

PNACH-801

**STRENGTHENING POLICY ANALYSIS:
ECONOMETRIC TESTS USING
MICROCOMPUTER SOFTWARE**

**Lawrence Haddad
M. Daniel Westbrook
Daniel Driscoll
Ellen Payongayong
Joshua Rozen
Melvyn Weeks**

MICROCOMPUTERS IN POLICY RESEARCH 2

INTERNATIONAL FOOD POLICY RESEARCH INSTITUTE

A

Copyright 1995 International Food Policy Research Institute

All rights reserved. Sections of this report may be reproduced without the express permission of but with acknowledgment to the International Food Policy Research Institute.

Library of Congress Cataloging-in-Publication Data

Strengthening policy analysis: econometric tests using microcomputer software / Lawrence Haddad . . . [et al.]

p. cm—(Microcomputers in policy research: 2)

Includes bibliographical references.

ISBN 0-89629-330-0

1. Econometrics—Computer programs. 2. SAS (Computer file) 3. GAUSS-386i.

I. Haddad, Lawrence. II. Series

HB139.S785 1995

330'.01'5195—dc20

95-5179
CIP

The following product names used in this manual are trademarked:

GAUSS and GAUSS-386, trademarks of Aptech Systems, Inc.
SAS and SAS/STAT, registered trademarks of SAS Institute, Inc.
SPSS/PC+, a trademark of SPSS, Inc.

**INTERNATIONAL
FOOD
POLICY
RESEARCH
INSTITUTE**

The International Food Policy Research Institute was established in 1975 to identify and analyze alternative national and international strategies and policies for meeting food needs of the developing world on a sustainable basis, with particular emphasis on low-income countries and on the poorer groups in those countries. While the research effort is geared to the precise objective of contributing to the reduction

of hunger and malnutrition, the factors involved are many and wide-ranging, requiring analysis of underlying processes and extending beyond a narrowly defined food sector. The Institute's research program reflects worldwide collaboration with governments and private and public institutions interested in increasing food production and improving the equity of its distribution. Research results are disseminated to policymakers, opinion formers, administrators, policy analysts, researchers, and others concerned with national and international food and agricultural policy.

IFPRI is a member of the Consultative Group on International Agricultural Research and receives support from Australia, Belgium, Canada, Centre de coopération internationale en recherche agronomique pour le développement (CIRAD), China, Denmark, Food and Agriculture Organization of the United Nations, Ford Foundation, France, German Agency for Technical Cooperation (GTZ), German Federal Ministry for Economic Cooperation (BMZ), India, Inter-American Development Bank, International Development Research Centre (Canada), International Fund for Agricultural Development, Japan, Netherlands, Norway, Overseas Development Institute, Philippines, Rockefeller Foundation, Spain, Sweden, Switzerland, United Kingdom, United Nations Development Programme, United Nations International Children's Emergency Fund, United States, and the World Bank.

CONTENTS

Preface

1. Introduction	1
2. Software Information	4
3. Data Handling	7
4. Specification Tests	20
5. Deficient Data Problems	116
Appendix 1: SPSS/PC+ Environment and Commands	138
Appendix 2: SAS PC Environment and Commands	143
Appendix 3: Gauss-386 Environment and Commands	150
Bibliography	154

TABLES

1. Resource requirements of SPSS/PC+, SAS, and GAUSS-386 . . .	6
2. Labels and summary descriptive statistics on all variables used in programs (DATA.ASC)	8

