value is less than 0.01, the hypothesis is "significant[24]"at the 99% Confidence level or "alpha = 0.01" level of significance. (The natural extension of this inference is that the hypothesis is significant at the 95% and 90% Confidence levels also.)

The region for the two–tailed test is "> 1.967 or < –1.967." In this example, the test is significant (at a 0.05 level of significance because the estimated T lies in the critical region. (The estimated T of "–7.465" lies in the region "< –1.967".)

(b) One–tailed (left-tail)

The hypothesis was:

— H_0 (Null Hypothesis): $u1 - u2 >= 5$

— Ha (Alternate hypothesis): $u1 - u2 < 5$, where *u1* is the mean of sample *s1* and *u2* the mean of sample *s2*.

The P value for the one–tailed test is "7.45 multiplied by the 13th point after the decimal" or "0.000000000000745." The relevant test here is the left–tail because the T statistic is a negative value. As the P value is less

[24] If a test is "significant" the implication is a "failure to accept" the null hypothesis. The test T statistic lies in the critical region. In informal terms, the alternate hypothesis is "correct."

than 0.01, the hypothesis is "significant" at the 99% Confidence level or "alpha = 0.01" level of significance. (The natural extension of this inference is that the hypothesis is significant at the 95% and 90% Confidence levels also.)

Another way to test the hypothesis is to compare the estimated T statistic to the critical region shown in the column "T Critical one–tail." The region for the left–tailed test is "< –1.649". In this example, the test is "significant[25]" at a .05 level of significance because the estimated T lies in the critical region. (The estimated T of "–7.465" lies in the region "< –1.649".)

(c) One–tailed (right-tail)

The hypothesis was:

— H_0 (Null Hypothesis): $u1 - u2 <= 5$

— Ha (Alternate hypothesis): $u1 - u2 > 5$, where $u1$ is the mean of sample $s1$ and $u2$ the mean of sample $s2$.

The region for the right–tailed test is "> 1.649". In this example, the test is not significant because the estimated T does not lie in the critical region. (The estimated T of "–7.465" is not in the region ">1.649".)

[25] If a test is "significant" the implication is a "failure to accept" the null hypothesis. The test T statistic lies in the critical region. In informal terms, the alternate hypothesis is "correct."

11.2.C T-TEST: TWO-SAMPLE ASSUMING EQUAL VARIANCES

This tool performs a two–sample student's T-test— under the assumption that the variances of both data sets are equal. The hypothesis and interpretation of results is the same as for the Two–Sample Assuming Unequal Variances. (See previous sub-section).

The next table shows the result this type of test[26].

11.3 PAIRED SAMPLE T-TESTS

This tool performs a paired two–sample T-test to deduce whether the difference between the sample means is statistically distinct from a hypothesized difference. This T-test form does not assume that the variances of both populations are equal. You can use a paired test when there is a natural pairing of observations in the samples, such as when a sample group is tested twice— before and after an experiment. The tested groups form a "Paired Sample" with the same respondents sampled "before" and "after" an event.

Go to the menu option TOOLS/DATA ANALYSIS[27]. Select the option "T-

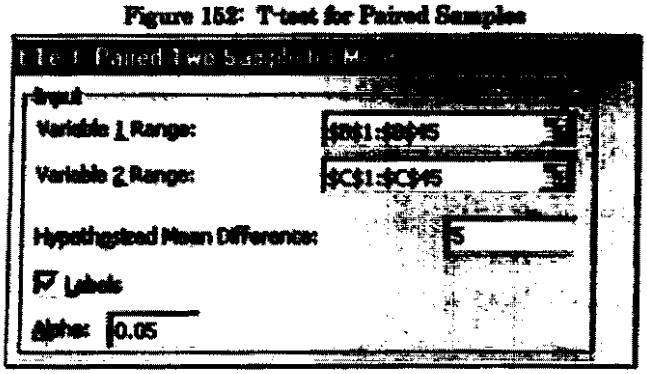
[26] I do not supply the sample data for most of the examples in chapter 42 to chapter 46. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

[27] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

test: Two–Sample Assuming Unequal Variances." The relevant dialog is shown in the next figure.

The range must consist of a single column or row and contain the same number of data points as the first range.

Figure 152: T-test for Paired Samples



Place the hypothesized difference in means into the checkbox "Hypothesized Mean Difference." In this example, one is using the hypothesis:

> "H_0 (Null Hypothesis): mean difference > 5". See the next figure for an example of setting the hypothesis for testing. Set a hypothesized mean difference of zero to test the standard hypothesis that the "Means for the two groups/samples are statistically different."
>
> The level of significance for the hypothesis tests should be placed in the checkbox "Alpha." If you desire a significance level of "alpha = .05" (that is, a Confidence level of 95%), then write in ".05" into the checkbox *Alpha*. The next figure illustrates this.

Possible hypothesis for testing

u1 is the mean of sample one. *u2* is the mean for sample two. The critical regions are based on a 5% significance level (or, equivalently, a 95% Confidence Interval)

(a) Two–tailed

The hypothesis

— H_0 (Null Hypothesis): $u1 - u2 = 0$

— Ha (Alternate hypothesis): $u1 - u2 <> 0$

Critical region:

— "Fail to accept" the null hypothesis if the absolute value of the calculated T is higher than 1.96. Examples of such T values are: "+2.12" and "–2.12."

— "Fail to reject" the null hypothesis if the absolute value of the calculated T is lower than 1.96. Examples of such T values are: "+1.78," "0.00" and "–1.78."

In short, if the absolute value of the T is higher than 1.96, then one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

(b) One–tailed (left-tail)

The hypothesis:

— H_0 (Null Hypothesis): $u1 - u2 >= 0$

— Ha (Alternate hypothesis): $u1 - u2 < 0$ (one–tailed)

Critical region:

— "Fail to accept" the null hypothesis if the value of the calculated T is lower than "–1.64." Examples of such T values are: "–2.12" and "–1.78."

— "Fail to reject" the null hypothesis if the absolute value of the calculated T is greater than "–1.64". Examples of such T values are: "+1.78" and "0.00."

In short, if the T is lower than "–1.64," then one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

(c) One–tailed (right-tail)

The hypothesis:

— H_0 (Null Hypothesis): $u1 - u2 <= 1$

— Ha (Alternate hypothesis): $u1 - u2 > 1$ (one–tailed)

Critical region:

— "Fail to accept" the null hypothesis if the value of the calculated T is greater than "+1.64." Examples of such T values are: "+2.12" and "+1.78."

— "Fail to reject" the null hypothesis if the absolute value of the calculated T is less than "+1.64." Examples of such T values are: "–1.78" and "0.00."

In short, if the T is greater than "+1.64," then one may conclude (with 95% Confidence) that the means of the samples differ by the hypothesized difference.

Excel calculates the P or Significance value for each test you run.

— If P is less than 0.10, then the test is significant at 90% Confidence (equivalently, the hypothesis that the means are equal can be rejected at the 90% level of Confidence). This criterion is considered too "loose" by some.

— If P is less than 0.05, then the test is significant at 95% Confidence (equivalently, the hypothesis that the means are equal can be rejected at the 95% level of Confidence). This is the standard criterion used.

— If P is less than 0.01, then the test is significant at 99% Confidence (equivalently, the hypothesis that the means are equal can be rejected at the 99% level of Confidence). This is the strictest criterion used.

You should memorize these criteria, as nothing is more helpful in interpreting the output from hypothesis tests (including all the tests intrinsic to every regression, ANOVA and other analysis). The output for

such a test is shown in the next table[38].

Table 35: Output from a T-test for Paired Samples. The text in italics has been inserted by the author.

	First sampling	*Second sampling*	
Mean	152	145	
Variance	126	114	
Observations	44	44	
Pearson Correlation	0.999693		
Hypothesized Mean Difference	5		
Df	43		
T Stat	26.76		*26.76 is the T estimated from the data*
One–tailed test			
P (T< = t) one–tail	0.00		*1.68 is the "T cut–off Critical Value" from T-Tables*
T Critical one–tail	1.68		
Two–tailed Test			
P (T< = t) two–tail	0.00		*2.02 is the "T cut–off Critical Value" from T-Tables*
T Critical two–tail	2.02		

Interpretation:

One–tailed test		
P (T< = t) one–tail	0.00	*Thus, significant at 99%*

[38] I do not supply the sample data for most of the examples in chapter 42 to chapter 46. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

One–tailed test		
T Critical one–tail (positive for positive tail test, negative for negative tail)	1.68 —1.68	*2.02 is the "T cut–off Critical Value" from T-Tables for alpha = 0.05 and Df = 43*
Inferential Analysis: — Fail to reject null (1-tailed for null hypothesizing in a negative direction: H_0 (Null Hypothesis): mean<5) — Fail to accept null if H0 (Null Hypothesis): mean>5.		
Two–tailed Test		
P (T< = t) two–tail	0.00	*Thus, significant at 99%*
T Critical two–tail (compare absolute value of T- stat from the data with this absolute value)	2.02	*This is the "T cut–off Critical Value" from T-Tables for alpha = 0.025 and Df = 43*
Inferential Analysis: — For two–tailed test, fail to accept null at 99% Confidence		

11.4 ANOVA

This tool performs simple analysis of variance (ANOVA) to test the hypothesis that means from two or more samples are equal (drawn from populations with the same mean). This technique expands on the tests for two means, such as the T-test.

Go to the menu option TOOLS/DATA ANALYSIS[29]. Select the option "ANOVA: Single Factor." The input range must consist of two or more

[29] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

adjacent ranges of data arranged in columns or rows. A sample output[30] is shown in the next few tables.

Figure 153: Single Factor ANOVA

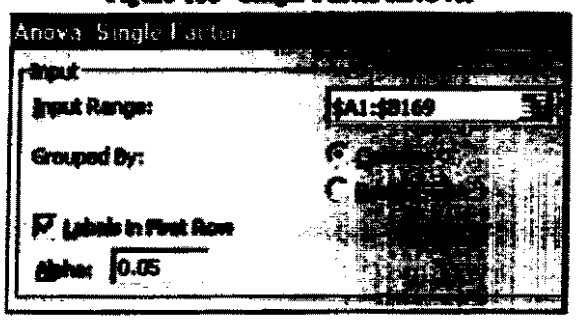


Table 36: Output from Single Factor ANOVA — a

ANOVA: Single Factor				
Groups	Count	Sum	Average	Variance
s1	168	1229.8	7.3	32.7
s2	168	1215.4	7.2	40.1

The first table shows some descriptive statistics for the samples.

Table 37: Output from Single Factor ANOVA — b

ANOVA					
Source of Variation	SS	Df	MS	F	P-value

[30] I do not supply the sample data for most of the examples in chapter 42 to chapter 46. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

ANOVA					
Between Groups	0.62	1	0.62	0.017	0.90
Within Groups	12158.65	334	36.403		
Total	12159.27	335			

Interpreting the output

The information on "Between Groups" is derived from the difference in means and variances across the groups. In an ANOVA, the number of groups may exceed two.

— The test is analyzing the variance as measured by the SS "Sum of Squares" of the "dependent" series. The total Sum of Squares is 12159.27. Of this, 0.62 can be explained by the differences across the means of the two groups. The other 12158.65 is explained by the differences across individual values of the "dependent" series.

- Sum of Squares = Sum of Squares for Between Groups + Sum of Squares for Within Groups

— The MS is the "Mean Sum of Squares" and is estimated by dividing the SS by the degrees of freedom. Therefore, the MS for "Between Groups" equals (0.62/1) = 0.62. (Note that "ANOVA = Analysis of Variance.") The MS for "Within Groups" equals (12158.65/334) = 36.403. The MS may be informally interpreted as "Sum of Squares Explained per Degree of Freedom."

- Mean Sum of Squares = (Sum of Squares)/ (Degrees of Freedom)

— The ANOVA uses an F-test to determine if "Between Groups" information (the number 0.62 in the column "Between Groups" Source of Variation MS) provides sufficient additional information to improve the ability of the data to explain the variance in the "dependent" series. The ANOVA is asking "Does the *Between Groups* Sum of Squares Explained per Degree of Freedom" divided by the "*Within Groups* Sum of Squares" provide an F that is large enough to justify the statement "The use of Between Groups information explains a statistically significant amount of the Sum of Squares of the dependent series."

- F = (Mean Sum of Squares *Between Groups*)/ (Mean Sum of Squares *Within Groups*)

— All ANOVA tests (including the ANOVA output from a regression) can be interpreted in the same way –

- F = [(Increase in ability of model to explain the Sum of Squares)/ (Degrees of Freedom) /
  (Total Sum of Squares) / (Degrees of Freedom)]

CHAPTER 12

REGRESSION

This chapter discusses the following topics:

— ASSUMPTIONS UNDERLYING REGRESSION MODELS

— CONDUCTING THE REGRESSION

This chapter requires the Analysis ToolPak Add-Ins; chapter 9 shows how to learn how to launch the Add-Ins.

12.1 ASSUMPTIONS UNDERLYING REGRESSION MODELS

The field of econometrics uses regression analysis to create quantitative models that can be used to predict the value of a series if one knows the value of several other variables. For example, the wage per hour can be predicted if one knows the values of the variables that constitute the regression equation. This is a big leap of faith from a correlation or Confidence interval estimate. In a correlation, the statistician is not presuming or implying any causality or deduction of causality. On the other hand, regression analysis is used so often (probably even abused) because of its supposed ability to link cause and effect. Skepticism of causal relationships is not only healthy but also important because real power of regression lies in a comprehensive interpretation of the results.

Regression models are used to test the statistical validity of causal relation presumed in theory or hypothesis. Regression can never be divorced from the hypothesis it is testing. The construction of the model has to be based upon the hypothesis, and not on the availability of the data. Therefore, if you believe you have a valid hypothesis, but do not have the correct data series to represent each factor in your hypothesis, the best practice is not running a regression analysis.

On the other hand, the method of throwing in all variables into the model and making the computer select the best model is a misleading technique that sadly has gained popularity because of the belief that the best model is the one that fits the data the best.

The best models can only be a subset of "valid models." (That is, models that have passed all diagnostic test for presumptions for conforming to the assumptions required by a regression.) Furthermore, note that if the model is shown to "not fit" the data, or the expected relationship between variables is estimated as negligible, you still have valid results. The variance between the hypothesis and the results is always important and can give rise to a new perspective relative to the hypothesis.

The process of interpretation is called inferential analysis and is far more important than the actual number punching. Inferential analysis also includes testing if the data and model have complied with the strong assumptions underlying a regression model.

The very veracity and validity depends upon several diagnostic tests. Unfortunately, many econometricians do not perform the diagnostic testing or simply lie about the inferences and conclusions derived from the model.

Our book "Interpreting Regression Output" provides a summary table (a

cheat–sheet for you!) that lists the implications of the invalidity of assumptions. (The book can be purchased at http://www.vjbooks.net). This summary provides, in one page, what other books have spread out over many chapters. Please use this table as a checklist before you interpret any model. Most statistic professors and textbooks teach the interpretation of regression results before discussing the issue of validity. You will save yourself a lot of grief if you always perform diagnostics after running a regression model.

Once you have a valid model, interpret the results in the logical sequence shown in the table interpreting regression output in our book "Interpreting Regression Output." This table will provide a framework and flowchart for interpretation thereby enabling a structured and comprehensive inferential analysis.

12.1.A ASSUMPTION 1: THE RELATIONSHIP BETWEEN ANY ONE INDEPENDENT SERIES AND THE DEPENDENT SERIES CAN BE CAPTURED BY A STRAIGHT LINE IN A 2–AXIS GRAPH

This is also called the assumption of linearity in the regression coefficients. (None of the regression coefficients — the betas — should have an exponential power or any other non— linear transformation.)

12.1.B ASSUMPTION 2: THE INDEPENDENT VARIABLES DO NOT CHANGE IF THE SAMPLING IS REPLICATED

The independent variables are truly independent— the model assumes is using deviations across the X variables to explain the dependent series. The regression attempts to explain the dependent series' variations across

the combination of values of the independent variables.

If repeated samples are used, the model predicts the same predicted dependent series for each combination of X values, but— across the samples— the observed Y may differ across the same combination of X values. (The gap between the predicted and observed Y values is the residual or error.)

12.1.C ASSUMPTION 3: THE SAMPLE SIZE MUST BE GREATER THAN THE NUMBER OF INDEPENDENT VARIABLES (N SHOULD BE GREATER THAN K-1)

This assumption ensures that a basic mathematical postulate is adhered to by the regression algorithm. A system of simultaneous equations is only "determined[21]"if the number of equations[22] is greater than the number of unknowns. That is, only if the number of regression coefficients— K minus 1, the subtraction accounting for the coefficient for the intercept).

What information is "known" prior to running the regression?

— All values of the independent variables are known first. In theory, the independent variables are the "experiment."

[21] That is, it can be solved to estimate the optimization parameters — the regression coefficients in the case of a regression

[22] The sample size N in the case of a regression

— Once the "experiment" is conducted, the values of the dependent series Y are known. (Not that this "experiment" analogy holds even if the data for the independent and dependent variables are obtained from the same data collection survey.)

— The regression minimizes the sum of the squared residuals, which is the same as minimizing the square of the difference between the observed and the predicted dependent series. The number of residuals equals the number of observations. Thus, the number of equations equals the number of observations.

What information is "unknown" prior to running the regression?

The regression coefficients — the betas — are unknown. Once the regression coefficients are known, one can estimate the predicted dependent variables, errors/residuals, R–square, etc. If X does not vary, then the series cannot have any role in explaining the variation in Y. The number of unknowns equals the number of regression coefficients.

12.1.D ASSUMPTION 4: NOT ALL THE VALUES OF ANY ONE INDEPENDENT SERIES CAN BE THE SAME

A model uses the effect of variation in X to explain variation in Y. If X does not vary, then the series cannot have any role in explaining the variation in Y.

Note that the formulas for estimating the regression coefficients — the betas — use the "squared deviations from mean" in the denominator of the formula. If the X values do not vary then all the values equal the mean implying that the "squared deviations from mean" is zero. This will

make the regression coefficient indeterminate because the denominator of the formula equals zero.

12.1.E ASSUMPTION 5: THE RESIDUAL OR DISTURBANCE ERROR TERMS FOLLOW SEVERAL RULES

This is the most important assumption, and most diagnostic tests are checking for the observance of this assumption. In several textbooks, you will find this assumption broken into parts, but I prefer to list the rules of Assumption 5:

Assumption 5a: The mean/average or expected value of the disturbance equals zero

If not, then you know that the model has a systemic bias, which makes it inaccurate, especially because one does not typically know what is causing the bias.

Assumption 5b: The disturbance terms all have the same variance

This assumption is also called homoskedasticity. Given that the expected value of any disturbance equals zero, if one disturbance has a higher variance than the other one, it implies that the observation underlying this high variance should be given less importance because its relative accuracy is suspect. (This is the reason that weighted regression is used to correct for the nonconformity with this rule.)

Assumption 5c: A disturbance term for one observation should have no relation with the disturbance terms for other observations or with any of the independent variables

The disturbance term must be truly random — one should not be able to predict or guess the value of any disturbance term given any of the information on the model data. The disturbance term is also called the error term. This error is assumed random. If this is not the case, then your model may have failed to capture all the underlying independent variables, incorrectly measured independent variables, or have correlation between successive observations in a series Sorted by one of the independent variables.

Typically, Time Series data series suffers from the problem of disturbance terms being related to the values of previous periods. It is for this reason that times series analysis requires special data manipulation procedures prior to creating any prediction model.

Assumption 5d: There is no specification bias

This is the most crucial assumption because a mistake in specifying the equation for regression is the responsibility of the statistician. One cannot blame the nature of the data for this problem. One type of specification bias is the use of an incorrect functional form. For example, you have a specification bias if you use a linear function when a logarithmic or exponential function should be used.

The other type of specification bias is when the model does not include a relevant data series. This is the most common type of error of oversight by because of the incorrect habit in creating a hypothesis only after looking at the available data. This approach may result in the exclusion of an important series that may not be in the available data set.

Remember that a regression is based on a hypothesis — you always define the hypothesis first. After that, look for data that can capture all of the variables in the hypothesis. If you do not find the data to represent an important factor, then you should not use regression analysis. Another bad habit is the dropping of variables from a model if the coefficient is seen to have no impact on the dependent series. It is better to have an irrelevant or excess series, then to drop a relevant series. In fact, the result that a factor has no impact on the dependent series often provides compelling insight.

Assumption 5e: The disturbance terms have a Normal Density Function

The use of the F-test for validating the model and the T-tests for validating individual coefficients is predicated on the presumption that the disturbance terms follow a Normal Density Function.

12.1.F ASSUMPTION 6: THERE ARE NO STRONG LINEAR RELATIONSHIPS AMONG THE INDEPENDENT VARIABLES

If the relationships are strong, then the regression estimation will not be able to isolate the impact of each independent series. Related to this is another rule: there should be no endogenity in the model. This means that none of the independent variables should be dependent on other variables. An independent series should not be a function of another independent series.

Every estimate in a regression is not only a point estimate of the parameter of the expected value of the parameter. The regression estimates the expected value (mean) of the parameter, its variance, and its Density Function (the assumption of normality provides the shape of

the Density Function). The mean and standard error are estimated by the model. There is a pair of such estimates for each coefficient (each BETA), each disturbance term, and each predicted value of the dependent series.

Note: The dependent series is that whose values you are trying to predict (or whose dependence on the independent variables is being studied). It is also referred to as the "Explained" or "Endogenous" series, or as the "Regressand."

The independent variables are used to explain the values of the dependent series. The values of the independent variables are not being explained/determined by the model — thus, they are "independent" of the model. The independent variables are also called "Explanatory" or "Exogenous" variables. They are also referred to as "Regressors."

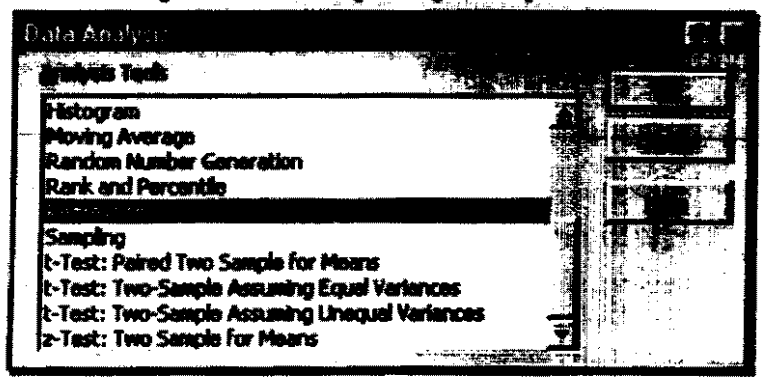
I do not show the details of regression analysis. Please refer to our book "Interpreting regression Output" available at http://www.vjbooks.net.

12.2 CONDUCTING THE REGRESSION

Go to the menu option TOOLS/DATA ANALYSIS[33]. Select the option "Regression" as shown in Figure 154.

[33] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

Figure 154: Selecting the regression procedure



Choose the exact cell references for the Y and X ranges. So do not choose "C:D;" instead, chooee C1:D235, as shown in Figure 155.

<u>Other restrictions:</u>

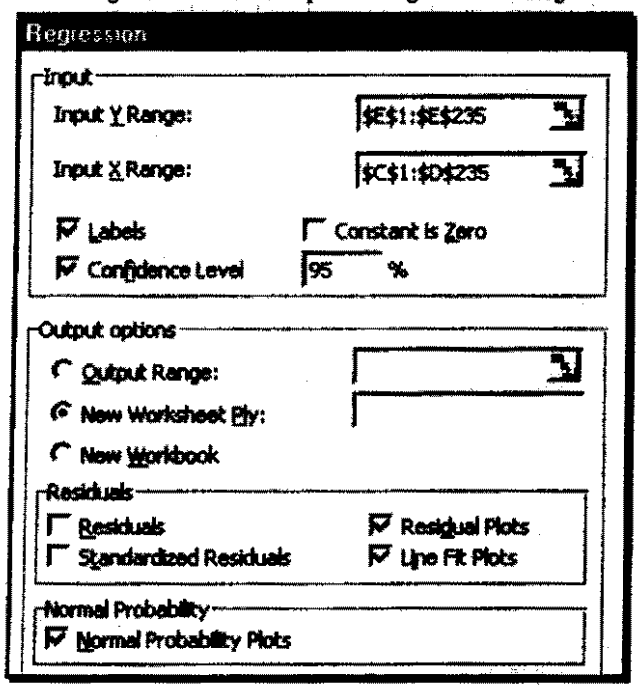
– All the X variables have to be in adjacent columns

and

– The data cannot have missing values

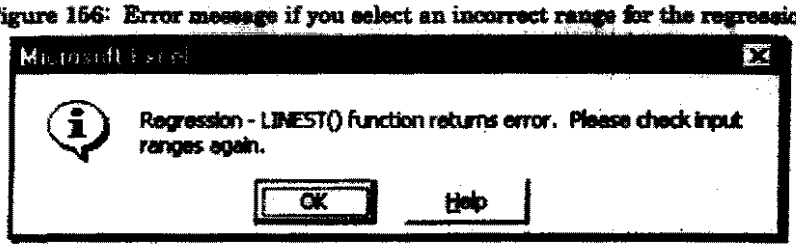
Choose all other options as shown in Figure 155.

Figure 155: The completed Regression dialog



There should be no missing values in the range defined. Otherwise, you get the error message shown in Figure 156.

Figure 156: Error message if you select an incorrect range for the regression



Warning! The statistical Add-In provided with Excel has many limitations— it does only a few procedures, has bugs, and cannot handle complex data. (For example, it cannot do a regression if there are any missing values.) Fortunately, some other companies have created Add-Ins that provides comprehensive statistics capabilities. Links to such Add-Ins can be accessed at the URL http://www.vjbooks.net/products/publications/Excel/Excel.htm..

I do not show the output or its detailed interpretation. Please refer to our book "Interpreting regression Output" available at http://www.vjbooks.net.

A brief summary of interpretation guidelines is presented in the next section.

12.3 BRIEF GUIDELINE FOR INTERPRETING REGRESSION OUTPUT

Table 38: Interpreting regression output

Name Of Statistic/ Chart	*What Does It Measure Or Indicate?*	*Critical Values*	*Comment*
Sig.-F	Whether the model as a whole is significant. It tests whether R-square is significantly different from zero	– below .01 for 99% confidence in the ability of the model to explain the dependent variable – below .05 for 95% confidence in the ability of the model to explain the dependent variable – below 0.1 for 90% confidence in the ability of the model to explain the dependent variable	The first statistic to look for in the output. If Sig.-F is insignificant, then the regression as a whole has failed. No more interpretation is necessary (although some disagree on this point). You must conclude that the "Dependent variable cannot be explained by the independent/explanatory variables." The next steps could be rebuilding the model, using more data points, etc.
RSS, ESS & TSS	The main function of these values lies in calculating test statistics like the	The ESS should be high compared to the TSS (the ratio equals the R-square). Note for	If the R-squares of two models are very similar or rounded off to zero or one, then you might prefer to use the F-test

Name Of Statistic/ Chart	*What Does It Measure Or Indicate?*	*Critical Values*	*Comment*
	F-test, etc.	interpreting the table, column "Sum of Squares": "Total" =TSS, "Regression" = ESS, and "Residual" = RSS	formula that uses RSS and ESS.
SE of Regression	The standard error of the estimate predicted dependent variable	There is no critical value. Just compare the std. error to the mean of the predicted dependent variable. The former should be small (<10%) compared to the latter.	You may wish to comment on the SE, especially if it is too large or small relative to the mean of the predicted/estimated values of the dependent variable.
-Square	Proportion of variation in the dependent variable that can be explained by the independent variables	Between 0 and 1. A higher value is better.	This often mis-used value should serve only as a summary measure of Goodness of Fit. Do not use it blindly as a criterion for model selection.
Adjusted R-square	Proportion of variance in the dependent variable that can be explained by the independent variables or R-square adjusted for # of independent variables	Below 1. A higher value is better	Another summary measure of Goodness of Fit. Superior to R-square because it is sensitive to the addition of irrelevant variables.
-Ratios	The reliability of our estimate of the individual beta	Look at the p-value (in the column "Sig.") it must be low: - below .01 for 99% confidence in	For a one-tailed test (at 95% confidence level), the critical value is (approximately) 1.65 for testing if the coefficient is greater than zero and (approximately) -1.65 for

Name Of Statistic/ Chart	*What Does It Measure Or Indicate?*	*Critical Values*	*Comment*
		the value of the estimated coefficient - below .05 for 95% confidence in the value of the estimated coefficient - below .1 for 90% confidence in the value of the estimated coefficient	testing if it is below zero.
Confidence Interval for beta	The 95% confidence band for each beta estimate	The upper and lower values give the 95% confidence limits for the coefficient	Any value within the confidence interval cannot be rejected (as the true value) at 95% degree of confidence
Charts: Scatter of predicted dependent variable and residual (Preferably after standardizing the series) Make a scatter chart manually after running the regression in Excel. **	Mis-specification and/or heteroskedasticity	There should be no discernible pattern. If there is a discernible pattern, then do the RESET and/or DW test for mis-specification or the White's test for heteroskedasticity	Extremely useful for checking for breakdowns of the classical assumptions, i.e. - for problems like mis-specification and/or heteroskedasticity. At the top of this table, we mentioned that the F-statistic is the first output to interpret. Some may argue that the PRED-RESID plot is more important.
Charts: plots of residuals against independent variables. (Preferably after standardizing	Heteroskedasticity	There should be no discernible pattern. If there is a discernible pattern, then perform a formal test.	Common in cross-sectional data. If a partial plot has a pattern, then that variable is a likely candidate for the cause of

Name Of Statistic/ Chart	*What Does It Measure Or Indicate?*	*Critical Values*	*Comment*
the series) Make a scatter chart manually after running regression**			heteroskedasticity.
Charts: Histograms of residuals. No need to standardize. Make an area chart after running the regression in Excel**	Provides an idea about the distribution of the residuals	The distribution should look like a normal distribution	A good way to observe the actual behavior of our residuals and to observe any severe problem in the residuals (which would indicate a breakdown of the classical assumptions)

** (a) Estimate the series "predicted" by using the regression formula:

Predicted_Y= constant + B_1X_1 +... + B_kX_k.

(b) Standardize the series of predicted values using the function INSERT/ FUNCTION/ STATISTICAL/ STANDARDIZE.

(c) Estimate the residual, by using the formula:

Residual= Y — Predicted_Y

(d) Standardize the series of residuals using the function INSERT/ FUNCTION/ STATISTICAL/ STANDARDIZE

(e) make the charts using the standardized series. See book two in this series — *Charting in Excel* — for more on making charts.

12.4 BREAKDOWN OF CLASSICAL ASSUMPTIONS: VALIDATION AND CORRECTION

Basic validation can be conducted using procedures mentioned in the previous table. Excel does not have procedures for more advanced testing. The corrective procedures are not available in Excel.

The validation and corrective procedures are available in Add-Ins for statistics. Links to such Add-Ins can be accessed at the URL http://www.vjbooks.net/products/publications/Excel/Excel.htm.

For more on this topic, please refer to our book "Interpreting regression Output" available at http://www.vjbooks.net.

CHAPTER 13

OTHER TOOLS FOR STATISTICS

This chapter briefly touches on the following topics:

— SAMPLING ANALYSIS

— RANDOM NUMBER GENERATION

— TIME SERIES

— EXPONENTIAL SMOOTHING, MOVING AVERAGE ANALYSIS

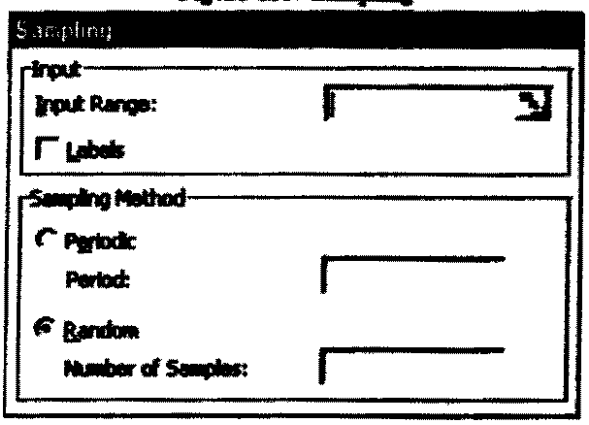
This chapter requires the Analysis ToolPak Add-Ins; chapter 9 shows how to learn how to launch the Add-Ins.

13.1 SAMPLING ANALYSIS

This tool creates a sample from a population by treating the input range as a population. You can use a representative sample when the population is too large to process or chart. You can also create a sample that contains only values fro a particular part of a cycle if you believe that the input data is periodic Excel draws samples from the first column, then the second column, and so on.

Access the feature through the menu path TOOLS/DATA ANALYSIS and choose the procedure "Sampling."

Figure 157: Sampling



Sampling Method: choose Periodic or Random to indicate the sampling interval you want.

Period: Enter the periodic interval at which you want sampling to take place. The interval value in the input range and every period's value thereafter are copied to the output column.

Random & Number of Samples: Number of random values you desire in the output column. Excel draws each value from a random position in the input range. (Consequently, a value may be drawn more than once.)

Output Range: Data is written in a single column below the cell.

> **Note**:
> If you selected *Periodic*, the number of values in the output table is equal to the number of values in the input range, divided by the sampling rate. If you selected *Random*, the number of values in the output table is equal to the number of samples.

13.2 RANDOM NUMBER GENERATION

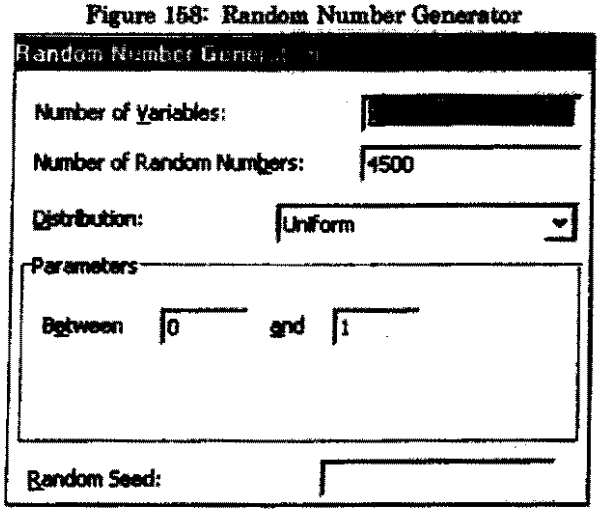
This tool fills a range with independent random numbers drawn from one of several Density Functions.

You can characterize a population with a Probability Density Function.

Select the option TOOLS/DATA ANALYSIS[34] and choose the procedure "Random Number Generation."

Number of Variables: Number of columns of values you want in the output table. If you do not enter a number, all columns in the output will be filled.

Figure 158: Random Number Generator



[34] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

Number of Random Numbers: Number of data points you want to see. Each point appears in a row of the output table. If you do not enter a number, all rows in the output range will be filled.

Distribution: choose the Density Function for defining the criterion for the Random Number generation.

Parameters: The base parameters for the generation process using the selected Density Function.

Figure 159: Choice of Density Functions

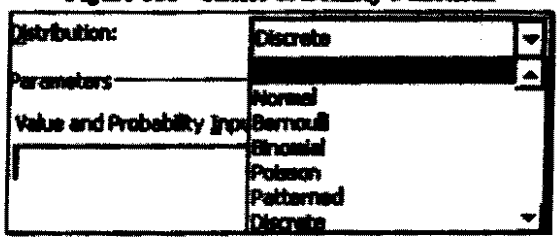
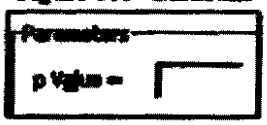
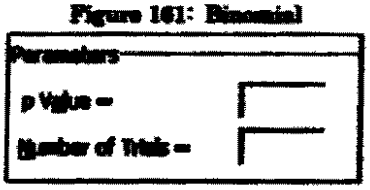


Table 39: Choice of Density Functions

Distribution	*Comment on setting parameters for random number generation*
Bernoulli	This Density Function is characterized by a probability of success (p value) on any given trial/observation. Figure 160: Bernoulli 
Binomial	This Density Function is characterized by a probability of success (p value) in any one trial for a number of trials. Figure 161: Binomial 
Discrete	Figure 162: Discrete Or Custom

Distribution	*Comment on setting parameters for random number generation*
Or Custom Density Function	The range must contain two columns: The left column contains values, and the right column contains probabilities associated with the value in that row. The sum of the probabilities must be 1. Note: You can use the function FREQUENCY (A1, A:A)/count (A:A) to generate the probability you see in column B. Density Function A: x, B: prob_x 2 3000 2.9% 3 3030 0.8% 4 3060 4.5% 5 3091 6.7% 6 3121 6.6% 7 3153 7.2% 8 3184 0.1% 9 3216 3.0% 10 3248 6.5% 11 3280 1.0% 12 3313 1.8% 13 3346 0.3% Figure 162: Parameters Parameters Value and Probability Input Range: $A$2:$B$31
Normal	This Density Function is characterized by a mean and a standard deviation.
Patterned	Figure 164: Patterned Random Number Generator Number of Variables: 1 Number of Random Numbers: 4500 Distribution: Patterned OK Cancel Help Parameters From 0 to 1000 in steps of 5 repeating each number 3 times repeating the sequence 1 times
Poisson	This Density Function is characterized by a value lambda, equal to (1/mean). Figure 165: Poisson Parameters Lambda =

230

Distribution	*Comment on setting parameters for random number generation*
Uniform	This distributing is characterized by lower and upper bounds. Excel draws variables from all values in the range. The probability of drawing a value is equal for all values in the range.

13.3 TIME SERIES

Exponential Smoothing

This tool and its formula predict a value based on the forecast for the prior period, adjusted for the error in that prior forecast. The tool uses the smoothing constant alpha, the magnitude of which determines how strongly forecasts respond to errors in the prior forecast.

Using the mouse, select the menu path TOOLS/DATA ANALYSIS[36] and choose the procedure "Exponential Smoothing."

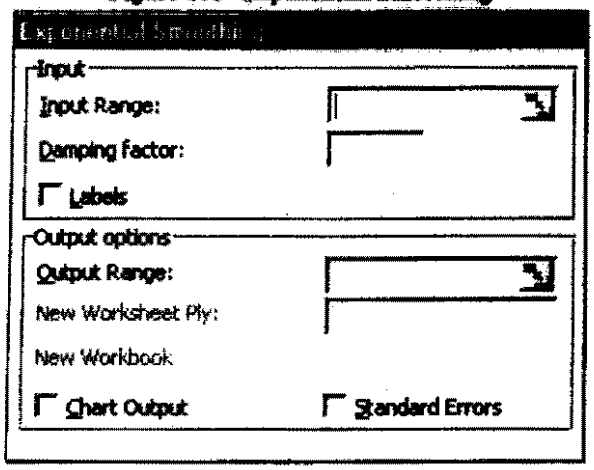
Damping: The factor you want to use as the exponential smoothing constant. The damping factor is a corrective factor that minimizes the instability of data collected across a population.

The default value for the damping factor is 0.3. Values of 0.2 to 0.3 are reasonable smoothing constants. These values indicate that the current forecast should be adjusted 20 to 30 percent for error in the prior forecast.

[36] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

Larger constants yield a faster response but can produce erratic projections. Smaller constants can result in long lags for forecast values.

Figure 166: Exponential Smoothing



Data Requirement: A single column or row with four or more cells with valid data.

Output: The output range must be on the same worksheet as the data in the input range. Enter the range reference for the upper— left cell of the output table (for example, "AD4"). You can obtain a column of Standard Errors by selecting the option "Standard Errors." If you want to chart the procedure's output — the actual values and forecasts –, select the option "Chart Output."

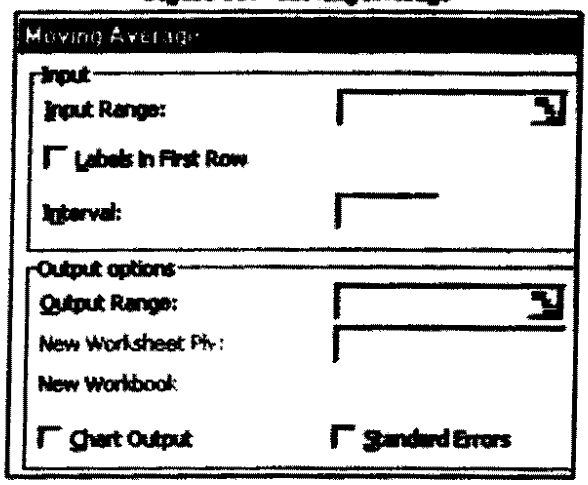
Moving Average analysis

This tool projects values in the forecast period based on "the average value of the series over a specific number of preceding periods." A moving average provides trend information that a simple average of all historical data would mask.

Select the option TOOLS/DATA ANALYSIS[36] and choose the procedure "Moving Average."

Interval: Number of values you want to include in the moving average. The default is three.

Figure 167: Moving Average



Data Requirement: A single column or row with four or more cells with valid data.

Output: The output range must be on the same worksheet as the data in the input range. Enter the range reference for the upper–left cell of the output table (for example, "AD4"). You can obtain a column of Standard Errors by selecting the option "Standard Errors." If you want to chart the procedure's output — the actual values and forecasts –, select the option

[36] If you do not see this option, then use TOOLS / ADD-INS to activate the Add-In for data analysis. Refer to section 41.4.

You have activated the "Analysis ToolPak." If you go to the menu TOOLS, you will see the option "SOLVER"— this option was not there before you accessed the Add-In. Please define a sample problem and try it on an Excel workbook[37].

Access the feature through the menu path TOOLS/SOLVER. The dialog shown in Figure 169 opens. The "Target Cell" contains the formula for the function you are attempting to optimize.

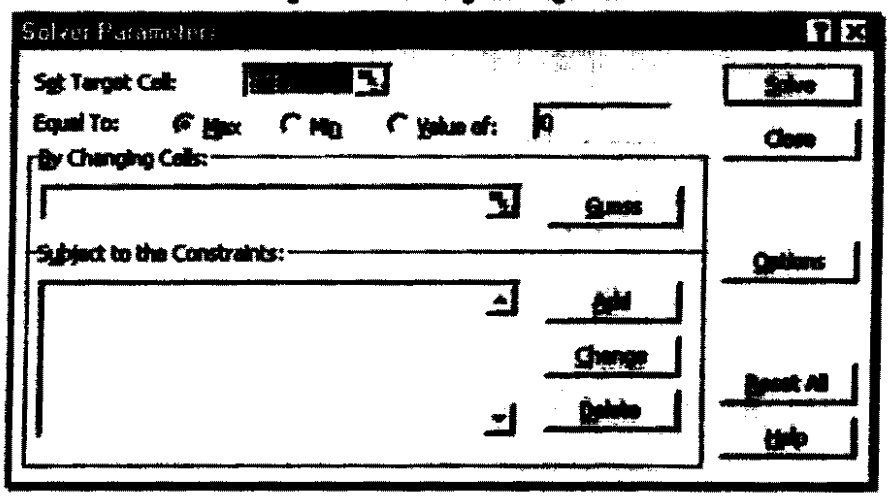
The "Equal to" area is where you choose the optimization criterion--

— Maximization (Max)

— Minimization (Min)

[37] I do not supply the sample data for most of the examples in chapter 42 to chapter 46. My experience is that many readers glaze over the examples and do not go through the difficult step of drawing inferences from a result if the sample data results are the same as those in the examples in the book.

Figure 169: Setting the target cell

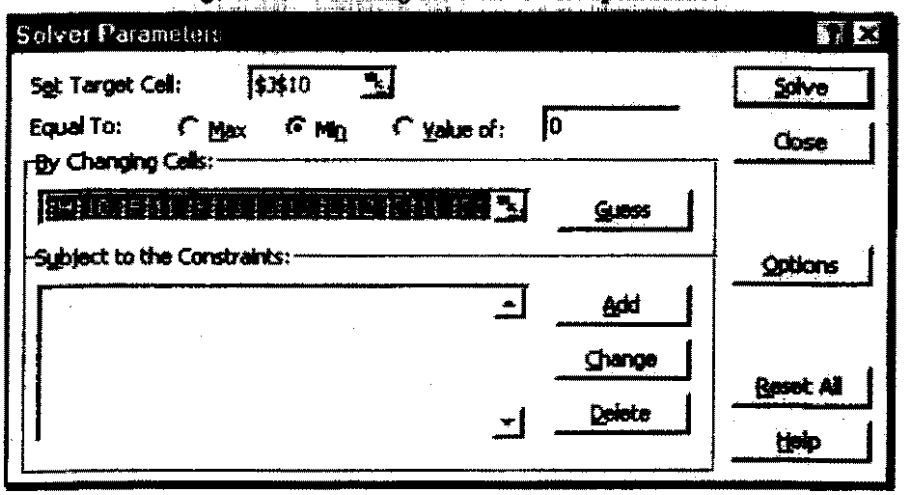


The choice parameters are the numbers the algorithm plays around with to find the max/min.

You have to tell Excel about the cells that contain these parameters. One can do it manually, or, an easier option is to click on the button "Guess."

Excel automatically chooses all the cell references for use in the formula in J10 (the target cell/objective function). This is illustrated in Figure 170.

Figure 170: Selecting the criterion for optimization

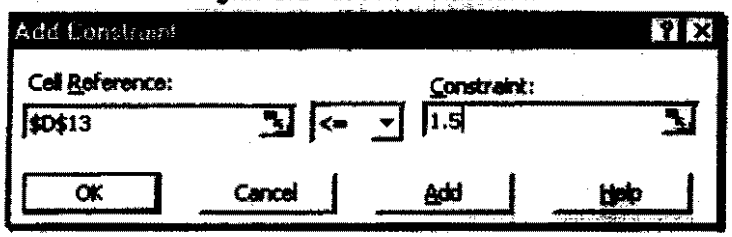


14.2 ADDING CONSTRAINTS

The optimization function has been defined, as have the "choice parameters." At this stage, you have to add the constraints.

Click on the button "Add" and write in a constraint as shown in Figure 171.

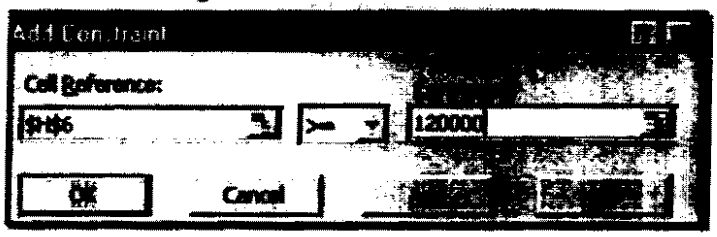
Figure 171: The first constraint



After defining the first constraint, click on the button "Add" (see Figure 171.) Write the second constraint— see Figure 172.

237

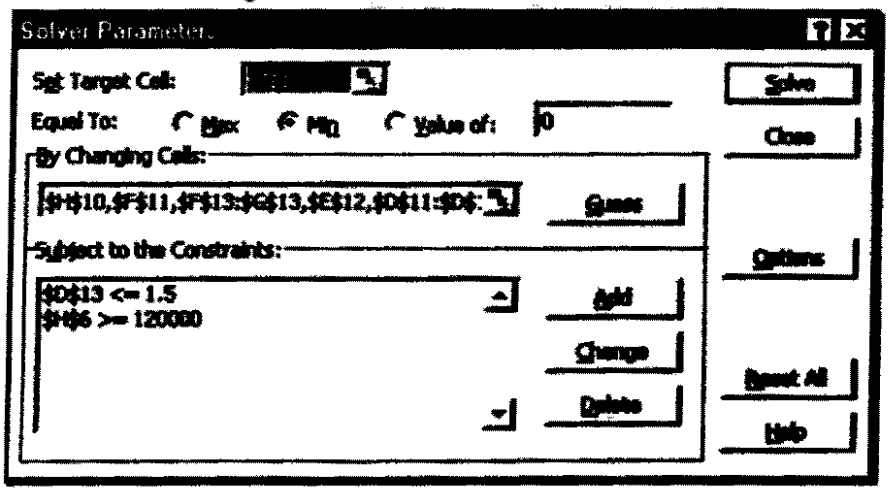
Figure 172: The second constraint



Continue with constraint definitions. After defining the last constraint, execute the dialog by clicking on the button OK (see Figure 172).