FIGURES

1. Sample ATOG program for converting ASCII data to GAUSS-386 data	10
2. Sample programs for reading ASCII files, in SAS PC	11
3. Sample programs for reading ASCII files, in SPSS/PC+	13
4. Sample programs for writing data to an ASCII data set, in GAUSS-386	15
5. Sample programs for writing data to an ASCII data set, in SAS PC	16
6. Sample programs for writing data to an ASCII data set, in SPSS/PC+	17
7. Sample program for reading data and reporting descriptive statistics, in GAUSS-386	18
8. Sample program for reading data and reporting descriptive statistics, in SAS PC	19
9. Sample program for reading data and reporting descriptive statistics, in SPSS/PC	19
10. Sample program for Goldfeld-Quandt test, in GAUSS-386	22
11. Sample program for Goldfeld-Quandt test, in SAS PC	25
12. Sample program for Goldfeld-Quandt test, in SPSS/PC+	26
13. Sample program for Breusch-Pagan test, in GAUSS-386	28
14. Sample program for Breusch-Pagan test, in SAS PC	30
15. Sample program for Breusch-Pagan test, in SPSS/PC+	31
16. Sample program for White test, in GAUSS-386	33
17. Sample program for White test, in SAS PC	36
18. Sample program for White test, in SPSS/PC+	37
19. Sample program for Jarque-Bera test, in GAUSS-386	39
20. Sample program for Jarque-Bera test, in SAS PC	41
21. Sample program for Jarque-Bera test, in SPSS/PC+	42
22. Sample programs for Hausman test and Hausman-Wu test, in GAUSS-386	47
23. Sample program for Hausman-Wu test, in SAS PC	54
24. Sample program for Hausman-Wu test, in SPSS/PC+	56
25. Sample program for Levi bounds test, in GAUSS-386	59
26. Sample program for Levi bounds test, in SAS PC	61
27. Sample program for Levi bounds test, in SPSS/PC+	62

28.	Sample program for nonnested F -test, in GAUSS-386	65
29.	Sample program for nonnested F -test, in SAS PC	69
30.	Sample program for nonnested F -test in SPSS/PC+	70
31.	Sample program for nonnested J -test, in GAUSS-386	73
32.	Sample program for nonnested J -test, in SAS PC	75
33.	Sample program for nonnested J -test, in SPSS/PC+	76
34.	Sample program for the Ramsey RESET Test, in GAUSS-386	79
35.	Sample program for the Ramsey RESET Test, in SAS PC	81
36.	Sample program for the Ramsey RESET Test, in SPSS/PC+	82
37.	Sample program for performing auxiliary regressions, in GAUSS-386	84
38.	Sample program for performing auxiliary regressions, in SAS PC	85
39.	Sample program for performing auxiliary regressions, in SPSS/PC+	86
40.	Sample program for determining the condition number, in GAUSS-386	88
41.	Sample program for determining the condition number, in SAS PC	90
42.	Sample program for determining the condition number, in SPSS/PC+	90
43.	Sample program for Chow test, in GAUSS-386	93
44.	Sample program for Chow test, in SAS PC	98
45.	Sample program for Chow test, in SPSS/PC+	99
46.	Sample program for Utts' Rainbow test, in GAUSS-386	102
47.	Sample program for Utts' Rainbow test, in SAS PC	105
48.	Sample program for Utts' Rainbow test, in SPSS/PC+	106
49.	Sample spline program, in GAUSS-386	109
50.	Sample spline program, in SAS PC	113
51.	Sample program for DFFITS calculation, in GAUSS-386	118
52.	Sample program for DFFITS calculation, in SAS PC	121
53.	Sample program for DFFITS calculation, in SPSS/PC+	122
54.	Sample program for estimating bounded influence, in GAUSS-386	124
55.	Sample program for estimating bounded influence, in SAS PC	128
56.	Sample program for estimating bounded influence, in SPSS/PC+	129
57.	Sample program for calculating first-order regressions when data are missing, in GAUSS-386	132
58.	Sample program for calculating first-order regressions when data are missing, in SAS PC	136
59.	Sample program for calculating first-order regressions when data are missing, in SPSS/PC+	137

PREFACE

Over the past decade, the increasing power and reliability of microcomputers and the development of sophisticated software designed specifically for use with them has led to significant changes in the way that socioeconomic data are collected and analyzed. The venue of the computations has shifted from off-site mainframes, dependent on highly trained operators and significant capital investment in supporting equipment, to desktop and laptop computers, dependent only on the occasional availability of electricity. This means that it is now feasible to quickly transfer new statistical techniques between IFPRI and IFPRI's collaborators in developing countries, that data manipulation costs of policy analysis have been substantially reduced, and that a new level of complexity and accuracy is now possible in the collection and analysis of household survey data in developing countries.

As with any new technology, however, there are substantial costs in time and money involved in learning the most efficient ways of using this new technology and then transmitting these lessons to others. This series, *Microcomputers in Policy Research*, represents IFPRI's collective ongoing experience in adapting microcomputer technology for use in food policy analysis in developing countries. The papers in the series are primarily for the purpose of sharing these lessons with potential users in developing countries, although persons and institutions in developed countries may also find them useful. The series is designed to provide hands-on methods for resolving statistical and data-collection problems encountered in food policy research. In our opinion, examples provide the best and clearest form of instruction; therefore, examples—including actual software codes wherever relevant—are used extensively throughout this series.

This second book in the series, *Strengthening Policy Analysis: Econometric Tests Using Microcomputer Software*, by Lawrence Haddad, M. Daniel Westbrook, Daniel Driscoll, Ellen Payongayong, Joshua Rozen, and Melvyn Weeks, is a manual outlining how to conduct some fairly basic econometric tests and procedures to determine the robustness of the estimated parameters upon which policy decisions are frequently based. It is based on IFPRI experiences with cross-section econometric analysis over the past 10 years. The authors address a number of issues relating to the choice of model variables, the choice of estimation method, and the sensitivity of results to missing or extreme data values. Examples are provided throughout, using comparable programs from SPSS/PC+™, SAS®, and GAUSS-386™.

Howarth Bouis, Lawrence Haddad, and Stephen Vosti
Editors

ACKNOWLEDGMENTS

This manual was conceived of in July 1991, and ever since it has represented a labor of love for a number of people whom we would like to acknowledge. For assisting in editing and formatting the document, we would like to express our deep gratitude to Jay Willis. For invaluable comments, we would like to thank Anna Alfano, Sumiter Broca, Lynn Brown, Julie Witcover, and Yisehac Yohannes. For expert programming advice, we are grateful to Dave Bruton. Finally, several people gave us considerable encouragement and support: Howarth Bouis, Just Faaland, Per Pinstруп-Andersen, Steve Vosti, and Nancy Walczak. The manual has benefited enormously from the input of all these individuals; however, we alone are responsible for all errors.

1 INTRODUCTION

Observe a swimmer trying to simultaneously submerge five inflatable beach balls. The swimmer struggles for some time. When the swimmer finally succeeds, he or she has a photograph taken. The swimmer quickly loses control and the balls explode above the surface of the water. The photographer is an econometrician.

—Anonymous econometrics professor

In the absence of comprehensive and well-presented empirical analyses, we had no option but to follow our political instincts.

—Anonymous policymaker

Improvements in econometric methods and the machines that run them, alongside dramatic increases in the quality and quantity of information available to inform policymakers, are bridging the gap between what is known and what is needed to guide policy. These developments make it easier for econometricians to model policy with some degree of confidence. The developments also make empirically based research more accessible to policy analysts. Likewise, policymakers can and should have more options than the anonymous policymaker quoted above—and, indeed, this new wealth of information forces them to look beyond political instincts for guidance. This manual contains critical structural support for the evolving bridge between policy needs and knowledge.

There is, however, both good and bad news associated with the explosion in the use of econometric procedures and tests in policy research witnessed in the past 10 years. The bad news is that this trend has led to a growing realization that estimated policy parameters are highly sensitive to the ways in which data are handled and the ways in which econometric models are constructed. The good news is that the ability to conduct tests that can gauge these sensitivities has improved with the emergence of powerful microcomputers, of statistical software that combines ease of use with statistical power, and of texts on *applied* econometrics. These developments permit econometricians and policy analysts to improve the accuracy and reliability of estimated policy parameters and, at the very least, to indicate where their models are most sensitive to specification error and departures from standard assumptions. The following example illustrates the usefulness of these tests for policy formulation.