> **Note:**
> The constraints are shown in the area "Subject to the Constraints" as shown in Figure 173.

Figure 173: The constraints for the Solver



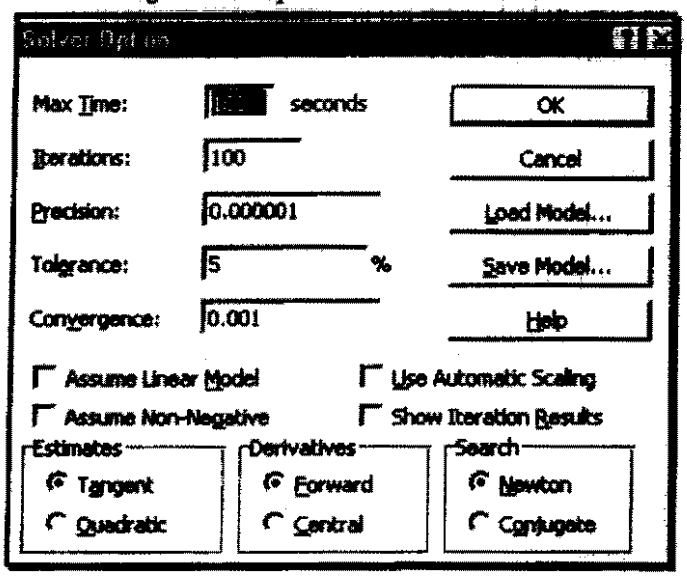
14.3 CHOOSING ALGORITHM OPTIONS

You need to choose the options for the analysis. So, click on the button "Options." The dialog shown in Figure 174 opens.

You may want to increase the iterations to 10,000. If you want to relax the requirements for preciseness, increase the value of "Precision" by removing some post-decimal zeros.

"Save Model" is used to save each optimization model. You can define several optimization problems in one workbook. The other options are beyond the scope of this book. Click on the button "Continue."

Figure 174: Options in the Solver Add-In



Running the Solver

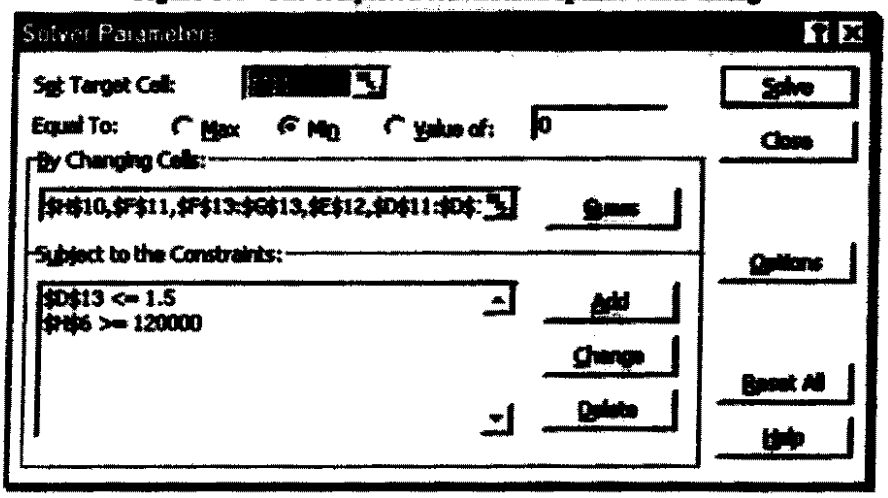
Execute the procedure by clicking on the button "Solve."

The following output can be read from the spreadsheet.

- the optimized value of the Objective Function (that is, the value of the formula in the cell defined in the box "Set Target Cell")

- is the combination of the choice variables (that is, those whose value is obtained from the cells defined in the dialog area "By Changing Cells")

Figure 175: The completed constrained optimization dialog



INDEX

#

μ 122

σ2122

A

A1 ..... 25, 28, 232

ABS 153

ADD–IN 161

ADD-INS . 17, 161, 163, 165, 170, 176, 178, 187, 190, 195, 199, 205, 219, 231, 234, 236, 240

ADD–INS INSTALLED WITH EXCEL 161

AND 35, 52, 109, 144, 161, 169

ANOVA . 129, 156, 163, 183, 187, 203, 205, 206, 207, 208

AUDITING 17, 76, 78, 79, 80, 81, 84

AUTOCORRECT 17

AUTOFORMAT 16

AVEDEV 156, 157

AVERAGE 66, 89, 90, 106, 155

AVERAGEA 106

B

BETADIST 113, 115, 132, 140, 141

BETAINV 133, 134, 140, 141

BINOMDIST 113, 140

BIVARIATE 169

C

CDF 109, 110, 111, 112, 113, 114, 115, 119, 120, 121, 123, 125, 127, 128, 129, 130, 132, 133, 134, 136, 137, 138, 140

CELL 25, 52, 89

CELL REFERENCE 25

CELLS 15, 16, 52

CENTRAL TENDENCY 89

CHIDIST 113, 115, 130, 131, 140, 141

CHIINV 131, 140, 141

CHI-SQUARE DENSITY FUNCTION. 109

CHOOSE 153

CLEAR 14

COLUMN 16, 52

COLUMNS 15, 52

COMMENT 16

COMMENTS 14, 35, 52

CONDITIONAL FORMATTING 16

CONFIDENCE 69, 70, 71, 109

CONFIDENCE INTERVAL 109

CONSOLIDATION 17

CONSTRAINTS 239

CONTROLLING CELL REFERENCE BEHAVIOR WHEN COPYING AND PASTING FORMULAE (USE OF THE 35

COPY 13, 36, 37, 38, 39, 42

COPYING AND PASTING 35, 36

COPYING AND PASTING A FORMULA TO OTHER CELLS IN A DIFFERENT ROW AND COLUMN 35

COPYING AND PASTING A FORMULA TO OTHER CELLS IN THE SAME COLUMN 35

COPYING AND PASTING A FORMULA TO OTHER CELLS IN THE SAME ROW 35

COPYING AND PASTING FORMULAS FROM ONE WORKSHEET TO ANOTHER 35

CORREL 67, 68, 157

CORRELATION 144, 169, 179, 180

COS 81, 82, 83, 84

COUNT 106, 144, 145, 146, 147

COUNTA 106, 144, 147

COUNTBLANK 144, 148

COUNTIF 144, 151, 152, 153

COUNTING AND SUMMING 144

COVAR 157

COVARIANCE 144

CROSS SERIES RELATIONS 144

CUMULATIVE DENSITY FUNCTION 109

CUSTOMIZE 17

242

INSERT ... 15, 44, 46, 47, 61, 63, 67, 69, 71, 72, 90, 91, 92, 93, 95, 96, 97, 99, 100, 104, 105, 120, 121, 123, 124, 125, 126, 129, 131, 132, 133, 134, 135, 136, 145, 147, 148, 149, 151, 152, 153, 155, 158, 159, 225

INVERSE MAPPING 109

K

KURT 105

KURTOSIS 89

L

LARGE 89, 98

LINKS 14

LN 154

LOG 115, 144, 155

LOG10 154

LOGINV 140, 141

LOGNORMDIST 113, 115, 140, 141

M

MACROS 17, 161

MAX 97, 98, 106

MAXA 97, 98, 106

MEDIAN 89, 95

MIN 98, 106

MINA 98, 106

MODE 84, 89, 95

MOVE OR COPY SHEET 14

MOVING AVERAGE 229

MULTIPLE RANGE REFERENCES 61

MULTIPLYING/DIVIDING/SUBTRACTING/ADDING ALL CELLS IN A RANGE BY A NUMBER 52

N

N 136, 146, 174, 214

NA 15, 44, 47, 83, 146

NEGBINOMDIST 113, 140

NORMAL DENSITY FUNCTION 109, 144

NORMDIST 113, 115, 119, 120, 140, 141

NORMINV 122, 123, 140, 141

NORMSDIST 113, 115, 123, 140, 141

NORMSINV 140, 141

NOT 35, 52, 161

O

OBJECT 14, 16

OBJECTIVE FUNCTION 239

OFFICE ASSISTANT 18

OFFICE CLIPBOARD 14

ONLINE COLLABORATION 17

OPEN 13

OPTIMIZATION 239

OPTIMIZATION CRITERION 239

OPTIONS 17, 26, 28, 35, 52

OR 89

P

PAGE BREAK 14, 15

PAGE BREAK PREVIEW 14

PAGE SETUP 13

PAIRED SAMPLE T-TEST 183

PASTE 13, 14, 35, 36, 37, 38, 40, 47, 49, 53, 54, 56, 57, 58, 62

PASTE SPECIAL 14, 53, 54, 56, 57, 58

PASTING ALL BUT THE BORDERS 52

PASTING COMMENTS 52

PASTING DATA VALIDATION 52

PASTING ONLY FORMATS 52

PASTING ONLY THE FORMULA 35, 52

PASTING THE RESULT OF A FORMULA, BUT NOT THE FORMULA ITSELF 35

PDF .109, 110, 112, 113, 119, 121, 127, 133, 134, 136, 137, 138, 140

PEARSON 157

PERCENTILE 89, 96, 97, 169

PERCENTRANK 99

PIVOT REPORT 17

POISSON 109, 113, 138, 140

PRECEDENTS 76

PRINT AREA 13

PRINT PREVIEW 13

PROBABILITY DENSITY FUNCTION109

PRODUCT 144, 149

PROPERTIES 13

PROTECTION 16

Q

QUARTILE 89, 96

R

R1C1 ..... 25, 28

RANDOM NUMBER GENERATION ..229

RANK 89, 100, 169

REDO 13

REFERENCES ALLOWED IN A FORMULA 25

REFERENCING A BLOCK OF CELLS..25

REFERENCING CELLS FROM ANOTHER WORKSHEET 25

REFERENCING CORRESPONDING BLOCKS OF CELLS / ROWS / COLUMNS FROM A SET OF WORKSHEETS 25

REFERENCING ENTIRE COLUMNS....25

REFERENCING ENTIRE ROWS 25

REFERENCING NON– ADJACENT CELLS 25

REGRESSION 211

REPLACE 14

ROW 16, 52

ROWS 15, 52

ROWS TO COLUMNS 52

RSQ 157

S

SAMPLING ANALYSIS 229

SAVE 13

SAVE AS 13

SAVE AS WEB PAGE 13

SAVE WORKSPACE 13

SCENARIOS 17

SEARCH 13

SHARE WORKBOOK 16

SHEET 16

SIGN 35, 153

SKEW 104

SKEWNESS 89

SMALL 99

SOLVER 239, 241

SORT 17

SPEECH 16

SPELLING 16

SPLIT 18

SPSS 3, 5, 170

SQRT 153

STANDARD DEVIATION 89

STANDARD NORMAL OR Z- DENSITY FUNCTION 109

STANDARDIZE 155, 225

STATA 5

STATUS BAR 14

STDEV 71, 72, 89, 100, 101, 106, 155

STDEVA 89, 100, 101, 102, 106

STDEVP 89, 101, 106

STDEVPA 89, 101, 106

STYLE 16, 25

SUBTOTALS 17

SUM 33, 144, 145, 148

SUM OF THE SQUARES OF DIFFERENCES ACROSS TWO VARIABLES 145

SUM OF THE SUM OF THE SQUARES OF TWO VARIABLES 144

SUMIF 144, 150, 151, 153

SUMPRODUCT 144, 149

SUMX2MY2 ..... 158, 159

SUMX2PY2 ..... 157, 158

SUMXMY2 158

T

T 23, 109, 115, 119, 125, 126, 127, 128, 163, 183, 189, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 218, 223

T- DENSITY FUNCTION 109

TABLE 17, 50

TDIST 113, 115, 125, 140, 141

TIME.....36, 229

TIME SERIES 229

TINV..... 121, 124, 126, 127, 128, 140, 141

TOOLBARS 14, 80

TOOLS16, 17, 26, 28, 76, 78, 79, 80, 81, 84, 163, 165, 170, 171, 173, 176, 178, 187, 190, 195, 199, 205, 219, 229, 231, 234, 236, 240, 241

TRACE 76, 78

TRACING THE CELL REFERENCES USED IN A FORMULA.....76

TRACING THE FORMULAS IN WHICH A PARTICULAR CELL IS REFERENCED.....76

TRIMMEAN.....91, 92, 94

TRIMMED MEAN.....89

TRUE..... 100, 101, 137, 145, 146, 148

T-TEST

TWO– SAMPLE ASSUMING EQUAL VARIANCES..... 183

TWO– SAMPLE ASSUMING UNEQUAL VARIANCES 183

T-TESTING MEANS WHEN THE TWO SAMPLES ARE FROM DISTINCT GROUPS 183

U

UNDO.....13, 49, 59

V

VALIDATION.....17

VALUE.....89, 146

VAR 89, 100, 106

VARA 89, 100, 101, 106

VARIANCE.....89

VARP.....89, 101, 106

VARPA 89, 101, 106

VIEW 14, 26, 27, 80

W

WEB.....17

WEIBULL..... 109, 113, 138, 140

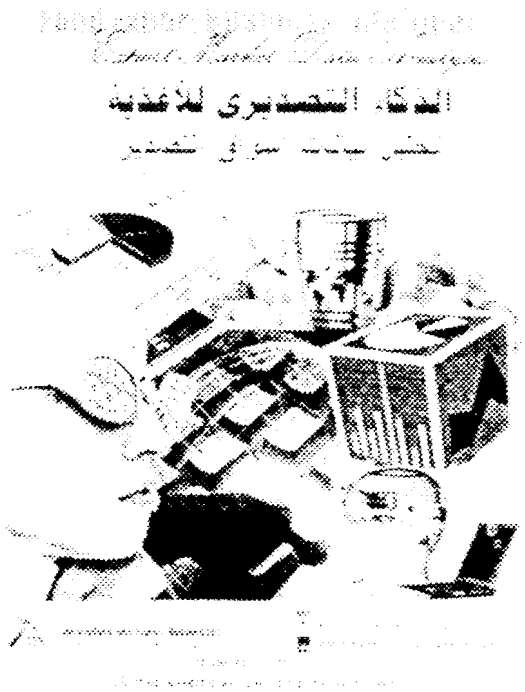
WINDOW 18, 76, 80

WORKSHEETS.....15, 36

Z

ZOOM 15

Z-TESTING FOR POPULATION MEANS 183

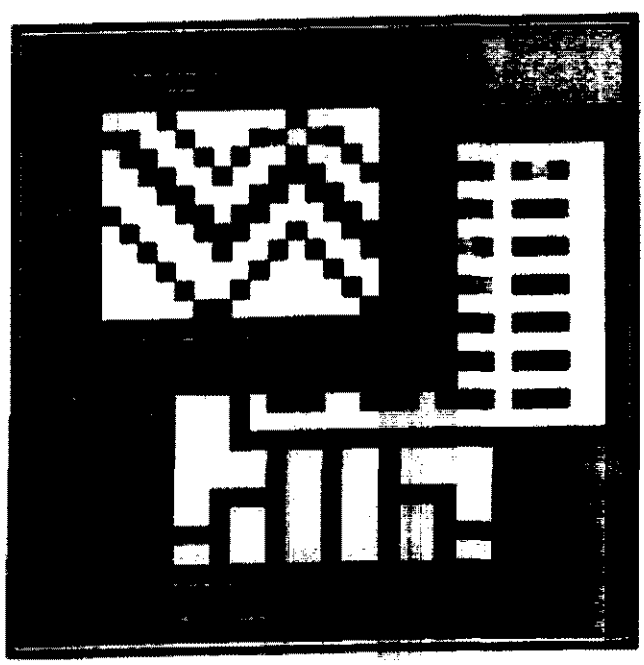


The Art of Market Signals Analysis Using E-Views

Nicholas R. Noble

EViews

User's Guide



EViews

Opening an *EViews* Workfile

Getting Data.
Existing Data files -

Click on

FILE-OPEN

Choose the correct workfile, and

Click **OPEN**

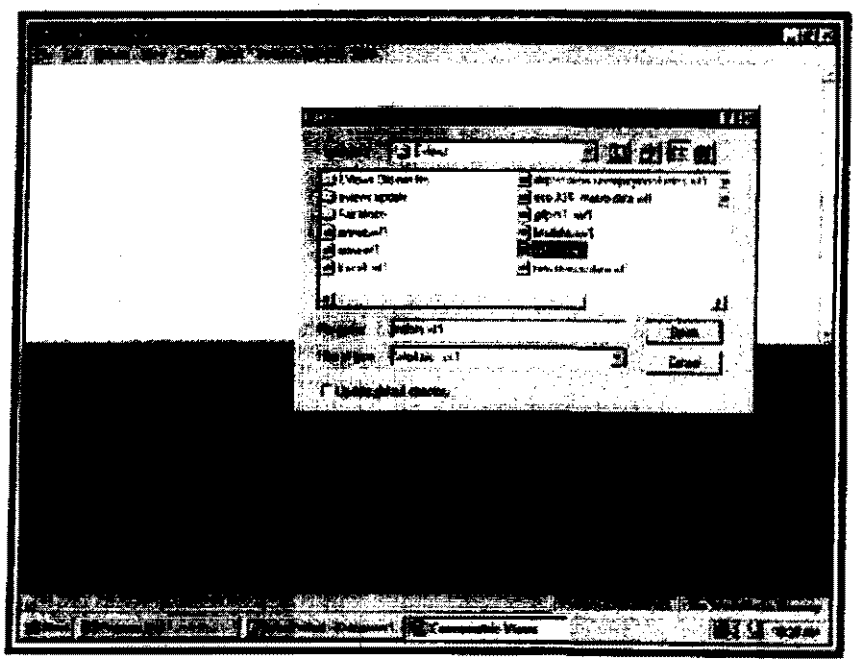


Figure 1

EViews will open the requested file, and provide a list of variables and objects in the file.

The open box on the screen is called the **workfile box.** (see Figure 2)

The buttons with the labels VIEW, PROCS, SAVE, etc. are referred to as the toolbar.

EViews has several different toolbars that we will use.

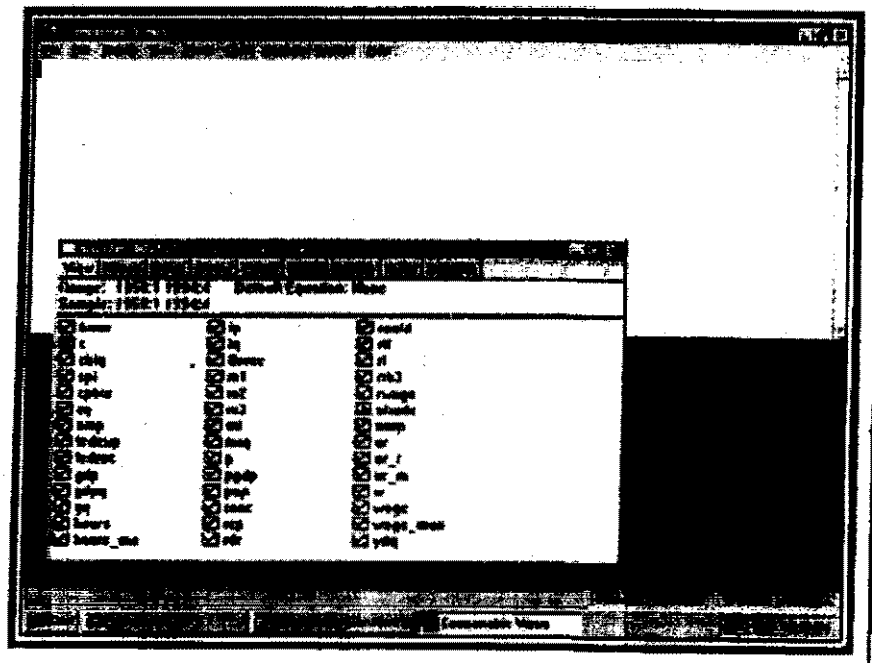


Figure 2

SAVING An *EViews* Workfile

To save a file,

Click on

FILE - SAVE

Alternatively, Click

SAVE on the toolbar on the workfile box.

or

FILE- SAVE AS

to change the name of the file

Note: To take the file with you, you must save the file to a diskette in the A:\ drive, instead of the computer's hard drive.

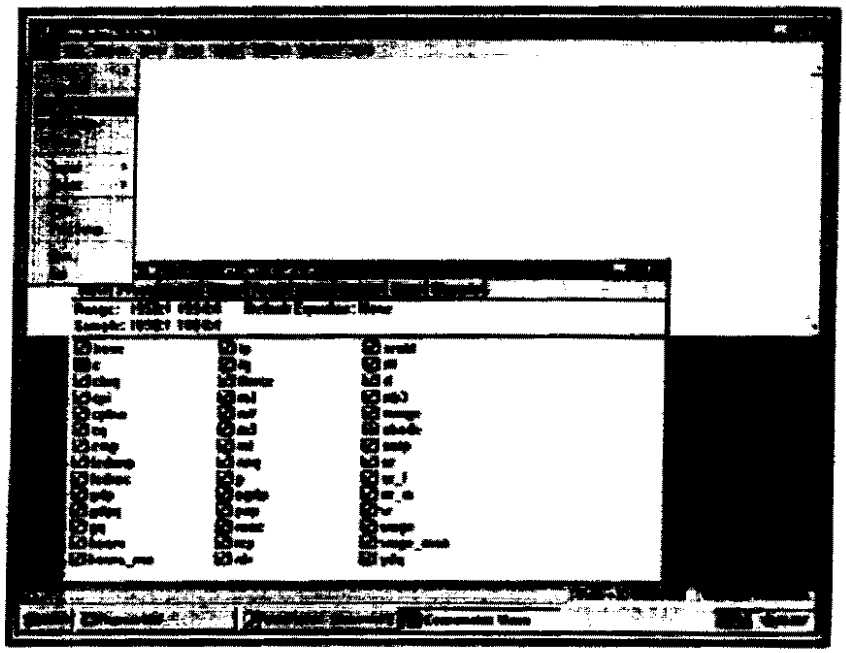


Figure 3

Choosing the Observations (Sample Size)

The portion of the data that EViews uses for analysis is determined by sample size.

Sample size can be changed from the defaults by using the QUICK menu. (Figure 4)

Click on

QUICK

then click on

SAMPLE

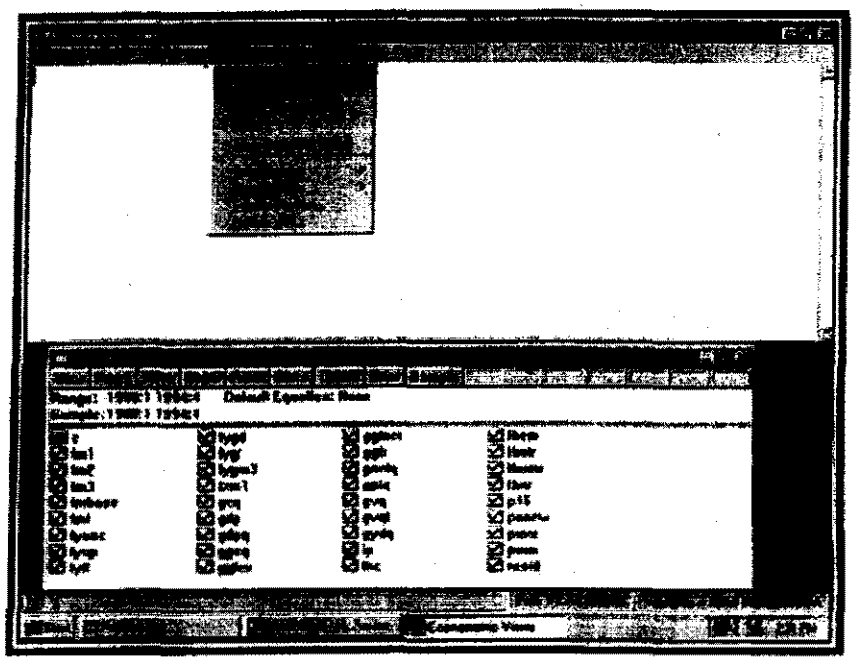


Figure 4

In the sample box (Figure 5) fill in the starting and ending date. You must use Proper EViews date form. The correct form for a date is:

Annual Data
1960 for the year 1960
Quarterly Data
1960:1 (or 60:1) for first quarter in 1960. Use 60:3 for third quarter.
Monthly Data
1960:4 (or 60:4) for April 1960. Use 60:10 for Oct. 1960.
Undated Data
Use observation numbers.

Skipping Data
To skip a set of observations, use four dates, start date 1, end date 1, start date 2, end date 2.

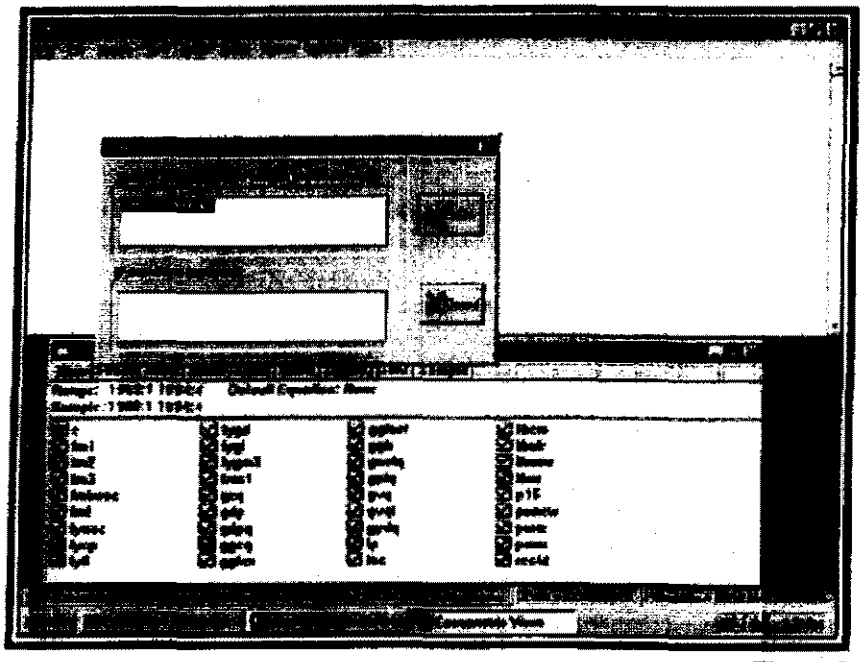


Figure 5

PRINTING DATA

To Print data, use the **QUICK** menu (Figure 4), and choose

SHOW

Type the variable name in the box. (Figure 6)

and Click on **OK**

Alternatively, click on SHOW in the workfile box.

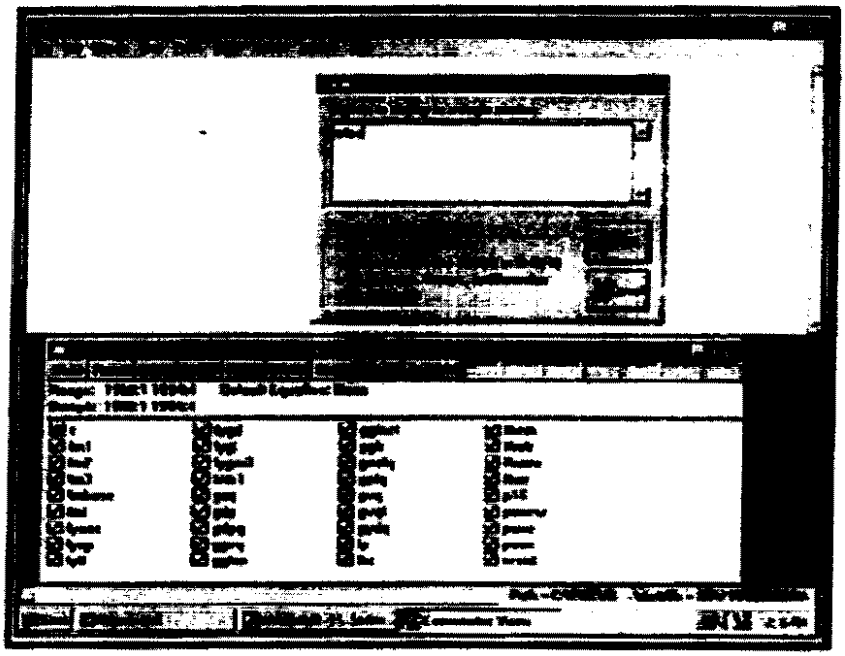


Figure 6

EViews will show the data in a spreadsheet form. (Figure 7)
This is called the Spreadsheet box.

Click on **PRINT** to send this information to the printer.

Alternatively, the Spreadsheet box will appear if you double-click on a variable in the workfile box.

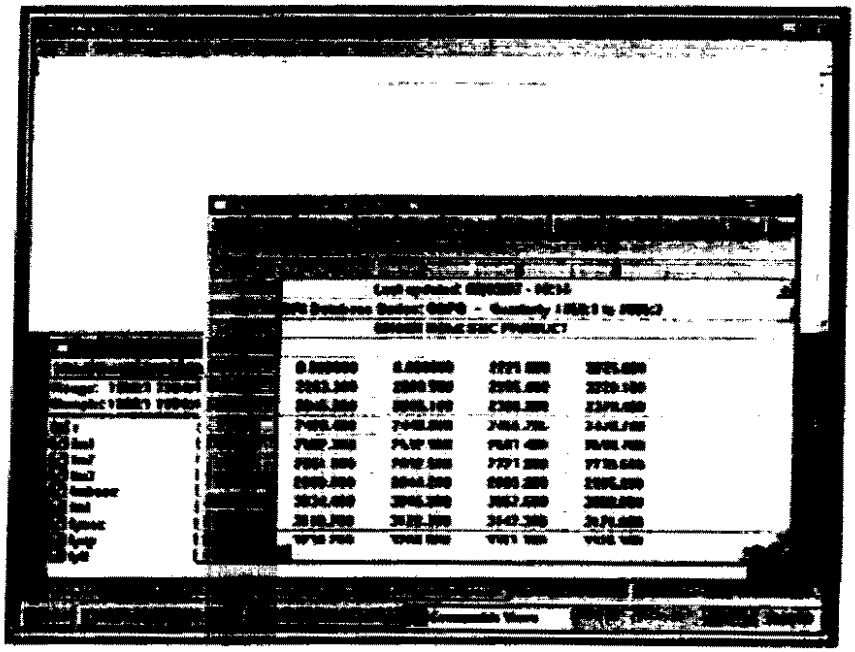


Figure 7

TO print more than one variable, choose **QUICK** and **SHOW**

and then enter the names of the variables you wish to print.
(Figure 8)

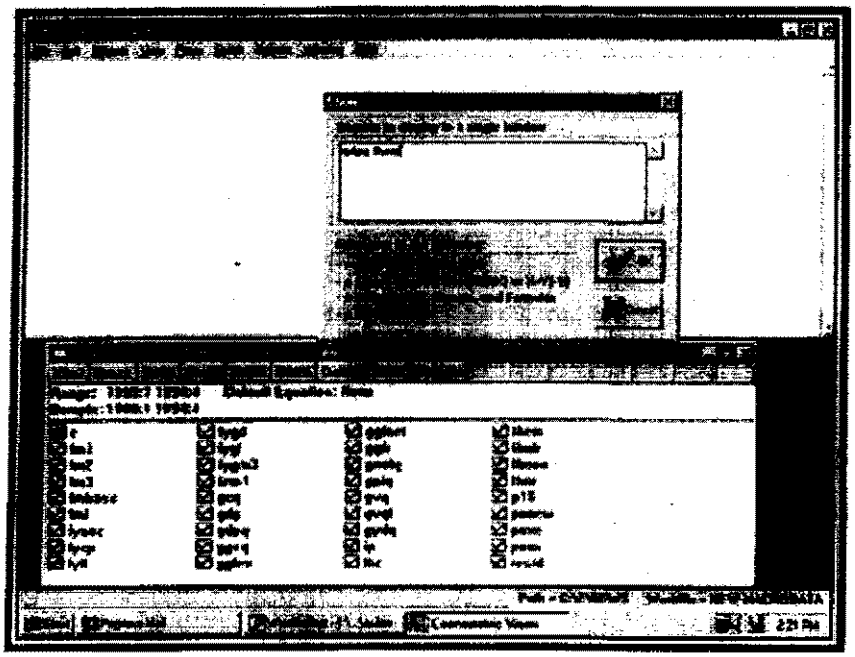


Figure 8

Use the **PRINT** button on the toolbar to send the data to the printer,

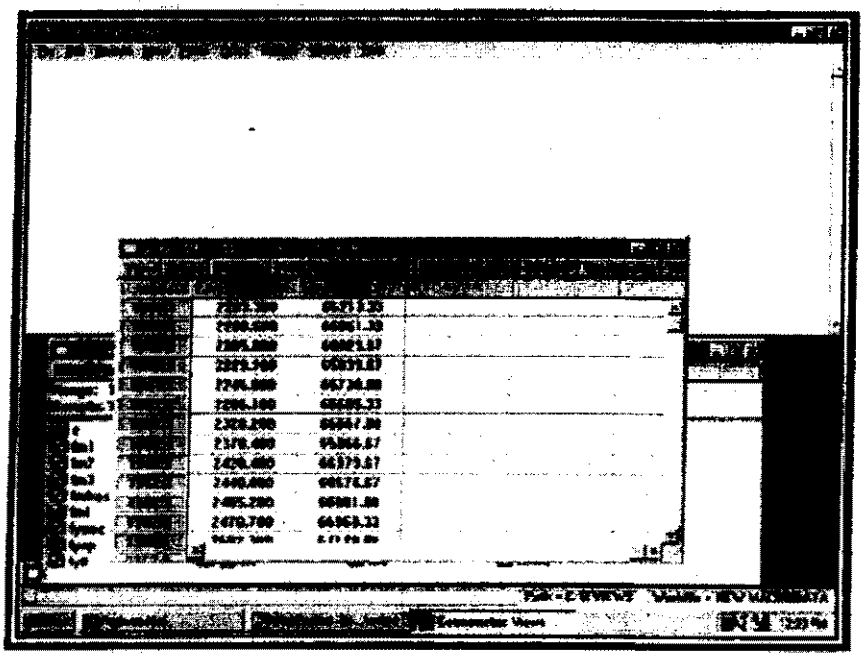


Figure 9

255

PLOTTING DATA

EViews does an excellent job plotting data. To plot two variables together, use the QUICK menu, and choose

GRAPH

Fill in the names of the variables in the box (figure 10) and click on OK

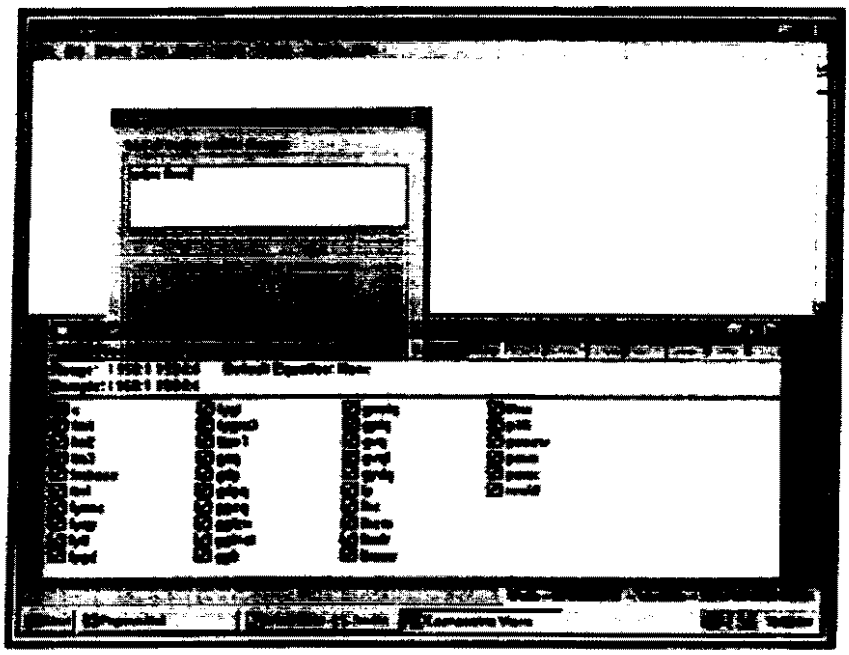


Figure 10

EViews has several graphic options. The default is a line graph.
If the variables that you plot are similar in magnitude, use

SINGLE SCALE

If the variables are different in magnitude, use one of the DUAL SCALES options

(Figure 11)

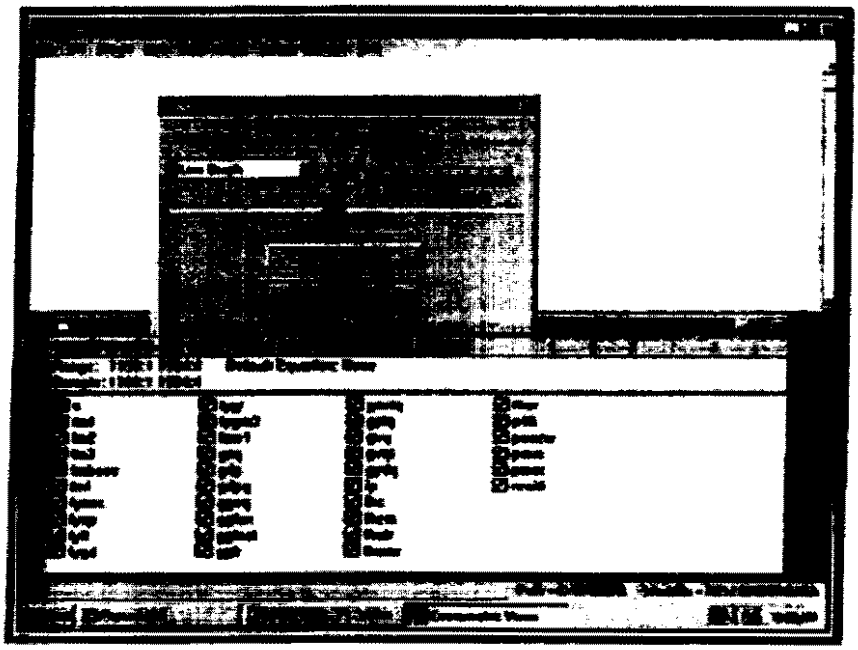


Figure 11

The resulting Graph can be printed using the

PRINT

button on the graphic toolbar.

The PrintSetup button determines the size and attributes of the printed graph.

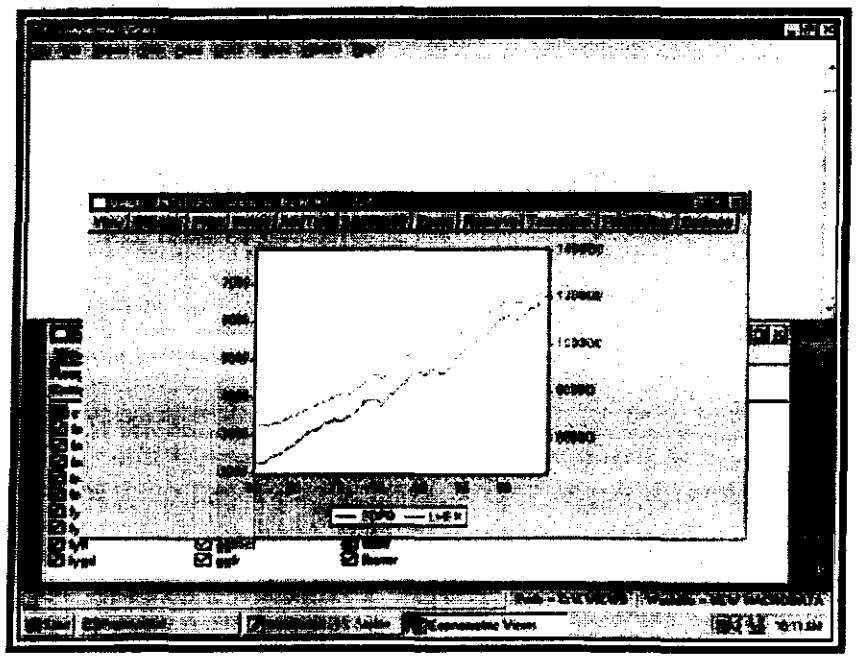


Figure 12

The AddText button (Figure 13) on the graphics toolbar allow the addition of a title or other information to the graph

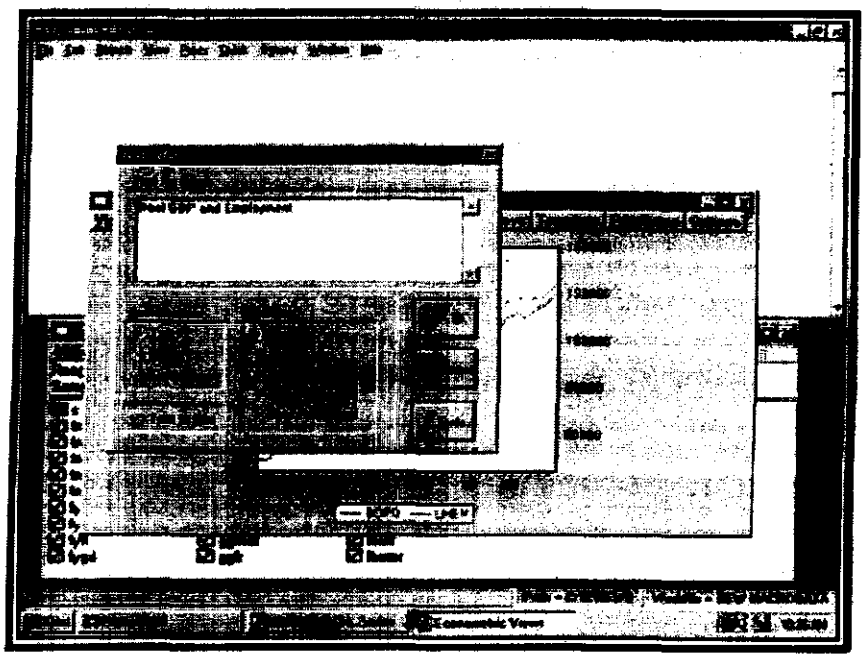


Figure 13

257

This information will also be printed along with the graph.

The name button on the toolbar will give the graph a name, and the graph can be stored in the workfile along with the data.

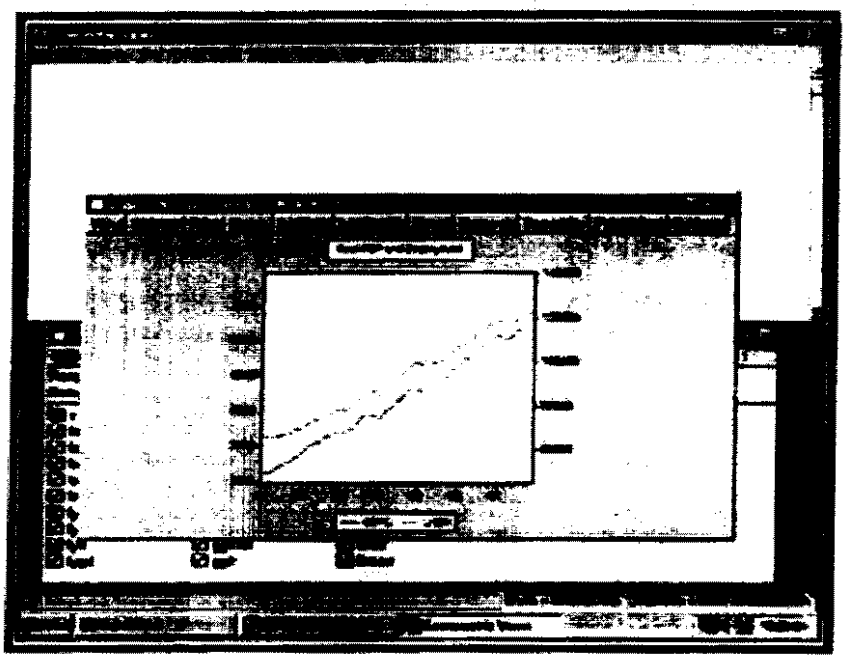


Figure 14

Using the add text allows for axis titles.

The AddShade allow time periods to be emphasized. (Figure 15)

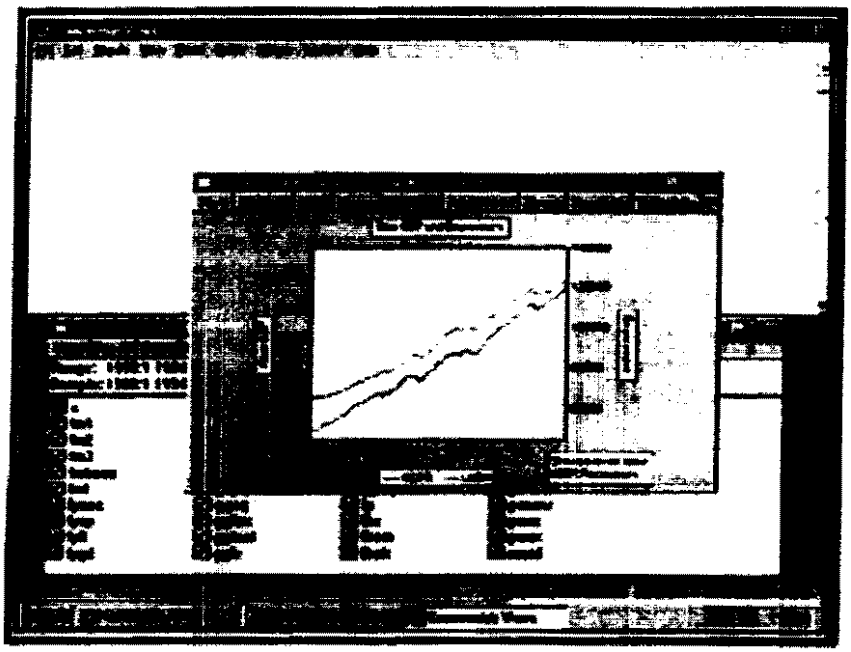


Figure 15

One other graphic option is a scattergram. A scattergram plots one variable on the y-axis and one variable on the x-axis.

To produce a scattergram, follow the steps in figure 10, but choose

SCATTER DIAGRAM

option. The click on OK.

Note the scale options play no role if scatter diagram is chosen.

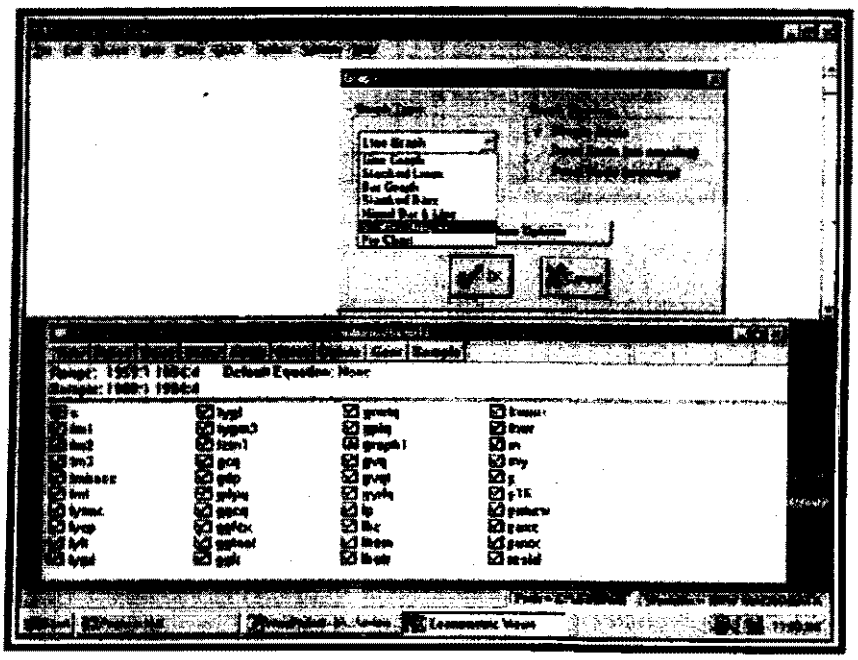


Figure 16

The variable listed first will be plotted on the y-axis, and the variable listed second will be plotted on the x-axis.