Until recently, one widely accepted notion about development was that poverty alleviation was necessary and sufficient for reductions in undernutrition to occur. The implication was that the effect of income-generation policies on household food consumption and nutrition status is strong. Recent econometric work (Behrman and Deolalikar 1987; Bouis

and Haddad 1992) has cast some doubt as to whether increasing income alone is sufficient to alleviate undernutrition. The policy choice revolves around the magnitude of the calorie-income elasticity, and by extension, the estimated coefficient of income (the marginal propensity to consume) when calorie consumption is the dependent variable. Bouis and Haddad (1992) found that, for the same households, two-stage least squares (2SLS) estimates differed from ordinary least squares (OLS) estimates. Calculation of the Levi bounds on the marginal propensity to consume indicated that income from the survey was measured with much error. A more formal Hausman-Wu test established that the differences between the OLS and 2SLS estimates were large enough to reject the use of OLS estimates because of their bias. The differences were due, in part, to the endogeneity of income on the right-hand side, which was caused in part by measurement error on income. The elasticity estimate was sensitive to the choice of estimator used. This sensitivity has consequences for policy formulation. If the larger elasticity estimates of approximately 0.5 are believed, policy can be more focused on income generation. If, on the other hand, the smaller elasticity estimates of approximately 0.1 are believed, the focus of policy perhaps should be expanded toward complementary factors for reducing undernutrition (such as education, community sanitation, water quality, and the availability of medical supplies), and toward the importance of other dimensions of undernutrition (such as micronutrient consumption and individual-level versus household-level consumption).

This manual outlines how to conduct some of these basic econometric specification tests and procedures, how to interpret the results, and how to modify the econometric approach as a result of the tests. The tests and procedures are largely confined to cross-section analyses, as opposed to time-series analyses. This reflects IFPRI's current research orientation as well as the nature of existing data used to support policy research in developing countries. The tests address a number of issues: (1) the validity of the assumption of normality and constant variance of the error term, (2) selection of the most appropriate explanatory variables to include in a model, (3) the appropriateness of the model under different structural conditions, (4) the need to account for measurement errors in explanatory variables, (5) how to detect and respond to outlier observations, and (6) what can be done (if anything) about missing data points.

Each test and procedure is described in terms of why, when, and how it might be used. Sample programs, with software code from SPSS/PC+,¹ the SAS system for personal computers (hereafter referred to as SAS PC), and GAUSS-386² are presented to demonstrate how the procedure can be executed with the sample data set (an ASCII file named DATA.ASC). These sample programs are also on the diskette that accompanies the manual. Keep in mind that there may be several alternative programming strategies in any given instance; generally only one is presented in this manual. At the end of each section, several

¹SPSS/PC+ for Windows, a recently released product, is not outlined in this manual.

²Companies producing the computer software mentioned in this manual are listed in the notes to Table 1, p. 6.

widely used econometrics textbooks are listed that discuss the tests provided in the manual—and alternatives that are not. Unless otherwise noted, all programs run in under 3 minutes using the sample data set on a DOS-based ZEOS 486DX2 desktop computer running at 66 megahertz. The econometric procedures selected are not exhaustive; rather, they reflect IFPRI's collective ongoing experience in using econometrics and widely available software packages for food policy analysis in developing countries.

In addition, it is important to remember that parameter estimates are sensitive to the quality of data used as well as the appropriateness of the econometric approach. To that end, the reader is encouraged to make use of the first paper in this Microcomputers in Policy Research series, *Designing a Data Entry and Verification System*, by Peter A. Tatian. Finally, it is hoped that readers will alert the authors to any errors found in this manual, together with their suggestions for additional materials to include in future versions of this manual, and in the series generally.

Standard econometric notation is used throughout this manual. In general, Arabic letters refer to data matrices and Greek letters refer to model parameters and to stochastic error terms. The basic model is written as follows:

$$y = X\beta + \epsilon.$$

In the model above, y is an $N \times 1$ vector of observations on the dependent variable; X is an $N \times K$ matrix of observations on the K explanatory variables (including the constant term); β is a $K \times 1$ vector of parameters; ϵ is an $N \times 1$ vector of unobservable stochastic disturbance terms; and N is the sample size.

It is generally assumed that the matrix X contains all of the appropriate regressors in the appropriate functional form, and that the classical normal assumptions concerning the stochastic disturbance terms hold: they have zero mean and are nonheteroskedastic, nonautocorrelated, uncorrelated with the regressors, and normally distributed. This manual is largely devoted to examining the definitions of X and to checking for heteroskedasticity and correlation between regressors and the stochastic disturbance term.

Extensions of this notation are required periodically in the manual and are introduced as needed. Usually, however, the dimensions of vectors and matrices, unless required for clarity, will not be repeated.

2 SOFTWARE INFORMATION

All the statistical software program files presented in this guide are in the form of "batch" files. Batch files are sets of software commands that can be created and edited in any text (ASCII)-editing package.

SPSS/PC+ SPSS/PC+ is a statistical package that allows easy access to data. SPSS/PC+ provides tools for reading, aggregating, merging, recoding, and creating data. In addition, SPSS/PC+ includes numerous econometric and statistical procedures.

There are three methods for executing SPSS/PC+ commands. The first method is a user-friendly menu system for building and executing commands. Second, the user can type commands at the SPSS/PC+ prompt in an interactive mode. Third, the user can create a text (ASCII) file and submit it for execution in batch mode. This last method can be done within SPSS/PC+, using the REVIEW text editor, or with any other text editor, such as EDLIN, NORTON EDITOR, or WordPerfect (saving the file as DOS text). For expositional purposes, this manual uses the latter format—although the SPSS/PC+ commands are identical—whichever method is chosen. Appendix 1 describes some interactive commands. A full exposition is given in Norusis (1990).

SPSS/PC+ is the software of choice if ease of use and low start-up costs are important to the user. The user-friendly interface makes SPSS/PC+ an ideal choice for someone with little or no programming experience.

To submit a batch program, type

```
SPSSPC filename
```

at the DOS prompt.

SAS PC The SAS system for personal computers is a statistical package that allows easy access to data. SAS PC provides tools for reading, aggregating, merging, recoding, and creating data. In addition, SAS PC includes numerous preprogrammed econometric and statistical procedures.

Although SAS PC can be run interactively, it is beyond the scope of this paper to describe its use (except for some notes in Appendix 2). Please consult the manual (SAS Institute Inc. 1988) for assistance.

SAS PC is somewhat more difficult for novices to learn than SPSS/PC+. However, it is slightly more comprehensive and powerful than SPSS/PC+. Programs written for SAS PC will run with almost no changes on other platforms, such as IBM mainframes and DEC VAXs.

To submit a batch program, type

SAS filename

at the DOS prompt.

GAUSS-386 GAUSS-386 is a programming language that uses syntax similar to that used in matrix algebra representations of econometric techniques. GAUSS-386 does not incorporate the extensive array of preprogrammed procedures found in SAS PC and SPSS/PC+, but allows much more flexibility and power to create specialized procedures.

Although GAUSS-386 can be run interactively, it is beyond the scope of this guide to describe its use (except for some notes in Appendix 3). Please consult the manual (Aptech Systems Inc. 1992) for assistance. To submit a batch program, type

GAUSS386 filename

at the DOS prompt.

SUMMARY Each of the three software packages featured throughout this guide has strengths and weaknesses. At IFPRI, these packages tend to be used in a complementary manner, rather than being treated as strict substitutes.³ Each software package is described in greater detail in the appendixes—SPSS/PC+, in Appendix 1; SAS PC, in Appendix 2; and GAUSS-386, in Appendix 3. Related to their different capacities, each package has different budgetary implications, both in terms of the direct cost of the software and the indirect costs of hardware requirements. These resource requirements are summarized in Table 1. Appendixes 1, 2, and 3 also summarize the common commands used in SAS PC, SPSS/PC+, and GAUSS-386, respectively, and the reader may find it useful to review the appropriate appendix before using the programs.⁴

³In fact, SAS PC and SPSS/PC+ data files can be converted into each other by a software program called DBMS/Copy Plus™ (by Concepts Software, Inc.).