Note the AddText button on the graphics toolbar can be used to add information to the graph as done in figure 15.

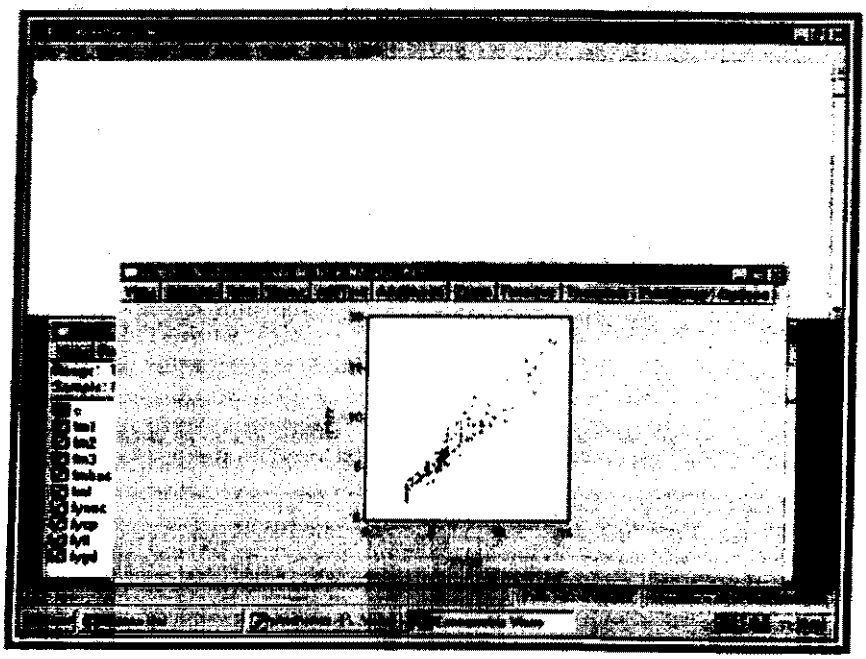


Figure 17

259

The option button on the graphics toolbar produces a menu of many graphics options to "fine-tune" the graph. These options are available on the line graph above also.

One interesting option with a scatter diagram is the **regression line option.** This option draws a straight line that "best" describes the data.

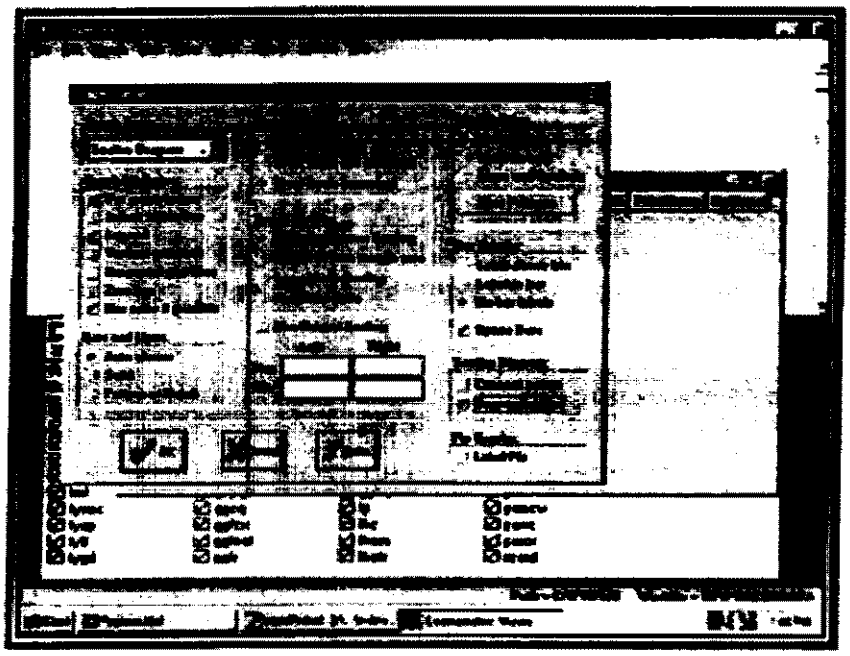


Figure 18

The scatter diagram can be printed with the

PRINT button on the toolbar.

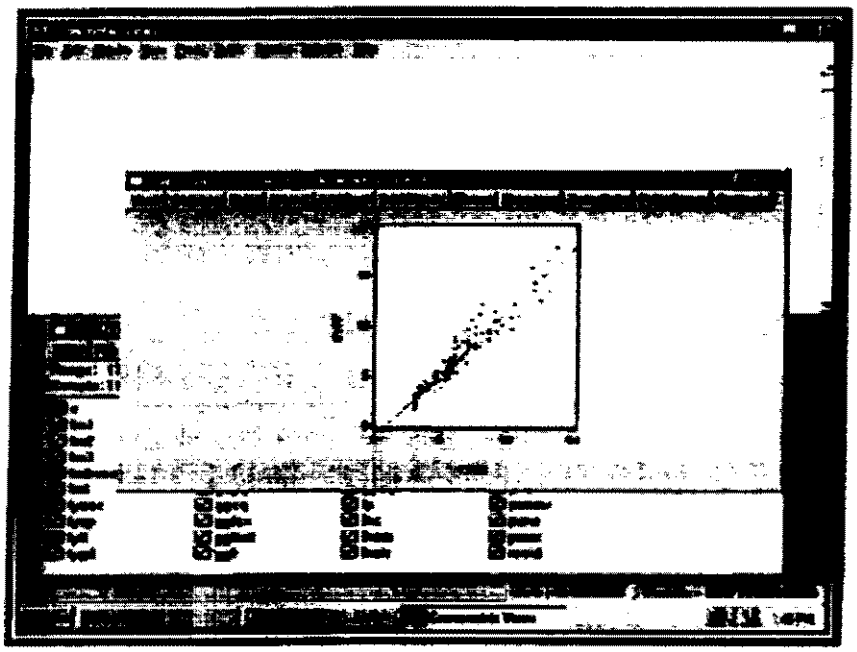


Figure 19

Creating New Variables

To create new variables in *EViews*, start with

Generate Series

on the Quick Menu (see Figure 4).

Enter the equation in the box to describe the variable that you wish to create.

Use a * for multiplication, use a / for division, use ** for raising to a power, and + and - for addition and subtraction. The example in figure 20 creates an inflation rate over a year span.

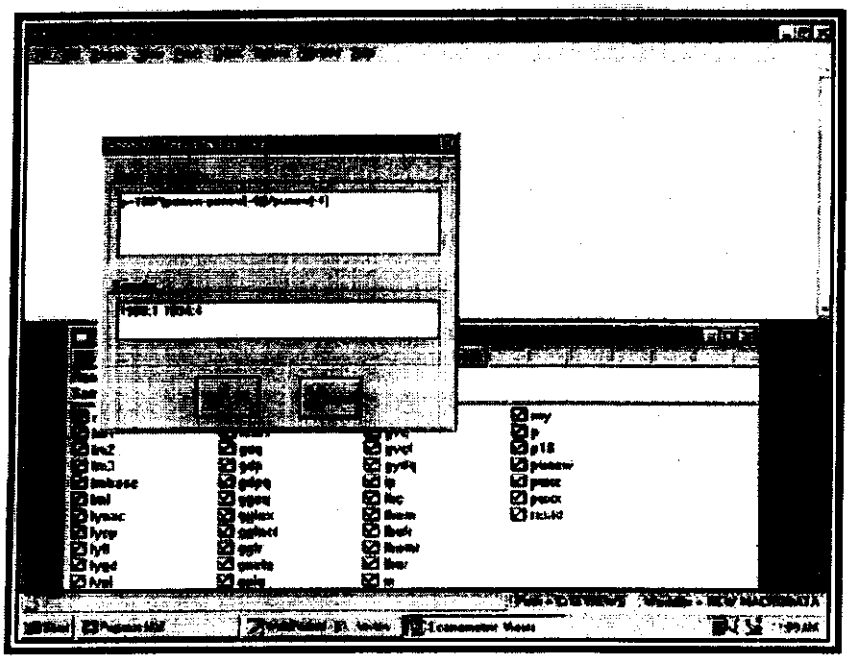


Figure 20

An alternative to the Quick Menu is to use the **GENR** button on the workfile toolbar.

After you compute a new series you should always plot and print the series as a check of your computational formula.

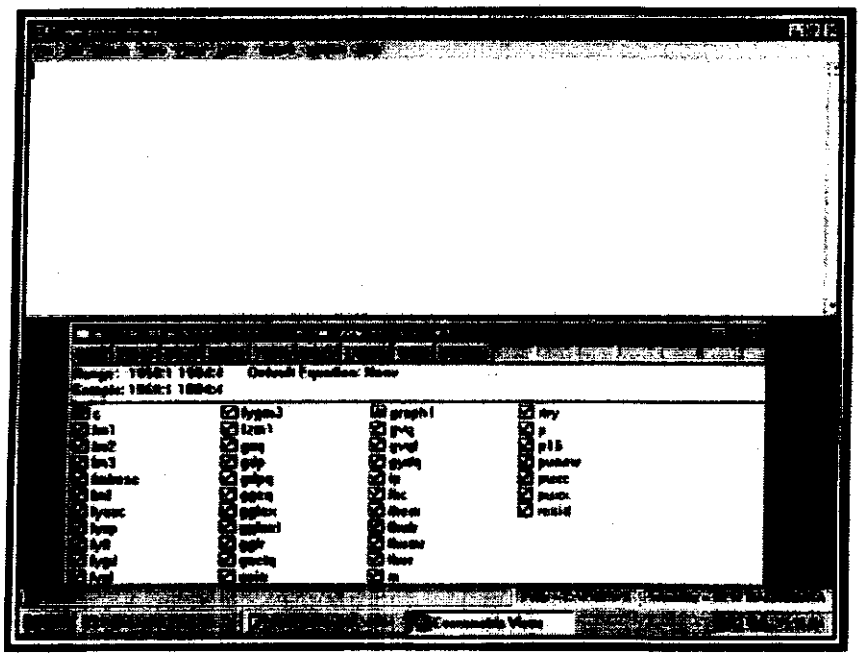


Figure 21

261

Estimating Regression Equations

To estimate a Regression equation, start with the QUICK MENU (figure 4) and choose

Estimate Equation...

If the equations to be estimated is:

$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$

Enter in the box,

Y C X

where C indicates to *EViews* to include a regression constant. The equation entered in the box estimates the federal funds rate as a function of the discount rate.

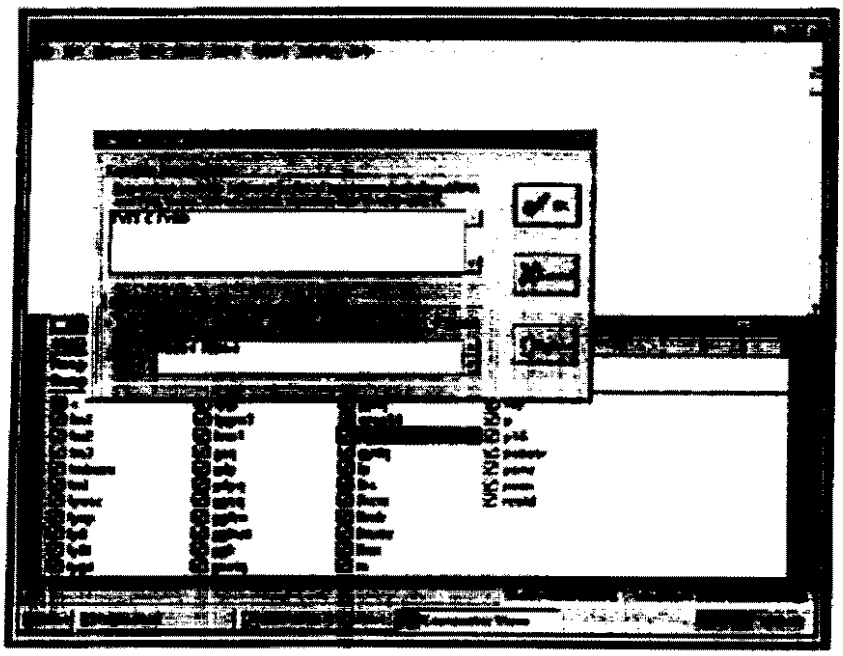


Figure 22

Figure 23 show standard *EViews* regression output. This is called the Regression box.

The PRINT button on the regression box toolbar will send the regression results to the printer.

Note date and time are included.

The name button will store the equation in the workfile.

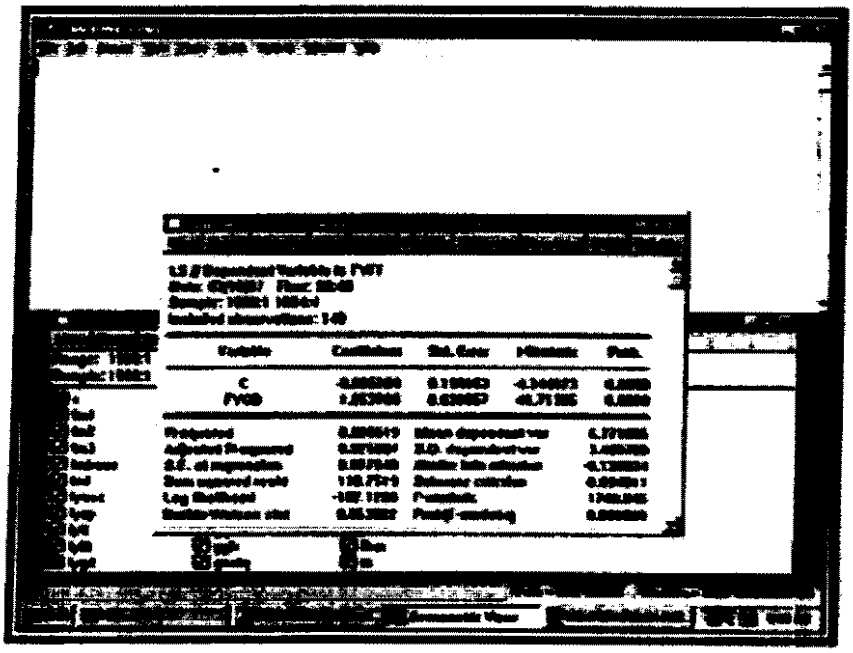


Figure 23

262

The **Resids** button on the regression box toolbar will generate time series graph of the actual and fitted(predicted) values and regression residuals. The PRINT button on the toolbar will now print this graphic. To get back to the regression results, click on Stats.

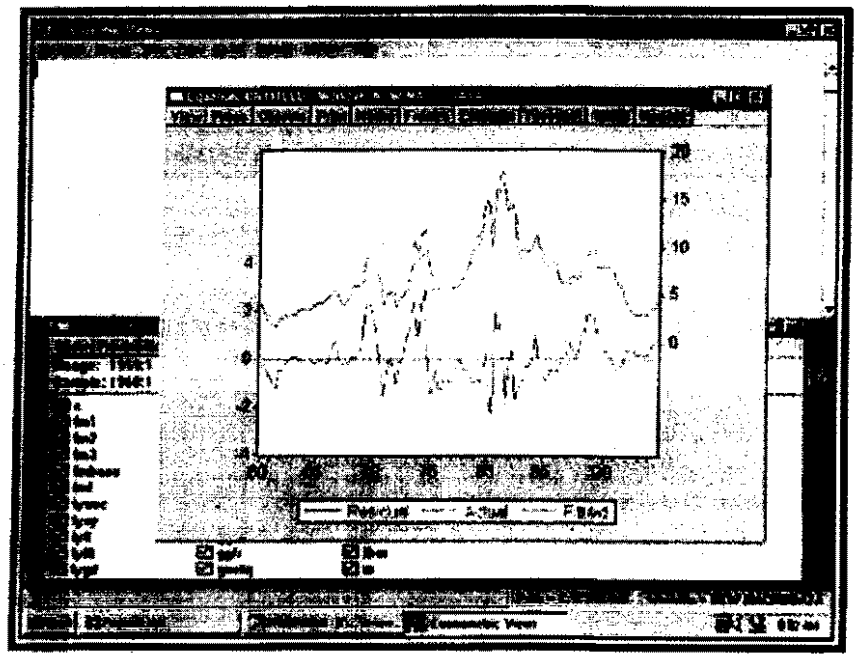


Figure 24

The residuals are stored in a series called resid. If you want to use this variable you must calculate a new variable based upon resid.

Use the Quick Menu, choose generate series, and enter a formula such as, err=resid. Now err is a variable that can be used in a regression equation, printed, plotted, etc.

Part III. Basic Single Equation Analysis

The following chapters describe the EViews features for basic single equation analysis.

- Chapter 11, "Basic Regression", beginning on page 259 outlines the basics of ordinary least squares estimation in EViews.
- Chapter 12, "Additional Regression Methods", on page 279 discusses weighted least squares, two-stage least squares and nonlinear least square estimation techniques.
- Chapter 13, "Time Series Regression", on page 303 describes single equation regression techniques for the analysis of time series data: testing for serial correlation, estimation of ARMAX and ARIMAX models, using polynomial distributed lags, and unit root tests for nonstationary time series.
- Chapter 14, "Forecasting from an Equation", beginning on page 343 outlines the fundamentals of using EViews to forecast from estimated equations.
- Chapter 15, "Specification and Diagnostic Tests", beginning on page 367 describes specification testing in EViews.

Additional single equation techniques for autoregressive conditional heteroskedasticity, and discrete and limited dependent variable models are described in Part IV. Part V documents multiple equation analysis.

Chapter 11. Basic Regression

Single equation regression is one of the most versatile and widely used statistical techniques. Here, we describe the use of basic regression techniques in EViews: specifying and estimating a regression model, performing simple diagnostic analysis, and using your estimation results in further analysis.

Subsequent chapters discuss testing and forecasting, as well as more advanced and specialized techniques such as weighted least squares, two-stage least squares (TSLS), nonlinear least squares, ARIMA/ARIMAX models, generalized method of moments (GMM), GARCH models, and qualitative and limited dependent variable models. These techniques and models all build upon the basic ideas presented in this chapter.

You will probably find it useful to own an econometrics textbook as a reference for the techniques discussed in this and subsequent documentation. Standard textbooks that we have found to be useful are listed below (in generally increasing order of difficulty):

- Pindyck and Rubinfeld (1991), *Econometric Models and Economic Forecasts*, 3rd edition.
- Johnston and DiNardo (1997), *Econometric Methods*, 4th Edition.
- Greene (1997), *Econometric Analysis*, 3rd Edition.
- Davidson and MacKinnon (1993), *Estimation and Inference in Econometrics*.

Where appropriate, we will also provide you with specialized references for specific topics.

Equation Objects

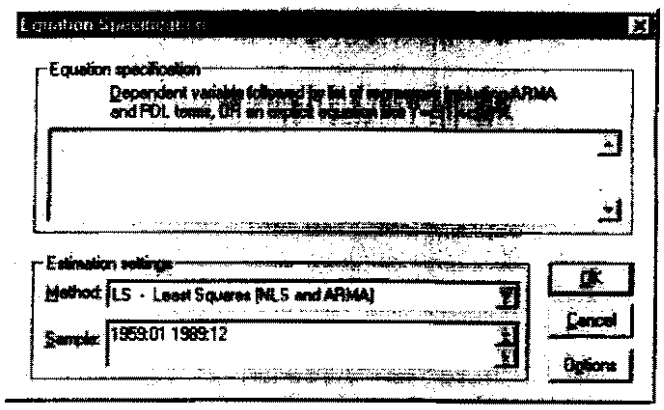
Single equation regression estimation in EViews is performed using the *equation object*. To create an equation object in EViews: select **Objects/New Object/Equation** or **Quick/Estimate Equation...** from the main menu, or simply type the keyword `equation` in the command window.

Next, you will specify your equation in the Equation Specification dialog box that appears, and select an estimation method. Below, we provide details on specifying equations in EViews. EViews will estimate the equation and display results in the equation window.

The estimation results are stored as part of the equation object so they can be accessed at any time. Simply open the object to display the summary results, or to access EViews tools for working with results from an equation object. For example, you can retrieve the sum-of-squares from any equation, or you can use the estimated equation as part of a multi-equation model.

Specifying an Equation in EViews

When you create an equation object, a specification dialog box is displayed.



You need to specify three things in this dialog: the equation specification, the estimation method, and the sample to be used in estimation.

In the upper edit box, you can specify the equation: the dependent (left-hand side) and independent (right-hand side) variables and the functional form. There are two basic ways of specifying an equation: "by list" and "by formula" or "by expression". The list method is easier but may only be used with unrestricted linear specifications; the formula method is more general and must be used to specify nonlinear models or models with parametric restrictions.

Specifying an Equation by List

The simplest way to specify a linear equation is to provide a list of variables that you wish to use in the equation. First, include the name of the dependent variable or expression, followed by a list of explanatory variables. For example, to specify a linear consumption function, CS regressed on a constant and INC, type the following in the upper field of the Equation Specification dialog:

```
cs c inc
```

Note the presence of the series name C in the list of regressors. This is a built-in EViews series that is used to specify a constant in a regression. EViews does not automatically include a constant in a regression so you must explicitly list the constant (or its equivalent) as a regressor. The internal series C does not appear in your workfile, and you may not use it outside of specifying an equation. If you need a series of ones, you can generate a new series, or use the number 1 as an auto-series.

You may have noticed that there is a pre-defined object C in your workfile. This is the *default coefficient vector*—when you specify an equation by listing variable names, EViews stores the estimated coefficients in this vector, in the order of appearance in the list. In the example above, the constant will be stored in C(1) and the coefficient on INC will be held in C(2).

Lagged series may be included in statistical operations using the same notation as in generating a new series with a formula—put the lag in parentheses after the name of the series. For example, the specification:

```
cs cs(-1) c inc
```

tells EViews to regress CS on its own lagged value, a constant, and INC. The coefficient for lagged CS will be placed in C(1), the coefficient for the constant is C(2), and the coefficient of INC is C(3).

You can include a consecutive range of lagged series by using the word "`to`" between the lags. For example,

```
cs c cs(-1 to -4) inc
```

regresses CS on a constant, CS(-1), CS(-2), CS(-3), CS(-4), and INC. If you don't include the first lag, it is taken to be zero. For example,

```
cs c inc(to -2) inc(-4)
```

regresses CS on a constant, INC, INC(-1), INC(-2), and INC(-4).

You may include auto-series in the list of variables. If the auto-series expressions contain spaces, they should be enclosed in parentheses. For example,

```
log(cs) c log(cs(-1)) ((inc+inc(-1)) / 2)
```

specifies a regression of the natural logarithm of CS on a constant, its own lagged value, and a two period moving average of INC.

Typing the list of series may be cumbersome, especially if you are working with many regressors. If you wish, EViews can create the specification list for you. First, highlight the dependent variable in the workfile window by single clicking on the entry. Next, CTRL-click on each of the explanatory variables to highlight them as well. When you are done selecting all of your variables, double click on any of the highlighted series, and select **Open/Equation...** The Equation Specification dialog box should appear with the names entered in the specification field. The constant C is automatically included in this list; you must delete the C if you do not wish to include the constant.

Specifying an Equation by Formula

You will need to specify your equation using a formula when the list method is not general enough for your specification. Many, but not all, estimation methods allow you to specify your equation using a formula.

An equation formula in EViews is a mathematical expression involving regressors and coefficients. To specify an equation using a formula, simply enter the expression in the dia-

log in place of the list of variables. EViews will add an implicit additive disturbance to this equation and will estimate the parameters of the model using least squares.

When you specify an equation by list, EViews converts this into an equivalent equation formula. For example, the list,

```
log(cs) c log(cs(-1)) log(inc)
```

is interpreted by EViews as,

```
log(cs) = c(1) + c(2)*log(cs(-1)) + c(3)*log(inc)
```

Equations do not have to have a dependent variable followed by an equal sign and then an expression. The " = " sign can be anywhere in the formula, as in:

```
log(urate) + c(1)*dmr = c(2)
```

The residuals for this equation are given by:

$$\epsilon = \log(\text{urate}) - c(1)\text{dmr} - c(2) . \tag{11.1}$$

EViews will minimize the sum-of-squares of these residuals.

If you wish, you can specify an equation as a simple expression, without a dependent variable and an equal sign. If there is no equal sign, EViews assumes that the entire expression is the disturbance term. For example, if you specify an equation as

```
c(1)*x + c(2)*y + 4*z
```

EViews will find the coefficient values that minimize the sum of squares of the given expression, in this case (C(1)*X + C(2)*Y + 4*Z). While EViews will estimate an expression of this type, since there is no dependent variable, some regression statistics (*e.g.* R-squared) are not reported and the equation cannot be used for forecasting. This restriction also holds for any equation that includes coefficients to the left of the equal sign. For example, if you specify,

```
x + c(1)*y = c(2)*z
```

EViews finds the values of C(1) and C(2) that minimize the sum of squares of (X + C(1)*Y–C(2)*Z). The estimated coefficients will be identical to those from an equation specified using:

```
x = -c(1)*y + c(2)*z
```

but some regression statistics are not reported.

The two most common motivations for specifying your equation by formula are to estimate restricted and nonlinear models. For example, suppose that you wish to constrain the

coefficients on the lags on the variable X to sum to one. Solving out for the coefficient restriction leads to the following linear model with parameter restrictions:

```
y = c(1) + c(2)*x + c(3)*x(-1) + c(4)*x(-2) +(1-c(2)-c(3)-
    c(4))*x(-3)
```

To estimate a nonlinear model, simply enter the nonlinear formula. EViews will automatically detect the nonlinearity and estimate the model using nonlinear least squares. For details, see "Nonlinear Least Squares" on page 289.

One benefit to specifying an equation by formula is that you can elect to use a different coefficient vector. To create a new coefficient vector, choose **Objects/New Object...** and select **Matrix-Vector-Coef** from the main menu, type in a name for the coefficient vector, and click **OK**. In the **New Matrix** dialog box that appears, select **Coefficient Vector** and specify how many rows there should be in the vector. The object will be listed in the workfile directory with the coefficient vector icon (the little α).

You may then use this coefficient vector in your specification. For example, suppose you created coefficient vectors A and BETA, each with a single row. Then you can specify your equation using the new coefficients in place of C:

```
log(cs) = a(1) + beta(1)*log(cs(-1))
```

Estimating an Equation in EViews

Estimation Methods

Having specified your equation, you now need to choose an estimation method. Click on the **Method:** entry in the dialog and you will see a drop-down menu listing estimation methods.

Standard, single-equation regression is performed using least squares. The other methods are described in subsequent chapters.

Equations estimated by ordinary least squares and two-stage least squares, GMM, and ARCH can be specified with a formula. Nonlinear equations are not allowed with binary, ordered, censored, and count models, or in equations with ARMA terms.

Estimation Sample

You should also specify the sample to be used in estimation. EViews will fill out the dialog with the current workfile sample, but you can change the sample for purposes of estimation by entering your sample string or object in the edit box (see "Samples" on page 60 for details). Changing the estimation sample does not affect the current workfile sample.

If any of the series used in estimation contain missing data, EViews will temporarily adjust the estimation sample of observations to exclude those observations (listwise exclusion). EViews notifies you that it has adjusted the sample by reporting the actual sample used in the estimation results:

```
Dependent Variable: Y
Method: Least Squares
Date: 08/19/97   Time: 10:24
Sample(adjusted): 1959:01 1989:12
Included observations: 340
Excluded observations: 32 after adjusting endpoints
```

Here we see the top of an equation output view. EViews reports that it has adjusted the sample. Out of the 372 observations in the period 1959:01–1989:12, EViews uses the 340 observations with observations for all of the relevant variables.

You should be aware that if you include lagged variables in a regression, the degree of sample adjustment will differ depending on whether data for the pre-sample period are available or not. For example, suppose you have nonmissing data for the two series M1 and IP over the period 1959:01–1989:12 and specify the regression as

```
m1 c ip ip(-1) ip(-2) ip(-3)
```

If you set the estimation sample to the period 1959:01–1989:12, EViews adjusts the sample to:

```
Dependent Variable: M1
Method: Least Squares
Date: 08/19/97   Time: 10:49
Sample: 1960:01 1989:12
Included observations: 360
```

since data for IP(-3) are not available until 1959:04. However, if you set the estimation sample to the period 1960:01–1989:12, EViews will not make any adjustment to the sample since all values of IP(-3) are available during the estimation sample.

Some operations, most notably estimation with MA terms and ARCH, do not allow missing observations in the middle of the sample. When executing these procedures, an error message is displayed and execution is halted if an NA is encountered in the middle of the sample. EViews handles missing data at the very start or the very end of the sample range by adjusting the sample endpoints and proceeding with the estimation procedure.

Estimation Options

EViews provides a number of estimation options. These options allow you to weight the estimating equation, to compute heteroskedasticity and auto-correlation robust covari-

ances, and to control various features of your estimation algorithm. These options are discussed in detail in "Estimation Options" on page 292.

Equation Output

When you click OK in the Equation Specification dialog, EViews displays the equation window displaying the estimation output view:

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 08/18/97 Time: 14:02
Sample: 1959:01 1989:12
Included observations: 372

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1.699912	0.164954	-10.30539	0.0000
LOG(IP)	1.765866	0.043546	40.55199	0.0000
TB3	-0.011895	0.004628	-2.570016	0.0106

R-squared	0.886416	Mean dependent var	5.863717
Adjusted R-squared	0.885800	S.D. dependent var	0.563903
S.E. of regression	0.187183	Akaike info criterion	-0.505429
Sum squared resid	12.92882	Schwarz criterion	-0.473825
Log likelihood	97.00980	F-statistic	1439.848
Durbin-Watson stat	0.008687	Prob(F-statistic)	0.000000

Using matrix notation, the standard regression may be written as:

$$y = X\beta + \epsilon \tag{11.2}$$

where y is a T-dimensional vector containing observations on the dependent variable, X is a $T \times k$ matrix of independent variables, β is a k-vector of coefficients, and ϵ is a T-vector of disturbances. T is the number of observations and k is the number of right-hand side regressors.

In the output above, y is log(M1), X consists of three variables C, log(IP), and TB3, where $T = 372$ and $k = 3$.

Coefficient Results

Regression Coefficients

The column labeled "Coefficient" depicts the estimated coefficients. The least squares regression coefficients b are computed by the standard OLS formula

$$b = (X'X)^{-1}X'y \tag{11.3}$$

If your equation is specified by list, the coefficients will be labeled in the "Variable" column with the name of the corresponding regressor; if your equation is specified by formula, EViews lists the actual coefficients, C(1), C(2), etc.

For the simple linear models considered here, the coefficient measures the marginal contribution of the independent variable to the dependent variable, holding all other variables fixed. If present, the coefficient of the C is the constant or intercept in the regression—it is the base level of the prediction when all of the other independent variables are zero. The other coefficients are interpreted as the slope of the relation between the corresponding independent variable and the dependent variable, assuming all other variables do not change.

Standard Errors

The "Std. Error" column reports the estimated standard errors of the coefficient estimates. The standard errors measure the statistical reliability of the coefficient estimates—the larger the standard errors, the more statistical noise in the estimates. If the errors are normally distributed, there are about 2 chances in 3 that the true regression coefficient lies within one standard error of the reported coefficient, and 95 chances out of 100 that it lies within two standard errors.

The covariance matrix of the estimated coefficients is computed as,

$$\mathrm{var}(b) = s^2(X'X)^{-1}; \quad s^2 = \hat{e}'\hat{e}/(T-k); \quad \hat{e} = y - Xb \tag{11.4}$$

where $\hat{e}$ is the residual. The standard errors of the estimated coefficients are the square roots of the diagonal elements of the coefficient covariance matrix. You can view the whole covariance matrix by choosing **View/Covariance Matrix**.

t-Statistics

The *t*-statistic, which is computed as the ratio of an estimated coefficient to its standard error, is used to test the hypothesis that a coefficient is equal to zero. To interpret the *t*-statistic, you should examine the probability of observing the *t*-statistic given that the coefficient is equal to zero. This probability computation is described below.

In cases where normality can only hold asymptotically, EViews will report a *z*-statistic instead of a *t*-statistic.

Probability

The last column of the output shows the probability of drawing a *t*-statistic (or a *z*-statistic) as extreme as the one actually observed, under the assumption that the errors are normally distributed, or that the estimated coefficients are asymptotically normally distributed.

This probability is also known as the *p-value* or the *marginal significance level*. Given a *p*-value, you can tell at a glance if you reject or accept the hypothesis that the true coefficient is zero against a two-sided alternative that it differs from zero. For example, if you are performing the test at the 5% significance level, a *p*-value lower than 0.05 is taken as evidence

to reject the null hypothesis of a zero coefficient. If you want to conduct a one-sided test, the appropriate probability is one-half that reported by EViews.

For the above example output, the hypothesis that the coefficient on TB3 is zero is rejected at the 5% significance level but not at the 1% level. However, if theory suggests that the coefficient on TB3 cannot be positive, then a one-sided test will reject the zero null hypothesis at the 1% level.

The p-values are computed from a t-distribution with $T-k$ degrees of freedom.

Summary Statistics

R-squared

The R-squared (R^2) statistic measures the success of the regression in predicting the values of the dependent variable within the sample. In standard settings, R^2 may be interpreted as the fraction of the variance of the dependent variable explained by the independent variables. The statistic will equal one if the regression fits perfectly, and zero if it fits no better than the simple mean of the dependent variable. It can be negative for a number of reasons. For example, if the regression does not have an intercept or constant, if the regression contains coefficient restrictions, or if the estimation method is two-stage least squares or ARCH.

EViews computes the (centered) R^2 as

$$R^2 = 1 - \frac{\hat{\varepsilon}'\hat{\varepsilon}}{(y-\bar{y})'(y-\bar{y})}; \quad \bar{y} = \sum_{t=1}^{T} y_t/T \tag{11.5}$$

where $\bar{y}$ is the mean of the dependent (left-hand) variable.

Adjusted R-squared

One problem with using R^2 as a measure of goodness of fit is that the R^2 will never decrease as you add more regressors. In the extreme case, you can always obtain an R^2 of one if you include as many independent regressors as there are sample observations.

The adjusted R^2, commonly denoted as $\bar{R}^2$, penalizes the R^2 for the addition of regressors which do not contribute to the explanatory power of the model. The adjusted R^2 is computed as

$$\bar{R}^2 = 1 - (1 - R^2)\frac{T-1}{T-k} \tag{11.6}$$

The $\bar{R}^2$ is never larger than the R^2, can decrease as you add regressors, and for poorly fitting models, may be negative.

Standard Error of the Regression (S.E. of regression)

The standard error of the regression is a summary measure based on the estimated variance of the residuals. The standard error of the regression is computed as

$$s = \sqrt{\frac{\hat{\varepsilon}'\hat{\varepsilon}}{(T-k)}} \tag{11.7}$$

Sum-of-Squared Residuals

The sum-of-squared residuals can be used in a variety of statistical calculations, and is presented separately for your convenience:

$$\hat{\varepsilon}'\hat{\varepsilon} = \sum_{t=1}^{T} (y_i - X_i'b)^2 \tag{11.8}$$

Log Likelihood

EViews reports the value of the log likelihood function (assuming normally distributed errors) evaluated at the estimated values of the coefficients. Likelihood ratio tests may be conducted by looking at the difference between the log likelihood values of the restricted and unrestricted versions of an equation.

The log likelihood is computed as

$$l = -\frac{T}{2}(1 + \log(2\pi) + \log(\hat{\varepsilon}'\hat{\varepsilon}/T)) \tag{11.9}$$

When comparing EViews output to that reported from other sources, note that EViews does not ignore constant terms.

Durbin-Watson Statistic

The Durbin-Watson statistic measures the serial correlation in the residuals. The statistic is computed as

$$\mathrm{DW} = \frac{\sum_{t=2}^{T} (\hat{\varepsilon}_t - \hat{\varepsilon}_{t-1})^2}{\sum_{t=1}^{T} \hat{\varepsilon}_t^2} \tag{11.10}$$

See Johnston and DiNardo (1997, Table D.5) for a table of the significance points of the distribution of the Durbin-Watson statistic.

As a rule of thumb, if the DW is less than 2, there is evidence of positive serial correlation. The DW statistic in our output is very close to one, indicating the presence of serial correlation in the residuals. See "Serial Correlation Theory" beginning on page 303 for a more

extensive discussion of the Durbin-Watson statistic and the consequences of serially correlated residuals.

There are better tests for serial correlation. In "Testing for Serial Correlation" on page 304, we discuss the Q-statistic, and the Breusch-Godfrey LM test, both of which provide a more general testing framework than the Durbin-Watson test.

Mean and Standard Deviation (S.D.) of the Dependent Variable

The mean and standard deviation of y are computed using the standard formulae:

$$\bar{y} = \sum_{t=1}^{T} y_t/T; \quad s_y = \sqrt{\sum_{t=1}^{T} (y_t - \bar{y})^2/(T-1)} \tag{11.11}$$

Akaike Information Criterion

The Akaike Information Criterion (AIC) is computed as:

$$\text{AIC} = -2l/T + 2k/T \tag{11.12}$$

where l is the log likelihood (given by Equation (11.9) on page 268).

The AIC is often used in model selection for non-nested alternatives—smaller values of the AIC are preferred. For example, you can choose the length of a lag distribution by choosing the specification with the lowest value of the AIC. See Appendix F, "Information Criteria", on page 683, for additional discussion.

Schwarz Criterion

The Schwarz Criterion (SC) is an alternative to the AIC that imposes a larger penalty for additional coefficients:

$$\text{SC} = -2l/T + (k \log T)/T \tag{11.13}$$

F-Statistic

The F-statistic reported in the regression output is from a test of the hypothesis that *all* of the slope coefficients (excluding the constant, or intercept) in a regression are zero. For ordinary least squares models, the F-statistic is computed as

$$F = \frac{R^2/(k-1)}{(1-R^2)/(T-k)} \tag{11.14}$$

Under the null hypothesis with normally distributed errors, this statistic has an F-distribution with $k-1$ numerator degrees of freedom and $T-k$ denominator degrees of freedom.

The p-value given just below the F-statistic, denoted **Prob(F-statistic)**, is the marginal significance level of the F-est. If the p-value is less than the significance level you are testing, say 0.05, you reject the null hypothesis that all slope coefficients are equal to zero. For the example above, the p-value is essentially zero, so we reject the null hypothesis that all of the regression coefficients are zero. Note that the F-est is a joint test so that even if all the t-statistics are insignificant, the F-statistic can be highly significant.

Working With Equation Statistics

The regression statistics reported in the estimation output view are stored with the equation and are accessible through special "@-functions". You can retrieve any of these statistics for further analysis by using these functions in genr, scalar, or matrix expressions. If a particular statistic is not computed for a given estimation method, the function will return an NA.

There are two kinds of "@-functions": those that return a scalar value, and those that return matrices or vectors.

Keywords that return scalar values

@aic	Akaike information criterion
@coefcov(i,j)	covariance of coefficient estimates i and j
@coefs(i)	i-th coefficient value
@dw	Durbin-Watson statistic
@f	F-statistic
@hq	Hannan-Quinn information criterion
@jstat	J-statistic — value of the GMM objective function (for GMM)
@logl	value of the log likelihood function
@meandep	mean of the dependent variable
@ncoef	number of estimated coefficients
@r2	R-squared statistic
@rbar2	adjusted R-squared statistic
@regobs	number of observations in regression
@schwarz	Schwarz information criterion
@sddep	standard deviation of the dependent variable
@se	standard error of the regression
@ssr	sum of squared residuals

@stderrs(i)	standard error for coefficient *i*
@tstats(i)	*t*-statistic value for coefficient *i*
c(i)	*i*-th element of default coefficient vector for equation (if applicable)

Keywords that return vector or matrix objects

@coefcov	matrix containing the coefficient covariance matrix
@coefs	vector of coefficient values
@stderrs	vector of standard errors for the coefficients
@tstats	vector of *t*-statistic values for coefficients

If you use these functions without reference to an equation object, EViews will use the default equation. For example, the command:

```
series y = @dw
```

creates a series named Y and assigns to every observation, the value of the Durbin-Watson statistic for the default equation.

We strongly recommend, however, that you prepend the name of an equation object and a "." to the statistic keyword. This instructs EViews to use the appropriate statistic for the named equation. For example:

```
series y = eq1.@dw
```

assigns to Y the value of the Durbin-Watson for the equation EQ1.

Functions that return a vector or matrix object should be assigned to the corresponding object type. For example, you should assign the results from `@tstats` to a vector:

```
vector tstats = eq1.@tstats
```

and the covariance matrix to a matrix:

```
matrix mycov = eq1.@cov
```

You can also access individual elements of these statistics:

```
scalar pvalue = 1-@cnorm(@abs(eq1.@tstats(4)))
scalar var1 = eq1.@covariance(1,1)
```

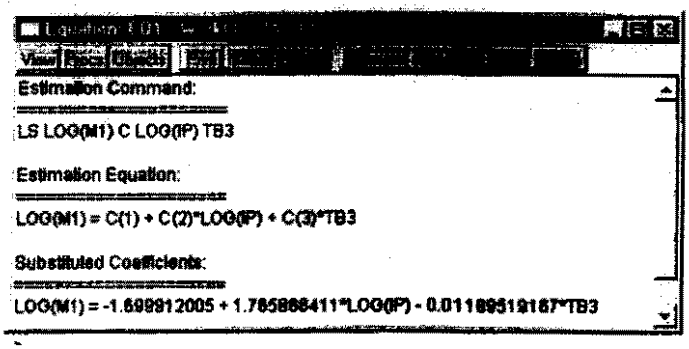
For documentation on using vectors and matrices in EViews, see Chapter 4, "Matrix Language", on page 55 of the *Command and Programming Reference*.

Working with Equations

Views of an Equation

- **Representations**. Displays the equation in three forms: EViews command form, as an algebraic equation with symbolic coefficients, and as an equation with the estimated values of the coefficients.

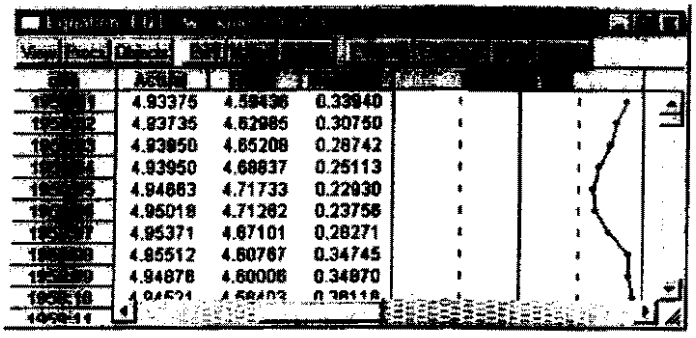
  You can cut-and-paste from the representations view into any application that supports the Windows clipboard.



- **Estimation Output**. Displays the equation output results described above.

- **Actual, Fitted, Residual**. These views display the actual and fitted values of the dependent variable and the residuals from the regression in tabular and graphical form. **Actual, Fitted, Residual Table** displays these values in table form.

  Note that the actual value is always the sum of the fitted value and the residual. **Actual, Fitted, Residual Graph** displays a standard EViews graph of the actual values, fitted values, and residuals. **Residual Graph** plots only the residuals, while the **Standardized Residual Graph** plots the residuals divided by the estimated residual standard deviation.



- **Gradients and Derivatives...** Provides views which describe the gradients of the objective function and the information about the computation of any derivatives of the regression function. Details on these views are provided in Appendix E, "Gradients and Derivatives", on page 675.

- **Covariance Matrix**. Displays the covariance matrix of the coefficient estimates as a spreadsheet view. To save this covariance matrix as a matrix object, use the `@cov` function.

- **Coefficient Tests, Residual Tests, and Stability Tests.** These are views for specification and diagnostic tests and are described in detail in Chapter 15, "Specification and Diagnostic Tests", beginning on page 367.

Procedures of an Equation

- **Specify/Estimate...** brings up the Equation Specification dialog box so that you can modify your specification. You can edit the equation specification, or change the estimation method or estimation sample.
- **Forecast...** forecasts or fits values using the estimated equation. Forecasting using equations is discussed in Chapter 14.
- **Make Residual Series...** saves the residuals from the regression as a series in the workfile. Depending on the estimation method, you may choose from three types of residuals: ordinary, standardized, and generalized. For ordinary least squares, only the ordinary residuals may be saved.
- **Make Regressor Group** creates an untitled group comprised of all the variables used in the equation (with the exception of the constant).
- **Make Gradient Group** creates a group containing the gradients of the objective function with respect to the coefficients of the model.
- **Make Derivative Group** creates a group containing the derivatives of the regression function with respect to the coefficients in the regression function.
- **Make Model** creates an untitled model containing a link to the estimated equation. This model can be solved in the usual manner. See Chapter 23, "Models", on page 601 for information on how to use models for forecasting and simulations.
- **Update Coefs from Equation** places the estimated coefficients of the equation in the coefficient vector. You can use this procedure to initialize starting values for various estimation procedures.

Default Equation

Following estimation, EViews often holds the estimated coefficients and their covariance matrix, the residuals, and some summary statistics in an untitled equation object. These results are available for use in a variety of subsequent computations including the specification and diagnostic tests described in Chapter 15, "Specification and Diagnostic Tests", beginning on page 367, and the computation of forecasts and model simulation in Chapter 14, "Forecasting from an Equation", on page 343 and Chapter 23, "Models", on page 601.

Untitled equations are not saved with the workfile. You may use the Name button on the equation toolbar to name your equation. The equation will be saved with the workfile

when the latter is saved. Once named, you can access the information in the equation at any time, even if you have just estimated several other models, or have not worked with the workfile for a long period of time.

For your convenience, EViews keeps track of a *default equation*. The default equation is the equation that is active or was the most recently active equation. The name of the default equation is shown at the upper right hand corner of the workfile window.

Residuals from an Equation

The residuals from the default equation are stored in a series object called RESID. RESID may be used directly as if it were a regular series, except in estimation.

RESID will be overwritten whenever you estimate an equation and will contain the residuals from the latest estimated equation. To save the residuals from a particular equation for later analysis, you should save them in a different series so they are not overwritten by the next estimation command. For example, you can copy the residuals into a regular EViews series called RES1 by the command

```
series res1 = resid
```

Even if you have already overwritten the RESID series, you can always create the desired series using EViews' built-in procedures if you still have the equation object. If your equation is named EQ1, open the equation window and select **Procs/Make Residual Series**, or enter

```
eq1.makeresid res1
```

to create the desired series.

Regression Statistics

You may refer to various regression statistics through the @-functions described above. For example, to generate a new series equal to FIT plus twice the standard error from the last regression, you can use the command

```
series plus = fit + 2*eq1.@se
```

To get the *t*-statistic for the second coefficient from equation EQ1, you could specify

```
eq1.@tstats(2)
```

To store the coefficient covariance matrix from EQ1 as a named symmetric matrix, you can use the command

```
sym ccov1 = eq1.@cov
```

See "Keywords that return scalar values" on page 270 for additional details.

Storing and Retrieving an Equation

As with other objects, equations may be stored to disk in data bank or database files. You can also fetch equations from these files.

Equations may also be copied-and-pasted to, or from, workfiles or databases.

EViews even allows you to access equations directly from your databases or another workfile. You can estimate an equation, store it in a database, and then use it to forecast in several workfiles.

See Chapter 3, "EViews Basics", beginning on page 33 and Chapter 6, "EViews Databases", beginning on page 107 for additional information about objects, databases, and object containers.

Using Estimated Coefficients

The coefficients of an equation are listed in the representations view. By default, EViews will use the C coefficient vector when you specify an equation, but you may explicitly use other coefficient vectors in defining your equation.

These stored coefficients may be used as scalars in generating data. While there are easier ways of generating fitted values (see "Forecasting from an Equation" on page 343), for purposes of illustration, note that we can use the coefficients to form the fitted values from an equation. The command:

```
series cshat = eq1.c(1) + eq1.c(2)*gdp
```

forms the fitted value of CS, CSHAT, from the OLS regression coefficients and the independent variables from the equation object EQ1.

Note that while EViews will accept a series generating equation which does not explicitly refer to a named equation:

```
series cshat = c(1) + c(2)*gdp
```

and will use the existing values in the C coefficient vector, we strongly recommend that you always use named equations to identify the appropriate coefficients. In general, C will contain the correct coefficient values only immediately following estimation or a coefficient update. Using a named equation, or selecting **Procs/Update coefs from equation**, guarantees that you are using the correct coefficient values.

An alternative to referring to the coefficient vector is to reference the `@coefs` elements of your equation (see page 270). For example, the examples above may be written as

```
series cshat=eq1.@coefs(1)+eq1.@coefs(2)*gdp
```

EViews assigns an index to each coefficient in the order that it appears in the representations view. Thus, if you estimate the equation

```
equation eq01.ls y=c(10)+b(5)*y(-1)+a(7)*inc
```

where B and A are also coefficient vectors, then

- eq01.@coefs(1) contains C(10)
- eq01.@coefs(2) contains B(5)
- eq01.@coefs(3) contains A(7)

This method should prove useful in matching coefficients to standard errors derived from the `@stderrs` elements of the equation (see Chapter 3, "Object, View and Procedure Reference", beginning on page 19 of the *Command and Programming Reference*). The `@coefs` elements allow you to refer to both the coefficients and the standard errors using a common index.

If you have used an alternative named coefficient vector in specifying your equation, you can also access the coefficient vector directly. For example, if you have used a coefficient vector named BETA, you can generate the fitted values by issuing the commands

```
equation eq02.ls cs=beta(1)+beta(2)*gdp
series cshat=beta(1)+beta(2)*gdp
```

where BETA is a coefficient vector. Again, however, we recommend that you use the `@coefs` elements to refer to the coefficients of EQ02. Alternatively, you can update the coefficients in BETA prior to use by selecting **Procs/Update coefs from equation** from the equation window. Note that EViews does not allow you to refer to the named equation coefficients EQ02.BETA(1) and EQ02.BETA(2). You must instead use the expressions, EQ02.@COEFS(1) and EQ02.@COEFS(2).

Estimation Problems

Exact Collinearity

If the regressors are very highly collinear, EViews may encounter difficulty in computing the regression estimates. In such cases, EViews will issue an error message "Near singular matrix." When you get this error message, you should check to see whether the regressors are *exactly* collinear. The regressors are exactly collinear if one regressor can be written as a linear combination of the other regressors. Under exact collinearity, the regressor matrix X does not have full column rank and the OLS estimator cannot be computed.

You should watch out for exact collinearity when you are using dummy variables in your regression. A set of mutually exclusive dummy variables and the constant term are exactly collinear. For example, suppose you have quarterly data and you try to run a regression with the specification

```
y c x @seas(1) @seas(2) @seas(3) @seas(4)
```

EViews will return a "Near singular matrix" error message since the constant and the four quarterly dummy variables are exactly collinear through the relation:

```
c = @seas(1) + @seas(2) + @seas(3) + @seas(4)
```

In this case, simply drop either the constant term or one of the dummy variables.

The textbooks listed above provide extensive discussion of the issue of collinearity.

Commands

To declare a new equation object, follow the equation command with a name for the equation object:

```
equation eq1
```

To estimate an equation by OLS, follow the equation name with a dot and the keyword "ls" or "est", the name of the dependent variable, and the names of the independent variables, each separated by a space:

```
eq1.ls cs c gdp cpi
```

regresses CS on a constant, GDP, and CPI.

Alternatively, you can specify the equation by a formula with an equal sign:

```
eq1.ls cs = c(1) + c(2)*gdp + c(3)*cpi
```

You can define and estimate an equation in one command:

```
equation eq_sale.ls sales c trend orders industry_growth
```

estimates the specified equation and stores the results in an equation named EQ_SALE.

See `ls` (p. 245) in the *Command and Programming Reference* for a complete list of commands and options for single equation least squares estimation in EViews.

Chapter 12. Additional Regression Methods

This chapter discusses weighted least squares, heteroskedasticity and autocorrelation consistent covariance estimation, two-stage least squares (TSLS), nonlinear least squares, and generalized method of moments (GMM). Note that most of these methods are also available in systems of equations; see Chapter 19.

Parts of this chapter refer to estimation of models which have autoregressive (AR) and moving average (MA) error terms. These concepts are discussed in greater depth in Chapter 13.

Weighted Least Squares

Suppose that you have heteroskedasticity of known form, and that there is a series w, whose values are proportional to the *reciprocals* of the error standard deviations. You can use weighted least squares, with weight series w, to correct for the heteroskedasticity.

EViews performs weighted least squares by first dividing the weight series by its mean, then multiplying all of the data for each observation by the scaled weight series. The scaling of the weight series is a normalization that has no effect on the parameter results, but makes the weighted residuals more comparable to the unweighted residuals. The normalization does imply, however, that EViews weighted least squares is not appropriate in situations where the scale of the weight series is relevant, as in frequency weighting.

Estimation is then completed by running a regression using the weighted dependent and independent variables to minimize the sum-of-squared residuals

$$S(\beta) = \sum_t w_t^2 (y_t - x_t'\beta)^2 \tag{12.1}$$

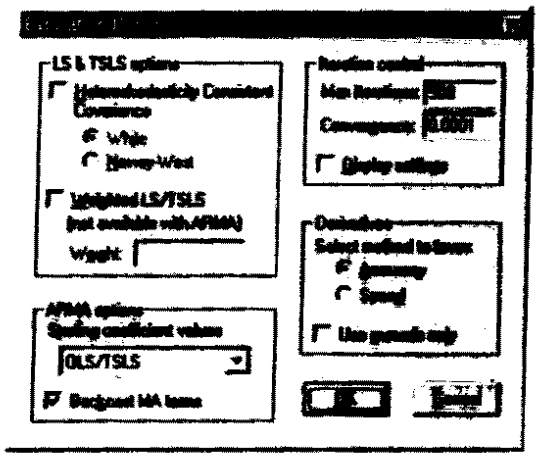
with respect to the k-dimensional vector of parameters β. In matrix notation, let W be a diagonal matrix containing the scaled w along the diagonal and zeroes elsewhere, and let y and X be the usual matrices associated with the left and right-hand side variables. The weighted least squares estimator is

$$b_{WLS} = (X'W'WX)^{-1} X'W'Wy, \tag{12.2}$$

and the estimated covariance matrix is

$$\hat{\Sigma}_{WLS} = s^2 (X'W'WX)^{-1}. \tag{12.3}$$

To estimate an equation using weighted least squares, first go to the main menu and select **Quick/Estimate Equation...**, then choose **LS—Least Squares (NLS and ARMA)** from the combo box. Enter your equation specification and sample in the edit boxes, then push the **Options** button and click on the **Weighted LS/TSLS** option.



Fill in the blank after **Weight:** with the name of the series containing your weights, and click on **OK**. Click on **OK** again to accept the dialog and estimate the equation.

Dependent Variable: LOG(X)
Method: Least Squares
Date: 10/15/97 Time: 11:10
Sample(adjusted): 1891 1983
Included observations: 93 after adjusting endpoints
Weighting series: POP

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.004233	0.012745	0.332082	0.7406
LOG(X(-1))	0.099840	0.112539	0.887163	0.3774
LOG(W(-1))	0.194219	0.421005	0.461322	0.6457

Weighted Statistics			
R-squared	0.016252	Mean dependent var	0.009762
Adjusted R-squared	-0.005609	S.D. dependent var	0.108487
S.E. of regression	0.108785	Akaike info criterion	-1.604274
Sum squared resid	1.026272	Schwarz criterion	-1.522577
Log likelihood	77.59873	F-statistic	0.743433
Durbin-Watson stat	1.948087	Prob(F-statistic)	0.478378

Unweighted Statistics			
R-squared	-0.002922	Mean dependent var	0.011083
Adjusted R-squared	-0.025209	S.D. dependent var	0.121357
S.E. of regression	0.122877	Sum squared resid	1.368883
Durbin-Watson stat	2.086969		

EViews will open an output window displaying the standard coefficient results, and both weighted and unweighted summary statistics. The weighted summary statistics are based on the fitted residuals, computed using the weighted data,

$$\hat{u}_t = w_t(y_t - x_t'b_{WLS}). \tag{12.4}$$

The unweighted summary results are based on the residuals computed from the original (unweighted) data,

$$u_t = y_t - x_t{}' b_{WLS}. \tag{12.5}$$

Following estimation, the unweighted residuals are placed in the RESID series.

If the residual variance assumptions are correct, the weighted residuals should show no evidence of heteroskedasticity. If the variance assumptions are correct, the unweighted residuals should be heteroskedastic, with the reciprocal of the standard deviation of the residual at each period t being proportional to w_t.

The weighting option will be ignored in equations containing ARMA specifications. Note also that the weighting option is not available for binary, count, censored and truncated, or ordered discrete choice models.

Heteroskedasticity and Autocorrelation Consistent Covariances

When the form of heteroskedasticity is not known, it may not be possible to obtain efficient estimates of the parameters using weighted least squares. OLS provides consistent parameter estimates in the presence of heteroskedasticity but the usual OLS standard errors will be incorrect and should not be used for inference.

Before we describe the techniques for HAC covariance estimation, note that:

- Using the White heteroskedasticity consistent or the Newey-West HAC consistent covariance estimates does not change the point estimates of the parameters, only the estimated standard errors.
- There is nothing to keep you from combining various methods of accounting for heteroskedasticity and serial correlation. For example, weighted least squares estimation might be accompanied by White or Newey-West covariance matrix estimates.

Heteroskedasticity Consistent Covariances (White)

White (1980) has derived a heteroskedasticity consistent covariance matrix estimator which provides correct estimates of the coefficient covariances in the presence of heteroskedasticity of unknown form. The White covariance matrix is given by:

$$\hat{\Sigma}_W = \frac{T}{T-k}(X'X)^{-1}\left(\sum_{t=1}^{T} u_t^2 x_t x_t{}'\right)(X'X)^{-1}, \tag{12.6}$$

where is T the number of observations, k is the number of regressors, and u_t is the least squares residual.

EViews provides you the option to use the White covariance estimator in place of the standard OLS formula. Open the equation dialog and specify the equation as before, then push the **Options** button. Next, click on the check box labeled **Heteroskedasticity Consistent**

Covariance and click on the **White** radio button. Accept the options and click **OK** to estimate the equation.

EViews will estimate your equation and compute the variances using White's covariance estimator. You can always tell when EViews is using White covariances, since the output display will include a line to document this fact:

Dependent Variable: LOG(C)
Method: Least Squares
Date: 10/15/97 Time: 11:11
Sample(adjusted): 1891 1983
Included observations: 93 after adjusting endpoints
Weighting series: POP
White Heteroskedasticity-Consistent Standard Errors & Covariance

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.004233	0.012519	0.338088	0.7361
LOG(X(-1))	0.099840	0.137262	0.727369	0.4689
LOG(W(-1))	0.194219	0.436544	0.444800	0.6575

HAC Consistent Covariances (Newey-West)

The White covariance matrix described above assumes that the residuals of the estimated equation are serially uncorrelated. Newey and West (1987) have proposed a more general covariance estimator that is consistent in the presence of both heteroskedasticity and autocorrelation of unknown form. The Newey-West estimator is given by

$$\hat{\Sigma}_{NW} = \frac{T}{T-k}(X'X)^{-1}\hat{\Omega}(X'X)^{-1}, \tag{12.7}$$

where

$$\hat{\Omega} = \frac{T}{T-k}\left\{\sum_{t=1}^{T} u_t^2 x_t x_t' + \sum_{v=1}^{q}\left(\left(1-\frac{v}{q+1}\right)\sum_{t=v+1}^{T}(x_t u_t u_{t-v} x_{t-v}' + x_{t-v} u_{t-v} u_t x_t')\right)\right\} \tag{12.8}$$

and q, the truncation lag, is a parameter representing the number of autocorrelations used in evaluating the dynamics of the OLS residuals u_t. Following the suggestion of Newey and West, EViews sets q to

$$q = \text{floor}(4(T/100)^{2/9}). \tag{12.9}$$

To use the Newey-West method, push the **Options** button in the estimation dialog box. Check the box labeled **Heteroskedasticity Consistent Covariance** and press the **Newey-West** radio button.

Two-stage Least Squares

A fundamental assumption of regression analysis is that the right-hand side variables are uncorrelated with the disturbance term. If this assumption is violated, both OLS and weighted LS are biased and inconsistent.

There are a number of situations where some of the right-hand side variables are correlated with disturbances. Some classic examples occur when:

- There are endogenously determined variables on the right-hand side of the equation.
- Right-hand side variables are measured with error.

For simplicity, we will refer to variables that are correlated with the residuals as *endogenous*, and variables that are not correlated with the residuals as *exogenous* or *predetermined*.

The standard approach in cases where right-hand side variables are correlated with the residuals is to estimate the equation using *instrumental variables* regression. The idea behind instrumental variables is to find a set of variables, termed *instruments*, that are both (1) correlated with the explanatory variables in the equation, and (2) uncorrelated with the disturbances. These instruments are used to eliminate the correlation between right-hand side variables and the disturbances.

Two-stage least squares (TSLS) is a special case of instrumental variables regression. As the name suggests, there are two distinct stages in two-stage least squares. In the first stage, TSLS finds the portions of the endogenous and exogenous variables that can be attributed to the instruments. This stage involves estimating an OLS regression of each variable in the model on the set of instruments. The second stage is a regression of the original equation, with all of the variables replaced by the fitted values from the first-stage regressions. The coefficients of this regression are the TSLS estimates.

You need not worry about the separate stages of TSLS since EViews will estimate both stages simultaneously using instrumental variables techniques. More formally, let Z be the matrix of instruments, and let y and X be the dependent and explanatory variables. Then the coefficients computed in two-stage least squares are given by,

$$b_{TSLS} = (X'Z(Z'Z)^{-1}Z'X)^{-1}X'Z(Z'Z)^{-1}Z'y, \tag{12.10}$$

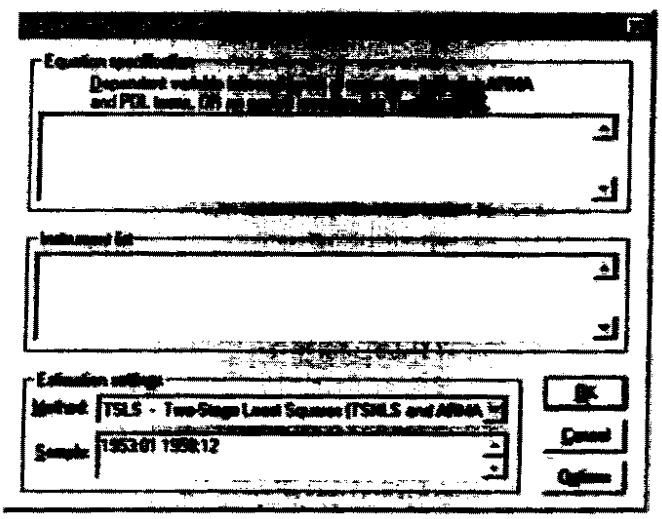
and the estimated covariance matrix of these coefficients is given by

$$\hat{\Sigma}_{TSLS} = s^2(X'Z(Z'Z)^{-1}Z'X)^{-1}, \tag{12.11}$$

where s^2 is the estimated residual variance (square of the standard error of the regression).

Estimating TSLS in EViews

To use two-stage least squares, open the equation specification box by choosing **Object/New Object/Equation...** or **Quick/Estimate Equation...**. Choose **TSLS** from the **Method:** combo box and the dialog will change to include an edit window where you will list the instruments. In the edit boxes, specify your dependent variable and independent variables and the list of instruments.



There are a few things to keep in mind as you enter your instruments:

- In order to calculate TSLS estimates, your specification must satisfy the *order condition* for identification, which says that there must be at least as many instruments as there are coefficients in your equation. There is an additional rank condition which must also be satisfied. See Davidson and MacKinnon (1994) and Johnston and DiNardo (1997) for additional discussion.
- For econometric reasons that we will not pursue here, any right-hand side variables that are not correlated with the disturbances can be used as instruments.
- The constant, C, is always a suitable instrument, so EViews will add it to the instrument list if you omit it.

For example, suppose you are interested in estimating a consumption equation relating consumption (CONS) to gross domestic product (GDP), lagged consumption (CONS(-1)), a trend variable (TIME) and a constant (C). GDP is endogenous and therefore correlated with the residuals. You may, however, believe that government expenditures (G), the log of the money supply (LM), lagged consumption, TIME, and C, are exogenous and uncorrelated with the disturbances, so that these variables may be used as instruments. Your equation specification is then,

```
cons c gdp cons(-1) time
```

and the instrument list is,

```
c gov cons(-1) time lm
```

This specification satisfies the order condition for identification, which requires that there are at least as many instruments (five) as there are coefficients (four) in the equation specification.

Furthermore, all of the variables in the consumption equation that are believed to be uncorrelated with the disturbances, (CONS(-1), TIME, and C), appear both in the equation specification and in the instrument list. Note that listing C as an instrument is redundant, since EViews automatically adds it to the instrument list.

Output from TSLS

Below, we present TSLS estimates from a regression of LOG(CS) on a constant and LOG(GDP), with the instrument list C LOG(CS(-1)) LOG(GDP(-1)):

Dependent Variable: LOG(CS)
Method: Two-Stage Least Squares
Date: 10/15/97 Time: 11:32
Sample(adjusted): 1947:2 1995:1
Included observations: 192 after adjusting endpoints
Instrument list: C LOG(CS(-1)) LOG(GDP(-1))

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1.209268	0.039151	-30.88699	0.0000
LOG(GDP)	1.094339	0.004924	222.2597	0.0000

R-squared	0.996168	Mean dependent var	7.480286
Adjusted R-squared	0.996148	S.D. dependent var	0.452990
S.E. of regression	0.028735	Sum squared resid	0.156888
F-statistic	49399.36	Durbin-Watson stat	0.102639
Prob(F-statistic)	0.000000		

EViews identifies the estimation procedure, as well as the list of instruments in the header. This information is followed by the usual coefficient, t-statistics, and asymptotic p-values.

The summary statistics reported at the bottom of the table are computed using the formulas outlined in Chapter 11. Bear in mind that all reported statistics are only asymptotically valid. For a discussion of the finite sample properties of TSLS, see Johnston and DiNardo (1997, pp. 355–358) or Davidson and MacKinnon (1984, pp. 221–224).

EViews uses the *structural residuals* $u_t = y_t - x_t{}'b_{TSLS}$ in calculating all of the summary statistics. For example, the standard error of the regression used in the asymptotic covariance calculation is computed as

$$s^2 = \sum_t u_t^2/(T-k)\,. \tag{12.12}$$

These structural residuals should be distinguished from the *second stage residuals* that you would obtain from the second stage regression if you actually computed the two-stage least squares estimates in two separate stages. The second stage residuals are given by

$\hat{u}_t = \hat{y}_t - \hat{x}_t' b_{TSLS}$, where the $\hat{y}_t$ and $\hat{x}_t$ are the fitted values from the first-stage regressions.

We caution you that some of the reported statistics should be interpreted with care. For example, since different equation specifications will have different instrument lists, the reported R^2 for TSLS can be negative even when there is a constant in the equation.

Weighted TSLS

You can combine TSLS with weighted regression. Simply enter your TSLS specification as above, then press the **Options** button, select the **Weighted LS/TSLS** option and enter the weighting series.

Weighted two-stage least squares is performed by multiplying all of the data, including the instruments, by the weight variable, and estimating TSLS on the transformed model. Equivalently, EViews then estimates the coefficients using the formula,

$$b_{WTSLS} = (X'W'WZ(Z'W'WZ)^{-1}Z'W'WX)^{-1} \cdot X'W'WZ(Z'W'WZ)^{-1}Z'W'Wy \tag{12.13}$$

The estimated covariance matrix is

$$\hat{\Sigma}_{WTSLS} = s^2(X'W'WZ(Z'W'WZ)^{-1}Z'W'WX)^{-1}. \tag{12.14}$$

TSLS with AR errors

You can adjust your TSLS estimates to account for serial correlation by adding AR terms to your equation specification. EViews will automatically transform the model to a nonlinear least squares problem, and estimate the model using instrumental variables. Details of this procedure may be found in Fair (1984, pp. 210–214). The output from TSLS with an AR(1) specification looks as follows:

Dependent Variable: LOG(CS)
Method: Two-Stage Least Squares
Date: 10/15/97 Time: 11:42
Sample(adjusted): 1947:2 1995:1
Included observations: 192 after adjusting endpoints
Convergence achieved after 4 iterations
Instrument list: C LOG(CS(-1)) LOG(GDP(-1))

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1.420705	0.203266	-6.989390	0.0000
LOG(GDP)	1.119858	0.025116	44.58782	0.0000
AR(1)	0.930900	0.022267	41.80595	0.0000

R-squared	0.999611	Mean dependent var	7.480286
Adjusted R-squared	0.999607	S.D. dependent var	0.462990
S.E. of regression	0.009175	Sum squared resid	0.015909
F-statistic	243139.7	Durbin-Watson stat	1.931027
Prob(F-statistic)	0.000000		
Inverted AR Roots	.93		

The **Options** button in the estimation box may be used to change the iteration limit and convergence criterion for the nonlinear instrumental variables procedure.

First-order AR errors

Suppose your specification is:

$$\begin{aligned} y_t &= x_t'\beta + w_t\gamma + u_t \\ u_t &= \rho_1 u_{t-1} + \epsilon_t \end{aligned} \tag{12.15}$$

where x_t is a vector of endogenous variables, and w_t is a vector of predetermined variables, which, in this context, may include lags of the dependent variable. z_t is a vector of instrumental variables not in w_t that is large enough to identify the parameters of the model.

In this setting, there are important technical issues to be raised in connection with the choice of instruments. In a widely quoted result, Fair (1970) shows that if the model is estimated using an iterative Cochrane-Orcutt procedure, all of the lagged left- and right-hand side variables $(y_{t-1}, x_{t-1}, w_{t-1})$ must be included in the instrument list to obtain consistent estimates. In this case, then the instrument list should include

$$(w_t, z_t, y_{t-1}, x_{t-1}, w_{t-1}) . \tag{12.16}$$

Despite the fact the EViews estimates the model as a nonlinear regression model, the first stage instruments in TSLS are formed as if running Cochrane-Orcutt. Thus, if you choose to omit the lagged left- and right-hand side terms from the instrument list, EViews will automatically add each of the lagged terms as instruments. This fact is noted in your output.

Higher Order AR errors

The AR(1) result extends naturally to specifications involving higher order serial correlation. For example, if you include a single AR(4) term in your model, the natural instrument list will be

$$(w_t, z_t, y_{t-4}, x_{t-4}, w_{t-4}) \tag{12.17}$$

If you include AR terms from 1 through 4, one possible instrument list is

$$(w_t, z_t, y_{t-1}, \ldots, y_{t-4}, x_{t-1}, \ldots, x_{t-4}, w_{t-1}, \ldots, w_{t-4}) \tag{12.18}$$

Note that while theoretically valid, this instrument list has a large number of overidentifying instruments, which may lead to computational difficulties and large finite sample biases (Fair (1984, p. 214), Davidson and MacKinnon (1993, pp. 222-224)). In theory, adding instruments should always improve your estimates, but as a practical matter this may not be so in small samples.

Examples

Suppose that you wish to estimate the consumption function by two-stage least squares, allowing for first-order serial correlation. You may then use two-stage least squares with the variable list,

```
cons c gdp ar(1)
```

and instrument list,

```
c gov log(m1) time cons(-1) gdp(-1)
```

Notice that the lags of both the dependent and endogenous variables (CONS(-1) and GDP(-1)), are included in the instrument list.

Similarly, consider the consumption function,

```
cons c cons(-1) gdp ar(1)
```

A valid instrument list is given by

```
c gov log(m1) time cons(-1) cons(-2) gdp(-1)
```

Here we treat the lagged left and right-hand side variables from the original specification as predetermined and add the lagged values to the instrument list.

Lastly, consider the specification,

```
cons c gdp ar(1) ar(2) ar(3) ar(4)
```

Adding all of the relevant instruments in the list, we have

```
c gov log(m1) time cons(-1) cons(-2) cons(-3) cons(-4) gdp(-1)
   gdp(-2) gdp(-3) gdp(-4)
```

TSLS with MA errors

You can also estimate two-stage least squares variable problems with MA error terms of various orders. To account for the presence of MA errors, simply add the appropriate terms to your specification prior to estimation.

Illustration

Suppose that you wish to estimate the consumption function by two-stage least squares, accounting for first-order moving average errors. You may then use two-stage least squares with the variable list,

```
cons c gdp ma(1)
```

and instrument list,

```
c gov log(m1) time
```

EViews will add both first and second lags of CONS and GDP to the instrument list.

Technical Details

Most of the technical details are identical to those outlined above for AR errors. EViews transforms the model that is nonlinear in parameters (employing backcasting, if appropriate) and then estimates the model using nonlinear instrumental variables techniques.

Note that EViews augments the instrument list appropriately by adding lagged left- and right-hand side variables. There is an approximately involved here, however, in a truncation of the lag structure. In principle, each MA term involves an infinite number of AR terms. Clearly it is impossible to add an infinite number of lags to the instrument list. Instead, EViews performs an ad hoc approximation by adding a truncated set of instruments involving the MA order and an additional lag. If for example, you have an MA(5), EViews will add lagged instruments corresponding to lags 5 and 6.

Nonlinear Least Squares

Suppose that we have the regression specification

$$y_t = f(x_t, \beta) + \epsilon_t, \tag{12.19}$$

where f is a general function of the explanatory variables x_t and the parameters β. Least squares estimation chooses the parameter values that minimize the sum of squared residuals:

$$S(\beta) = \sum_t (y_t - f(x_t, \beta))^2 = (y - f(X, \beta))'(y - f(X, \beta)) \tag{12.20}$$

We say that a model is *linear in parameters* if the derivatives of f with respect to the parameters do not depend upon β; if the derivatives are functions of β, we say that the model is *nonlinear in parameters*.

For example, consider the model given by

$$y_t = \beta_1 + \beta_2 \log L_t + \beta_3 \log K_t + \epsilon_t . \tag{12.21}$$

it is easy to see that this model is linear in its parameters, implying that it can be estimated using ordinary least squares.

In contrast, the equation specification

$$y_t = \beta_1 L_t^{\beta_2} K_t^{\beta_3} + \epsilon_t \tag{12.22}$$

has derivatives that depend upon the elements of β. There is no way to rearrange the terms in this model so that ordinary least squares can be used to minimize the sum-of-squared residuals. We must use nonlinear least squares techniques to estimate the parameters of the model.

Nonlinear least squares minimizes the sum-of-squared residuals with respect to the choice of parameters β. While there is no closed form solution for the parameter estimates, the estimates satisfy the first-order conditions:

$$(G(\beta))'(y - f(X, \beta)) = 0 , \tag{12.23}$$

where $G(\beta)$ is the matrix of first derivatives of $f(X, \beta)$ with respect to β (to simplify notation we suppress the dependence of G upon X). The estimated covariance matrix is given by

$$\hat{\Sigma}_{NLLS} = s^2 (G(b_{NLLS})' G(b_{NLLS}))^{-1} . \tag{12.24}$$

where b_{NLLS} are the estimated parameters. For additional discussion of nonlinear estimation, see Pindyck and Rubinfeld (1991, pp. 231-245) or Davidson and MacKinnon (1993).

Estimating NLS Models in EViews

It is easy to tell EViews that you wish to estimate the parameters of a model using nonlinear least squares. EViews automatically applies nonlinear least squares to any regression equation that is nonlinear in its coefficients. Simply select **Object/New Object/Equation**, enter the equation in the equation specification dialog box and click **OK**. EViews will do all of the work of estimating your model using an iterative algorithm.

A full technical discussion of iterative estimation procedures is provided in Appendix D, "Estimation Algorithms and Options", beginning on page 663.

Specifying Nonlinear Least Squares

For nonlinear regression models, you will have to enter your specification in equation form using EViews expressions that contain direct references to coefficients. You may use elements of the default coefficient vector C (e.g. C(1), C(2), C(34), C(87)), or you can define and use other coefficient vectors. For example,

```
y = c(1) + c(2)*(k^c(3)+l^c(4))
```

is a nonlinear specification that uses the first through the fourth elements of the default coefficient vector, C.

To create a new coefficient vector, select **Objects/New Object/Matrix-Vector-Coef/Coefficient Vector** in the main menu and provide a name. You may now use this coefficient vector in your specification. For example, if you create a coefficient vector named CF, you can rewrite the specification above as

```
y = cf(11) + cf(12)*(k^cf(13)+l^cf(14))
```

which uses the eleventh through the fourteenth elements of CF.

You can also use multiple coefficient vectors in your specification:

```
y = c(11) + c(12)*(k^cf(1)+l^cf(2))
```

which uses both C and CF in the specification.

It is worth noting that EViews implicitly adds an additive disturbance to your specification. For example, the input

```
y = (c(1)*x + c(2)*z + 4)^2
```

is interpreted as $y_t = (c(1)x_t + c(2)z_t + 4)^2 + \epsilon_t$, and EViews will minimize

$$S(c(1), c(2)) = \sum_t (y_t - (c(1)x_t + c(2)z_t + 4)^2)^2. \tag{12.25}$$

If you wish, the equation specification may be given by a simple expression that does not include a dependent variable. For example, the input

```
(c(1)*x + c(2)*z + 4)^2
```

is interpreted by EViews as $-(c(1)x_t + c(2)z_t + 4)^2 = \epsilon_t$, and EViews will minimize

$$S(c(1), c(2)) = \sum_t (-(c(1)x_t + c(2)z_t + 4)^2)^2. \tag{12.26}$$

While EViews will estimate the parameters of this last specification, the equation cannot be used for forecasting and cannot be included in a model. This restriction also holds for

any equation that includes coefficients to the left of the equal sign. For example, if you specify

```
x + c(1)*y = z^c(2)
```

EViews will find the values of C(1) and C(2) that minimize the sum of squares of the implicit equation

$$x_t + c(1)y_t - z_t^{c(2)} = \epsilon_t, \tag{12.27}$$

but the estimated equation cannot be used in forecasting or included in a model, since there is no dependent variable.

Estimation Options

Starting Values. Iterative estimation procedures require starting values for the coefficients of the model. There are no general rules for selecting starting values for parameters. The closer to the true values the better, so if you have reasonable guesses for parameter values, these can be useful. In some cases, you can obtain good starting values by estimating a restricted version of the model using least squares. In general, however, you will have to experiment in order to find starting values.

EViews uses the values in the coefficient vector at the time you begin the estimation procedure as starting values for the iterative procedure. It is easy to examine and change these coefficient starting values.

To see the starting values, double click on the coefficient vector in the workfile directory. If the values appear to be reasonable, you can close the window and proceed with estimating your model.

If you wish to change the starting values, first make certain that the spreadsheet view of your coefficients is in edit mode, then enter the coefficient values. When you are finished setting the initial values, close the coefficient vector window and estimate your model.

You may also set starting coefficient values from the command window using the PARAM command. Simply enter the PARAM keyword, following by each coefficient and desired value:

```
param c(1) 153 c(2) .68 c(3) .15
```

sets C(1) = 153, C(2) = .68, and C(3) = .15.

See Appendix D, "Estimation Algorithms and Options" on page 663, for further details.

Derivative Methods. Estimation in EViews requires computation of the derivatives of the regression function with respect to the parameters. EViews provides you with the option of computing analytic expressions for these derivatives (if possible), or computing finite difference numeric derivatives in cases where the derivative is not constant. Furthermore, if

numeric derivatives are computed, you can choose whether to favor speed of computation (fewer function evaluations) or whether to favor accuracy (more function evaluations). Additional issues associated with ARIMA models are discussed in "Estimation Options" on page 318.

Iteration and Convergence Options. You can control the iterative process by specifying convergence criterion and the maximum number of iterations. Press the **Options** button in the equation dialog box and enter the desired values.

EViews will report that the estimation procedure has converged if the convergence test value is below your convergence tolerance. See "Iteration and Convergence Options" on page 669 for details.

In most cases, you will not need to change the maximum number of iterations. However, for some difficult to estimate models, the iterative procedure will not converge within the maximum number of iterations. If your model does not converge within the allotted number of iterations, simply click on the **Estimate** button, and, if desired, increase the maximum number of iterations. Click on **OK** to accept the options, and click on **OK** to begin estimation. EViews will start estimation using the last set of parameter values as starting values.

These options may also be set from the global options dialog. See Appendix A, "Estimation Defaults" on page 649.

Output from NLS

Once your model has been estimated, EViews displays an equation output screen showing the results of the nonlinear least squares procedure. Below is the output from a regression of LOG(CS) on C, and the Box-Cox transform of GDP:

Dependent Variable: LOG(CS)
Method: Least Squares
Date: 10/15/97 Time: 11:51
Sample(adjusted): 1947:1 1995:1
Included observations: 193 after adjusting endpoints
Convergence achieved after 80 iterations
LOG(CS)= C(1)+C(2)*(GDP^C(3)-1)/C(3)

	Coefficient	Std. Error	t-Statistic	Prob.
C(1)	2.851780	0.279033	10.22024	0.0000
C(2)	0.257592	0.041147	6.260254	0.0000
C(3)	0.182959	0.020201	9.056824	0.0000

R-squared	0.997252	Mean dependent var	7.476058
Adjusted R-squared	0.997223	S.D. dependent var	0.465503
S.E. of regression	0.024532	Akaike info criterion	-4.562220
Sum squared resid	0.114350	Schwarz criterion	-4.511505
Log likelihood	443.2542	F-statistic	34469.84
Durbin-Watson stat	0.134628	Prob(F-statistic)	0.000000

If the estimation procedure has converged, EViews will report this fact, along with the number of iterations that were required. If the iterative procedure did not converge, EViews will report "Convergence not achieved after" followed by the number of iterations attempted.

Below the line describing convergence, EViews will repeat the nonlinear specification so that you can easily interpret the estimated coefficients of your model.

EViews provides you with all of the usual summary statistics for regression models. Provided that your model has converged, the standard statistical results and tests are *asymptotically* valid.

Weighted NLS

Weights can be used in nonlinear estimation in a manner analogous to weighted linear least squares. To estimate an equation using weighted nonlinear least squares, enter your specification, press the **Options** button and click on the **Weighted LS/TSLS** option. Fill in the blank after **Weight:** with the name of the weight series and then estimate the equation.

EViews minimizes the sum of the weighted squared residuals:

$$S(\beta) = \sum_t w_t^2 (y_t - f(x_t, \beta))^2 = (y - f(X, \beta))' W' W (y - f(X, \beta)) \quad (12.28)$$

with respect to the parameters β, where w_t are the values of the weight series and W is the matrix of weights. The first-order conditions are given by

$$(G(\beta))' W' W (y - f(X, \beta)) = 0 \quad (12.29)$$

and the covariance estimate is computed as

$$\hat{\Sigma}_{WNLLS} = s^2 (G(b_{WNLLS})' W' W G(b_{WNLLS}))^{-1}. \quad (12.30)$$

NLS with AR errors

EViews will estimate nonlinear regression models with autoregressive error terms. Simply select **Objects/New Object/Equation...** or **Quick/Estimate Equation...** and specify your model using EViews expressions, followed by an additive term describing the AR correction enclosed in square brackets. The AR term should consist of a coefficient assignment for each AR term, separated by commas. For example, if you wish to estimate

$$\begin{aligned} CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= c_3 u_{t-1} + c_4 u_{t-2} + \epsilon_t \end{aligned} \quad (12.31)$$

you should enter the specification

```
cs = c(1) + gdp^c(2) + [ar(1)=c(3), ar(2)=c(4)]
```

See "How EViews Estimates AR Models" on page 310 for additional details. EViews does not currently estimate nonlinear models with MA errors, nor does it estimate weighted models with AR terms—if you add AR terms to a weighted nonlinear model, the weighting series will be ignored.

Nonlinear TSLS

Nonlinear two-stage least squares refers to an instrumental variables procedure for estimating nonlinear regression models involving functions of endogenous and exogenous variables and parameters. Suppose we have the usual nonlinear regression model:

$$y_t = f(x_t, \beta) + \epsilon_t, \tag{12.32}$$

where β is a k-dimensional vector of parameters, and x_t contains both exogenous and endogenous variables. In matrix form, if we have $m \geq k$ instruments z_t, nonlinear two-stage least squares minimizes

$$S(\beta) = (y - f(X, \beta))'Z(Z'Z)^{-1}Z'(y - f(X, \beta)) \tag{12.33}$$

with respect to the choice of β.

While there is no closed form solution for the parameter estimates, the parameter estimates satisfy the first-order conditions:

$$G(\beta)'Z(Z'Z)^{-1}Z'(y - f(X, \beta)) = 0 \tag{12.34}$$

with estimated covariance given by

$$\hat{\Sigma}_{TSNLLS} = s^2(G(b_{TSNLLS})'Z(Z'Z)^{-1}Z'G(b_{TSNLLS}))^{-1}. \tag{12.35}$$

How to Estimate Nonlinear TSLS in EViews

EViews performs the estimation procedure in a single step so that you don't have to perform the separate stages yourself. Simply select **Object/New Object/Equation...** or **Quick/Estimate Equation...** Choose **TSLS** from the **Method:** combo box, enter your nonlinear specification and the list of instruments. Click **OK**.

With nonlinear two-stage least squares estimation, you have a great deal of flexibility with your choice of instruments. Intuitively you want instruments that are correlated with $G(\beta)$. Since G is nonlinear, you may begin to think about using more than just the exogenous and predetermined variables as instruments. Various nonlinear functions of these variables, for example, cross-products and powers, may also be valid instruments. One should be aware, however, of the possible finite sample biases resulting from using too many instruments.

Weighted Nonlinear Two-stage Least Squares

Weights can be used in nonlinear two-stage least squares estimation. Simply add weighting to your nonlinear TSLS specification above by pressing the **Options** button, selecting **Weighted LS/TSLS** option, and entering the name of the weight series.

The objective function for weighted TSLS is,

$$S(\beta) = (y - f(X, \beta))'W'WZ(Z'W'WZ)^{-1}Z'W'W(y - f(X, \beta)). \quad (12.36)$$

The reported standard errors are based on the covariance matrix estimate given by

$$\hat{\Sigma}_{WTSNLLS} = s^2(G(b)'W'WZ(Z'W'WZ)^{-1}Z'W'WG(b))^{-1} \quad (12.37)$$

where $b = b_{WTSNLLS}$. Note that if you add AR or MA terms to a weighted specification, the weighting series will be ignored.

Nonlinear Two-stage Least Squares with AR errors

While we will not go into much detail here, note that EViews can estimate non-linear TSLS models where there are autoregressive error terms. EViews does not currently estimate nonlinear models with MA errors.

To estimate your model, simply open your equation specification window, and enter your nonlinear specification, including all AR terms, and provide your instrument list. For example, you could enter the regression specification

```
cs = exp(c(1) + gdp^c(2)) + [ar(1)=c(3)]
```

with the instrument list

```
c gov
```

EViews will transform the nonlinear regression model as described in "Estimating AR Models" on page 307, and then estimate nonlinear TSLS on the transformed specification using the instruments C and GOV. For nonlinear models with AR errors, EViews uses a Gauss-Newton algorithm. See "Optimization Algorithms" on page 663 for further details.

Solving Estimation Problems

EViews may not be able to estimate your nonlinear equation on the first attempt. Sometimes, the nonlinear least squares procedure will stop immediately. Other times, EViews may stop estimation after several iterations without achieving convergence. EViews might even report that it cannot improve the sums-of-squares. While there are no specific rules on how to proceed if you encounter these estimation problems, there are a few general areas you might want to examine.

Starting Values

If you experience problems with the very first iteration of a nonlinear procedure, the problem is almost certainly related to starting values. See the discussion above for how to examine and change your starting values.

Model Identification

If EViews goes through a number of iterations and then reports that it encounters a "Near Singular Matrix", you should check to make certain that your model is identified. Models are said to be non-identified if there are multiple sets of coefficients which identically yield the minimized sum-of-squares value. If this condition holds, it is impossible to choose between the coefficients on the basis of the minimum sum-of-squares criterion.

For example, the nonlinear specification,

$$y_t = \beta_1\beta_2 + \beta_2^2 x_t + \epsilon_t \tag{12.38}$$

is not identified, since any coefficient pair (β_1, β_2) is indistinguishable from the pair $(-\beta_1, -\beta_2)$ in terms of the sum-of-squared residuals.

For a thorough discussion of identification of nonlinear least squares models, see Davidson and MacKinnon (1993, Sections 2.3, 5.2 and 6.3).

Convergence Criterion

EViews may report that it is unable to improve the sums-of-squares. This result may be evidence of non-identification or model misspecification. Alternatively, it may be the result of setting your convergence criterion too low, which can occur if your nonlinear specification is particularly complex.

If you wish to change the convergence criterion, enter the new value in the Options dialog. Be aware that increasing this value increases the possibility that you will stop at a local minimum, and may hide misspecification or non-identification of your model.

See "Setting Estimation Options" on page 666, for further details.

Generalized Method of Moments (GMM)

The starting point of GMM estimation is a theoretical relation that the parameters should satisfy. The idea is to choose the parameter estimates so that the theoretical relation is satisfied as "closely" as possible. The theoretical relation is replaced by its sample counterpart and the estimates are chosen to minimize the weighted distance between the theoretical and actual values. GMM is a robust estimator in that, unlike maximum likelihood estimation, it does not require information of the exact distribution of the disturbances. In fact, many common estimators in econometrics can be considered as special cases of GMM.

The theoretical relation that the parameters should satisfy are usually *orthogonality conditions* between some (possibly nonlinear) function of the parameters $f(\theta)$ and a set of instrumental variables z_t:

$$E(f(\theta)'Z) = 0, \tag{12.39}$$

where θ are the parameters to be estimated. The GMM estimator selects parameter estimates so that the sample correlations between the instruments and the function f are as close to zero as possible, as defined by the criterion function:

$$J(\theta) = (m(\theta))'Am(\theta), \tag{12.40}$$

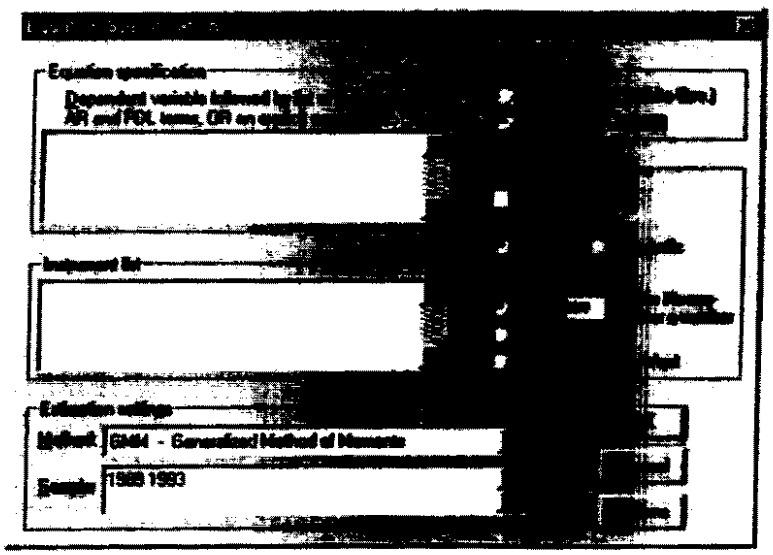
where $m(\theta) = f(\theta)'Z$ and A is a weighting matrix. Any symmetric positive definite matrix A will yield a consistent estimate of q. However, it can be shown that a necessary (but not sufficient) condition to obtain an (asymptotically) efficient estimate of q is to set A equal to the inverse of the covariance matrix of the sample moments m.

Many standard estimators, including all of the system estimators provided in EViews, can be set up as special cases of GMM. For example, the ordinary least squares estimator can be viewed as a GMM estimator, based upon the conditions that each of the right-hand variables is uncorrelated with the residual.

Estimation by GMM in EViews

To estimate an equation by GMM, either create a new equation object by selecting **Object/New Object/Equation**, or press the **Estimate** button in the toolbar of an existing equation. From the Equation Specification dialog choose **Estimation Method: GMM**. The estimation specification dialog will change as depicted below.

To obtain GMM estimates in EViews, you need to write the moment condition as an orthogonality condition between an expression including the parameters and a set of instrumental variables. There are two ways you can write the orthogonality condition: with and without a dependent variable.



If you specify the equation either by listing vari-

able names or by an expression with an equal sign, EViews will interpret the moment condition as an orthogonality condition between the instruments and the residuals defined by the equation. If you specify the equation by an expression without an equal sign, EViews will orthogonalize that expression to the set of instruments.

You must also list the names of the instruments in the **Instrument List** field box of the Equation Specification dialog box. For the GMM estimator to be identified, there must be at least as many instrumental variables as there are parameters to estimate. EViews will always include the constant in the list of instruments.

For example, if you type

Equation Specification: `y c x`

Instrument list: `c z w`

the orthogonality conditions given by

$$\begin{aligned} \sum(y_t - c(1) - c(2)x_t) &= 0 \\ \sum(y_t - c(1) - c(2)x_t)z_t &= 0 \\ \sum(y_t - c(1) - c(2)x_t)w_t &= 0 \end{aligned} \tag{12.41}$$

If you enter an expression

Equation Specification: `c(1)*log(y)+x^c(2)`

Instrument list: `c z z(-1)`

the orthogonality conditions are

$$\begin{aligned} \sum(c(1)\log y_t + x_t^{c(2)}) &= 0 \\ \sum(c(1)\log y_t + x_t^{c(2)})z_t &= 0 \\ \sum(c(1)\log y_t + x_t^{c(2)})z_{t-1} &= 0 \end{aligned} \tag{12.42}$$

On the right part of the Equation Specification dialog are the options for selecting the weighting matrix A in the objective function. If you select **Weighting Matrix: Cross section (White Cov)**, the GMM estimates will be robust to heteroskedasticity of unknown form.

If you select **Weighting Matrix: Time series (HAC)**, the GMM estimates will be robust to heteroskedasticity and autocorrelation of unknown form. For the HAC option, you have to specify the kernel type and bandwidth.

- The **Kernel Options** determine the functional form of the kernel used to weight the autocovariances in computing the weighting matrix.

- The **Bandwidth Selection** option determines how the weights given by the kernel change with the lags of the autocovariances in the computation of the weighting matrix. If you select **Fixed** bandwidth, you may either enter a number for the bandwidth or type nw to use Newey and West's fixed bandwidth selection criterion.
- The **Prewhitening** option runs a preliminary VAR(1) prior to estimation to "soak up" the correlation in the moment conditions.

The technical notes in "Generalized Method of Moments (GMM)" on page 515 describe these options in more detail.

Example

Tauchen (1986) considers the problem of estimating the taste parameters β, γ from the Euler equation

$$(\beta R_{t+1} w_{t+1}^{-\gamma} - 1)' z_t = 0 \tag{12.43}$$

where we use instruments $z_t = (1, w_t, w_{t-1}, r_t, r_{t-1})'$. To estimate the parameters β, γ by GMM, fill in the Equation Specification dialog as

Equation Specification: `c(1)*r(+1)*w(+1)^(-c(2))-1`

Instrument list: `c w w(-1) r r(-1)`

The estimation result using the default HAC **Weighting Matrix** option looks as follows:

Dependent Variable: Implicit Equation
Method: Generalized Method of Moments
Date: 08/26/97 Time: 14:02
Sample(adjusted): 1891 1982
Included observations: 92 after adjusting endpoints
No prewhitening
Bandwidth: Fixed (3)
Kernel: Bartlett
Convergence achieved after: 7 weight matricies, 7 total coef iterations
C(1)*R(+1)*W(+1)^(-C(2))-1
Instrument list: C W W(-1) R R(-1)

	Coefficient	Std. Error	t-Statistic	Prob.
C(1)	0.934086	0.018751	49.81000	0.0000
C(2)	1.366396	0.741802	1.841995	0.0688
S.E. of regression	0.154084	Sum squared resid		2.136780
Durbin-Watson stat	1.903837	J-statistic		0.054523

Note that when you specify an equation without a dependent variable, EViews does not report some of the regression statistics such as the R-squared. The *J*-statistic reported at the bottom of the table is the minimized value of the objective function. The *J*-statistic can be used to carry out hypothesis tests from GMM estimation; see Newey and West (1987a). A simple application of the *J*-statistic is to test the validity of overidentifying restrictions when you have more instruments than parameters to estimate. In this example, we have

five instruments to estimate two parameters and so there are three overidentifying restrictions. Under the null hypothesis that the overidentifying restrictions are satisfied, the J-statistic times the number of regression observations is asymptotically χ^2 with degrees of freedom equal to the number of overidentifying restrictions. You can compute the test statistic as a named scalar in EViews using the commands

```
scalar overid=eq_gmm.@regobs*eq_gmm.@jstat
scalar overid_p=1-@cchisq(overid,3)
```

where EQ_GMM is the name of the equation containing the GMM estimates. The second command computes the p-value of the test statistic as a named scalar OVERID_P. To view the value of OVERID_P, double click on its name; the value will be displayed in the status line at the bottom of the EViews window.

Commands

To estimate an equation by weighted least squares, specify the weighting series in parentheses with the `w=` option after the `ls` command:

```
eq1.ls(w=1/pop) cs c gdp cpi
```

To estimate an equation by two-stage least squares, follow the `tsls` command with the dependent variable, the independent variables, an @ sign, and a list of instruments:

```
equation eq2.tsls cs c gdp @ c cs(-1) gdp(-1)
```

To estimate an equation by GMM, follow the `gmm` command with the dependent variable, the independent variables, an @ sign, and a list of instruments that define the orthogonality conditions:

```
equation eq3.gmm cs c gdp @ c cs(-1) gdp(-1)
```

You can set the starting values prior to nonlinear estimation by the command

```
param c(1) 1.5 c(2) 1
```

or

```
c(1) = 1.5
c(2) = 1
```

To declare a coefficient vector, specify the number of rows in parentheses and provide a name:

```
coef(4) beta
```

declares a four element coefficient vector named BETA filled with zeros.

See "Equation" on page 21 of the *Command and Programming Reference* for a complete list of commands and options for single equation estimation in EViews.

Chapter 13. Time Series Regression

In this section we discuss single equation regression techniques that are important for the analysis of time series data: testing for serial correlation, estimation of ARMA models, using polynomial distributed lags, and testing for unit roots in potentially nonstationary time series.

The chapter focuses on the specification and estimation of time series models. A number of related topics are discussed elsewhere: standard multiple regression techniques are discussed in Chapters 11 and 12, forecasting and inference are discussed extensively in Chapters 14 and 15, vector autoregressions are discussed in Chapter 20, and state space models and the Kalman filter are discussed in Chapter 22.

Serial Correlation Theory

A common finding in time series regressions is that the residuals are correlated with their own lagged values. This serial correlation violates the standard assumption of regression theory that disturbances are not correlated with other disturbances. The primary problems associated with serial correlation are:

- OLS is no longer efficient among linear estimators. Furthermore, since prior residuals help to predict current residuals, we can take advantage of this information to form a better prediction of the dependent variable.
- Standard errors computed using the textbook OLS formula are not correct, and are generally understated.
- If there are lagged dependent variables on the right-hand side, OLS estimates are biased and inconsistent.

EViews provides tools for detecting serial correlation and estimation methods that take account of its presence.

In general, we will be concerned with specifications of the form:

$$\begin{aligned} y_t &= x_t'\beta + u_t \\ u_t &= z_{t-1}'\gamma + \epsilon_t \end{aligned} \tag{13.1}$$

where x_t is a vector of explanatory variables observed at time t, z_{t-1} is a vector of variables known in the previous period, β and γ are vectors of parameters, u_t is a disturbance term, and ϵ_t is the innovation in the disturbance. The vector z_{t-1} may contain lagged values of u, lagged values of ϵ, or both.

The disturbance u_t is termed the *unconditional residual*. It is the residual based on the structural component ($x_t\beta$) but not using the information contained in z_{t-1}. The innovation ϵ_t is also known as the *one-period ahead forecast error* or the *prediction error*. It is the difference between the actual value of the dependent variable and a forecast made on the basis of the independent variables and the past forecast errors.

The First-Order Autoregressive Model

The simplest and most widely used model of serial correlation is the first-order autoregressive, or AR(1), model. The AR(1) model is specified as

$$\begin{aligned} y_t &= x_t'\beta + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{13.2}$$

The parameter ρ is the first-order serial correlation coefficient. In effect, the AR(1) model incorporates the residual from the past observation into the regression model for the current observation.

Higher-Order Autoregressive Models

More generally, a regression with an autoregressive process of order p, AR(p) error is given by

$$\begin{aligned} y_t &= x_t'\beta + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_p u_{t-p} + \epsilon_t \end{aligned} \tag{13.3}$$

The autocorrelations of a stationary AR(p) process gradually die out to zero, while the partial autocorrelations for lags larger than p are zero.

Testing for Serial Correlation

Before you use an estimated equation for statistical inference (*e.g.* hypothesis tests and forecasting), you should generally examine the residuals for evidence of serial correlation. EViews provides several methods of testing a specification for the presence of serial correlation.

The Durbin-Watson Statistic

EViews reports the Durbin-Watson (DW) statistic as a part of the standard regression output. The Durbin-Watson statistic is a test for first-order serial correlation. More formally, the DW statistic measures the linear association between adjacent residuals from a regression model. The Durbin-Watson is a test of the hypothesis $\rho = 0$ in the specification:

$$u_t = \rho u_{t-1} + \epsilon_t. \tag{13.4}$$

If there is no serial correlation, the DW statistic will be around 2. The DW statistic will fall below 2 if there is positive serial correlation (in the worst case, it will be near zero). If there is negative correlation, the statistic will lie somewhere between 2 and 4.

Positive serial correlation is the most commonly observed form of dependence. As a rule of thumb, with 50 or more observations and only a few independent variables, a DW statistic below about 1.5 is a strong indication of positive first order serial correlation. See Johnston and DiNardo (1997, Chapter 6.6.1) for a thorough discussion on the Durbin-Watson test and a table of the significance points of the statistic.

There are three main limitations of the DW test as a test for serial correlation. First, the distribution of the DW statistic under the null hypothesis depends on the data matrix x. The usual approach to handling this problem is to place bounds on the critical region, creating a region where the test results are inconclusive. Second, if there are lagged dependent variables on the right-hand side of the regression, the DW test is no longer valid. Lastly, you may only test the null hypothesis of no serial correlation against the alternative hypothesis of first-order serial correlation.

Two other tests of serial correlation—the Q-statistic and the Breusch-Godfrey LM test—overcome these limitations, and are preferred in most applications.

Correlograms and Q-statistics

If you select **View/Residual Tests/Correlogram-Q-statistics** on the equation toolbar, EViews will display the autocorrelation and partial autocorrelation functions of the residuals, together with the Ljung-Box Q-statistics for high-order serial correlation. If there is no serial correlation in the residuals, the autocorrelations and partial autocorrelations at all lags should be nearly zero, and all Q-statistics should be insignificant with large p-values.

Note that the p-values of the Q-statistics will be computed with the degrees of freedom adjusted for the inclusion of ARMA terms in your regression. There is evidence that some care should be taken in interpreting the results of a Ljung-Box test applied to the residuals from an ARMAX specification (see Dezhbaksh, 1990, for simulation evidence on the finite sample performance of the test in this setting).

Details on the computation of correlograms and Q-statistics are provided in greater detail in Chapter 7, "Series", on page 169.

Serial Correlation LM Test

Selecting **View/Residual Tests/Serial Correlation LM Test...** carries out the Breusch-Godfrey Lagrange multiplier test for general, high-order, ARMA errors. In the Lag Specification dialog box, you should enter the highest order of serial correlation to be tested.

The null hypothesis of the test is that there is no serial correlation in the residuals up to the specified order. EViews reports a statistic labeled "F-statistic" and an "Obs*R-squared"

(NR^2—the number of observations times the R-square) statistic. The NR^2 statistic has an asymptotic χ^2 distribution under the null hypothesis. The distribution of the F-statistic is not known, but is often used to conduct an informal test of the null.

See "Serial Correlation LM Test" on page 305 for further discussion of the serial correlation LM test.

Example

As an example of the application of these testing procedures, consider the following results from estimating a simple consumption function by ordinary least squares:

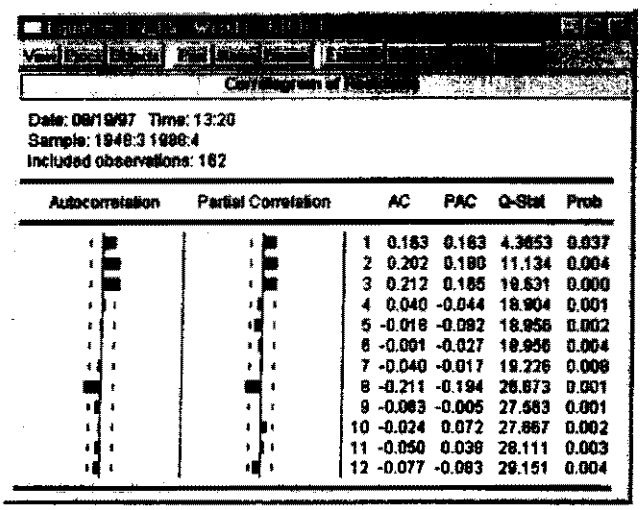
Dependent Variable: CS
Method: Least Squares
Date: 08/19/97 Time: 13:03
Sample: 1948:3 1988:4
Included observations: 162

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-9.227624	5.896177	-1.564487	0.1197
GDP	0.038732	0.017205	2.251193	0.0257
CS(-1)	0.952049	0.024484	38.88516	0.0000

R-squared	0.999525	Mean dependent var	1781.875
Adjusted R-squared	0.999521	S.D. dependent var	694.5419
S.E. of regression	13.53003	Akaike info criterion	8.068045
Sum squared resid	29106.82	Schwarz criterion	8.123223
Log likelihood	-650.3497	F-statistic	212047.1
Durbin-Watson stat	1.672255	Prob(F-statistic)	0.000000

A quick glance at the results reveals that the coefficients are statistically significant and the fit is very tight. However, if the error term is serially correlated, the estimated OLS standard errors are invalid and the estimated coefficients will be biased and inconsistent due to the presence of a lagged dependent variable on the right-hand side. The Durbin-Watson statistic is not appropriate as a test for serial correlation in this case, since there is a lagged dependent variable on the right-hand side of the equation.

Selecting View/Residual Tests/Correlogram-Q-statistics from this equation produces the following view:



Date: 08/19/97 Time: 13:20
Sample: 1948:3 1998:4
Included observations: 162

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.183	0.183	4.3653	0.037
		2	0.202	0.180	11.134	0.004
		3	0.212	0.185	18.631	0.000
		4	0.040	-0.044	18.904	0.001
		5	-0.018	-0.092	18.956	0.002
		6	-0.001	-0.027	18.956	0.004
		7	-0.040	-0.017	19.226	0.008
		8	-0.211	-0.194	26.873	0.001
		9	-0.083	-0.005	27.583	0.001
		10	-0.024	0.072	27.667	0.002
		11	-0.050	0.038	28.111	0.003
		12	-0.077	-0.083	29.151	0.004

The correlogram has spikes at lags up to three and at lag eight. The Q-statistics are significant at all lags, indicating significant serial correlation in the residuals.

Selecting **View/Residual Tests/Serial Correlation LM Test...** and entering a lag of 4 yields the following result:

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	3.654696	Probability	0.007109
Obs*R-squared	13.96215	Probability	0.007417

The test rejects the hypothesis of no serial correlation up to order four. The Q-statistic and the LM test both indicate that the residuals are serially correlated and the equation should be re-specified before using it for hypothesis tests and forecasting.

Estimating AR Models

Before you use the tools described in this section, you may first wish to examine your model for other signs of misspecification. Serial correlation in the errors may be evidence of serious problems with your specification. In particular, you should be on guard for an excessively restrictive specification that you arrived at by experimenting with ordinary least squares. Sometimes, adding improperly excluded variables to your regression will eliminate the serial correlation.

For a discussion of the efficiency gains from the serial correlation correction and some Monte-Carlo evidence, see Rao and Griliches (1969).

First-Order Serial Correlation

To estimate an AR(1) model in EViews, open an equation by selecting **Quick/Estimate Equation...** and enter your specification as usual, adding the expression "AR(1)" to the end of your list. For example, to estimate a simple consumption function with AR(1) errors,

$$\begin{aligned} CS_t &= c_1 + c_2 GDP_t + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{13.5}$$

you should specify your equation as

```
cs c gdp ar(1)
```

EViews automatically adjusts your sample to account for the lagged data used in estimation, estimates the model, and reports the adjusted sample along with the remainder of the estimation output.

Higher-Order Serial Correlation

Estimating higher order AR models is only slightly more complicated. To estimate an AR(k), you should enter your specification, followed by expressions for each AR term you wish to include. If you wish to estimate a model with autocorrelations from one to five:

$$\begin{aligned} CS_t &= c_1 + c_2 GDP_t + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_5 u_{t-5} + \epsilon_t \end{aligned} \tag{13.6}$$

you should enter

```
cs c gdp ar(1) ar(2) ar(3) ar(4) ar(5)
```

By requiring that you enter all of the autocorrelations you wish to include in your model, EViews allows you great flexibility in restricting lower order correlations to be zero. For example, if you have quarterly data and want to include a single term to account for seasonal autocorrelation, you could enter

```
cs c gdp ar(4)
```

Nonlinear Models with Serial Correlation

EViews can estimate nonlinear regression models with additive AR errors. For example, suppose you wish to estimate the following nonlinear specification with an AR(2) error:

$$\begin{aligned} CS_t &= c_1 + GDP_t^{c_2} + u_t \\ u_t &= c_3 u_{t-1} + c_4 u_{t-2} + \epsilon_t \end{aligned} \tag{13.7}$$

Simply specify your model using EViews expressions, followed by an additive term describing the AR correction enclosed in square brackets. The AR term should contain a coefficient assignment for each AR lag, separated by commas:

```
cs = c(1) + gdp^c(2) + [ar(1)=c(3), ar(2)=c(4)]
```

EViews transforms this nonlinear model by differencing, and estimates the transformed nonlinear specification using a Gauss-Newton iterative procedure (see "How EViews Estimates AR Models" on page 310).

Two-Stage Regression Models with Serial Correlation

By combining two-stage least squares or two-stage nonlinear least squares with AR terms, you can estimate models where there is correlation between regressors and the innovations as well as serial correlation in the residuals.

If the original regression model is linear, EViews uses the Marquardt algorithm to estimate the parameters of the transformed specification. If the original model is nonlinear, EViews uses Gauss-Newton to estimate the AR corrected specification.

For further details on the algorithms and related issues associated with the choice of instruments, see the discussion in "TSLS with AR errors" beginning on page 286.

Output from AR Estimation

When estimating an AR model, some care must be taken in interpreting your results. While the estimated coefficients, coefficient standard errors, and t-statistics may be interpreted in the usual manner, results involving residuals differ from those computed in OLS settings.

To understand these differences, keep in mind that there are two different residuals associated with an AR model. The first are the estimated *unconditional residuals*,

$$\hat{u}_t = y_t - x_t'b, \tag{13.8}$$

which are computed using the original variables, and the estimated coefficients, b. These residuals are the errors that you would observe if you made a prediction of the value of y_t using contemporaneous information, but ignoring the information contained in the lagged residual.

Normally, there is no strong reason to examine these residuals, and EViews does not automatically compute them following estimation.

The second set of residuals are the estimated *one-period ahead forecast errors*, $\hat{\varepsilon}$. As the name suggests, these residuals represent the forecast errors you would make if you computed forecasts using a prediction of the residuals based upon past values of your data, in addition to the contemporaneous information. In essence, you improve upon the uncondi-

tional forecasts and residuals by taking advantage of the predictive power of the lagged residuals.

For AR models, the residual-based regression statistics—such as the R^2, the standard error of regression, and the Durbin-Watson statistic— reported by EViews are based on the one-period ahead forecast errors, $\tilde{e}$.

A set of statistics that is unique to AR models is the estimated AR parameters, $\hat{\rho}_i$. For the simple AR(1) model, the estimated parameter $\hat{\rho}$ is the serial correlation coefficient of the unconditional residuals. For a stationary AR(1) model, the true ρ lies between -1 (extreme negative serial correlation) and $+1$ (extreme positive serial correlation). The stationarity condition for general AR(p) processes is that the inverted roots of the lag polynomial lie inside the unit circle. EViews reports these roots as **Inverted AR Roots** at the bottom of the regression output. There is no particular problem if the roots are imaginary, but a stationary AR model should have all roots with modulus less than one.

How EViews Estimates AR Models

Textbooks often describe techniques for estimating AR models. The most widely discussed approaches, the Cochrane-Orcutt, Prais-Winsten, Hatanaka, and Hildreth-Lu procedures, are multi-step approaches designed so that estimation can be performed using standard *linear* regression. All of these approaches suffer from important drawbacks which occur when working with models containing lagged dependent variables as regressors, or models using higher-order AR specifications; see Davidson and MacKinnon (1994, pp. 329–341), Greene (1997, p. 600–607).

EViews estimates AR models using nonlinear regression techniques. This approach has the advantage of being easy to understand, generally applicable, and easily extended to nonlinear specifications and models that contain endogenous right-hand side variables. Note that the nonlinear least squares estimates are asymptotically equivalent to maximum likelihood estimates and are asymptotically efficient.

To estimate an AR(1) model, EViews transforms the linear model

$$\begin{aligned} y_t &= x_t'\beta + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{13.9}$$

into the nonlinear model,

$$y_t = \rho y_{t-1} + (x_t - \rho x_{t-1})'\beta + \epsilon_t, \tag{13.10}$$

by substituting the second equation into the first, and rearranging terms. The coefficients ρ and β are estimated simultaneously by applying a Marquardt nonlinear least squares algorithm to the transformed equation. See Appendix D, "Estimation Algorithms and Options", on page 663 for details on nonlinear estimation.

For a nonlinear AR(1) specification, EViews transforms the nonlinear model

$$\begin{aligned} y_t &= f(x_t, \beta) + u_t \\ u_t &= \rho u_{t-1} + \epsilon_t \end{aligned} \tag{13.11}$$

into the alternative nonlinear specification

$$y_t = \rho y_{t-1} + f(x_t, \beta) - \rho f(x_{t-1}, \beta) + \epsilon_t \tag{13.12}$$

and estimates the coefficients using a Marquardt nonlinear least squares algorithm.

Higher order AR specifications are handled analogously. For example, a nonlinear AR(3) is estimated using nonlinear least squares on the equation

$$\begin{aligned} y_t = {} & (\rho_1 y_{t-1} + \rho_2 y_{t-2} + \rho_3 y_{t-3}) + f(x_t, \beta) - \rho_1 f(x_{t-1}, \beta) \\ & - \rho_2 f(x_{t-2}, \beta) - \rho_3 f(x_{t-3}, \beta) + \epsilon_t \end{aligned} \tag{13.13}$$

For details, see Fair (1984, pp. 210–214), and Davidson and MacKinnon (1996, pp. 331–341).

ARIMA Theory

ARIMA (autoregressive integrated moving average) models are generalizations of the simple AR model that use three tools for modeling the serial correlation in the disturbance:

- The first tool is the autoregressive, or AR, term. The AR(1) model introduced above uses only the first-order term but, in general, you may use additional, higher-order AR terms. Each AR term corresponds to the use of a lagged value of the residual in the forecasting equation for the unconditional residual. An autoregressive model of order p, AR(p) has the form

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_p u_{t-p} + \epsilon_t . \tag{13.14}$$

- The second tool is the integration order term. Each integration order corresponds to differencing the series being forecast. A first-order integrated component means that the forecasting model is designed for the first difference of the original series. A second-order component corresponds to using second differences, and so on.

- The third tool is the MA, or moving average term. A moving average forecasting model uses lagged values of the forecast error to improve the current forecast. A first-order moving average term uses the most recent forecast error, a second-order term uses the forecast error from the two most recent periods, and so on. An MA(q) has the form:

$$u_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} . \tag{13.15}$$

Please be aware that some authors and software packages use the opposite sign convention for the θ coefficients so that the signs of the MA coefficients may be reversed.

The autoregressive and moving average specifications can be combined to form an ARMA(p, q) specification

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \ldots + \rho_p u_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} \tag{13.16}$$

Although econometricians typically use ARIMA models applied to the residuals from a regression model, the specification can also be applied directly to a series. This latter approach provides a univariate model, specifying the conditional mean of the series as a constant, and measuring the residuals as differences of the series from its mean.

Principles of ARIMA Modeling (Box-Jenkins 1976)

In ARIMA forecasting, you assemble a complete forecasting model by using combinations of the three building blocks described above. The first step in forming an ARIMA model for a series of residuals is to look at its autocorrelation properties. You can use the correlogram view of a series for this purpose, as outlined in "Correlogram" on page 167.

This phase of the ARIMA modeling procedure is called *identification* (not to be confused with the same term used in the simultaneous equations literature). The nature of the correlation between current values of residuals and their past values provides guidance in selecting an ARIMA specification.

The autocorrelations are easy to interpret—each one is the correlation coefficient of the current value of the series with the series lagged a certain number of periods. The partial autocorrelations are a bit more complicated; they measure the correlation of the current and lagged series after taking into account the predictive power of all the values of the series with smaller lags. The partial autocorrelation for lag 6, for example, measures the added predictive power of u_{t-6} when $u_1, \ldots, u_{t-5}$ are already in the prediction model. In fact, the partial autocorrelation is precisely the regression coefficient of u_{t-6} in a regression where the earlier lags are also used as predictors of u_t.

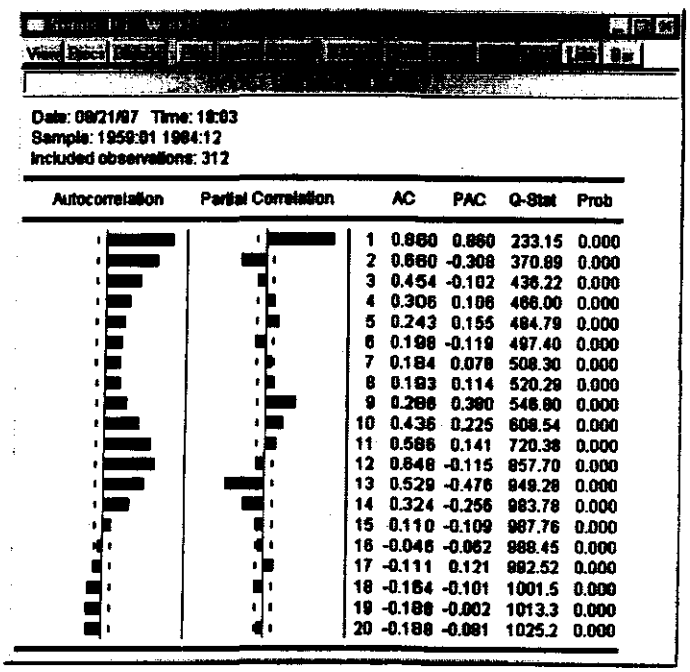
If you suspect that there is a distributed lag relationship between your dependent (left-hand) variable and some other predictor, you may want to look at their cross correlations before carrying out estimation.

The next step is to decide what kind of ARIMA model to use. If the autocorrelation function dies off smoothly at a geometric rate, and the partial autocorrelations were zero after one lag, then a first-order autoregressive model is appropriate. Alternatively, if the autocorrelations were zero after one lag and the partial autocorrelations declined geometrically, a first-order moving average process would seem appropriate. If the autocorrelations appear

to have a seasonal pattern, this would suggest the presence of a seasonal ARMA structure (see "Seasonal ARMA Terms" on page 316).

For example, we can examine the correlogram of the DRI Basics housing series in the HS.WF1 workfile by selecting **View/Correlogram...** from the HS series toolbar:

The "wavy" cyclical correlogram with a seasonal frequency suggests fitting a seasonal ARMA model to HS.



	AC	PAC	Q-Stat	Prob
1	0.860	0.860	233.15	0.000
2	0.660	-0.308	370.89	0.000
3	0.454	-0.102	436.22	0.000
4	0.306	0.106	466.00	0.000
5	0.243	0.155	484.79	0.000
6	0.198	-0.119	497.40	0.000
7	0.184	0.078	508.30	0.000
8	0.193	0.114	520.29	0.000
9	0.266	0.380	546.80	0.000
10	0.436	0.225	608.54	0.000
11	0.586	0.141	720.38	0.000
12	0.648	-0.115	857.70	0.000
13	0.529	-0.476	949.28	0.000
14	0.324	-0.256	983.78	0.000
15	0.110	-0.109	987.76	0.000
16	-0.046	-0.062	988.45	0.000
17	-0.111	0.121	992.52	0.000
18	-0.164	-0.101	1001.5	0.000
19	-0.188	-0.002	1013.3	0.000
20	-0.188	-0.091	1025.2	0.000

The goal of ARIMA analysis is a parsimonious representation of the process governing the residual. You should use only enough AR and MA terms to fit the properties of the residuals. The Akaike information criterion and Schwarz criterion provided with each set of estimates may also be used as a guide for the appropriate lag order selection.

After fitting a candidate ARIMA specification, you should verify that there are no remaining autocorrelations that your model has not accounted for. Examine the autocorrelations and the partial autocorrelations of the innovations (the residuals from the ARIMA model) to see if any important forecasting power has been overlooked. EViews provides views for diagnostic checks after estimation.

Estimating ARIMA Models

EViews estimates general ARIMA specifications that allow for right-hand side explanatory variables. Despite the fact that these models are sometimes termed ARIMAX specifications, we will refer to this general class of models as ARIMA.

To specify your ARIMA model, you will:

- Difference your dependent variable, if necessary, to account for the order of integration.
- Describe your structural regression model (dependent variables and regressors) and add any AR or MA terms, as described above.

Differencing

The d operator can be used to specify differences of series. To specify first differencing, simply include the series name in parentheses after d. For example, `d(gdp)` specifies the first difference of GDP, or GDP–GDP(–1).

More complicated forms of differencing may be specified with two optional parameters, n and s. `d(x,n)` specifies the n-th order difference of the series X:

$$d(x, n) = (1 - L)^n x, \tag{13.17}$$

where L is the lag operator. For example, `d(gdp,2)` specifies the second order difference of GDP:

```
d(gdp,2) = gdp - 2*gdp(-1) + gdp(-2)
```

`d(x,n,s)` specifies n-th order ordinary differencing of X with a seasonal difference at lag s:

$$d(x, n, s) = (1 - L)^n (1 - L^s) x. \tag{13.18}$$

For example, `d(gdp,0,4)` specifies zero ordinary differencing with a seasonal difference at lag 4, or GDP–GDP(–4).

If you need to work in logs, you can also use the `dlog` operator, which returns differences in the log values. For example, `dlog(gdp)` specifies the first difference of log(GDP) or log(GDP)–log(GDP(–1)). You may also specify the n and s options as described for the simple d operator, `dlog(x,n,s)`.

There are two ways to estimate integrated models in EViews. First, you may generate a new series containing the differenced data, and then estimate an ARMA model using the new data. For example, to estimate a Box-Jenkins ARIMA(1, 1, 1) model for M1, you can enter:

```
series dm1 = d(m1)
ls dm1 c ar(1) ma(1)
```

Alternatively, you may include the difference operator d directly in the estimation specification. For example, the same ARIMA(1,1,1) model can be estimated by the one-line command

```
ls d(m1) c ar(1) ma(1)
```

The latter method should generally be preferred for an important reason. If you define a new variable, such as DM1 above, and use it in your estimation procedure, then when you forecast from the estimated model, EViews will make forecasts of the dependent variable DM1. That is, you will get a forecast of the differenced series. If you are really interested in forecasts of the level variable, in this case M1, you will have to manually transform the

forecasted value and adjust the computed standard errors accordingly. Moreover, if any other transformation or lags of M1 are included as regressors, EViews will not know that they are related to DM1. If, however, you specify the model using the difference operator expression for the dependent variable, `d(m1)`, the forecasting procedure will provide you with the option of forecasting the level variable, in this case M1.

The difference operator may also be used in specifying exogenous variables and can be used in equations without ARMA terms. Simply include them in the list of regressors in addition to the endogenous variables. For example,

```
d(cs,2) c d(gdp,2) d(gdp(-1),2) d(gdp(-2),2) time
```

is a valid specification that employs the difference operator on both the left-hand and right-hand sides of the equation.

ARMA Terms

The AR and MA parts of your model will be specified using the keywords `ar` and `ma` as part of the equation. We have already seen examples of this approach in our specification of the AR terms above, and the concepts carry over directly to MA terms.

For example, to estimate a second-order autoregressive and first-order moving-average error process ARMA(2,1), you would include expressions for the AR(1), AR(2), and MA(1) terms along with your other regressors:

```
c gov ar(1) ar(2) ma(1)
```

Once again, you need not use the AR and MA terms consecutively. For example, if you want to fit a fourth-order autoregressive model to take account of seasonal movements, you could use AR(4) by itself:

```
c gov ar(4)
```

You may also specify a pure moving average model by using only MA terms. Thus,

```
c gov ma(1) ma(2)
```

indicates an MA(2) model for the residuals.

The traditional Box-Jenkins or ARMA models do not have any right-hand side variables except for the constant. In this case, your list of regressors would just contain a C in addition to the AR and MA terms. For example,

```
c ar(1) ar(2) ma(1) ma(2)
```

is a standard Box-Jenkins ARMA (2,2).

Seasonal ARMA Terms

Box and Jenkins (1976) recommend the use of seasonal autoregressive (SAR) and seasonal moving average (SMA) terms for monthly or quarterly data with systematic seasonal movements. A SAR(p) term can be included in your equation specification for a seasonal autoregressive term with lag p. The lag polynomial used in estimation is the product of the one specified by the AR terms and the one specified by the SAR terms. The purpose of the SAR is to allow you to form the product of lag polynomials.

Similarly, SMA(q) can be included in your specification to specify a seasonal moving average term with lag q. The lag polynomial used in estimation is the product of the one defined by the MA terms and the one specified by the SMA terms. As with the SAR, the SMA term allows you to build up a polynomial that is the product of underlying lag polynomials.

For example, a second-order AR process without seasonality is given by

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \epsilon_t, \tag{13.19}$$

which can be represented using the lag operator L, $L^n x_t = x_{t-n}$ as

$$(1 - \rho_1 L - \rho_2 L^2) u_t = \epsilon_t. \tag{13.20}$$

You can estimate this process by including `ar(1)` and `ar(2)` terms in the list of regressors. With quarterly data, you might want to add a `sar(4)` expression to take account of seasonality. If you specify the equation as

```
sales c inc ar(1) ar(2) sar(4)
```

then the estimated error structure would be:

$$(1 - \rho_1 L - \rho_2 L^2)(1 - \phi L^4) u_t = \epsilon_t. \tag{13.21}$$

The error process is equivalent to:

$$u_t = \rho_1 u_{t-1} + \rho_2 u_{t-2} + \phi u_{t-4} - \phi\rho_1 u_{t-5} - \phi\rho_2 u_{t-6} + \epsilon_t. \tag{13.22}$$

The parameter ϕ is associated with the seasonal part of the process. Note that this is an AR(6) process with nonlinear restrictions on the coefficients.

As another example, a second-order MA process without seasonality may be written

$$u_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}, \tag{13.23}$$

or using lag operators,

$$u_t = (1 + \theta_1 L + \theta_2 L^2) \epsilon_t. \tag{13.24}$$

You can estimate this second-order process by including both the MA(1) and MA(2) terms in your equation specification.

With quarterly data, you might want to add `sma(4)` to take account of seasonality. If you specify the equation as

```
cs c ad ma(1) ma(2) sma(4)
```

then the estimated model is:

$$\begin{aligned} CS_t &= \beta_1 + \beta_2 AD_t + u_t \\ u_t &= (1 + \theta_1 L + \theta_2 L^2)(1 + \omega L^4)\epsilon_t \end{aligned} \tag{13.25}$$

The error process is equivalent to

$$u_t = \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \omega\epsilon_{t-4} + \omega\theta_1\epsilon_{t-5} + \omega\theta_2\epsilon_{t-6}\,. \tag{13.26}$$

The parameter w is associated with the seasonal part of the process. This is just an MA(6) process with nonlinear restrictions on the coefficients. You can also include both SAR and SMA terms.

Output from ARIMA Estimation

The output from estimation with AR or MA specifications is the same as for ordinary least squares, with the addition of a lower block that shows the reciprocal roots of the AR and MA polynomials. If we write the general ARMA model using the lag polynomial $\rho(L)$ and $\theta(L)$ as

$$\rho(L)u_t = \theta(L)\epsilon_t, \tag{13.27}$$

then the reported roots are the roots of the polynomials

$$\rho(x^{-1}) = 0 \qquad \text{and} \qquad \theta(x^{-1}) = 0\,. \tag{13.28}$$

The roots, which may be imaginary, should have modulus no greater than one. The output will display a warning message if any of the roots violate this condition.

If ρ has a real root whose absolute value exceeds one or a pair of complex reciprocal roots outside the unit circle (that is, with modulus greater than one), it means that the autoregressive process is explosive.

If θ has reciprocal roots outside the unit circle, we say that the MA process is *noninvertible*, which makes interpreting and using the MA results difficult. However, noninvertibility poses no substantive problem, since as Hamilton (1994a, p. 65) notes, there is always an equivalent representation for the MA model where the reciprocal roots lie inside the unit circle. Accordingly, you should re-estimate your model with different starting values until you get a moving average process that satisfies invertibility. Alternatively, you may wish to turn off MA backcasting (see "Backcasting MA terms" on page 320).

If the estimated MA process has roots with modulus close to one, it is a sign that you may have over-differenced the data. The process will be difficult to estimate and even more difficult to forecast. If possible, you should re-estimate with one less round of differencing.

Consider the following example output from ARMA estimation:

Dependent Variable: R
Method: Least Squares
Date: 08/14/97 Time: 16:53
Sample(adjusted): 1954:06 1993:07
Included observations: 470 after adjusting endpoints
Convergence achieved after 25 iterations
Backcast: 1954:01 1954:05

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	8.614804	0.961559	8.958208	0.0000
AR(1)	0.983011	0.009127	107.7077	0.0000
SAR(4)	0.941898	0.018788	50.13275	0.0000
MA(1)	0.513572	0.040236	12.76402	0.0000
SMA(4)	-0.960369	0.000601	-1598.423	0.0000

R-squared	0.991557	Mean dependent var	6.978830
Adjusted R-squared	0.991484	S.D. dependent var	2.919807
S.E. of regression	0.269420	Akaike info criterion	0.225489
Sum squared resid	33.75296	Schwarz criterion	0.269867
Log likelihood	-47.96992	F-statistic	13652.76
Durbin-Watson stat	2.099958	Prob(F-statistic)	0.000000

Inverted AR Roots	.99	.98		
Inverted MA Roots	.99	-.00+.99i	-.00-.99i	-.51
	-.99			

This estimation result corresponds to the following specification:

$$
\begin{aligned}
y_t &= 8.61 + u_t \\
(1-0.98L)(1-0.94L^4)u_t &= (1+0.51L)(1-0.96L^4)\epsilon_t
\end{aligned}
\tag{13.29}
$$

or equivalently, to

$$
\begin{aligned}
y_t = {} & 0.0088 + 0.98y_{t-1} + 0.94y_{t-4} - 0.92y_{t-5} + \epsilon_t \\
& + 0.51\epsilon_{t-1} - 0.96\epsilon_{t-4} - 0.49\epsilon_{t-5}
\end{aligned}
\tag{13.30}
$$

Note that the signs of the MA terms may be reversed from those in textbooks. Note also that the inverted roots have moduli very close to one, which is typical for many macro time series models.

Estimation Options

ARMA estimation employs the same nonlinear estimation techniques described earlier for AR estimation. These nonlinear estimation techniques are discussed further in Chapter 12, "Additional Regression Methods", on page 290.

You may need to use the **Estimation Options** dialog box to control the iterative process. EViews provides a number of options that allow you to control the iterative procedure of the estimation algorithm. In general, you can rely on the EViews choices but on occasion you may wish to override the default settings.

Iteration Limits and Convergence Criterion

Controlling the maximum number of iterations and convergence criterion are described in detail in "Iteration and Convergence Options" on page 669.

Derivative Methods

EViews always computes the derivatives of AR coefficients analytically and the derivatives of the MA coefficients using finite difference numeric derivative methods. For other coefficients in the model, EViews provides you with the option of computing analytic expressions for derivatives of the regression equation (if possible) or computing finite difference numeric derivatives in cases where the derivative is not constant. Furthermore, you can choose whether to favor speed of computation (fewer function evaluations) or whether to favor accuracy (more function evaluations) in the numeric derivative computation.

Starting Values for ARMA Estimation

As discussed above, models with AR or MA terms are estimated by nonlinear least squares. Nonlinear estimation techniques require starting values for all coefficient estimates. Normally, EViews determines its own starting values and for the most part this is an issue that you need not be concerned about. However, there are a few times when you may want to override the default starting values.

First, estimation will sometimes halt when the maximum number of iterations is reached, despite the fact that convergence is not achieved. Resuming the estimation with starting values from the previous step causes estimation to pick up where it left off instead of starting over. You may also want to try different starting values to ensure that the estimates are a global rather than a local minimum of the squared errors. You might also want to supply starting values if you have a good idea of what the answers should be, and want to speed up the estimation process.

To control the starting values for ARMA estimation, click on the **Options** button in the Equation Specification dialog. Among the options which EViews provides are several alternatives for setting starting values that you can see by accessing the drop-down menu labeled **Starting Coefficient Values for ARMA**.

EViews' default approach is **OLS/TSLS**, which runs a preliminary estimation without the ARMA terms and then starts nonlinear estimation from those values. An alternative is to use fractions of the OLS or TSLS coefficients as starting values. You can choose **.8**, **.5**, **.3**, or you can start with all coefficient values set equal to zero.

The final starting value option is **User Supplied**. Under this option, EViews uses the coefficient values that are in the coefficient vector. To set the starting values, open a window for the coefficient vector C by double clicking on the icon, and editing the values.

To properly set starting values, you will need a little more information about how EViews assigns coefficients for the ARMA terms. As with other estimation methods, when you specify your equation as a list of variables, EViews uses the built-in C coefficient vector. It assigns coefficient numbers to the variables in the following order:

- First are the coefficients of the variables, in order of entry.
- Next come the AR terms in the order you typed them.
- The SAR, MA, and SMA coefficients follow, in that order.

Thus the following two specifications will have their coefficients in the same order:

```
y c x ma(2) ma(1) sma(4) ar(1)
y sma(4)c ar(1) ma(2) x ma(1)
```

You may also assign values in the C vector using the `param` command:

```
param c(1) 50 c(2) .8 c(3) .2 c(4) .6 c(5) .1 c(6) .5
```

The starting values will be 50 for the constant, 0.8 for X, 0.2 for AR(1), 0.6 for MA(2), 0.1 for MA(1) and 0.5 for SMA(4). Following estimation, you can always see the assignment of coefficients by looking at the **Representations** view of your equation.

You can also fill the C vector from any estimated equation (without typing the numbers) by choosing **Procs/Update Coefs from Equation** in the equation toolbar.

Backcasting MA terms

By default, EViews backcasts MA terms (Box and Jenkins, 1976). Consider an MA(q) model of the form

$$\begin{aligned} y_t &= X_t'\beta + u_t \\ u_t &= \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \ldots + \theta_q\epsilon_{t-q} \end{aligned} \tag{13.31}$$

Given initial values, $\hat{\beta}$ and $\hat{\phi}$, EViews first computes the unconditional residuals $\hat{u}_t$ for $t = 1, 2, \ldots, T$, and uses the backward recursion:

$$\tilde{\epsilon}_t = \hat{u}_t - \hat{\phi}_1\tilde{\epsilon}_{t+1} - \ldots - \hat{\phi}_q\tilde{\epsilon}_{t+q} \tag{13.32}$$

to compute backcast values of ϵ to $\epsilon_{-(q-1)}$. To start this recursion, the q values for the innovations *beyond* the estimation sample are set to zero:

$$\tilde{\epsilon}_{T+1} = \tilde{\epsilon}_{T+2} = \ldots = \tilde{\epsilon}_{T+q} = 0. \tag{13.33}$$

Next, a forward recursion is used to estimate the values of the innovations

$$\hat{\epsilon}_t = \hat{u}_t - \hat{\phi}_1\hat{\epsilon}_{t-1} - \ldots - \hat{\phi}_q\hat{\epsilon}_{t-q}, \tag{13.34}$$

using the backcasted values of the innovations (to initialize the recursion) and the actual residuals. If your model also includes AR terms, EViews will ρ-difference the $\hat{u}_t$ to eliminate the serial correlation prior to performing the backcast.

Lastly, the sum of squared residuals (SSR) is formed as a function of the β and ϕ, using the fitted values of the lagged innovations:

$$\text{ssr}(\beta, \phi) = \sum_{t=p+1}^{T} (y_t - X_t'\beta - \phi_1\hat{\epsilon}_{t-1} - \ldots - \phi_q\hat{\epsilon}_{t-q})^2. \tag{13.35}$$

This expression is minimized with respect to β and ϕ.

The backcast step, forward recursion, and minimization procedures, are repeated until the estimates of β and ϕ converge.

If backcasting is turned off, the values of the pre-sample ϵ are set to zero:

$$\epsilon_{-(q-1)} = \ldots = \hat{\epsilon}_0 = 0, \tag{13.36}$$

and forward recursion is used to solve for the remaining values of the innovations.

Dealing with Estimation Problems

Since EViews uses nonlinear least squares algorithms to estimate ARMA models, all of the discussion in Chapter 12, "Solving Estimation Problems" on page 296, is applicable, especially the advice to try alternative starting values.

There are a few other issues to consider that are specific to estimation of ARMA models.

First, MA models are notoriously difficult to estimate. In particular, you should avoid high order MA terms unless absolutely required for your model as they are likely to cause estimation difficulties. For example, a single large spike at lag 57 in the correlogram does not necessarily require you to include an MA(57) term in your model unless you know there is something special happening every 57 periods. It is more likely that the spike in the correlogram is simply the product of one or more outliers in the series. By including many MA terms in your model, you lose degrees of freedom, and may sacrifice stability and reliability of your estimates.

If the underlying roots of the MA process have modulus close to one, you may encounter estimation difficulties, with EViews reporting that it cannot improve the sum-of-squares or that it failed to converge in the maximum number of iterations. This behavior may be a sign that you have over-differenced the data. You should check the correlogram of the series to determine whether you can re-estimate with one less round of differencing.

Lastly, if you continue to have problems, you may wish to turn off MA backcasting.

TSLS with ARIMA errors

Two-stage least squares or instrumental variable estimation with ARIMA poses no particular difficulties.

For a detailed discussion of how to estimate TSLS specifications with ARMA errors, see "Two-stage Least Squares" on page 283.

Nonlinear Models with ARMA errors

EViews will estimate nonlinear ordinary and two-stage least squares models with autoregressive error terms. For details, see the extended discussion in "Nonlinear Least Squares" beginning on page 289.

EViews does not currently estimate nonlinear models with MA errors. You can, however, use the state space object to specify and estimate these models (see "ARMAX(2, 3) with a Random Coefficient" on page 586).

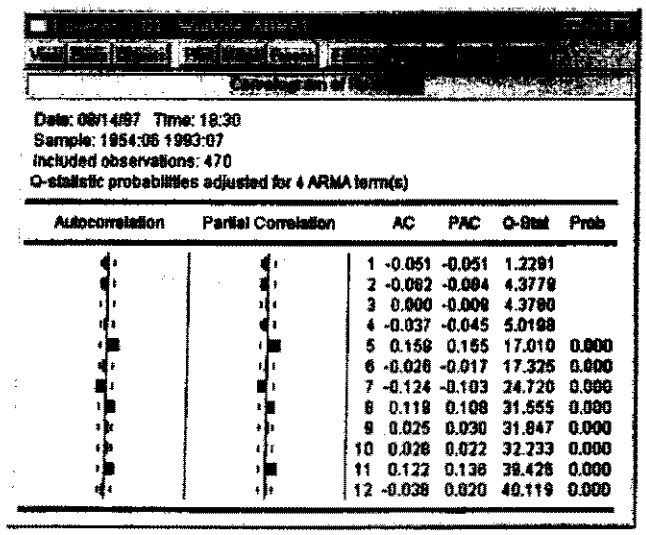
Weighted Models with ARMA errors

EViews does not have procedures to automatically estimate weighted models with ARMA error terms—if you add AR terms to a weighted model, the weighting series will be ignored. You can, of course, always construct the weighted series and then perform estimation using the weighted data and ARMA terms.

Diagnostic Evaluation

If your ARMA model is correctly specified, the residuals from the model should be nearly white noise. This means that there should be no serial correlation left in the residuals. The Durbin-Watson statistic reported in the regression output is a test for AR(1) in the absence of lagged dependent variables on the right-hand side. As discussed above, more general tests for serial correlation in the residuals can be carried out with **View/Residual Tests/Correlogram-Q-statistic** and **View/Residual Tests/Serial Correlation LM Test...**.

For the example seasonal ARMA model, the residual correlogram looks as follows:



Date: 08/14/97 Time: 18:30
Sample: 1954:06 1993:07
Included observations: 470
Q-statistic probabilities adjusted for 4 ARMA term(s)

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.051	-0.051	1.2291	
		2	-0.082	-0.084	4.3779	
		3	0.000	-0.008	4.3780	
		4	-0.037	-0.045	5.0198	
		5	0.158	0.155	17.010	0.000
		6	-0.026	-0.017	17.325	0.000
		7	-0.124	-0.103	24.720	0.000
		8	0.119	0.108	31.555	0.000
		9	0.025	0.030	31.847	0.000
		10	0.026	0.022	32.233	0.000
		11	0.122	0.136	39.426	0.000
		12	-0.038	0.020	40.119	0.000

The correlogram has a significant spike at lag 5 and all subsequent Q-statistics are highly significant. This result clearly indicates the need for respecification of the model.

Polynomial Distributed Lags (PDLs)

A distributed lag is a relation of the type

$$y_t = w_t\delta + \beta_0 x_t + \beta_1 x_{t-1} + \ldots + \beta_k x_{t-k} + \epsilon_t \tag{13.37}$$

The coefficients β describe the lag in the effect of x on y. In many cases, the coefficients can be estimated directly using this specification. In other cases, the high collinearity of current and lagged values of x will defeat direct estimation.

You can reduce the number of parameters to be estimated by using polynomial distributed lags (PDLs) to impose a smoothness condition on the lag coefficients. Smoothness is expressed as requiring that the coefficients lie on a polynomial of relatively low degree. A polynomial distributed lag model with order p restricts the β coefficients to lie on a p-th order polynomial of the form

$$\beta_j = \gamma_1 + \gamma_2(j - \bar{c}) + \gamma_3(j - \bar{c})^2 + \ldots + \gamma_{p+1}(j - \bar{c})^p \tag{13.38}$$

for $j = 1, 2, \ldots, k$, where $\bar{c}$ is a pre-specified constant given by

$$\bar{c} = \begin{cases} (k)/2 & \text{if } p \text{ is even} \\ (k-1)/2 & \text{if } p \text{ is odd} \end{cases} \tag{13.39}$$

The PDL is sometimes referred to as an Almon lag. The constant $\bar{c}$ is included only to avoid numerical problems that can arise from collinearity and does not affect the estimates of β.

This specification allows you to estimate a model with k lags of x using only p parameters (if you choose $p > k$, EViews will return a "Near Singular Matrix" error).

If you specify a PDL, EViews substitutes Equation (13.38) into Equation (13.37) yielding

$$y_t = \alpha + \gamma_1 z_1 + \gamma_2 z_2 + \ldots + \gamma_{p+1} z_{p+1} + \epsilon_t \tag{13.40}$$

where

$$\begin{aligned} z_1 &= x_t + x_{t-1} + \ldots + x_{t-k} \\ z_2 &= -\bar{c}x_t + (1-\bar{c})x_{t-1} + \ldots + (k-\bar{c})x_{t-k} \\ &\ldots \\ z_{p+1} &= (-\bar{c})^p x_t + (1-\bar{c})^p x_{t-1} + \ldots + (k-\bar{c})^p x_{t-k} \end{aligned} \tag{13.41}$$

Once we estimate γ from Equation (13.40), we can recover the parameters of interest β, and their standard errors using the relationship described in Equation (13.38). This procedure is straightforward since β is a linear transformation of γ.

The specification of a polynomial distributed lag has three elements: the length of the lag k, the degree of the polynomial (the highest power in the polynomial) p, and the constraints that you want to apply. A near end constraint restricts the one-period lead effect of x on y to be zero:

$$\beta_{-1} = \gamma_1 + \gamma_2(-1-\bar{c}) + \ldots + \gamma_{p+1}(-1-\bar{c})^p = 0\,. \tag{13.42}$$

A far end constraint restricts the effect of x on y to die off beyond the number of specified lags:

$$\beta_{k+1} = \gamma_1 + \gamma_2(k+1-\bar{c}) + \ldots + \gamma_{p+1}(k+1-\bar{c})^p = 0\,. \tag{13.43}$$

If you restrict either the near or far end of the lag, the number of γ parameters estimated is reduced by one to account for the restriction; if you restrict both the near and far end of the lag, the number of γ parameters is reduced by two.

By default, EViews does not impose constraints.

How to Estimate Models Containing PDLs

You specify a polynomial distributed lag by the pdl term, with the following information in parentheses, each separated by a comma in this order:

- The name of the series.
- The lag length (the number of lagged values of the series to be included).
- The degree of the polynomial.
- A numerical code to constrain the lag polynomial (optional):

1	constrain the near end of the lag to zero.
2	constrain the far end.
3	constrain both ends.

You may omit the constraint code if you do not want to constrain the lag polynomial. Any number of `pdl` terms may be included in an equation. Each one tells EViews to fit distributed lag coefficients to the series and to constrain the coefficients to lie on a polynomial.

For example,

```
ls sales c pdl(orders,8,3)
```

fits SALES to a constant, and a distributed lag of current and eight lags of ORDERS, where the lag coefficients of ORDERS lie on a third degree polynomial with no endpoint constraints. Similarly,

```
ls div c pdl(rev,12,4,2)
```

fits DIV to a distributed lag of current and 12 lags of REV, where the coefficients of REV lie on a 4th degree polynomial with a constraint at the far end.

The `pdl` specification may also be used in two-stage least squares. If the series in the `pdl` is exogenous, you should include the PDL of the series in the instruments as well. For this purpose, you may specify `pdl(*)` as an instrument; all `pdl` variables will be used as instruments. For example, if you specify the TSLS equation as

```
sales c inc pdl(orders(-1),12,4)
```

with instruments

```
fed fed(-1) pdl(*)
```

the distributed lag of ORDERS will be used as instruments together with FED and FED(-1).

Polynomial distributed lags cannot be used in nonlinear specifications.

Example

The distributed lag model of industrial production (IP) on money (M1) yields the following results:

Dependent Variable: IP
Method: Least Squares
Date: 08/15/97 Time: 17:09
Sample(adjusted): 1960:01 1989:12
Included observations: 360 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	40.67568	0.823886	49.37171	0.0000
M1	0.129699	0.214574	0.604449	0.5459
M1(-1)	-0.045962	0.378907	-0.121944	0.9030
M1(-2)	0.033183	0.397099	0.083563	0.9335
M1(-3)	0.010621	0.405861	0.026169	0.9791
M1(-4)	0.031425	0.418805	0.075035	0.9402
M1(-5)	-0.048847	0.431728	-0.113143	0.9100
M1(-6)	0.053880	0.440753	0.122245	0.9028
M1(-7)	-0.015240	0.438123	-0.034844	0.9721
M1(-8)	-0.024902	0.423546	-0.058795	0.9531
M1(-9)	-0.028048	0.413540	-0.067825	0.9460
M1(-10)	0.030806	0.407523	0.075593	0.9398
M1(-11)	0.018509	0.389133	0.047564	0.9621
M1(-12)	-0.057373	0.228026	-0.250728	0.8022

R-squared	0.852398	Mean dependent var	71.72679
Adjusted R-squared	0.846852	S.D. dependent var	19.53083
S.E. of regression	7.643137	Akaike info criterion	6.943808
Sum squared resid	20212.47	Schwarz criterion	7.094732
Log likelihood	-1235.849	F-statistic	153.7030
Durbin-Watson stat	0.006255	Prob(F-statistic)	0.000000

Taken individually, none of the coefficients on lagged M1 are statistically different from zero. Yet the regression as a whole has a reasonable R^2 with a very significant F-statistic (though with a very low Durbin-Watson statistic). This is a typical symptom of high collinearity among the regressors and suggests fitting a polynomial distributed lag model.

To estimate a fifth-degree polynomial distributed lag model with no constraints, enter the commands:

```
smpl 59.1 89.12
ls ip c pdl(m1,12,5)
```

The following result is reported at the top of the equation window:

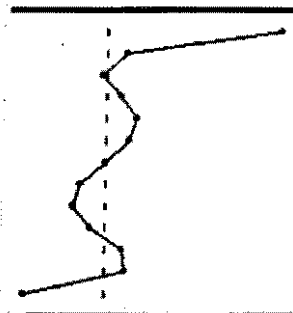
Dependent Variable: IP
Method: Least Squares
Date: 08/15/97 Time: 17:53
Sample(adjusted): 1960:01 1989:12
Included observations: 360 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	40.67311	0.815195	49.89374	0.0000
PDL01	-4.66E-05	0.055566	-0.000839	0.9993
PDL02	-0.015625	0.062884	-0.248479	0.8039
PDL03	-0.000160	0.013909	-0.011485	0.9908
PDL04	0.001862	0.007700	0.241788	0.8091
PDL05	2.58E-05	0.000408	0.063211	0.9496
PDL06	-4.93E-05	0.000180	-0.273611	0.7845

R-squared	0.852371	Mean dependent var	71.72679
Adjusted R-squared	0.849862	S.D. dependent var	19.53063
S.E. of regression	7.567664	Akaike info criterion	6.904899
Sum squared resid	20216.15	Schwarz criterion	6.980462
Log likelihood	-1235.882	F-statistic	339.6882
Durbin-Watson stat	0.008026	Prob(F-statistic)	0.000000

This portion of the view reports the estimated coefficients γ of the polynomial in Equation (13.38) on page 323. The terms PDL01, PDL02, PDL03, ..., correspond to $z_1, z_2, \ldots$ in Equation (13.40).

The implied coefficients of interest β_j in equation (1) are reported at the bottom of the table, together with a plot of the estimated polynomial:



Lag Distribution of M1	i	Coefficient	Std. Error	T-Statistic
	0	0.10270	0.14677	0.69970
	1	0.01159	0.10948	0.10587
	2	-0.00215	0.10138	-0.02123
	3	0.00920	0.06150	0.14955
	4	0.01766	0.07435	0.23756
	5	0.01363	0.06974	0.19547
	6	-4.7E-05	0.05557	-0.00084
	7	-0.01399	0.07080	-0.19764
	8	-0.01821	0.07537	-0.24158
	9	-0.00798	0.06399	-0.12475
	10	0.01017	0.10454	0.09726
	11	0.01260	0.11069	0.11386
	12	-0.04737	0.15693	-0.30182
Sum of Lags		0.08780	0.00297	29.5534

The Sum of Lags reported at the bottom of the table is the sum of the estimated coefficients on the distributed lag and has the interpretation of the long run effect of M1 on IP, assuming stationarity.

Note that selecting **View/Coefficient Tests** for an equation estimated with PDL terms tests the restrictions on γ, not on β. In this example, the coefficients on the fourth- (PDL05) and fifth-order (PDL06) terms are individually insignificant and very close to zero. To test

the joint significance of these two terms, click **View/Coefficient Tests/Wald-Coefficient Restrictions...** and type

```
c(6)=0, c(7)=0
```

in the Wald Test dialog box (see "Wald Test (Coefficient Restrictions)" on page 368 for an extensive discussion of Wald tests in EViews). EViews displays the result of the joint test:

Wald Test:
Equation: IP_PDL

Test Statistic	Value	df	Probability
F-statistic	0.030852	(2, 353)	0.9808
Chi-square	0.078704	2	0.9808

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
C(6)	2.59E-05	2448.827
C(7)	-4.93E-05	5560.537

Restrictions are linear in coefficients.

There is no evidence to reject the null hypothesis, suggesting that you could have fit a lower order polynomial to your lag structure.

Nonstationary Time Series

The theory behind ARMA estimation is based on stationary time series. A series is said to be (weakly or covariance) *stationary* if the mean and autocovariances of the series do not depend on time. Any series that is not stationary is said to be *nonstationary*.

A common example of a nonstationary series is the *random walk*:

$$y_t = y_{t-1} + \epsilon_t, \tag{13.44}$$

where ϵ is a stationary random disturbance term. The series y has a constant forecast value, conditional on t, and the variance is increasing over time. The random walk is a difference stationary series since the first difference of y is stationary:

$$y_t - y_{t-1} = (1-L)y_t = \epsilon_t. \tag{13.45}$$

A difference stationary series is said to be *integrated* and is denoted as $I(d)$ where d is the order of integration. The order of integration is the number of unit roots contained in the series, or the number of differencing operations it takes to make the series stationary. For the random walk above, there is one unit root, so it is an I(1) series. Similarly, a stationary series is I(0).

Standard inference procedures do not apply to regressions which contain an integrated dependent variable or integrated regressors. Therefore, it is important to check whether a

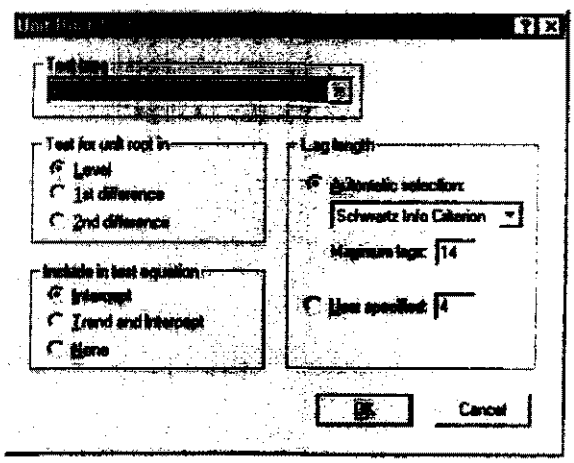
series is stationary or not before using it in a regression. The formal method to test the stationarity of a series is the unit root test.

Unit Root Tests

EViews provides you with a variety of powerful tools for testing a series (or the first or second difference of the series) for the presence of a unit root. In addition to the existing Augmented Dickey-Fuller (1979) and Phillips-Perron (1998) tests, EViews now allows you to compute the GLS-detrended Dickey-Fuller (Elliot, Rothenberg, and Stock, 1996), Kwiatkowski, Phillips, Schmidt, and Shin (KPSS, 1992), Elliott, Rothenberg, and Stock Point Optimal (ERS, 1996), and Ng and Perron (NP, 2001) unit root tests. All of these tests are available as a view of a series.

Performing Unit Root Tests in EViews

The following discussion assumes that you are familiar with the basic forms of the unit root tests, and the associated options. We provide theoretical background for these tests in "Basic Unit Root Theory" beginning on page 333, and document the settings used when performing these tests.



To begin, double click on the series name to open the series window, and choose **View/Unit Root Test...**

You must specify four sets of options to carry out a unit root test. The first three settings (on the left-hand side of the dialog) determine the basic form of the unit root test. The fourth set of options (on the right-hand side of the dialog) consist of test specific advanced settings. You only need concern yourself with these settings if you wish to customize the calculation of your unit root test.

First, you should use the topmost combo box to select the type of unit root test that you wish to perform. You may choose one of six tests: ADF, DFGLS, PP, KPSS, ERS, and NP.

Next, specify whether you wish to test for a unit root in the level, first difference, or second difference of the series.

Lastly, choose your exogenous regressors. You can choose to include a constant, a constant and linear trend, or neither (there are limitations on these choices for some of the tests).

You can click on **OK** to compute the test using the specified settings, or you can customize your test using the advanced settings portion of the dialog.

The advanced settings for both the ADF and DFGLS tests allow you to specify how lagged difference terms p are to be included in the ADF test equation. You may choose to let EViews automatically select p, or you may specify a fixed positive integer value (if you choose automatic selection, you are given the additional option of selecting both the information criterion and maximum number of lags to be used in the selection procedure).

In this case, we have chosen to estimate an ADF test that includes a constant in the test regression and employs automatic lag length selection using a Schwarz Information Criterion (BIC) and a maximum lag length of 14. Applying these settings to data on the U. S. one-month Treasury bill rate for the period from March 1953 to July 1971, we can replicate Example 9.2 of Hayashi (2000, p. 596). The results are described below.

The first part of the unit root output provides information about the form of the test (the type of test, the exogenous variables, and lag length used), and contains the test output, associated critical values, and in this case, the p-value:

Null Hypothesis: TBILL has a unit root
Exogenous: Constant
Lag Length: 1 (Automatic based on SIC, MAXLAG=14)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-1.417410	0.5734
Test critical values:	1% level	-3.458088	
	5% level	-2.874435	
	10% level	-2.573719	

*MacKinnon (1996) one-sided p-values.

The ADF statistic value is -1.417 and the associated one-sided p-value (for a test with 221 observations) is .573. In addition, EViews reports the critical values at the 1%, 5% and 10% levels. Notice here that the statistic t_α value is greater than the critical values so that we do not reject the null at conventional test sizes.

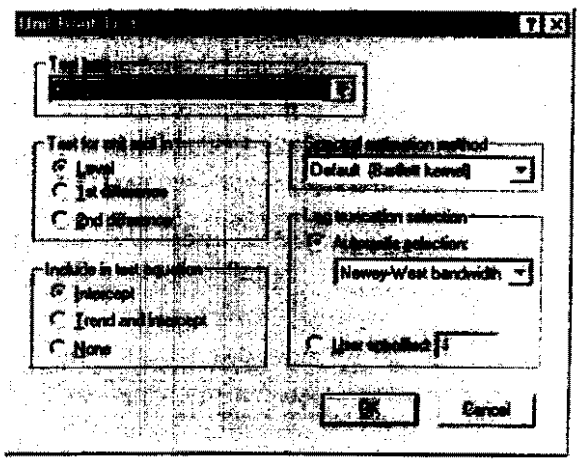
The second part of the output shows the intermediate test equation that EViews used to calculate the ADF statistic:

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(TBILL)
Method: Least Squares
Date: 02/07/02 Time: 12:29
Sample: 1953:03 1971:07
Included observations: 221

Variable	Coefficient	Std. Error	t-Statistic	Prob.
TBILL(-1)	-0.022951	0.016192	-1.417410	0.1578
D(TBILL(-1))	-0.203330	0.067007	-3.034470	0.0027
C	0.088398	0.056934	1.552526	0.1220

R-squared	0.053856	Mean dependent var	0.013826
Adjusted R-squared	0.045175	S.D. dependent var	0.379758
S.E. of regression	0.371081	Akaike info criterion	0.868688
Sum squared resid	30.01882	Schwarz criterion	0.914817
Log likelihood	-92.99005	F-statistic	6.204410
Durbin-Watson stat	1.976361	Prob(F-statistic)	0.002395

If you had choosen to perform any of the other unit root tests (PP, KPSS, ERS, NP), the right side of the dialog would show the different options associated with the specified test. The options are associated with the method used to estimate the zero frequency spectrum term, f_0, that is used in constructing the particular test statistic. As before, you only need pay attention to these settings if you wish to change from the EViews defaults.



Here we have selected the PP test in the combo box. Note that the right-hand side of the dialog has changed, and now features a combo box for selecting the spectral estimation method. You may use this combo box to choose between various kernel or AR regression based estimators for f_0. The entry labeled "Default" will show you the default estimator for the specific unit root test—here we see that the PP default uses a kernel sum-of-covariances estimator with Bartlett weights. If, instead, you had selected a NP test, the default entry would be "AR spectral-GLS".

Lastly, you can control the lag length or bandwidth used for your spectral estimator. If you select one of the kernel estimation methods (Bartlett, Parzen, Quadratic Spectral), the dialog will give you a choice between using Newey-West or Andrews automatic bandwidth selection methods, or providing a user specified bandwidth. If, instead, you choose one of the AR spectral density estimation methods (AR Spectral - OLS, AR Spectral - OLS detrended, AR Spectral - GLS detrended), the dialog will prompt you to choose from various automatic lag length selection methods (using information criteria) or to provide a

user specified lag length. See "Automatic Bandwidth and Lag Length Selection" on page 340.

Once you have chosen the appropriate settings for your test, click on the OK button. EViews reports the test statistic along with output from the corresponding test regression. For these tests, EViews reports the uncorrected estimate of the residual variance and the estimate of the frequency zero spectrum f_0 (labeled as the "HAC corrected variance") in addition to the basic output. Running a PP test using the TBILL series yields:

Null Hypothesis: TBILL has a unit root
Exogenous: Constant
Bandwidth: 3.82 (Andrews using Bartlett kernel)

		Adj. t-Stat	Prob.*
Phillips-Perron test statistic		-1.519035	0.5223
Test critical values:	1% level	-3.459888	
	5% level	-2.874435	
	10% level	-2.573719	

*MacKinnon (1996) one-sided p-values.

Residual variance (no correction)	0.141589
HAC corrected variance (Bartlett kernel)	0.107615

As with the ADF test, we fail to reject the null hypothesis of a unit root in the TBILL series at conventional significance levels.

Note that your test output will differ somewhat for alternative test specifications. For example, the KPSS output only provides the asymptotic critical values tabulated by KPSS:

Null Hypothesis: TBILL is stationary
Exogenous: Constant
Bandwidth: 11 (Newey-West Fixed using Bartlett kernel)

		LM-Stat.
Kwiatkowski-Phillips-Schmidt-Shin test statistic		1.537310
Asymptotic critical values*:	1% level	0.739000
	5% level	0.463000
	10% level	0.347000

*Kwiatkowski-Phillips-Schmidt-Shin (1992, Table 1)

Residual variance (no correction)	2.415089
HAC corrected variance (Bartlett kernel)	26.11028

Similarly, the NP test output will contain results for all four test statistics, along with the NP tabulated critical values.

A word of caution. You should note that the critical values reported by EViews are valid only for unit root tests of a data series, and will be invalid if the series is based on estimated values. For example, Engle and Granger (1987) proposed a two-step method to test

for cointegration. The test amounts to testing for a unit root in the residuals of a first stage regression. Since these residuals are estimates of the disturbance term, the asymptotic distribution of the test statistic differs from the one for ordinary series. The correct critical values for a subset of the tests may be found in Davidson and MacKinnon (1993, Table 20.2).

Basic Unit Root Theory

The following discussion outlines the basics features of unit root tests. By necessity, the discussion will be brief. Users who require detail should consult the original sources and standard references (see, for example, Davidson and MacKinnon, 1993, Chapter 20, Hamilton, 1994, Chapter 17, and Hayashi, 2000, Chapter 9).

Consider a simple AR(1) process:

$$y_t = \rho y_{t-1} + x_t'\delta + \epsilon_t, \tag{13.46}$$

where x_t are optional exogenous regressors which may consist of constant, or a constant and trend, ρ and δ are parameters to be estimated, and the ϵ_t are assumed to be white noise. If $|\rho| \geq 1$, y is a nonstationary series and the variance of y increases with time and approaches infinity. If $|\rho| < 1$, y is a (trend-)stationary series. Thus, the hypothesis of (trend-)stationarity can be evaluated by testing whether the absolute value of ρ is strictly less than one.

The unit root tests that EViews provides generally test the null hypothesis H_0: $\rho = 1$ against the one-sided alternative H_1: $\rho < 1$. In some cases, the null is tested against a point alternative. In contrast, the KPSS Lagrange Multiplier test evaluates the null of H_0: $\rho < 1$ against the alternative H_1: $\rho = 1$.

The Augmented Dickey-Fuller (ADF) Test

The standard DF test is carried out by estimating Equation (13.46) after subtracting y_{t-1} from both sides of the equation:

$$\Delta y_t = \alpha y_{t-1} + x_t'\delta + \epsilon_t, \tag{13.47}$$

where $\alpha = \rho - 1$. The null and alternative hypotheses may be written as

$$\begin{aligned} H_0&: \alpha = 0 \\ H_1&: \alpha < 0 \end{aligned} \tag{13.48}$$

and evaluated using the conventional t-ratio for α:

$$t_\alpha = \hat{\alpha}/(se(\hat{\alpha})) \tag{13.49}$$

where $\hat{\alpha}$ is the estimate of α, and $se(\hat{\alpha})$ is the coefficient standard error.

Dickey and Fuller (1979) show that under the null hypothesis of a unit root, this statistic does not follow the conventional Student's t-distribution, and they derive asymptotic

results and simulate critical values for various test and sample sizes. More recently, MacKinnon (1991, 1996) implements a much larger set of simulations than those tabulated by Dickey and Fuller. In addition, MacKinnon estimates response surfaces for the simulation results, permitting the calculation of Dickey-Fuller critical values and p-values for arbitrary sample sizes. The more recent MacKinnon critical value calculations are used by EViews in constructing test output.

The simple Dickey-Fuller unit root test described above is valid only if the series is an AR(1) process. If the series is correlated at higher order lags, the assumption of white noise disturbances ϵ_t is violated. The Augmented Dickey-Fuller (ADF) test constructs a parametric correction for higher-order correlation by assuming that the y series follows an AR(p) process and adding p lagged difference terms of the dependent variable y to the right-hand side of the test regression:

$$\Delta y_t = \alpha y_{t-1} + x_t'\delta + \beta_1 \Delta y_{t-1} + \beta_2 \Delta y_{t-2} + \ldots + \beta_p \Delta y_{t-p} + v_t. \qquad (13.50)$$

This augmented specification is then used to test (13.48) using the t-ratio (13.49). An important result obtained by Fuller is that the asymptotic distribution of the t-ratio for α is independent of the number of lagged first differences included in the ADF regression. Moreover, while the assumption that y follows an autoregressive (AR) process may seem restrictive, Said and Dickey (1984) demonstrate that the ADF test is asymptotically valid in the presence of a moving average (MA) component, provided that sufficient lagged difference terms are included in the test regression.

You will face two practical issues in performing an ADF test. First, you must choose whether to include exogenous variables in the test regression. You have the choice of including a constant, a constant and a linear time trend, or neither, in the test regression. One approach would be to run the test with both a constant and a linear trend since the other two cases are just special cases of this more general specification. However, including irrelevant regressors in the regression will reduce the power of the test to reject the null of a unit root. The standard recommendation is to choose a specification that is a plausible description of the data under both the null and alternative hypotheses. See, Hamilton (1994a, p. 501) for discussion.

Second, you will have to specify the number of lagged difference terms (which we will term the "lag length") to be added to the test regression (0 yields the standard DF test; integers greater than 0 correspond to ADF tests). The usual (though not particularly useful) advice is to include a number of lags sufficient to remove serial correlation in the residuals. EViews provides both automatic and manual lag length selection options. For details, see "Automatic Bandwidth and Lag Length Selection" beginning on page 340.

Dickey-Fuller Test with GLS Detrending (DFGLS)

As noted above, you may elect to include a constant, or a constant and a linear time trend, in your ADF test regression. For these two cases, ERS (1996) propose a simple modification

of the ADF tests in which the data are detrended so that explanatory variables are "taken out" of the data prior to running the test regression.

ERS define a quasi-difference of y_t that depends on the value a representing the specific point alternative against which we wish to test the null:

$$d(y_t|a) = \begin{cases} y_t & \text{if } t = 1 \\ y_t - a y_{t-1} & \text{if } t > 1 \end{cases} \tag{13.51}$$

Next, consider an OLS regression of the quasi-differenced data $d(y_t|a)$ on the quasi-differenced $d(x_t|a)$:

$$d(y_t|a) = d(x_t|a)'\delta(a) + \eta_t \tag{13.52}$$

where x_t contains either a constant, or a constant and trend, and let $\hat{\delta}(a)$ be the OLS estimates from this regression.

All that we need now is a value for a. ERS recommend the use of $a = \bar{a}$, where

$$\bar{a} = \begin{cases} 1 - 7/T & \text{if } x_t = \{1\} \\ 1 - 13.5/T & \text{if } x_t = \{1, t\} \end{cases} \tag{13.53}$$

We now define the *GLS detrended data*, y_t^d using the estimates associated with the $\bar{a}$:

$$y_t^d \equiv y_t - x_t'\hat{\delta}(\bar{a}) \tag{13.54}$$

Then the DFGLS test involves estimating the standard ADF test equation, (13.50), after substituting the GLS detrended y_t^d for the original y_t:

$$\Delta y_t^d = \alpha y_{t-1}^d + \beta_1 \Delta y_{t-1}^d + \ldots + \beta_p y_{t-p}^d + v_t \tag{13.55}$$

Note that since the y_t^d are detrended, we do not include the x_t in the DFGLS test equation. As with the ADF test, we consider the t-ratio for $\hat{\alpha}$ from this test equation.

While the DFGLS t-ratio follows a Dickey-Fuller distribution in the constant only case, the asymptotic distribution differs when you include both a constant and trend. ERS (1996, Table 1, p. 825) simulate the critical values of the test statistic in this latter setting for $T = \{50, 100, 200, \infty\}$. Thus, the EViews lower tail critical values use the MacKinnon simulations for the no constant case, but are interpolated from the ERS simulated values for the constant and trend case. The null hypothesis is rejected for values that fall below these critical values.

The Phillips-Perron (PP) Test

Phillips and Perron (1988) propose an alternative (nonparametric) method of controlling for serial correlation when testing for a unit root. The PP method estimates the non-augmented DF test equation (13.47), and modifies the t-ratio of the α coefficient so that

serial correlation does not affect the asymptotic distribution of the test statistic. The PP test is based on the statistic:

$$\tilde{t}_\alpha = t_\alpha\left(\frac{\gamma_0}{f_0}\right)^{1/2} - \frac{T(f_0-\gamma_0)(se(\hat{\alpha}))}{2f_0^{1/2}s} \tag{13.56}$$

where $\hat{\alpha}$ is the estimate, and t_α the t-ratio of α, $se(\hat{\alpha})$ is coefficient standard error, and s is the standard error of the test regression. In addition, γ_0 is a consistent estimate of the error variance in (13.47) (calculated as $(T-k)s^2/T$, where k is the number of regressors). The remaining term, f_0, is an estimator of the residual spectrum at frequency zero.

There are two choices you will have make when performing the PP test. First, you must choose whether to include a constant, a constant and a linear time trend, or neither, in the test regression. Second, you will have to choose a method for estimating f_0. EViews supports estimators for f_0 based on kernel-based sum-of-covariances, or on autoregressive spectral density estimation. See "Frequency Zero Spectrum Estimation" beginning on page 338 for details.

The asymptotic distribution of the PP modified t-ratio is the same as that of the ADF statistic. EViews reports MacKinnon lower-tail critical and p-values for this test.

The Kwiatkowski, Phillips, Schmidt, and Shin (KPSS) Test

The KPSS (1992) test differs from the other unit root tests described here in that the series y_t is assumed to be (trend-) stationary under the null. The KPSS statistic is based on the the residuals from the OLS regression of y_t on the exogenous variables x_t:

$$y_t = x_t'\delta + u_t \tag{13.57}$$

The LM statistic is be defined as:

$$LM = \sum_t S(t)^2/(T^2 f_0) \tag{13.58}$$

where f_0, is an estimator of the residual spectrum at frequency zero and where $S(t)$ is a cumulative residual function:

$$S(t) = \sum_{r=1}^{t} \hat{u}_r \tag{13.59}$$

based on the residuals $\hat{u}_t = y_t - x_t'\hat{\delta}(0)$. We point out that the estimator of δ used in this calculation differs from the estimator for δ used by GLS detrending since it is based on a regression involving the original data, and not on the quasi-differenced data.

To specify the KPSS test, you must specify the set of exogenous regressors x_t and a method for estimating f_0. See "Frequency Zero Spectrum Estimation" on page 338 for discussion.

The reported critical values for the LM test statistic are based upon the asymptotic results presented in KPSS (Table 1, p. 166).

Elliot, Rothenberg, and Stock Point Optimal (ERS) Test

The ERS Point Optimal test is based on the quasi-differencing regression defined in Equations (13.52). Define the residuals from (13.52) as $\hat{\eta}_t(a) = d(y_t|a) - d(x_t|a)'\hat{\delta}(a)$, and let $\mathrm{SSR}(a) = \sum \hat{\eta}_t^2(a)$ be the sum-of-squared residuals function. The ERS (feasible) point optimal test statistic of the null that $\alpha = 1$ against the alternative that $\alpha = \bar{a}$, is then defined as

$$P_T = (\mathrm{SSR}(\bar{a}) - \bar{a}\mathrm{SSR}(1))/f_0 \tag{13.60}$$

where f_0, is an estimator of the residual spectrum at frequency zero.

To compute the ERS test you must specify the set of exogenous regressors x_t and a method for estimating f_0 (see "Frequency Zero Spectrum Estimation" on page 338).

Critical values for the ERS test statistic are computed by interpolating the simulation results provided by ERS (1996, Table 1, p. 825) for $T = \{50, 100, 200, \infty\}$.

Ng and Perron (NP) Tests

Ng and Perron (2001) construct four test statistics that are based upon the GLS detrended data y_t^d. These test statistics are modified forms of Phillips and Perron Z_α and Z_t statistics, the Bhargava (1986) R_1 statistic, and the ERS Point Optimal statistic. First, define the term:

$$\kappa = \sum_{t=2}^{T} (y_{t-1}^d)^2 / T^2 \tag{13.61}$$

The modified statistics may then be written as

$$\begin{aligned}
MZ_\alpha^d &= (T^{-1}(y_T^d)^2 - f_0)/(2\kappa) \\
MZ_t^d &= MZ_\alpha \times MSB \\
MSB^d &= (\kappa/f_0)^{1/2} \\
MP_T^d &= \begin{cases} (\bar{c}^2\kappa - \bar{c}T^{-1}(y_T^d)^2)/f_0 & \text{if } x_t = \{1\} \\ (\bar{c}^2\kappa + (1-\bar{c})T^{-1}(y_T^d)^2)/f_0 & \text{if } x_t = \{1, t\} \end{cases}
\end{aligned} \tag{13.62}$$

where

$$\bar{c} = \begin{cases} -7 & \text{if } x_t = \{1\} \\ -13.5 & \text{if } x_t = \{1, t\} \end{cases} \tag{13.63}$$

The NP tests require a specification for x_t and a choice of method for estimating f_0 (see

Frequency Zero Spectrum Estimation

Many of the unit root tests described above require a consistent estimate of the residual spectrum at frequency zero. EViews supports two classes of estimators for f_0: kernel-based sum-of-covariances estimators, and autoregressive spectral density estimators.

Kernel Sum-of-Covariances Estimation

The kernel-based estimator of the frequency zero spectrum is based on a weighted sum of the autocovariances, with the weights are defined by a kernel function. The estimator takes the form

$$\hat{f}_0 = \sum_{j=-(T-1)}^{T-1} \hat{\gamma}(j) \cdot K(j/l) \tag{13.64}$$

where l is a bandwidth parameter (which acts as a truncation lag in the covariance weighting), K is a kernel function, and where $\hat{\gamma}(j)$, the j-th sample autocovariance of the residuals $\hat{u}_t$, is defined as

$$\hat{\gamma}(j) = \sum_{t=j+1}^{T} (\hat{u}_t \hat{u}_{t-j})/T \tag{13.65}$$

Note that the residuals $\hat{u}_t$ that EViews uses in estimating the autocovariance functions in (13.65) will differ depending on the specified unit root test:

Unit root test	Source of $\hat{u}_t$ residuals for kernel estimator
ADF, DFGLS	*not applicable.*
PP, ERS Point Optimal, NP	residuals from the Dickey-Fuller test equation, (13.47).
KPSS	residuals from the OLS test equation, (13.57).

EViews supports the following kernel functions:

Bartlett:	$K(x) = \begin{cases} 1-\lvert x\rvert & \text{if } \lvert x\rvert \le 1 \\ 0 & \text{otherwise} \end{cases}$
Parzen:	$K(x) = \begin{cases} 1-6x^2+6\lvert x\rvert^3 & \text{if } 0 \le \lvert x\rvert \le (1/2) \\ 2(1-\lvert x\rvert)^3 & \text{if } (1/2) < \lvert x\rvert \le 1 \\ 0 & \text{otherwise} \end{cases}$
Quadratic Spectral	$K(x) = \frac{25}{12\pi^2 x^2}\left(\frac{\sin(6\pi x/5)}{6\pi x/5} - \cos(6\pi x/5)\right)$

The properties of these kernels are described in Andrews (1991).

As with most kernel estimators, the choice of the bandwidth parameter l is of considerable importance. EViews allows you to specify a fixed parameter, or to have EViews select one using a data-dependent method. Automatic bandwidth parameter selection is discussed in "Automatic Bandwidth and Lag Length Selection" beginning on page 340.

Autoregressive Spectral Density Estimator

The autoregressive spectral density estimator at frequency zero is based upon the residual variance and estimated coefficients from the auxiliary regression:

$$\Delta\tilde{y}_t = \alpha\tilde{y}_{t-1} + \varphi \cdot \tilde{x}_t'\delta + \beta_1\Delta\tilde{y}_{t-1} + \ldots + \beta_p\Delta\tilde{y}_{t-p} + u_t \tag{13.66}$$

EViews provides three autoregressive spectral methods: OLS, OLS detrending, and GLS detrending, corresponding to difference choices for the data $\tilde{y}_t$. The following table summarizes the auxiliary equation estimated by the various AR spectral density estimators:

AR spectral method	Auxiliary AR regression specification
OLS	$\tilde{y}_t = y_t$, and $\varphi = 1$, $\tilde{x}_t = x_t$.
OLS detrended	$\tilde{y}_t = y_t - x_t'\hat{\delta}(0)$, and $\varphi = 0$.
GLS detrended	$\tilde{y}_t = y_t - x_t'\hat{\delta}(\bar{a}) = y_t^d$. and $\varphi = 0$.

where $\hat{\delta}(a)$ are the coefficient estimates from the regression defined in (13.52).

The AR spectral estimator of the frequency zero spectrum is defined as:

$$\hat{f}_0 = \hat{\sigma}_u^2/(1-\hat{\beta}_1-\hat{\beta}_2-\ldots-\hat{\beta}_p) \tag{13.67}$$

where $\hat{\sigma}_u^2 = \sum \hat{u}_t^2/T$ is the residual variance, and $\hat{\beta}$ are the estimates from (13.66). We note here that EViews uses the non-degree-of-freedom estimator of the residual variance. As a result, spectral estimates comuted in EViews may differ slightly from those obtained from other sources.

Not surprisingly, the spectrum estimator is sensitive to the number of lagged difference terms in the auxiliary equation. You may either specify a fixed parameter, or to have EViews automatically select one based on an information criterion. Automatic lag length selection is examined in "Automatic Bandwidth and Lag Length Selection" on page 340.

Default Settings

By default, EViews will choose the estimator of f_0 used by the authors of a given test specification. You may, of course, override the default settings and choose from either family of estimation methods. The default settings are listed below:

Unit root test	Frequency zero spectrum default method
ADF, DFGLS	*not applicable*
PP, KPSS	Kernel (Bartlett) sum-of-covariances
ERS Point Optimal	AR spectral regression (OLS)
NP	AR spectral regression (GLS-detrended)

Automatic Bandwidth and Lag Length Selection

There are three distinct situations in which EViews can automatically compute a bandwidth or a lag length parameter.

The first situation occurs when you are selecting the bandwidth parameter l for the kernel-based estimators of f_0. For the kernel estimators, EViews provides you with the option of using the Newey-West (1994) or the Andrews (1991) data-based automatic bandwidth parameter methods. See the original sources for details. For those familiar with the Newey-West procedure, we note that EViews uses the lag selection parameter formulae given in the corresponding first lines of Table II-C. The Andrews method is based on an AR(1) specification.

The latter two occur when the unit root test requires estimation of a regression with a parametric correction for serial correlation as in the ADF and DFGLS test equation regressions, and in the AR spectral estimator for f_0. In all of these cases, p lagged difference terms are added to a regression equation. The automatic selection methods choose p (less than the specified maximum) to minimize one of the following criteria:

Information criterion	Definition
Akaike (AIC)	$-2(l/T) + 2k/T$
Schwarz (SIC)	$-2(l/T) + k\log(T)/T$
Hannan-Quinn (HQ)	$-2(l/T) + 2k\log(\log(T))/T$
Modified AIC (MAIC)	$-2(l/T) + 2(k+\tau)/T$
Modified SIC (MSIC)	$-2(l/T) + (k+\tau)\log(T)/T$
Modified Hannan-Quinn (MHQ)	$-2(l/T) + 2(k+\tau)\log(\log(T))/T$

where the modification factor τ is computed as

$$\tau = \alpha^2 \sum_t \tilde{y}_{t-1}^2 / \hat{\sigma}_u^2 \tag{13.68}$$

for $\tilde{y}_t = y_t$, when computing the ADF test equation, and for $\tilde{y}_t$ as defined in "Autoregressive Spectral Density Estimator" on page 339, when estimating f_0. NP (2001) propose and examine the modified criteria, concluding with a recommendation of the MAIC.

For the information criterion selection methods, you must also specify an upper bound to the lag length. By default, EViews chooses a maximum lag of

$$k_{\max} = \text{int}(12(T/100)^{1/4}) \tag{13.69}$$

See Hayashi (2000, p. 594) for a discussion of the selection of this upper bound.

Commands

The command

```
equation eq_gdp.ls gdp c ar(1) ar(2) ma(1) ma(2)
```

fits an ARMA(2,2) model to the GDP series and stores the results in the equation object named EQ_GDP.

```
eq1.auto(4)
```

tests for serial correlation of up to order four in the residuals from equation EQ1.

```
eq1.correlogram(12)
```

displays the correlogram for the residuals in EQ1 up to lag 12.

```
equation eq2.ls gdp c pdl(m1,12,3)
```

fits a third degree polynomial to the coefficients of M1 up to twelve lags.

```
gdp.uroot(lag=4,const)
```

runs the ADF unit root test including a constant and four lags of first differences.

```
gdp.uroot(pp,trend,hac=bt,b=4.2)
```

runs the Phillips-Perron unit root test including a constant and linear trend with a Bartlett kernel and bandwidth of 4.2.

Chapter 14. Forecasting from an Equation

This chapter describes procedures for forecasting and computing fitted values from a single equation. The techniques described here are for forecasting with equation objects estimated using regression methods. Forecasts from equations estimated by specialized techniques, such as ARCH, binary, ordered, tobit, and count methods, are discussed in the corresponding chapters. Forecasting from a series using exponential smoothing methods is explained in "Exponential Smoothing" on page 190, and forecasting using multiple equations and models is described in Chapter 23, "Models", on page 601.

Forecasting from Equations in EViews

To illustrate the process of forecasting from an estimated equation, we begin with a simple example. Suppose we have data on the logarithm of monthly housing starts (HS) and the logarithm of the S&P index (SP) over the period 1959:01–1996:01. The data are contained in a workfile with range 1959:01–1998:12.

We estimate a regression of HS on a constant, SP, and the lag of HS, with an AR(1) to correct for residual serial correlation, using data for the period 1959:01–1990:12, and then use the model to forecast housing starts under a variety of settings. Following estimation, the equation results are held in the equation object EQ01:

Dependent Variable: HS
Method: Least Squares
Date: 10/18/97 Time: 12:44
Sample(adjusted): 1959:03 1990:01
Included observations: 371 after adjusting endpoints
Convergence achieved after 4 iterations

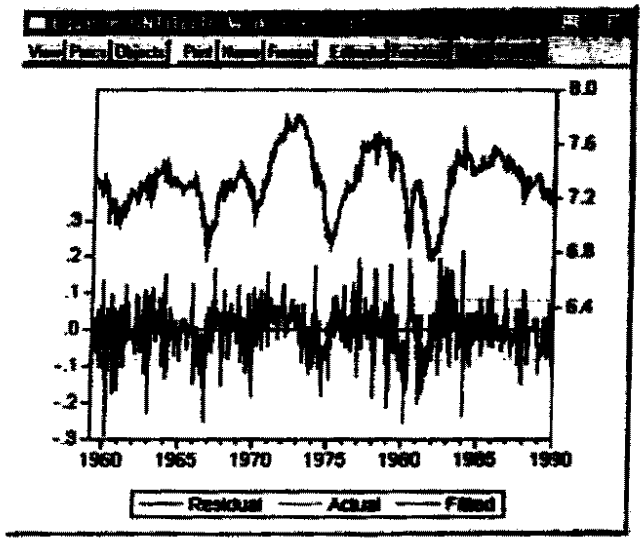
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.321924	0.117278	2.744975	0.0063
HS(-1)	0.952653	0.016218	58.74157	0.0000
SP	0.005222	0.007588	0.688249	0.4917
AR(1)	-0.271254	0.052114	-5.205027	0.0000

R-squared	0.861373	Mean dependent var	7.324051
Adjusted R-squared	0.860240	S.D. dependent var	0.220996
S.E. of regression	0.082618	Akaike info criterion	-2.138453
Sum squared resid	2.505050	Schwarz criterion	-2.096230
Log likelihood	400.6830	F-statistic	760.1338
Durbin-Watson stat	2.013460	Prob(F-statistic)	0.000000

Inverted AR Roots	-.27

Note that the estimation sample is adjusted by two observations to account for the first difference of the lagged endogenous variable used in deriving AR(1) estimates for this model.

To get a feel for the fit of the model, select **View/Actual, Fitted, Residual...**, then choose **Actual, Fitted, Residual Graph**:

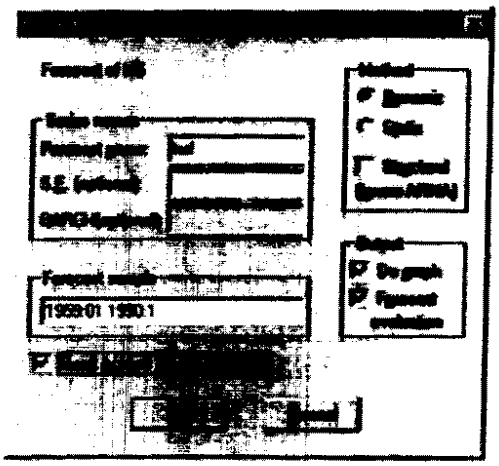


The actual and fitted values depicted on the upper portion of the graph are virtually indistinguishable. This view provides little control over the process of producing fitted values, and does not allow you to save your fitted values. These limitations are overcome by using EViews built-in forecasting procedures to compute fitted values for the dependent variable.

How to Perform a Forecast

To forecast HS from this equation, push the **Forecast** button on the equation toolbar, or select **Procs/Forecast...**

You should provide the following information:



- **Series names.**
- **Forecasted series.** Fill in the edit box with the name to be given to the forecasted dependent variable. EViews suggests a name, but you can change it to any valid series name. The name should be different from the name of the dependent variable, since the forecast procedure will overwrite the data in the specified series.
- **S.E. (optional).** If desired, you may provide a name for the series to be filled with the forecast standard errors. If you do not provide a name, no forecast errors will be saved.

- **GARCH (optional)**. For models estimated by ARCH, you will be given a further option of saving forecasts of the conditional variances (GARCH terms). See Chapter 16 for a discussion of GARCH estimation.

- **Forecasting method**. You have a choice between the following methods:

  Dynamic—calculates *multi-step forecasts* starting from the first period in the forecast sample.

  Static—calculates a sequence of *one-step ahead forecasts*, using actual, rather than forecasted values for lagged dependent variables.

  and you can set the following options:

  1. **Structural**—instructs EViews to ignore any ARMA terms in the equation when forecasting. By default, when your equation has ARMA terms, both dynamic and static solution methods form forecasts of the residuals. If you select **Structural**, all forecasts will ignore the residuals and will form predictions using only the structural part of the model.

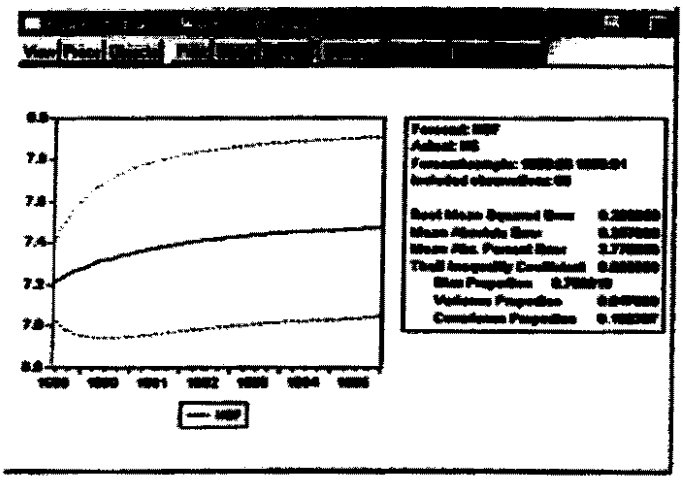
  2. **Sample range**. You must specify the sample to be used for the forecast. By default, EViews sets this sample to be the workfile sample. By specifying a sample outside the sample used in estimating your equation (the *estimation sample*), you can instruct EViews to produce out-of-sample forecasts.

  Note that you are responsible for supplying the values for the independent variables in the out-of-sample forecasting period. For static forecasts, you must also supply the values for any lagged dependent variables.

- **Output**. You can choose to see the forecast output as a graph or a numerical forecast evaluation, or both. Forecast evaluation is only available if the forecast sample includes observations for which the dependent variable is observed.

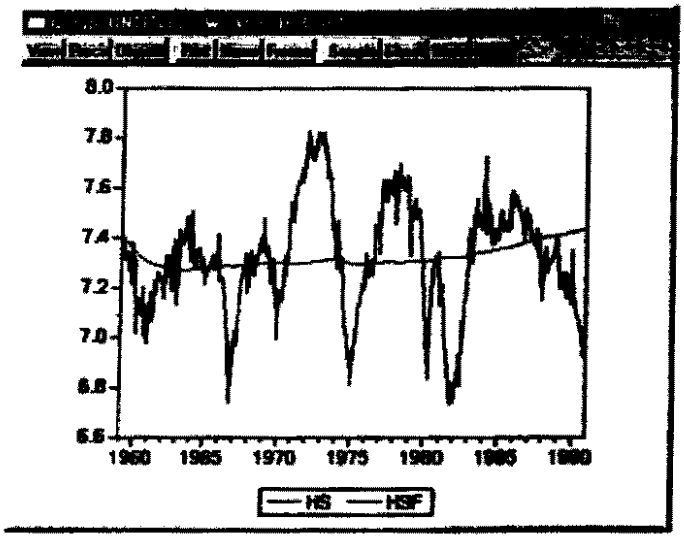
Illustration

Suppose we produce a dynamic forecast using EQ01 over the sample 1959:01 to 1996:01. The forecast values will be placed in the series HSF, and EViews will display both a graph of the forecasts and the plus and minus two standard error bands, as well as a forecast evaluation:



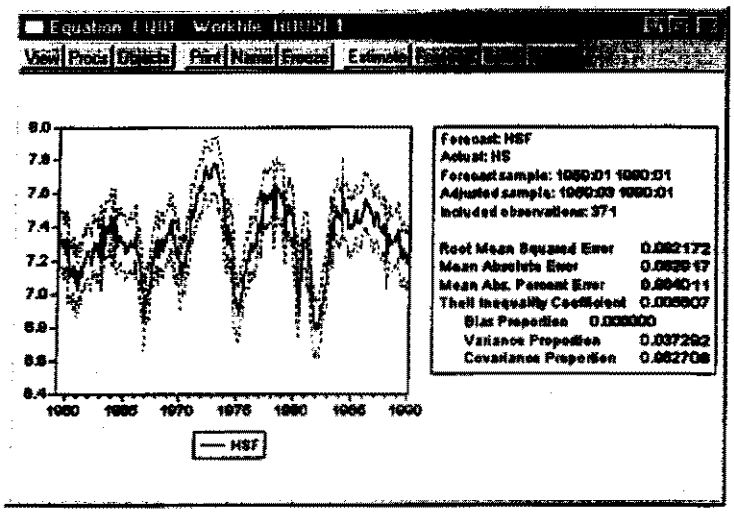
As noted in the output, the forecast values are saved in the series HSF. Since HSF is a standard EViews series, you may examine your forecasts using all of the standard tools for working with series objects.

We can examine the actual versus fitted values by creating a group containing HS and HSF, and plotting the two series. Click on **Quick/Show...** and enter HS and HSF. Then select **View/Graph/Line** to display the two series.

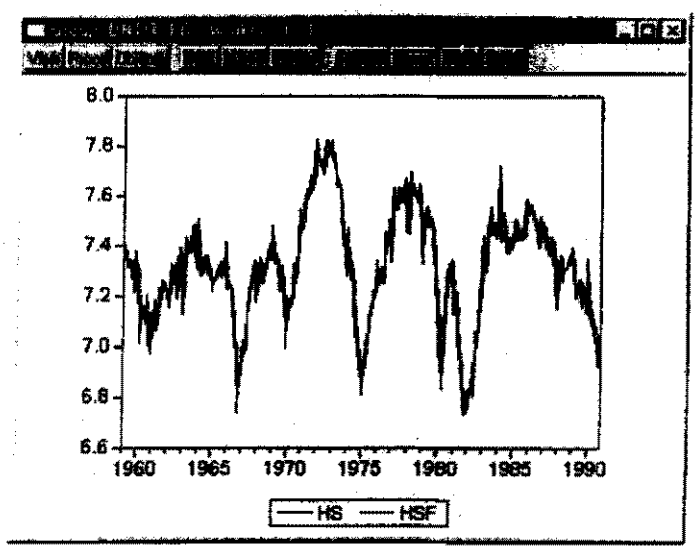


This is a dynamic forecast over the entire period from 1959:01 through 1996:01. For every period, the previously forecasted values for HS(-1) are used in forming a forecast of the subsequent value of HS. Note the considerable difference between this actual and fitted graph and the **Actual, Fitted, Residual Graph** depicted above.

To perform a series of one-step ahead forecasts, click on **Forecast** on the equation toolbar, and select **Static** forecasts. EViews will display the forecast results:

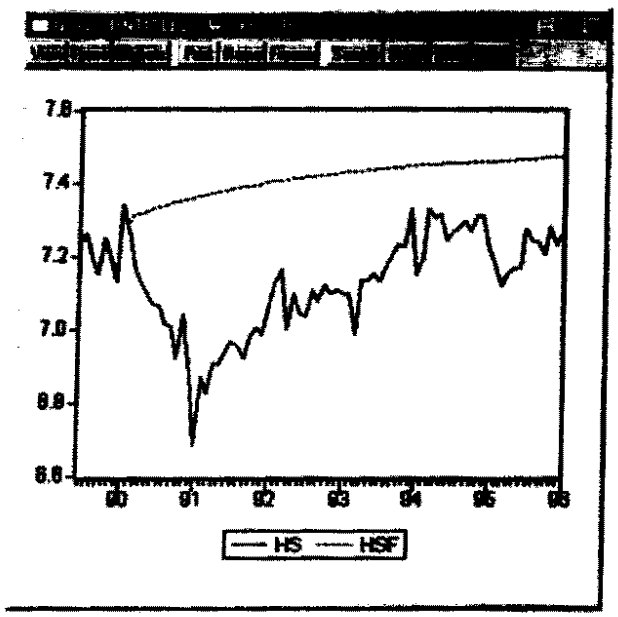


We can also compare the actual and fitted values from the static forecast by examining a line graph of a group containing HS and HSF.



The one-step ahead static forecasts are more accurate than the dynamic forecasts since, for each period, the actual value of HS(-1) is used in forming the forecast of HS. These one-step ahead static forecasts are the same forecasts used in the **Actual, Fitted, Residual Graph** displayed above.

Lastly, we construct a dynamic forecast beginning in 1990:02 (the first period following the estimation sample) and ending in 1996:01. Keep in mind that data are available for SP for this entire period. The plot of the actual and the forecast values for 1989:01 to 1996:01 is given by:

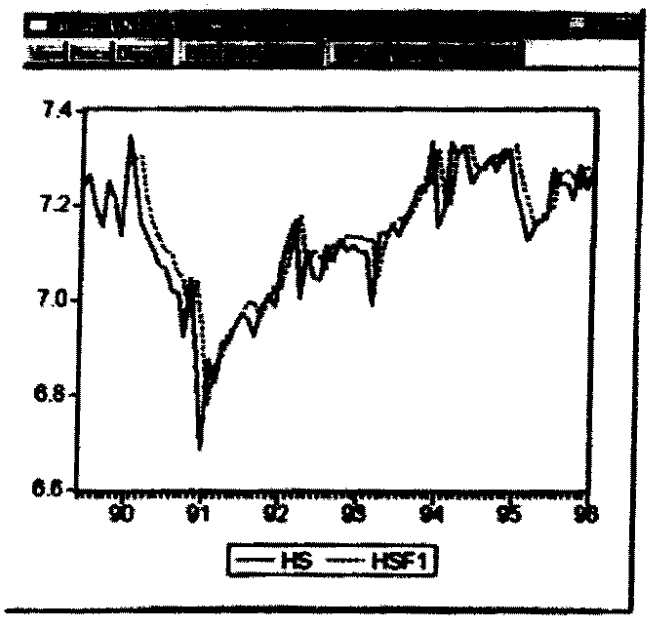


EViews backfills the forecast series prior to the forecast sample (up through 1990:01) and then dynamically forecasts HS for each subsequent period through 1996:01. This is the forecast that you would have constructed if, in 1990:01, you predicted values of HS from 1990:02 through 1996:01, given knowledge about the entire path of SP over that period.

The corresponding static forecast is displayed below:

Again, EViews backfills the values of the forecast series, HSF1, through 1990:01. This forecast is the one you would have constructed if, in 1990:01, you used all available data to estimate a model, and then used this estimated model to perform one-step ahead forecasts every month for the next six years.

The remainder of this chapter focuses on the details associated with the construction of these forecasts, and the corresponding forecast evaluations.



Forecast Basics

EViews stores the forecast results in the series specified in the **Forecast name** field. We will refer to this series as the *forecast series*.

The *forecast sample* specifies the observations for which EViews will try to compute fitted or forecasted values. If the forecast is not computable, a missing value will be returned. In some cases, EViews will carry out automatic adjustment of the sample to prevent a forecast consisting entirely of missing values (see "Adjustment for Missing Values" on page 350, below). Note that the forecast sample may or may not overlap with the sample of observations used to estimate the equation.

For values not included in the forecast sample, there are two options. By default, EViews fills in the actual values of the dependent variable. If you turn off the **Insert actuals for out-of-sample** option, out-of-forecast-sample values will be filled with NAs.

As a consequence of these rules, *all data in the forecast series will be overwritten during the forecast procedure*. Existing values in the forecast series will be lost.

Computing Forecasts

For each observation in the forecast sample, EViews computes the fitted value of the dependent variable using the estimated parameters, the right-hand side exogenous variables, and, either the actual or estimated values for lagged endogenous variables and residuals. The method of constructing these forecasted values depends upon the estimated model and user specified settings.

To illustrate the forecasting procedure, we begin with a simple linear regression model with no lagged endogenous right-hand side variables, and no ARMA terms. Suppose that you have estimated the following equation specification:

```
y c x z
```

Now click on **Forecast**, specify a forecast period, and click **OK**.

For every observation in the forecast period, EViews will compute the fitted value of Y using the estimated parameters and the corresponding values of the regressors, X and Z:

$$\hat{y}_t = \hat{c}(1) + \hat{c}(2)x_t + \hat{c}(3)z_t. \tag{14.1}$$

You should make certain that you have valid values for the exogenous right-hand side variables for all observations in the forecast period. If any data are missing in the forecast sample, the corresponding forecast observation will be an NA.

Adjustment for Missing Values

There are two cases when a missing value will be returned for the forecast value. First, if any of the regressors have a missing value, and second, if any of the regressors are out of the range of the workfile. This includes the implicit error terms in AR models.

In the case of forecasts with no dynamic components in the specification (*i.e.* with no lagged endogenous or ARMA error terms), a missing value in the forecast series will not affect subsequent forecasted values. In the case where there are dynamic components, however, a single missing value in the forecasted series will propagate throughout all future values of the series.

As a convenience feature, EViews will move the starting point of the sample forward where necessary, until a valid forecast value is obtained. Without these adjustments, the user would have to figure out the appropriate number of presample values to skip, otherwise the forecast would consist entirely of missing values. For example, suppose you wanted to forecast dynamically from the following equation specification:

```
y c y(-1) ar(1)
```

If you specified the beginning of the forecast sample to the beginning of the workfile range, EViews will adjust forward the forecast sample by 2 observations, and will use the pre-forecast-sample values of the lagged variables (the loss of 2 observations occurs because the residual loses one observation due to the lagged endogenous variable so that the forecast for the error term can begin only from the third observation.)

Forecast Errors and Variances

Suppose the "true" model is given by:

$$y_t = x_t'\beta + \epsilon_t, \tag{14.2}$$

where ϵ_t is an independent, and identically distributed, mean zero random disturbance, and β is a vector of unknown parameters. Below, we relax the restriction that the ϵ's be independent.

The true model generating y is not known, but we obtain estimates b of the unknown parameters β. Then, setting the error term equal to its mean value of zero, the (point) forecasts of y are obtained as

$$\hat{y}_t = x_t'b. \tag{14.3}$$

Forecasts are made with error, where the error is simply the difference between the actual and forecasted value $e_t = y_t - x_t'b$. Assuming that the model is correctly specified, there are two sources of forecast error: residual uncertainty and coefficient uncertainty.

Residual Uncertainty

The first source of error, termed *residual* or *innovation uncertainty*, arises because the innovations ϵ in the equation are unknown for the forecast period, and are replaced with their expectations. While the residuals are zero in expected value, the individual values are non-zero; the larger the variation in the individual errors, the greater the overall error in the forecasts.

The standard measure of this variation is the standard error of the regression (labeled "S.E. of regression" in the equation output). Residual uncertainty is usually the largest source of forecast error.

In dynamic forecasts, innovation uncertainty is compounded by the fact that lagged dependent variables and ARMA terms depend on lagged innovations. EViews also sets these equal to their expected values, which differ randomly from realized values. This additional source of forecast uncertainty tends to rise over the forecast horizon, leading to a pattern of increasing forecast errors. Forecasting with lagged dependent variables and ARMA terms is discussed in more detail below.

Coefficient Uncertainty

The second source of forecast error is *coefficient uncertainty*. The estimated coefficients b of the equation deviate from the true coefficients β in a random fashion. The standard error of the estimated coefficient, given in the regression output, is a measure of the precision with which the estimated coefficients measure the true coefficients.

The effect of coefficient uncertainty depends upon the exogenous variables. Since the estimated coefficients are multiplied by the exogenous variables x in the computation of forecasts, the more the exogenous variables deviate from their mean values, the greater is the forecast uncertainty.

Forecast Variability

The variability of forecasts is measured by the forecast standard errors. For a single equation without lagged dependent variables or ARMA terms, the forecast standard errors are computed as

$$\text{forecast se} = s\sqrt{1 + x_t'(X'X)^{-1}x_t} \tag{14.4}$$

where s is the standard error of regression. These standard errors account for both innovation (the first term) and coefficient uncertainty (the second term). Point forecasts made from linear regression models estimated by least squares are optimal in the sense that they have the smallest forecast variance among forecasts made by linear unbiased estimators. Moreover, if the innovations are normally distributed, the forecast errors have a t-distribution and forecast intervals can be readily formed.

If you supply a name for the forecast standard errors, EViews computes and saves a series of forecast standard errors in your workfile. You can use these standard errors to form forecast intervals. If you choose the **Do graph** option for output, EViews will plot the forecasts with plus and minus two standard error bands. These two standard error bands provide an approximate 95% forecast interval; if you (hypothetically) make many forecasts, the actual value of the dependent variable will fall inside these bounds 95 percent of the time.

Additional Details

EViews accounts for the additional forecast uncertainty generated when lagged dependent variables are used as explanatory variables (see "Forecasts with Lagged Dependent Variables" on page 355).

There are several other special cases, involving dependent variables that are defined by expression, where coefficient uncertainty is ignored. These cases are described in "Forecasting Equations with Formulas" on page 359.

Forecast standard errors derived from equations estimated by nonlinear least squares and equations that include PDL (polynomial distributed lag) terms only account for the residual uncertainty ("Forecasting with Nonlinear and PDL Specifications" on page 364).

Forecast Evaluation

Suppose we construct a dynamic forecast for HS over the period 1990:02 to 1996:01 using our estimated housing equation. If the **Forecast evaluation** option is checked, and there are actual data for the forecasted variable for the forecast sample, EViews reports a table of statistical results evaluating the forecast:

Forecast: HSF
Actual: HS
Sample: 1990:02 1996:01
Include observations: 72

Root Mean Squared Error	0.318700
Mean Absolute Error	0.297261
Mean Absolute Percentage Error	4.205880
Theil Inequality Coefficient	0.021917
Bias Proportion	0.888882
Variance Proportion	0.082804
Covariance Proportion	0.047214

Note that EViews cannot compute a forecast evaluation if there are no data for the dependent variable for the forecast sample.

The forecast evaluation is saved in one of two formats. If you turn on the **Do graph** option, the forecasts are included along with a graph of the forecasts. If you wish to display the evaluations in their own table, you should turn off the **Do graph** option in the Forecast dialog box.

Suppose the forecast sample is $j = T+1, T+2, \ldots, T+h$, and denote the actual and forecasted value in period t as y_t and $\hat{y}_t$, respectively. The reported forecast error statistics are computed as follows:

Root Mean Squared Error	$\sqrt{\sum_{t=T+1}^{T+h} (\hat{y}_t - y_t)^2 / h}$
Mean Absolute Error	$\sum_{t=T+1}^{T+h} \lvert \hat{y}_t - y_t \rvert / h$
Mean Absolute Percentage Error	$100 \sum_{t=T+1}^{T+h} \left\lvert \frac{\hat{y}_t - y_t}{y_t} \right\rvert / h$
Theil Inequality Coefficient	$\frac{\sqrt{\sum_{t=T+1}^{T+h} (\hat{y}_t - y_t)^2 / h}}{\sqrt{\sum_{t=T+1}^{T+h} \hat{y}_t^2 / h} + \sqrt{\sum_{t=T+1}^{T+h} y_t^2 / h}}$

The first two forecast error statistics depend on the scale of the dependent variable. These should be used as relative measures to compare forecasts for the same series across different models; the smaller the error, the better the forecasting ability of that model according to that criterion. The remaining two statistics are scale invariant. The Theil inequality coefficient always lies between zero and one, where zero indicates a perfect fit.

The mean squared forecast error can be decomposed as

$$\sum (\hat{y}_t - y_t)^2 / h = ((\sum \hat{y}_t / h) - \bar{y})^2 + (s_{\hat{y}} - s_y)^2 + 2(1-r) s_{\hat{y}} s_y \tag{14.5}$$

where $\sum \hat{y}_t / h$, $\bar{y}$, $s_{\hat{y}}$, s_y are the means and (biased) standard deviations of $\hat{y}_t$ and y, and r is the correlation between $\hat{y}$ and y. The proportions are defined as:

Bias Proportion	$\frac{((\sum \hat{y}_t / h) - \bar{y})^2}{\sum (\hat{y}_t - y_t)^2 / h}$
Variance Proportion	$\frac{(s_{\hat{y}} - s_y)^2}{\sum (\hat{y}_t - y_t)^2 / h}$
Covariance Proportion	$\frac{2(1-r) s_{\hat{y}} s_y}{\sum (\hat{y}_t - y_t)^2 / h}$

- The bias proportion tells us how far the mean of the forecast is from the mean of the actual series.
- The variance proportion tells us how far the variation of the forecast is from the variation of the actual series.
- The covariance proportion measures the remaining unsystematic forecasting errors.

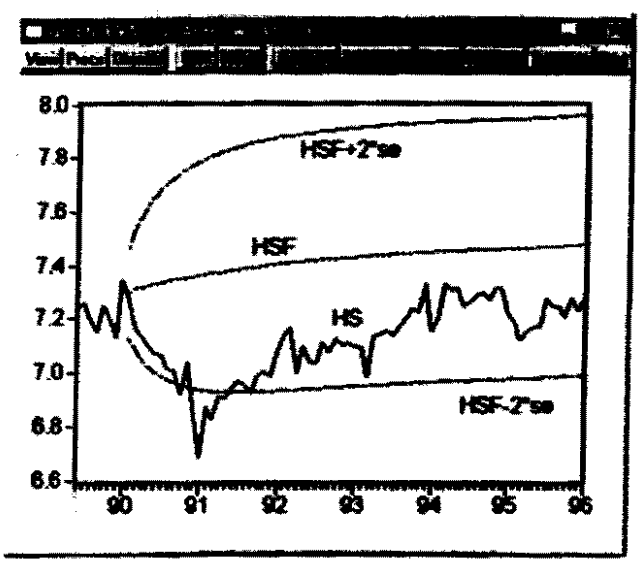
Note that the bias, variance, and covariance proportions add up to one.

If your forecast is "good", the bias and variance proportions should be small so that most of the bias should be concentrated on the covariance proportions. For additional discussion of forecast evaluation, see Pindyck and Rubinfeld (1991, Chapter 12).

For the example output, the bias proportion is large, indicating that the mean of the forecasts does a poor job of tracking the mean of the dependent variable. To check this, we will plot the forecasted series together with the actual series in the forecast sample with the two standard error bounds. Suppose we saved the forecasts and their standard errors as HSF and HSFSE, respectively. Then the plus and minus two standard error series can be generated by the commands

```
smpl 1990:02 1996:01
series hsf_high = hsf + 2*hsfse
series hsf_low = hsf - 2*hsfse
```

Create a group containing the four series. You can highlight the four series HS, HSF, HSF_HIGH, and HSF_LOW, double click on the selected area, and select **Open Group**, or you can select **Quick/Show...** and enter the four series names. Once you have the group open, select **View/Graph/Line**.



The forecasts completely miss the downturn at the start of the 1990's, but, subsequent to the recovery, track the trend reasonably well from 1992 to 1996.

Forecasts with Lagged Dependent Variables

Forecasting is complicated by the presence of lagged dependent variables on the right-hand side of the equation. For example, we can augment the earlier specification to include the first lag of Y:

```
y c x z y(-1)
```

and click on the **Forecast** button and fill out the series names in the dialog as above. There is some question, however, as to how we should evaluate the lagged value of Y that appears on the right-hand side of the equation. There are two possibilities: dynamic forecasting and static forecasting.

Dynamic Forecasting

If you select dynamic forecasting, EViews will perform a multi-step forecast of Y, beginning at the start of the forecast sample. For our single lag specification above:

- The initial observation in the forecast sample will use the actual value of lagged Y. Thus, if S is the first observation in the forecast sample, EViews will compute

$$\hat{y}_S = \hat{c}(1) + \hat{c}(2)x_S + \hat{c}(3)z_S + \hat{c}(4)y_{S-1}, \tag{14.6}$$

  where y_{S-1} is the value of the lagged endogenous variable in the period prior to the start of the forecast sample. This is the one-step ahead forecast.

- Forecasts for subsequent observations will use the previously *forecasted* values of Y:

$$\hat{y}_{S+k} = \hat{c}(1) + \hat{c}(2)x_{S+k} + \hat{c}(3)z_{S+k} + \hat{c}(4)\hat{y}_{S+k-1}. \tag{14.7}$$

- These forecasts may differ significantly from the one-step ahead forecasts.

If there are additional lags of Y in the estimating equation, the above algorithm is modified to account for the non-availability of lagged forecasted values in the additional period. For example, if there are three lags of Y in the equation:

- The first observation (S) uses the actual values for all three lags, y_{S-3}, y_{S-2}, and y_{S-1}.
- The second observation ($S+1$) uses actual values for y_{S-2} and, y_{S-1} and the forecasted value $\hat{y}_S$ of the first lag of y_{S+1}.
- The third observation ($S+2$) will use the actual values for y_{S-1}, and forecasted values $\hat{y}_{S+1}$ and $\hat{y}_S$ for the first and second lags of y_{S+2}.
- All subsequent observations will use the forecasted values for all three lags.

The selection of the start of the forecast sample is very important for dynamic forecasting. The dynamic forecasts are true multi-step forecasts (from the start of the forecast sample), since they use the recursively computed forecast of the lagged value of the dependent vari-

able. These forecasts may be interpreted as the forecasts for subsequent periods that would be computed using information available at the start of the forecast sample.

Dynamic forecasting requires that data for the exogenous variables be available for every observation in the forecast sample, and that values for any lagged dependent variables be observed at the start of the forecast sample (in our example, y_{S-1}, but more generally, any lags of y). If necessary, the forecast sample will be adjusted.

Any missing values for the explanatory variables will generate an NA for that observation and in all subsequent observations, via the dynamic forecasts of the lagged dependent variable.

Static Forecasting

Static forecasting performs a series of one-step ahead forecasts of the dependent variable:

- For each observation in the forecast sample, EViews computes

$$\hat{y}_{S+k} = \hat{c}(1) + \hat{c}(2)x_{S+k} + \hat{c}(3)z_{S+k} + \hat{c}(4)y_{S+k-1} \tag{14.8}$$

  always using the actual value of the lagged endogenous variable.

Static forecasting requires that data for both the exogenous and any lagged endogenous variables be observed for every observation in the forecast sample. As above, EViews will, if necessary, adjust the forecast sample to account for pre-sample lagged variables. If the data are not available for any period, the forecasted value for that observation will be an NA. The presence of a forecasted value of NA does not have any impact on forecasts for subsequent observations.

A Comparison of Dynamic and Static Forecasting

Both methods will always yield identical results in the first period of a multi-period forecast. Thus, two forecast series, one dynamic and the other static, should be identical for the first observation in the forecast sample.

The two methods will differ for subsequent periods only if there are lagged dependent variables or ARMA terms.

Forecasting with ARMA Errors

Forecasting from equations with ARMA components involves some additional complexities. When you use the AR or MA specifications, you will need to be aware of how EViews handles the forecasts of the lagged residuals which are used in forecasting.

Structural Forecasts

By default, EViews will forecast values for the residuals using the estimated ARMA structure, as described below.

For some types of work, you may wish to assume that the ARMA errors are always zero. If you select the structural forecast option by checking **Structural (ignore ARMA)**, EViews computes the forecasts assuming that the errors are always zero. If the equation is estimated without ARMA terms, this option has no effect on the forecasts.

Forecasting with AR Errors

For equations with AR errors, EViews adds forecasts of the residuals from the equation to the forecast of the structural model that is based on the right-hand side variables.

In order to compute an estimate of the residual, EViews requires estimates or actual values of the lagged residuals. For the first observation in the forecast sample, EViews will use pre-sample data to compute the lagged residuals. If the pre-sample data needed to compute the lagged residuals are not available, EViews will adjust the forecast sample, and backfill the forecast series with actual values (see the discussion of "Adjustment for Missing Values" on page 350).

If you choose the **Dynamic** option, both the lagged dependent variable and the lagged residuals will be forecasted dynamically. If you select **Static**, both will be set to the actual lagged values. For example, consider the following AR(2) model:

$$\begin{aligned} y_t &= x_t'\beta + u_t \\ u_t &= \rho_1 u_{t-1} + \rho_2 u_{t-2} + \epsilon_t \end{aligned} \tag{14.9}$$

Denote the fitted residuals as $e_t = y_t - x_t'b$, and suppose the model was estimated using data up to $t = S-1$. Then, provided that the x_t values are available, the static and dynamic forecasts for $t = S, S+1, \ldots$, are given by:

	static	dynamic
$\hat{y}_S$	$x_S'b + \hat{\rho}_1 e_{S-1} + \hat{\rho}_2 e_{S-2}$	$x_S'b + \hat{\rho}_1 e_{S-1} + \hat{\rho}_2 e_{S-2}$
$\hat{y}_{S+1}$	$x_{S+1}'b + \hat{\rho}_1 e_S + \hat{\rho}_2 e_{S-1}$	$x_{S+1}'b + \hat{\rho}_1 \hat{u}_S + \hat{\rho}_2 e_{S-1}$
$\hat{y}_{S+2}$	$x_{S+2}'b + \hat{\rho}_1 e_{S+1} + \hat{\rho}_2 e_S$	$x_{S+2}'b + \hat{\rho}_1 \hat{u}_{S+1} + \hat{\rho}_2 \hat{u}_S$

where the residuals $\hat{u}_t = \hat{y}_t - x_t'b$ are formed using the forecasted values of y_t. For subsequent observations, the dynamic forecast will always use the residuals based upon the multi-step forecasts, while the static forecast will use the one-step ahead forecast residuals.

Forecasting with MA Errors

In general, you need not concern yourselves with the details of MA forecasting, since EViews will do all of the work for you. For those of you who are interested in the details of dynamic forecasting, however, the following discussion should aid you in relating EViews results with those obtained from other sources.

The first step in computing forecasts using MA terms is to obtain fitted values for the innovations in the pre-forecast sample period. For example, if you are forecasting the values of y, beginning in period S, with a simple MA(q),

$$\hat{y}_S = \epsilon_S + \hat{\phi}_1\epsilon_{S-1} + \ldots + \hat{\phi}_q\epsilon_{S-q}, \tag{14.10}$$

you will need values for the lagged innovations, $\epsilon_{S-1}, \epsilon_{S-2}, \ldots, \epsilon_{S-q}$.

To compute these pre-forecast innovations, EViews will first assign values for the q innovations prior to the start of the *estimation* sample, $\epsilon_0, \epsilon_{-1}, \epsilon_{-2}, \ldots, \epsilon_{-q}$. If your equation is estimated with backcasting turned on, EViews will perform backcasting to obtain these values. If your equation is estimated with backcasting turned off, or if the forecast sample precedes the estimation sample, the initial values will be set to zero.

Given the initial values, EViews will fit the values of subsequent innovations, $\epsilon_1, \epsilon_2, \ldots, \epsilon_q, \ldots, \epsilon_{S-1}$, using forward recursion. The backcasting and recursion procedures are described in detail in the discussion of backcasting in ARMA models in "Backcasting MA terms" on page 320.

Note the difference between this procedure and the approach for AR errors outlined above, in which the forecast sample is adjusted forward and the pre-forecast values are set to actual values.

The choice between dynamic and static forecasting has two primary implications:

- Once the q pre-sample values for the innovations are computed, dynamic forecasting sets subsequent innovations to zero. Static forecasting extends the forward recursion through the end of the estimation sample, allowing for a series of one-step ahead forecasts of both the structural model and the innovations.
- When computing static forecasts, EViews uses the entire estimation sample to backcast the innovations. For dynamic MA forecasting, the backcasting procedure uses observations from the beginning of the estimation sample to either the beginning of the forecast period, or the end of the estimation sample, whichever comes first.

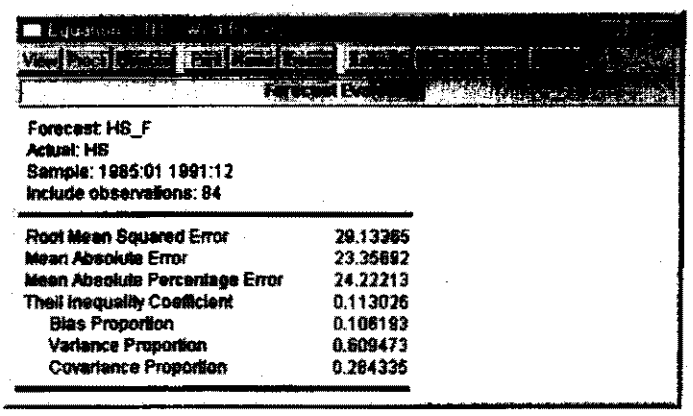
Example

As an example of forecasting from ARMA models, consider forecasting the monthly new housing starts (HS) series. The estimation period is 1959:01–1984:12 and we forecast for

the period 1985:01–1991:12. We estimated the following simple multiplicative seasonal autoregressive model:

```
hs c ar(1) sar(12)
```

To forecast from this estimated model, click **Forecast** on the equation toolbar. The forecast evaluation statistics for the model are shown below:



Forecast: HS_F
Actual: HS
Sample: 1985:01 1991:12
Include observations: 84

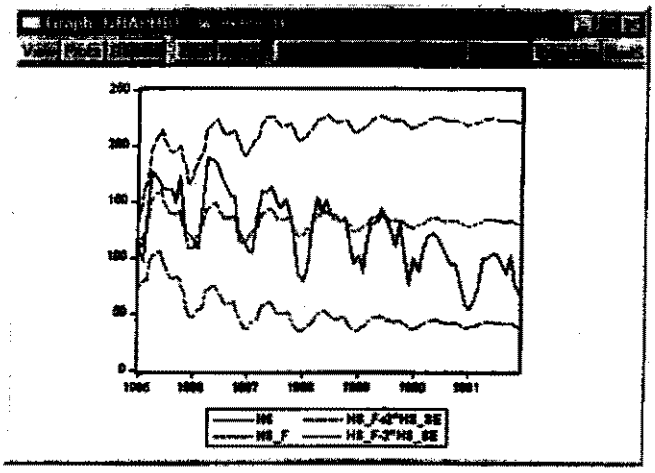
Root Mean Squared Error	29.13365
Mean Absolute Error	23.35692
Mean Absolute Percentage Error	24.22213
Theil Inequality Coefficient	0.113026
Bias Proportion	0.106193
Variance Proportion	0.609473
Covariance Proportion	0.284335

The large variance proportion indicates that the forecasts are not tracking the variation in the actual HS series. To plot the actual and forecasted series together with the two standard error bands, you can type

```
smpl 1985:01 1991:12
plot hs hs_f hs_f+2*hs_se hs_f-2*hs_se
```

where HS_F and HS_SE are the forecasts and standard errors of HS.

As indicated by the large variance proportion, the forecasts track the seasonal movements in HS only at the beginning of the forecast sample and quickly flattens out to the mean forecast value.



Forecasting Equations with Formulas

EViews allows estimation and forecasting with equations where the left-hand variable is a transformation specified by a formula. When forecasting from equations with formulas on the

left-hand side, three things determine the forecasting procedures and options that are available:

- whether the formula is linear or nonlinear
- whether the formula includes lagged variables
- whether the formula includes estimated coefficients

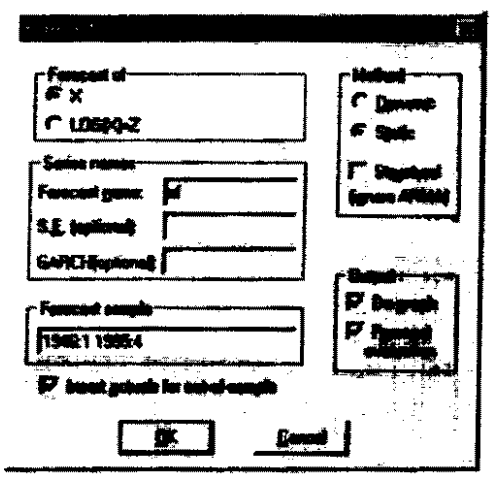
Point Forecasts

EViews always provides you with the option to forecast the transformed dependent variable. If the transformation can be normalized and solved for the first series in the formula, then EViews also provides you with the option to forecast the normalized series.

For example, suppose you estimated an equation with the specification

```
(log(x)+z) c y
```

If you press the Forecast button, the Forecast dialog looks like this:

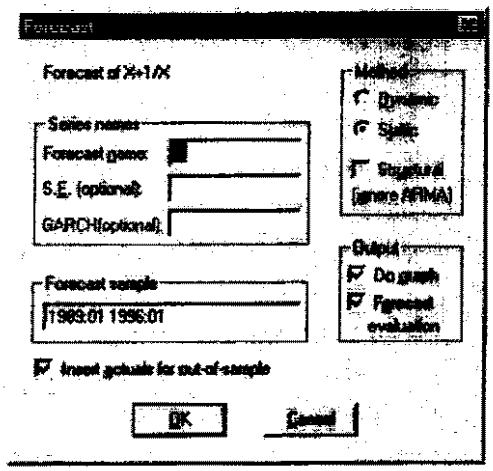


Notice that the dialog provides you with two choices for the series to forecast: the normalized series, X, and the dependent variable, LOG(X) + Z. X is the normalized series since it is the first series that appears on the left-hand side of the equation.

However, if you specify the equation as

```
x+1/x = c(1) + c(2)*y
```

EViews will not be able to normalize the dependent variable and the Forecast dialog looks like this:



The dialog only allows you to forecast the transformed dependent variable, since EViews does not know how to normalize and solve for X. Note also that only static forecasts are available for this case. This restriction holds since EViews will not be able to solve for any lagged values of X on the right hand-side.

If the formula can be normalized, EViews will compute the forecasts of the transformed dependent variable by transforming the forecasts of the normalized series. This has important consequences when the formula includes lagged series. For example, consider the following two models:

```
series dy = d(y)
equation eq1.ls d(y) c x
equation eq2.ls dy c x
```

The dynamic forecasts of the first difference D(Y) from the first equation will be numerically identical to those for DY from the second equation. However, the static forecasts for D(Y) from the two equations will not be identical. This is because in the first equation, EViews knows that the dependent variable is a transformation of Y, so it will use the actual lagged value of Y in computing the static forecast of the first difference D(Y). In the second equation, EViews simply views DY as an ordinary series, so that C and X are used to compute the static forecast.

Plotted Standard Errors

When you select **Do graph** in the forecast dialog, EViews will plot the forecasts, along with plus and minus two standard error bands. When you estimate an equation with an expression for the left-hand side, EViews will plot the standard error bands for either the normalized or the unnormalized expression, depending upon which term you elect to forecast.

If you elect to predict the normalized dependent variable, EViews will automatically account for the nonlinearity in the standard error transformation. The next section provides additional details on the procedure used to normalize the upper and lower bounds.

Saved Forecast Standard Errors

If you provide a name in this edit box, EViews will store the standard errors of the underlying series or expression that you chose to forecast.

When the dependent variable of the equation is a simple series or a formula involving only *linear* transformations, the saved standard errors will be exact (except where the forecasts do not account for coefficient uncertainty, as described below). If the dependent variable involves nonlinear transformations, the saved forecast standard errors will be exact if you choose to forecast the entire formula. If you choose to forecast the underlying endogenous series, the forecast uncertainty cannot be computed exactly, and EViews will provide a linear (first-order) approximation to the forecast standard errors.

Consider the following equations involving a formula dependent variable:

```
d(y) c x
log(y) c x
```

For the first equation you may choose to forecast either Y or D(Y). In both cases, the forecast standard errors will be exact, since the expression involves only linear transformations. The two standard errors will, however, differ in dynamic forecasts since the forecast standard errors for Y take into account the forecast uncertainty from the lagged value of Y. In the second example, the forecast standard errors for log(Y) will be exact. If, however, you request a forecast for Y itself, the standard errors saved in the series will be the approximate (linearized) forecast standard errors for Y.

Note that when EViews displays a graph view of the forecasts together with standard error bands, the standard error bands are always exact. Thus, in forecasting the underlying dependent variable in a nonlinear expression, the standard error bands will not be the same as those you would obtain by constructing series using the linearized standard errors saved in the workfile.

Suppose in our second example above that you store the forecast of Y and its standard errors in the workfile as the series YHAT and SE_YHAT. Then the *approximate* two standard error bounds can be generated manually as:

```
series yhat_high1 = yhat + 2*se_yhat
series yhat_low1 = yhat - 2*se_yhat
```

These forecast error bounds will be symmetric about the point forecasts YHAT.

On the other hand, when EViews plots the forecast error bounds of Y, it proceeds in two steps. It first obtains the forecast of log(Y) and its standard errors (say LYHAT and SE_LYHAT) and forms the forecast error bounds on log(Y):

```
lyhat + 2*se_lyhat
lyhat - 2*se_lyhat
```

It then normalizes (inverts the transformation) of the two standard error bounds to obtain the prediction interval for Y:

```
series yhat_high2 = exp(lyhat + 2*se_lyhat)
series yhat_low2 = exp(lyhat - 2*se_lyhat)
```

Because this transformation is a non-linear transformation, these bands will not be symmetric around the forecast.

To take a more complicated example, suppose that you generate the series DLY and LY, and then estimate three equivalent models:

```
series dly =dlog(y)
series ly = log(y)
equation eq1.ls dlog(y) c x
equation eq2.ls d(ly) c x
equation eq3.ls dly c x
```

The estimated equations from the three models are numerically identical. If you choose to forecast the underlying dependent (normalized) series from each model, EQ1 will forecast Y, EQ2 will forecast LY (the log of Y), and EQ3 will forecast DLY (the log of the first difference of Y, log(Y)-logY(-1)). The forecast standard errors saved from EQ1 will be linearized approximations to the forecast standard error of Y, while those from the latter two will be exact for the forecast standard error of log Y and the log of the first difference of Y.

Static forecasts from all three models are identical because the forecasts from previous periods are not used in calculating this period's forecast when performing static forecasts. For dynamic forecasts, the log of the forecasts from EQ1 will be identical to those from EQ2 and the log first difference of the forecasts from EQ1 will be identical to the first difference of the forecasts from EQ2 and to the forecasts from EQ3. For static forecasts, the log first difference of the forecasts from EQ1 will be identical to the first difference of the forecasts from EQ2. However, these forecasts differ from those obtained from EQ3 because EViews does not know that the generated series DLY is actually a difference term so that it does not use the dynamic relation in the forecasts.

A final word of caution: when you have lagged dependent variables, you should avoid referring to the lagged series before the current series in a dependent variable expression. For example, consider the two equation specifications:

```
d(y) c x
-y(-1)+y c x
```

Both models have the first difference of Y as the dependent variable and the estimation results are identical for the two models. However, if you forecast Y from the second model, EViews will try to calculate the forecasts of Y using leads of the actual series Y. These forecasts of Y will differ from those produced by the first model, and may not be what you expected.

Forecasting with Nonlinear and PDL Specifications

As explained above, forecast errors can arise from two sources: coefficient uncertainty and innovation uncertainty. For linear regression models, the forecast standard errors account for both coefficient and innovation uncertainty. However, if the model is nonlinear in the parameters (or if it contains a PDL specification), then the standard errors ignore coefficient uncertainty. EViews will display a message in the status line at the bottom of the EViews window when forecast standard errors only account for innovation uncertainty.

For example, consider the three specifications

```
log(y) c x
y = c(1) + c(2)*x
y = exp(c(1)*x)
y c x pdl(z, 4, 2)
```

Forecast standard errors from the first and second models account for both coefficient and innovation uncertainty since both models are linear in the coefficients. The third and fourth specifications have forecast standard errors that account only for residual uncertainty.

One additional case requires mention. Suppose you have the specification:

```
y-c(1) = c(3) + c(2)*x
```

Despite the fact that this specification is linear in the parameters, EViews will ignore coefficient uncertainty. Forecast standard errors for any specification that contains coefficients on the left-hand side of the equality will only reflect residual uncertainty.

Commands

To obtain static (one-step ahead) forecasts, follow the name of the estimated equation, a dot, the command `fit`, a name for the fitted series, and optionally a name for the standard errors of the fitted values:

```
eq1.fit yhat yhat_se
```

To obtain dynamic forecasts, follow the name of the estimated equation, a period, the command `forecast`, a name for the forecasts, and optionally a name for the standard errors of the forecasts:

```
eq1.forecast yh yh_se
```

See the *Command and Programming Reference* for a complete list of commands and options available for forecasting.

Chapter 15. Specification and Diagnostic Tests

Empirical research is usually an interactive process. The process begins with a specification of the relationship to be estimated. Selecting a specification usually involves several choices: the variables to be included, the functional form connecting these variables, and if the data are time series, the dynamic structure of the relationship between the variables.

Inevitably, there is uncertainty regarding the appropriateness of this initial specification. Once you estimate your equation, EViews provides tools for evaluating the quality of your specification along a number of dimensions. In turn, the results of these tests influence the chosen specification, and the process is repeated.

This chapter describes the extensive menu of specification test statistics that are available as views or procedures of an equation object. While we attempt to provide you with sufficient statistical background to conduct the tests, practical considerations ensure that many of the descriptions are incomplete. We refer you to standard statistical and econometric references for further details.

Background

Each test procedure described below involves the specification of a null hypothesis, which is the hypothesis under test. Output from a test command consists of the sample values of one or more test statistics and their associated probability numbers (*p*-values). The latter indicate the probability of obtaining a test statistic whose absolute value is greater than or equal to that of the sample statistic if the null hypothesis is true. Thus, low *p*-values lead to the rejection of the null hypothesis. For example, if a *p*-value lies between 0.05 and 0.01, the null hypothesis is rejected at the 5 percent but not at the 1 percent level.

Bear in mind that there are different assumptions and distributional results associated with each test. For example, some of the test statistics have exact, finite sample distributions (usually t or F-distributions). Others are large sample test statistics with asymptotic χ^2 distributions. Details vary from one test to another and are given below in the description of each test.

Types of Tests

The **View** button on the equation toolbar gives you a choice among three categories of tests to check the specification of the equation.

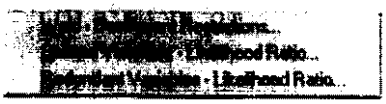
Additional tests are discussed elsewhere in the *User's Guide*. These tests include unit root tests ("Performing Unit Root Tests in EViews" on page 329), the Granger causality test ("Granger Causality" on

page 222), tests specific to binary, order, censored, and count models (Chapter 17, "Discrete and Limited Dependent Variable Models", on page 421), and the Johansen test for cointegration ("How to Perform a Cointegration Test" on page 538).

Coefficient Tests

These tests evaluate restrictions on the estimated coefficients, including the special case of tests for omitted and redundant variables.

Wald Test (Coefficient Restrictions)



The Wald test computes a test statistic based on the unrestricted regression. The Wald statistic measures how close the unrestricted estimates come to satisfying the restrictions under the null hypothesis. If the restrictions are in fact true, then the unrestricted estimates should come close to satisfying the restrictions.

How to Perform Wald Coefficient Tests

To demonstrate the calculation of Wald tests in EViews, we consider simple examples. Suppose a Cobb-Douglas production function has been estimated in the form:

$$\log Q = A + \alpha \log L + \beta \log K + \epsilon, \tag{15.1}$$

where Q, K and L denote value-added output and the inputs of capital and labor respectively. The hypothesis of constant returns to scale is then tested by the restriction: $\alpha + \beta = 1$.

Estimation of the Cobb-Douglas production function using annual data from 1947 to 1971 provided the following result:

Dependent Variable: LOG(Q)
Method: Least Squares
Date: 08/11/97 Time: 16:56
Sample: 1947 1971
Included observations: 25

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-2.327939	0.410601	-5.669595	0.0000
LOG(L)	1.591175	0.167740	9.485970	0.0000
LOG(K)	0.239604	0.105390	2.273498	0.0331

R-squared	0.983672	Mean dependent var	4.767586
Adjusted R-squared	0.982187	S.D. dependent var	0.326086
S.E. of regression	0.043521	Akaike info criterion	-3.318997
Sum squared resid	0.041669	Schwarz criterion	-3.172732
Log likelihood	44.48746	F-statistic	662.6619
Durbin-Watson stat	0.637300	Prob(F-statistic)	0.000000

The sum of the coefficients on LOG(L) and LOG(K) appears to be in excess of one, but to determine whether the difference is statistically relevant, we will conduct the hypothesis test of constant returns.

To carry out a Wald test, choose **View/Coefficient Tests/Wald-Coefficient Restrictions...** from the equation toolbar. Enter the restrictions into the edit box, with multiple coefficient restrictions separated by commas. The restrictions should be expressed as equations involving the estimated coefficients and constants (you may not include series names). The coefficients should be referred to as C(1), C(2), and so on, unless you have used a different coefficient vector in estimation.

To test the hypothesis of constant returns to scale, type the following restriction in the dialog box:

```
c(2) + c(3) = 1
```

and click **OK**. EViews reports the following result of the Wald test:

Wald Test:
Equation: EQ1

Test Statistic	Value	df	Probability
Chi-square	120.0177	1	0.0000
F-statistic	120.0177	(1, 22)	0.0000

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
-1 + C(2) + C(3)	0.830779	0.075834

Restrictions are linear in coefficients.

EViews reports an F-statistic and a Chi-square statistic with associated p-values. See "Wald Test Details" on page 371 for a discussion of these statistics. In addition, EViews reports the value of the normalized (homogeneous) restriction and an associated standard error. In this example, we have a single linear restriction so the two test statistics are identical, with the p-value indicating that we can decisively reject the null hypothesis of constant returns to scale.

To test more than one restriction, separate the restrictions by commas. For example, to test the hypothesis that the elasticity of output with respect to labor is 2/3 and the elasticity with respect to capital is 1/3, enter the restrictions as

```
c(2)=2/3, c(3)=1/3
```

and EViews reports

Wald Test:
Equation: EQ1

Test Statistic	Value	df	Probability
Chi-square	53.99105	2	0.0000
F-statistic	26.99553	(2, 22)	0.0000

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
-2/3 + C(2)	0.924508	0.167740
-1/3 + C(1)	-2.661272	0.410601

Restrictions are linear in coefficients.

Note that in addition to the test statistic summary, we report the values of both of the normalized restrictions, along with their standard errors (the square roots of the diagonal elements of the restriction covariance matrix).

As an example of a nonlinear model with a nonlinear restriction, we estimate a production function of the form

$$\log Q = \beta_1 + \beta_2 \log(\beta_3 K^{\beta_4} + (1-\beta_3)L^{\beta_4}) + \epsilon \tag{15.2}$$

and test the constant elasticity of substitution (CES) production function restriction $\beta_2 = 1/\beta_4$. This is an example of a nonlinear restriction. To estimate the (unrestricted) nonlinear model, you should select **Quick/Estimate Equation...** and then enter the following specification:

```
log(q) = c(1) + c(2)*log(c(3)*k^c(4)+(1-c(3))*l^c(4))
```

To test the nonlinear restriction, choose **View/Coefficient Tests/Wald-Coefficient Restrictions...** from the equation toolbar and type the following restriction in the Wald Test dialog box:

```
c(2)=1/c(4)
```

The results are presented below:

Wald Test:
Equation: EQ2

Test Statistic	Value	df	Probability
Chi-square	0.028508	1	0.8659
F-statistic	0.028508	(1, 21)	0.8675

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
C(2) - 1/C(4)	1.292163	7.653088

Delta method computed using analytic derivatives.

Since this is a nonlinear equation, we focus on the Chi-square statistic which fails to reject the null hypothesis. Note that EViews reports that it used the delta method (with analytic derivatives) to compute the Wald restriction variance for the nonlinear restriction.

It is well-known that nonlinear Wald tests are not invariant to the way that you specify the nonlinear restrictions. In this example, the nonlinear restriction $\beta_2 = 1/\beta_4$ may equivalently be written as $\beta_2\beta_4 = 1$ or $\beta_4 = 1/\beta_2$ (for nonzero β_2 and β_4). For example, entering the restriction as

```
c(2)*c(4)=1
```

yields:

Wald Test:
Equation: EQ2

Test Statistic	Value	df	Probability
Chi-square	104.5599	1	0.0000
F-statistic	104.5599	(1, 21)	0.0000

Null Hypothesis Summary:

Normalized Restriction (= 0)	Value	Std. Err.
-1 + C(2)*C(4)	0.836330	0.081801

Delta method computed using analytic derivatives.

so that the test now decisively rejects the null hypothesis. We hasten to add that type of inconsistency is not unique to EViews, but is a more general property of the Wald test. Unfortunately, there does not seem to be a general solution to this problem (see Davidson and MacKinnon, 1993, Chapter 13).

Wald Test Details

Consider a general nonlinear regression model

$$y = f(\beta) + \epsilon \tag{15.3}$$

where y and ϵ are T-vectors and β is a k-vector of parameters to be estimated. Any restrictions on the parameters can be written as

$$H_0\colon\ g(\beta) = 0, \tag{15.4}$$

where g is a smooth function, $g\colon R^k \to R^q$, imposing q restrictions on β. The Wald statistic is then computed as

$$W = g(\beta)'\left(\frac{\partial g(\beta)}{\partial \beta} V(b) \frac{\partial g(\beta)}{\partial \beta'}\right) g(\beta)|_{\beta = b} \tag{15.5}$$

where T is the number of observations and b is the vector of unrestricted parameter estimates, and where $\hat{V}$ is an estimate of the b covariance. In the standard regression case, $\hat{V}$ is given by

$$\hat{V}(b) = s^2\left(\frac{\partial f(\beta)}{\partial \beta}\frac{\partial f(\beta)}{\partial \beta'}\right)^{-1}\Bigg|_{\beta = b} \tag{15.6}$$

where u is the vector of unrestricted residuals, and s^2 is the usual estimator of the unrestricted residual variance, $s^2 = (u'u)/(N-k)$, but the estimator of V may differ. For example, $\hat{V}$ may be a robust variance matrix estimator computing using White or Newey-West techniques.

More formally, under the null hypothesis H_0, the Wald statistic has an asymptotic $\chi^2(q)$ distribution, where q is the number of restrictions under H_0.

For the textbook case of a linear regression model

$$y = X\beta + \epsilon \tag{15.7}$$

and linear restrictions

$$H_0\colon\ R\beta - r = 0, \tag{15.8}$$

where R is a known $q \times k$ matrix, and r is a q-vector, respectively. The Wald statistic in Equation (15.5) reduces to

$$W = (Rb - r)'(Rs^2(X'X)^{-1}R')^{-1}(Rb - r), \tag{15.9}$$

which is asymptotically distributed as $\chi^2(q)$ under H_0.

If we further assume that the errors ϵ are independent and identically normally distributed, we have an exact, finite sample F-statistic:

$$F = \frac{W}{q} = \frac{(\tilde{u}'\tilde{u} - u'u)/q}{(u'u)/(T-k)}, \tag{15.10}$$

where $\tilde{u}$ is the vector of residuals from the restricted regression. In this case, the F-statistic compares the residual sum of squares computed with and without the restrictions imposed.

We remind you that the expression for the finite sample F-statistic in (15.10) is for standard linear regression, and is not valid for more general cases (nonlinear models, ARMA specifications, or equations where the variances are estimated using other methods such as Newey-West or White). In non-standard settings, the reported F-statistic (which EViews always computes computes as W/q), does not possess the desired finite-sample properties. In these cases, while asymptotically valid, the F-statistic results should be viewed as illustrative and for comparison purposes only.

Omitted Variables

This test enables you to add a set of variables to an existing equation and to ask whether the set makes a significant contribution to explaining the variation in the dependent variable. The null hypothesis H_0 is that the additional set of regressors are not jointly significant.

The output from the test is an F-statistic and a likelihood ratio (LR) statistic with associated p-values, together with the estimation results of the unrestricted model under the alternative. The F-statistic is based on the difference between the residual sums of squares of the restricted and unrestricted regressions and is only valid in linear regression based settings. The LR statistic is computed as

$$LR = -2(l_r - l_u) \tag{15.11}$$

where l_r and l_u are the maximized values of the (Gaussian) log likelihood function of the unrestricted and restricted regressions, respectively. Under H_0, the LR statistic has an asymptotic χ^2 distribution with degrees of freedom equal to the number of restrictions (the number of added variables).

Bear in mind that:

- The omitted variables test requires that the same number of observations exist in the original and test equations. If any of the series to be added contain missing observations over the sample of the original equation (which will often be the case when you add lagged variables), the test statistics cannot be constructed.
- The omitted variables test can be applied to equations estimated with linear LS, TSLS, ARCH (mean equation only), binary, ordered, censored, truncated, and count models. The test is available only if you specify the equation by listing the regressors, not by a formula.

To perform an LR test in these settings, you can estimate a separate equation for the unrestricted and restricted models over a common sample, and evaluate the LR statistic and p-value using scalars and the `@cchisq` function, as described above.

How to Perform an Omitted Variables Test

To test for omitted variables, select **View/Coefficient Tests/Omitted Variables-Likelihood Ratio...** In the dialog that opens, list the names of the test variables, each separated by at least one space. Suppose, for example, that the initial regression is

```
ls log(q) c log(l) log(k)
```

If you enter the list

```
log(m) log(e)
```

in the dialog, then EViews reports the results of the unrestricted regression containing the two additional explanatory variables, and displays statistics testing the hypothesis that the coefficients on the new variables are jointly zero. The top part of the output depicts the test results:

Omitted Variables: LOG(M) LOG(E)			
F-statistic	4.267478	Probability	0.028611
Log likelihood ratio	8.884940	Probability	0.011767

The F-statistic has an exact finite sample F-distribution under H_0 for linear models if the errors are independent and identically distributed normal random variables. The numerator degrees of freedom is the number of additional regressors and the denominator degrees of freedom is the number of observations less the total number of regressors. The log likelihood ratio statistic is the LR test statistic and is asymptotically distributed as a χ^2 with degrees of freedom equal to the number of added regressors.

In our example, the tests reject the null hypothesis that the two series do not belong to the equation at a 5% significance level, but cannot reject the hypothesis at a 1% significance level.

Redundant Variables

The redundant variables test allows you to test for the statistical significance of a subset of your included variables. More formally, the test is for whether a subset of variables in an equation all have zero coefficients and might thus be deleted from the equation. The redundant variables test can be applied to equations estimated by linear LS, TSLS, ARCH (mean equation only), binary, ordered, censored, truncated, and count methods. The test is available only if you specify the equation by listing the regressors, not by a formula.

How to Perform a Redundant Variables Test

To test for redundant variables, select **View/Coefficient Tests/Redundant Variables-Likelihood Ratio...** In the dialog that appears, list the names of each of the test variables, separated by at least one space. Suppose, for example, that the initial regression is

```
ls log(q) c log(l) log(k) log(m) log(e)
```

If you type the list

```
log(m) log(e)
```

in the dialog, then EViews reports the results of the restricted regression dropping the two regressors, followed by the statistics associated with the test of the hypothesis that the coefficients on the two variables are jointly zero.

The test statistics are the F-statistic and the Log likelihood ratio. The F-statistic has an exact finite sample F-distribution under H_0 if the errors are independent and identically distributed normal random variables and the model is linear. The numerator degrees of freedom are given by the number of coefficient restrictions in the null hypothesis. The denominator degrees of freedom are given by the total regression degrees of freedom. The LR test is an asymptotic test, distributed as a χ^2 with degrees of freedom equal to the number of excluded variables under H_0. In this case, there are two degrees of freedom.

Residual Tests

EViews provides tests for serial correlation, normality, heteroskedasticity, and autoregressive conditional heteroskedasticity in the residuals from your estimated equation. Not all of these tests are available for every specification.

Correlograms and Q-statistics

This view displays the autocorrelations and partial autocorrelations of the equation residuals up to the specified number of lags. Further details on these statistics and the Ljung-Box Q-statistics that are also computed are provided in Chapter 7, "Q-Statistics" on page 169.

This view is available for the residuals from least squares, two-stage least squares, nonlinear least squares and binary, ordered, censored, and count models. In calculating the probability values for the Q-statistics, the degrees of freedom are adjusted to account for estimated ARMA terms.

To display the correlograms and Q-statistics, push **View/Residual Tests/Correlogram-Q-statistics** on the equation toolbar. In the Lag Specification dialog box, specify the number of lags you wish to use in computing the correlogram.

Correlograms of Squared Residuals

This view displays the autocorrelations and partial autocorrelations of the squared residuals up to any specified number of lags and computes the Ljung-Box Q-statistics for the corresponding lags. The correlograms of the squared residuals can be used to check autoregressive conditional heteroskedasticity (ARCH) in the residuals; see also "ARCH LM Test" on page 377, below.

If there is no ARCH in the residuals, the autocorrelations and partial autocorrelations should be zero at all lags and the Q-statistics should not be significant; see Chapter 7, page 167, for a discussion of the correlograms and Q-statistics.

This view is available for equations estimated by least squares, two-stage least squares, and nonlinear least squares estimation. In calculating the probability for Q-statistics, the degrees of freedom are adjusted for the inclusion of ARMA terms.

To display the correlograms and Q-statistics of the squared residuals, push **View/Residual Tests/Correlogram Squared Residuals** on the equation toolbar. In the Lag Specification dialog box that opens, specify the number of lags over which to compute the correlograms.

Histogram and Normality Test

This view displays a histogram and descriptive statistics of the residuals, including the Jarque-Bera statistic for testing normality. If the residuals are normally distributed, the histogram should be bell-shaped and the Jarque-Bera statistic should not be significant; see Chapter 7, page 153, for a discussion of the Jarque-Bera test. This view is available for residuals from least squares, two-stage least squares, nonlinear least squares, and binary, ordered, censored, and count models.

To display the histogram and Jarque-Bera statistic, select **View/Residual Tests/Histogram-Normality**. The Jarque-Bera statistic has a χ^2 distribution with two degrees of freedom under the null hypothesis of normally distributed errors.

Serial Correlation LM Test

This test is an alternative to the Q-statistics for testing serial correlation. The test belongs to the class of asymptotic (large sample) tests known as Lagrange multiplier (LM) tests.

Unlike the Durbin-Watson statistic for AR(1) errors, the LM test may be used to test for higher order ARMA errors and is applicable whether or not there are lagged dependent variables. Therefore, we recommend its use (in preference to the DW statistic) whenever you are concerned with the possibility that your errors exhibit autocorrelation.

The null hypothesis of the LM test is that there is no serial correlation up to lag order p, where p is a pre-specified integer. The local alternative is ARMA(r, q) errors, where the number of lag terms $p = \max(r, q)$. Note that this alternative includes both AR(p) and MA(p) error processes, so that the test may have power against a variety of alternative autocorrelation structures. See Godfrey (1988), for further discussion.

The test statistic is computed by an auxiliary regression as follows. First, suppose you have estimated the regression

$$y_t = X_t\beta + \epsilon_t \tag{15.12}$$

where b are the estimated coefficients and ϵ are the errors. The test statistic for lag order p is based on the auxiliary regression for the residuals $e = y - X\hat{\beta}$:

$$e_t = X_t\gamma + \left(\sum_{s=1}^{p} \alpha_s e_{t-s}\right) + v_t. \tag{15.13}$$

Following the suggestion by Davidson and MacKinnon (1993), EViews sets any presample values of the residuals to 0. This approach does not affect the asymptotic distribution of the statistic, and Davidson and MacKinnon argue that doing so provides a test statistic which has better finite sample properties than an approach which drops the initial observations.

This is a regression of the residuals on the original regressors X and lagged residuals up to order p. EViews reports two test statistics from this test regression. The F-statistic is an omitted variable test for the joint significance of all lagged residuals. Because the omitted variables are residuals and not independent variables, the exact finite sample distribution of the F-statistic under H_0 is still not known, but we present the F-statistic for comparison purposes.

The Obs*R-squared statistic is the Breusch-Godfrey LM test statistic. This LM statistic is computed as the number of observations, times the (uncentered) R^2 from the test regression. Under quite general conditions, the LM test statistic is asymptotically distributed as a $\chi^2(p)$.

The serial correlation LM test is available for residuals from either least squares or two-stage least squares estimation. The original regression may include AR and MA terms, in which case the test regression will be modified to take account of the ARMA terms. Testing in 2SLS settings involves additional complications, see Wooldridge (1990) for details.

To carry out the test, push **View/Residual Tests/Serial Correlation LM Test...** on the equation toolbar and specify the highest order of the AR or MA process that might describe the serial correlation. If the test indicates serial correlation in the residuals, LS standard errors are invalid and should not be used for inference.

ARCH LM Test

This is a Lagrange multiplier (LM) test for autoregressive conditional heteroskedasticity (ARCH) in the residuals (Engle 1982). This particular specification of heteroskedasticity was motivated by the observation that in many financial time series, the magnitude of residuals appeared to be related to the magnitude of recent residuals. ARCH in itself does not invalidate standard LS inference. However, ignoring ARCH effects may result in loss of efficiency; see Chapter 16 for a discussion of estimation of ARCH models in EViews.

The ARCH LM test statistic is computed from an auxiliary test regression. To test the null hypothesis that there is no ARCH up to order q in the residuals, we run the regression

$$e_t^2 = \beta_0 + \left(\sum_{s=1}^{q} \beta_s e_{t-s}^2 \right) + v_t, \tag{15.14}$$

where e is the residual. This is a regression of the squared residuals on a constant and lagged squared residuals up to order q. EViews reports two test statistics from this test regression. The F-statistic is an omitted variable test for the joint significance of all lagged squared residuals. The Obs*R-squared statistic is Engle's LM test statistic, computed as the number of observations times the R^2 from the test regression. The exact finite sample distribution of the F-statistic under H_0 is not known but the LM test statistic is asymptotically distributed $\chi^2(q)$ under quite general conditions. The ARCH LM test is available for equations estimated by least squares, two-stage least squares, and nonlinear least squares.

To carry out the test, push **View/Residual Tests/ARCH LM Test...** on the equation toolbar and specify the order of ARCH to be tested against.

White's Heteroskedasticity Test

This is a test for heteroskedasticity in the residuals from a least squares regression (White, 1980). Ordinary least squares estimates are consistent in the presence heteroskedasticity, but the conventional computed standard errors are no longer valid. If you find evidence of heteroskedasticity, you should either choose the robust standard errors option to correct the standard errors (see "Heteroskedasticity Consistent Covariances (White)" on page 281) or you should model the heteroskedasticity to obtain more efficient estimates using weighted least squares.

White's test is a test of the null hypothesis of no heteroskedasticity against heteroskedasticity of some unknown general form. The test statistic is computed by an auxiliary regression, where we regress the squared residuals on all possible (nonredundant) cross products of the regressors. For example, suppose we estimated the following regression:

$$y_t = b_1 + b_2 x_t + b_3 z_t + e_t \tag{15.15}$$

where the b are the estimated parameters and e the residual. The test statistic is then based on the auxiliary regression:

$$e_t^2 = \alpha_0 + \alpha_1 x_t + \alpha_2 z_t + \alpha_3 x_t^2 + \alpha_4 z_t^2 + \alpha_5 x_t z_t + v_t. \tag{15.16}$$

EViews reports two test statistics from the test regression. The F-statistic is an omitted variable test for the joint significance of all cross products, excluding the constant. It is presented for comparison purposes.

The Obs*R-squared statistic is White's test statistic, computed as the number of observations times the centered R^2 from the test regression. The exact finite sample distribution of the F-statistic under H_0 is not known, but White's test statistic is asymptotically dis-

tributed as a χ^2 with degrees of freedom equal to the number of slope coefficients (excluding the constant) in the test regression.

White also describes this approach as a general test for model misspecification, since the null hypothesis underlying the test assumes that the errors are both homoskedastic and independent of the regressors, and that the linear specification of the model is correct. Failure of any one of these conditions could lead to a significant test statistic. Conversely, a non-significant test statistic implies that none of the three conditions is violated.

When there are redundant cross-products, EViews automatically drops them from the test regression. For example, the square of a dummy variable is the dummy variable itself, so that EViews drops the squared term to avoid perfect collinearity.

To carry out White's heteroskedasticity test, select **View/Residual Tests/White Heteroskedasticity**. EViews has two options for the test: cross terms and no cross terms. The cross terms version of the test is the original version of White's test that includes all of the cross product terms (in the example above, $x_t z_t$). However, with many right-hand side variables in the regression, the number of possible cross product terms becomes very large so that it may not be practical to include all of them. The no cross terms option runs the test regression using only squares of the regressors.

Specification and Stability Tests

EViews provides a number of test statistic views that examine whether the parameters of your model are stable across various subsamples of your data.

One recommended empirical technique is to split the T observations in your data set of observations into T_1 observations to be used for estimation, and $T_2 = T - T_1$ observations to be used for testing and evaluation. Using all available sample observations for estimation promotes a search for a specification that best fits that specific data set, but does not allow for testing predictions of the model against data that have not been used in estimating the model. Nor does it allow one to test for parameter constancy, stability and robustness of the estimated relationship. In time series work you will usually take the first T_1 observations for estimation and the last T_2 for testing. With cross section data you may wish to order the data by some variable, such as household income, sales of a firm, or other indicator variables and use a sub-set for testing.

There are no hard and fast rules for determining the relative sizes of T_1 and T_2. In some cases there may be obvious points at which a break in structure might have taken place—a war, a piece of legislation, a switch from fixed to floating exchange rates, or an oil shock. Where there is no reason *a priori* to expect a structural break, a commonly used rule-of-thumb is to use 85 to 90 percent of the observations for estimation and the remainder for testing.

EViews provides built-in procedures which facilitate variations on this type of analysis.

Chow's Breakpoint Test

The idea of the breakpoint Chow test is to fit the equation separately for each subsample and to see whether there are significant differences in the estimated equations. A significant difference indicates a structural change in the relationship. For example, you can use this test to examine whether the demand function for energy was the same before and after the oil shock. The test may be used with least squares and two-stage least squares regressions.

To carry out the test, we partition the data into two or more subsamples. Each subsample must contain more observations than the number of coefficients in the equation so that the equation can be estimated. The Chow breakpoint test compares the sum of squared residuals obtained by fitting a single equation to the entire sample with the sum of squared residuals obtained when separate equations are fit to each subsample of the data.

EViews reports two test statistics for the Chow breakpoint test. The F-statistic is based on the comparison of the restricted and unrestricted sum of squared residuals and in the simplest case involving a single breakpoint, is computed as

$$F = \frac{(\tilde{u}'\tilde{u} - (u_1'u_1 + u_2'u_2))/k}{(u_1'u_1 + u_2'u_2)/(T-2k)}, \tag{15.17}$$

where $\tilde{u}'\tilde{u}$ is the restricted sum of squared residuals, $u_i'u_i$ is the sum of squared residuals from subsample i, T is the total number of observations, and k is the number of parameters in the equation. This formula can be generalized naturally to more than one breakpoint. The F-statistic has an exact finite sample F-distribution if the errors are independent and identically distributed normal random variables.

The log likelihood ratio statistic is based on the comparison of the restricted and unrestricted maximum of the (Gaussian) log likelihood function. The LR test statistic has an asymptotic χ^2 distribution with degrees of freedom equal to $(m-1)k$ under the null hypothesis of no structural change, where m is the number of subsamples.

One major drawback of the breakpoint test is that each subsample requires at least as many observations as the number of estimated parameters. This may be a problem if, for example, you want to test for structural change between wartime and peacetime where there are only a few observations in the wartime sample. The Chow forecast test, discussed below, should be used in such cases.

To apply the Chow breakpoint test, push **View/Stability Tests/Chow Breakpoint Test...** on the equation toolbar. In the dialog that appears, list the dates or observation numbers for the breakpoints. For example, if your original equation was estimated from 1950 to 1994, entering

```
1960
```

in the dialog specifies two subsamples, one from 1950 to 1959 and one from 1960 to 1994. Typing

```
1960 1970
```

specifies three subsamples, 1950 to 1959, 1960 to 1969, and 1970 to 1994.

Chow's Forecast Test

The Chow forecast test estimates two models—one using the full set of data T, and the other using a long subperiod T_1. A long difference between the two models casts doubt on the stability of the estimated relation over the sample period. The Chow forecast test can be used with least squares and two-stage least squares regressions.

EViews reports two test statistics for the Chow forecast test. The F-statistic is computed as

$$F = \frac{(\tilde{u}'\tilde{u} - u'u)/T_2}{u'u/(T_1 - k)}, \tag{15.18}$$

where $\tilde{u}'\tilde{u}$ is the residual sum of squares when the equation is fitted to all T sample observations, $u'u$ is the residual sum of squares when the equation is fitted to T_1 observations, and k is the number of estimated coefficients. This F-statistic follows an exact finite sample F-distribution if the errors are independent, and identically, normally distributed.

The log likelihood ratio statistic is based on the comparison of the restricted and unrestricted maximum of the (Gaussian) log likelihood function. Both the restricted and unrestricted log likelihood are obtained by estimating the regression using the whole sample. The restricted regression uses the original set of regressors, while the unrestricted regression adds a dummy variable for each forecast point. The LR test statistic has an asymptotic χ^2 distribution with degrees of freedom equal to the number of forecast points T_2 under the null hypothesis of no structural change.

To apply Chow's forecast test, push **View/Stability Tests/Chow Forecast Test...** on the equation toolbar and specify the date or observation number for the beginning of the forecasting sample. The date should be within the current sample of observations.

As an example, suppose we estimate a consumption function using quarterly data from 1947:1 to 1994:4 and specify 1973:1 as the first observation in the forecast period. The test reestimates the equation for the period 1947:1 to 1972:4, and uses the result to compute the prediction errors for the remaining quarters, and reports the following results:

Chow Forecast Test: Forecast from 1973:1 to 1994:4			
F-statistic	0.708348	Probability	0.951073
Log likelihood ratio	91.57088	Probability	0.376108

Neither of the forecast test statistics reject the null hypothesis of no structural change in the consumption function before and after 1973:1.

If we test the same hypothesis using the Chow breakpoint test, the result is

Chow Breakpoint Test: 1973:1			
F-statistic	38.39198	Probability	0.000000
Log likelihood ratio	65.75468	Probability	0.000000

Note that both of the breakpoint test statistics decisively reject the hypothesis from above. This example illustrates the possibility that the two Chow tests may yield conflicting results.

Ramsey's RESET Test

RESET stands for *Regression Specification Error Test* and was proposed by Ramsey (1969). The classical normal linear regression model is specified as

$$y = X\beta + \epsilon, \tag{15.19}$$

where the disturbance vector ϵ is presumed to follow the multivariate normal distribution $N(0, \sigma^2 I)$. Specification error is an omnibus term which covers any departure from the assumptions of the maintained model. Serial correlation, heteroskedasticity, or non-normality of all violate the assumption that the disturbances are distributed $N(0, \sigma^2 I)$. Tests for these specification errors have been described above. In contrast, RESET is a general test for the following types of specification errors:

- Omitted variables; X does not include all relevant variables.
- Incorrect functional form; some or all of the variables in y and X should be transformed to logs, powers, reciprocals, or in some other way.
- Correlation between X and ϵ, which may be caused, among other things, by measurement error in X, simultaneity, or the presence of lagged y values and serially correlated disturbances.

Under such specification errors, LS estimators will be biased and inconsistent, and conventional inference procedures will be invalidated. Ramsey (1969) showed that any or all of these specification errors produce a non-zero mean vector for ϵ. Therefore, the null and alternative hypotheses of the RESET test are

$$H_0\colon \epsilon \sim N(0, \sigma^2 I)$$
$$H_1\colon \epsilon \sim N(\mu, \sigma^2 I) \qquad \mu \neq 0 \tag{15.20}$$

The test is based on an augmented regression

$$y = X\beta + Z\gamma + \epsilon. \tag{15.21}$$

The test of specification error evaluates the restriction $\gamma = 0$. The crucial question in constructing the test is to determine what variables should enter the Z matrix. Note that the Z matrix may, for example, be comprised of variables that are not in the original specification, so that the test of $\gamma = 0$ is simply the omitted variables test described above.

In testing for incorrect functional form, the nonlinear part of the regression model may be some function of the regressors included in X. For example, if a linear relation

$$y = \beta_0 + \beta_1 X + \epsilon, \tag{15.22}$$

is specified instead of the true relation

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon \tag{15.23}$$

the augmented model has $Z = X^2$ and we are back to the omitted variable case. A more general example might be the specification of an additive relation

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon \tag{15.24}$$

instead of the (true) multiplicative relation

$$y = \beta_0 X_1^{\beta_1} X_2^{\beta_2} + \epsilon. \tag{15.25}$$

A Taylor series approximation of the multiplicative relation would yield an expression involving powers and cross-products of the explanatory variables. Ramsey's suggestion is to include powers of the predicted values of the dependent variable (which are, of course, linear combinations of powers and cross-product terms of the explanatory variables) in Z:

$$Z = [\hat{y}^2, \hat{y}^3, \hat{y}^4, \ldots] \tag{15.26}$$

where $\hat{y}$ is the vector of fitted values from the regression of y on X. The superscripts indicate the powers to which these predictions are raised. The first power is not included since it is perfectly collinear with the X matrix.

Output from the test reports the test regression and the F-statistic and log likelihood ratio for testing the hypothesis that the coefficients on the powers of fitted values are all zero. A study by Ramsey and Alexander (1984) showed that the RESET test could detect specification error in an equation which was known *a priori* to be misspecified but which nonetheless gave satisfactory values for all the more traditional test criteria—goodness of fit, test for first order serial correlation, high t-ratios.

To apply the test, select **View/Stability Tests/Ramsey RESET Test...** and specify the number of fitted terms to include in the test regression. The fitted terms are the powers of the fitted values from the original regression, starting with the square or second power. For example, if you specify 1, then the test will add $\hat{y}^2$ in the regression and if you specify 2, then the test will add $\hat{y}^2$ and $\hat{y}^3$ in the regression and so on. If you specify a large number of fitted terms, EViews may report a near singular matrix error message since the powers of the fitted values are likely to be highly collinear. The Ramsey RESET test is applicable only to an equation estimated by least squares.

Recursive Least Squares

In recursive least squares the equation is estimated repeatedly, using ever larger subsets of the sample data. If there are k coefficients to be estimated in the b vector, then the first k observations are used to form the first estimate of b. The next observation is then added to the data set and $k+1$ observations are used to compute the second estimate of b. This process is repeated until all the T sample points have been used, yielding $T-k+1$ estimates of the b vector. At each step the last estimate of b can be used to predict the next value of the dependent variable. The one-step ahead forecast error resulting from this prediction, suitably scaled, is defined to be a *recursive residual*.

More formally, let X_{t-1} denote the $(t-1)\times k$ matrix of the regressors from period 1 to period $t-1$, and y_{t-1} the corresponding vector of observations on the dependent variable. These data up to period $t-1$ give an estimated coefficient vector, denoted by b_{t-1}. This coefficient vector gives you a forecast of the dependent variable in period t. The forecast is $x_t'b$, where x_t' is the row vector of observations on the regressors in period t. The forecast error is $y_t - x_t'b$, and the forecast variance is given by:

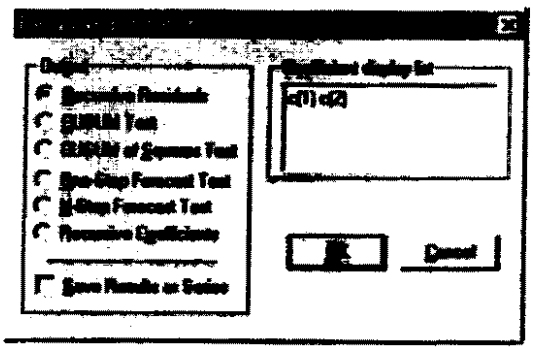
$$\sigma^2(1 + x_t'(X_t'X_t)^{-1}x_t). \tag{15.27}$$

The recursive residual w_t is defined in EViews as

$$w_t = \frac{(y_t - x_t'b)}{(1 + x_t'(X_t'X_t)^{-1}x_t)^{1/2}}. \tag{15.28}$$

These residuals can be computed for $t = k+1, \ldots, T$. If the maintained model is valid, the recursive residuals will be independently and normally distributed with zero mean and constant variance σ^2.

To calculate the recursive residuals, press **View/Stability Tests/Recursive Estimates (OLS only)...** on the equation toolbar. There are six options available for the recursive estimates view. The recursive estimates view is only available for equations estimated by ordinary least squares without AR and MA terms. The **Save Results as Series** option allows you to save the recursive residuals and recursive coefficients as named series in the workfile; see "Save Results as Series" on page 388.



Recursive Residuals

This option shows a plot of the recursive residuals about the zero line. Plus and minus two standard errors are also shown at each point. Residuals outside the standard error bands suggest instability in the parameters of the equation.

CUSUM Test

The CUSUM test (Brown, Durbin, and Evans, 1975) is based on the cumulative sum of the recursive residuals. This option plots the cumulative sum together with the 5% critical lines. The test finds parameter instability if the cumulative sum goes outside the area between the two critical lines.

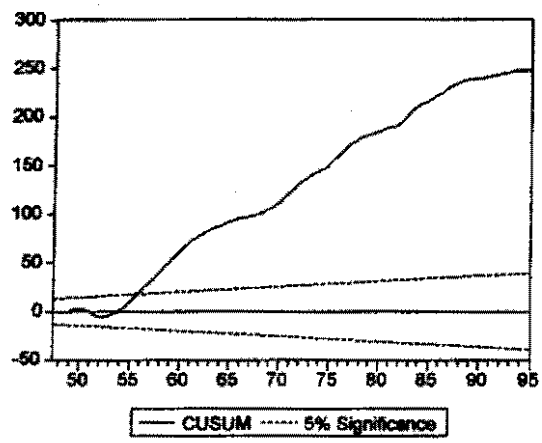
The CUSUM test is based on the statistic

$$W_t = \sum_{r=k+1}^{t} w_r / s, \tag{15.29}$$

for $t = k+1, \ldots, T$, where w is the recursive residual defined above, and s is the standard error of the regression fitted to all T sample points. If the β vector remains constant from period to period, $E(W_t) = 0$, but if β changes, W_t will tend to diverge from the zero mean value line. The significance of any departure from the zero line is assessed by reference to a pair of 5% significance lines, the distance between which increases with t. The 5% significance lines are found by connecting the points

$$[k, \pm -0.948(T-k)^{1/2}] \quad \text{and} \quad [T, \pm 3 \times 0.948(T-k)^{1/2}]. \tag{15.30}$$

Movement of W_t outside the critical lines is suggestive of coefficient instability. A sample CUSUM test is given below.



The test clearly indicates instability in the equation during the sample period.

CUSUM of Squares Test

The CUSUM of squares test (Brown, Durbin, and Evans, 1975) is based on the test statistic

$$W_t = \left(\sum_{r=k+1}^{t} w_r^2 \right) / \left(\sum_{r=k+1}^{T} w_r^2 \right). \tag{15.31}$$

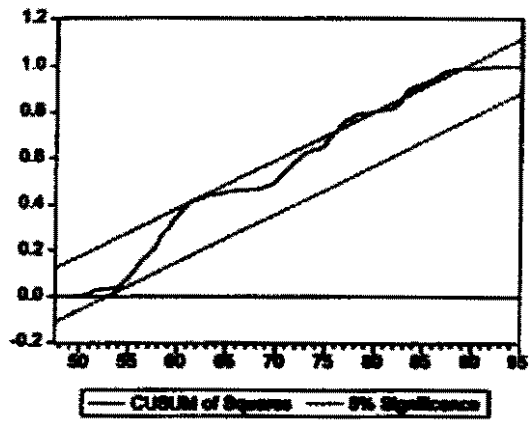
The expected value of S under the hypothesis of parameter constancy is

$$\mathrm{E}(S_t) = (t-k)/(T-k) \tag{15.32}$$

which goes from zero at $t = k$ to unity at $t = T$. The significance of the departure of S from its expected value is assessed by reference to a pair of parallel straight lines around the expected value. See Brown, Durbin, and Evans (1975) or Johnston and DiNardo (1997, Table D.8) for a table of significance lines for the CUSUM of squares test.

The CUSUM of squares test provides a plot of S_t against t and the pair of 5 percent critical lines. As with the CUSUM test, movement outside the critical lines is suggestive of parameter or variance instability.

The cumulative sum of squares is generally within the 5% significance lines, suggesting that the residual variance is somewhat stable.

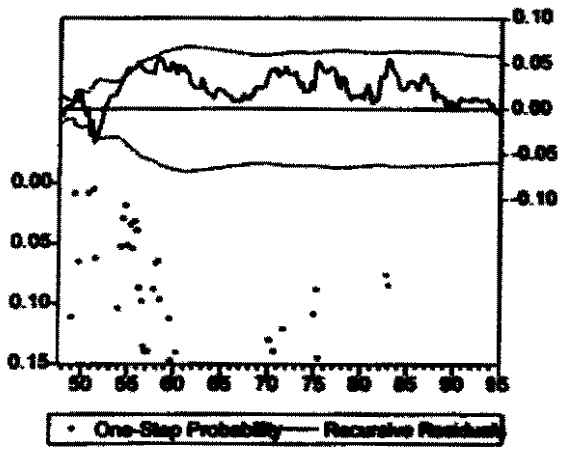


One-Step Forecast Test

If you look back at the definition of the recursive residuals given above, you will see that each recursive residual is the error in a one-step ahead forecast. To test whether the value of the dependent variable at time t might have come from the model fitted to all the data up to that point, each error can be compared with its standard deviation from the full sample.

The **One-Step Forecast Test** option produces a plot of the recursive residuals and standard errors and the sample points whose probability value is at or below 15 percent. The plot can help you spot the periods when your equation is least successful. For example, the one-step ahead forecast test might look like this:

The upper portion of the plot (right vertical axis) repeats the recursive residuals and standard errors displayed by the **Recursive Residuals** option. The lower portion of the plot (left vertical axis) shows the probability values for those sample points where the hypothesis of parameter constancy would be rejected at the 5, 10, or 15 percent levels. The points with p-values less the 0.05 correspond to those points where the recursive residuals go outside the two standard error bounds.



For the test equation, there is evidence of instability early in the sample period.

N-Step Forecast Test

This test uses the recursive calculations to carry out a sequence of Chow Forecast tests. In contrast to the single Chow Forecast test described earlier, this test does not require the specification of a forecast period— it automatically computes all feasible cases, starting with the smallest possible sample size for estimating the forecasting equation and then adding one observation at a time. The plot from this test shows the recursive residuals at

the top and significant probabilities (based on the F-statistic) in the lower portion of the diagram.

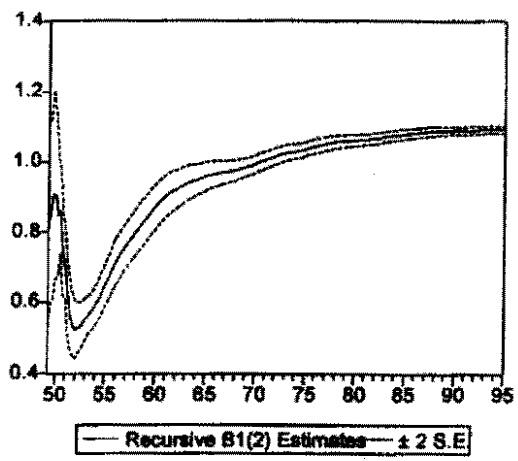
Recursive Coefficient Estimates

This view enables you to trace the evolution of estimates for any coefficient as more and more of the sample data are used in the estimation. The view will provide a plot of selected coefficients in the equation for all feasible recursive estimations. Also shown are the two standard error bands around the estimated coefficients.

If the coefficient displays significant variation as more data is added to the estimating equation, it is a strong indication of instability. Coefficient plots will sometimes show dramatic jumps as the postulated equation tries to digest a structural break.

To view the recursive coefficient estimates, click the **Recursive Coefficients** option and list the coefficients you want to plot in the **Coefficient Display List** field of the dialog box. The recursive estimates of the marginal propensity to consume (coefficient C(2)), from the sample consumption function are provided below:

The estimated propensity to consume rises steadily as we add more data over the sample period, approaching a value of one.



Save Results as Series

The **Save Results as Series** checkbox will do different things depending on the plot you have asked to be displayed. When paired with the **Recursive Coefficients** option, **Save Results as Series** will instruct EViews to save all recursive coefficients and their standard errors in the workfile as named series. EViews will name the coefficients using the next available name of the form, R_C1, R_C2, ..., and the corresponding standard errors as R_C1SE, R_C2SE, and so on.

If you check the **Save Results as Series** box with any of the other options, EViews saves the recursive residuals and the recursive standard errors as named series in the workfile. EViews will name the residual and standard errors as R_RES and R_RESSE, respectively.

Note that you can use the recursive residuals to reconstruct the CUSUM and CUSUM of squares series.

Applications

In this section, we show how to carry out other specification tests in EViews. For brevity, the discussion is based on commands, but most of these procedures can also be carried out using the menu system.

A Wald test of structural change with unequal variance

The F-statistics reported in the Chow tests have an F-distribution only if the errors are independent and identically normally distributed. This restriction implies that the residual variance in the two subsamples must be equal.

Suppose now that we wish to compute a Wald statistic for structural change with unequal subsample variances. Denote the parameter estimates and their covariance matrix in subsample i as b_i and V_i for $i = 1, 2$. Under the assumption that b_1 and b_2 are independent normal, the difference $b_1 - b_2$ has mean zero and variance $V_1 + V_2$. Therefore, a Wald statistic for the null hypothesis of no structural change and independent samples can be constructed as

$$W = (b_1 - b_2)'(V_1 + V_2)^{-1}(b_1 - b_2), \tag{15.33}$$

which has an asymptotic χ^2 distribution with degrees of freedom equal to the number of estimated parameters in the b vector.

To carry out this test in EViews, we estimate the model in each subsample and save the estimated coefficients and their covariance matrix. For example, suppose we have a quarterly sample of 1947:1–1994:4 and wish to test whether there was a structural change in the consumption function in 1973:1. First, estimate the model in the first sample and save the results by the commands

```
coef(2) b1
smpl 1947:1 1972:4
equation eq_1.ls log(cs)=b1(1)+b1(2)*log(gdp)
sym v1=eq_1.@cov
```

The first line declares the coefficient vector, B1, into which we will place the coefficient estimates in the first sample. Note that the equation specification in the third line explicitly refers to elements of this coefficient vector. The last line saves the coefficient covariance matrix as a symmetric matrix named V1. Similarly, estimate the model in the second sample and save the results by the commands

```
coef(2) b2
smpl 1973.1 1994.4
equation eq_2.ls log(cs)=b2(1)+b2(2)*log(gdp)
sym v2=eq_2.@cov
```

To compute the Wald statistic, use the command

```
matrix wald=@transpose(b1-b2)*@inverse(v1+v2)*(b1-b2)
```

The Wald statistic is saved in the 1×1 matrix named WALD. To see the value, either double click on WALD or type "show wald". You can compare this value with the critical values from the χ^2 distribution with 2 degrees of freedom. Alternatively, you can compute the p-value in EViews using the command

```
scalar wald_p=1-@cchisq(wald(1,1),2)
```

The p-value is saved as a scalar named WALD_P. To see the p-value, double click on WALD_P or type "show wald_p". The p-value will be displayed in the status line at the bottom of the EViews window.

The Hausman test

A widely used class of tests in econometrics is the Hausman test. The underlying idea of the Hausman test is to compare two sets of estimates, one of which is consistent under both the null and the alternative and another which is consistent only under the null hypothesis. A large difference between the two sets of estimates is taken as evidence in favor of the alternative hypothesis.

Hausman (1978) originally proposed a test statistic for endogeneity based upon a direct comparison of coefficient values. Here we illustrate the version of the Hausman test proposed by Davidson and MacKinnon (1989, 1993), which carries out the test by running an auxiliary regression.

The following equation was estimated by OLS:

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 08/13/97 Time: 14:12
Sample(adjusted): 1959:02 1995:04
Included observations: 435 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.022699	0.004443	-5.108528	0.0000
LOG(IP)	0.011630	0.002585	4.499706	0.0000
DLOG(PPI)	-0.024886	0.042754	-0.582071	0.5608
TB3	-0.000366	9.91E-05	-3.692675	0.0003
LOG(M1(-1))	0.996578	0.001210	823.4440	0.0000

R-squared	0.999953	Mean dependent var	5.844581
Adjusted R-squared	0.999953	S.D. dependent var	0.670596
S.E. of regression	0.004601	Akaike info criterion	-7.913714
Sum squared resid	0.009102	Schwarz criterion	-7.866871
Log likelihood	1726.233	F-statistic	2304897.
Durbin-Watson stat	1.265920	Prob(F-statistic)	0.000000

Suppose we are concerned that industrial production (IP) is endogenously determined with money (M1) through the money supply function. If this were the case, then OLS estimates will be biased and inconsistent. To test this hypothesis, we need to find a set of instrumental variables that are correlated with the "suspect" variable IP but not with the error term of the money demand equation. The choice of the appropriate instrument is a crucial step. Here we take the unemployment rate (URATE) and Moody's AAA corporate bond yield (AAA) as instruments.

To carry out the Hausman test by artificial regression, we run two OLS regressions. In the first regression, we regress the suspect variable (log) IP on all exogenous variables and instruments and retrieve the residuals:

```
ls log(ip) c dlog(ppi) tb3 log(m1(-1)) urate aaa
series res_ip=resid
```

Then in the second regression, we re-estimate the money demand function including the residuals from the first regression as additional regressors. The result is:

Dependent Variable: LOG(M1)
Method: Least Squares
Date: 08/13/97 Time: 15:28
Sample(adjusted): 1959:02 1995:04
Included observations: 435 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-0.007145	0.007473	-0.956158	0.3395
LOG(IP)	0.001560	0.004672	0.333832	0.7387
DLOG(PPI)	0.020233	0.045935	0.440466	0.6598
TB3	-0.000185	0.000121	-1.527775	0.1273
LOG(M1(-1))	1.001093	0.002123	471.4694	0.0000
RES_IP	0.014428	0.005593	2.579628	0.0102

R-squared	0.999954	Mean dependent var	5.944681
Adjusted R-squared	0.999954	S.D. dependent var	0.670696
S.E. of regression	0.004571	Akaike info criterion	-7.924612
Sum squared resid	0.008963	Schwarz criterion	-7.868300
Log likelihood	1729.581	F-statistic	1888171.
Durbin-Watson stat	1.307838	Prob(F-statistic)	0.000000

If the OLS estimates are consistent, then the coefficient on the first stage residuals should not be significantly different from zero. In this example, the test (marginally) rejects the hypothesis of consistent OLS estimates (to be more precise, this is an asymptotic test and you should compare the t-statistic with the critical values from the standard normal).

Non-nested Tests

Most of the tests discussed in this chapter are nested tests in which the null hypothesis is obtained as a special case of the alternative hypothesis. Now consider the problem of choosing between the following two specifications of a consumption function:

$$
\begin{aligned}
H_1: \quad & CS_t = \alpha_1 + \alpha_2 GDP_t + \alpha_3 GDP_{t-1} + \epsilon_t \\
H_2: \quad & CS_t = \beta_1 + \beta_2 GDP_t + \beta_3 CS_{t-1} + \epsilon_t
\end{aligned}
\tag{15.34}
$$

These are examples of non-nested models since neither model may be expressed as a restricted version of the other.

The *J*-test proposed by Davidson and MacKinnon (1993) provides one method of choosing between two non-nested models. The idea is that if one model is the correct model, then the fitted values from the other model should not have explanatory power when estimating that model. For example, to test model H_1 against model H_2, we first estimate model H_2 and retrieve the fitted values:

```
equation eq_cs2.ls cs c gdp cs(-1)
eq_cs2.fit cs2
```

The second line saves the fitted values as a series named CS2. Then estimate model H_1 including the fitted values from model H_2. The result is:

Dependent Variable: CS
Method: Least Squares
Date: 8/13/97 Time: 00:49
Sample(adjusted): 1947:2 1994:4
Included observations: 191 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	7.313232	4.391305	1.665389	0.0975
GDP	0.278749	0.029278	9.520694	0.0000
GDP(-1)	-0.314540	0.029287	-10.73978	0.0000
CS2	1.048470	0.019684	53.26506	0.0000

R-squared	0.999833	Mean dependent var	1953.966
Adjusted R-squared	0.999830	S.D. dependent var	848.4387
S.E. of regression	11.05357	Akaike info criterion	7.664104
Sum squared resid	22847.93	Schwarz criterion	7.732215
Log likelihood	-727.9219	F-statistic	373074.4
Durbin-Watson stat	2.253186	Prob(F-statistic)	0.000000

The fitted values from model H_2 enter significantly in model H_1 and we reject model H_1.

We must also test model H_2 against model H_1. Estimate model H_1, retrieve the fitted values, and estimate model H_2 including the fitted values from model H_1. The results of this "reverse" test are given by:

Dependent Variable: CS
Method: Least Squares
Date: 08/13/97 Time: 16:58
Sample(adjusted): 1947:2 1994:4
Included observations: 191 after adjusting endpoints

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1427.716	132.0349	-10.81318	0.0000
GDP	5.170543	0.476803	10.84419	0.0000
CS(-1)	0.977296	0.018348	53.26808	0.0000
CS1	-7.292771	0.679043	-10.73878	0.0000

R-squared	0.999833	Mean dependent var	1953.988
Adjusted R-squared	0.999830	S.D. dependent var	846.4387
S.E. of regression	11.05357	Akaike info criterion	7.664104
Sum squared resid	22847.93	Schwarz criterion	7.732215
Log likelihood	-727.9219	F-statistic	373074.4
Durbin-Watson stat	2.253186	Prob(F-statistic)	0.000000

The fitted values are again statistically significant and we reject model H_2.

In this example, we reject both specifications, against the alternatives, suggesting that another model for the data is needed. It is also possible that we fail to reject both models, in which case the data do not provide enough information to discriminate between the two models.

Commands

All of the specification and diagnostic tests explained in this chapter are available in command form as views of a named equation. Follow the equation name with a dot and the view name of the test. For example, to carry out the Wald test of whether the third and fourth coefficients of the equation object EQ1 are both equal to zero, type

```
eq1.wald c(3)=0,c(4)=0
```

To carry out the serial correlation LM test of the residuals in equation EQ_Y up to 4 lags, type

```
eq_y.auto(4)
```

To display the recursive residuals of equation EQM1, type

```
eqm1.rls(r)
```

See "Equation" on page 21 of the *Command and Programming Reference* for a complete list of commands and options available for equation objects.