⁴Throughout the sample programs, the file naming convention shown in Table 2 is adopted.

Table 1—Resource requirements of SPSS/PC+, SAS, and GAUSS-386

Software Name	SPSS/PC+	SAS	GAUSS-386
Version	4.0.1	6.04	3.0
Company name	SPSS, Inc.	SAS Institute Inc.	Aptech Systems, Inc
Address	444 N. Michigan Ave. Chicago, Illinois 60611 U.S.A.	SAS Circle Cary, North Carolina 27512 U.S.A.	23804 SE Kent-Kangley Road Maple Valley, Washington 98038 U.S.A.
Telephone	U.S.A. 312-329-3500	U.S.A. 919-677-8000	U.S.A. 206-432-7855
FAX	U.S.A. 312-329-3668	U.S.A. 919-677-8123	U.S.A. 206-432-7832
Technical support telephone number	U.S.A. 312-329-3410	U.S.A. 919-677-8008	U.S.A. 206-432-7855
Type of microchip processor required	286 (386 recommended)	286 (386 recommended)	386 plus coprocessor
Memory requirement	640 kilobytes	640 kilobytes (2 megabytes recommended)	4 megabytes
Hard disk space	5-8 megabytes	15-25 megabytes	4 megabytes
Priced ^a in U.S.A.	\$1,090 (for Base and Stat modules)	\$1,670 first year (for Base SAS and SAS/STAT); \$595 each additional year (license)	\$995

^aPrices may vary over time and location.

3 DATA HANDLING

A DESCRIPTION OF THE SAMPLE DATA SET

The data used in all of the following programs are taken from surveys of rural households residing in Bukidnon Province in the Philippines. Households were surveyed four times at four-month intervals (1984–85), and data were collected on a wide range of topics, including landholdings, expenditure patterns, food intake, housing characteristics, assets, schooling, and food prices. See Bouis and Haddad (1992) for a more detailed description of how the data were collected. The data set consists of four observations on each of 406 rural households ($N = 1,624$) that were present for all survey rounds and whose livelihood depended primarily on the production of either corn or sugarcane.

This data set was selected from among many at IFPRI, not because it is any cleaner or gives "better" results than others, but simply because it is the data set that the authors are most familiar with. Table 2 provides labels and summary descriptive statistics for the 28 variables used in the following procedures. The data are provided as an ASCII file, DATA.ASC, on the diskette.

Table 2—Labels and summary descriptive statistics on all variables used in programs (DATA.ASC)

Variable	Label	N	Minimum	Maximum	Mean	Standard Deviation
Dependent variable:						
Y1	Household calorie intake per capita per day (from 24-hour recall)	1,624	268.00000	4,583.27	1,714.30	563.08983
Y2	Household calorie intake per capita per day (from food expenditure data)	1,624	309.65000	5,457.72	1,754.75	653.06000
Explanatory variable:						
X1	Retail price of shelled corn per kilogram (1984 pesos; simple average of respondents for barrio)	1,624	3.20000	6.30000	4.49232	0.68510
X2	Retail price of milled rice per kilogram (1984 pesos; simple average of respondents for barrio)	1,624	4.65000	7.31000	5.79466	0.52136
X3	Cultivated area per capita (average of four survey rounds, in hectares)	1,624	0	2.60000	0.37816	0.39453
X4	Zero-one dummy for presence of electricity for house	1,624	0	1.00000	0.28818	0.45305
X5	Zero-one dummy for improved quality of flooring materials for house	1,624	0	1.00000	0.13547	0.34233
X6	Zero-one dummy for improved quality of roofing materials for house	1,624	0	1.00000	0.70936	0.45420
X7	Zero-one dummy for improved quality of materials used for house framing and walls	1,624	0	1.00000	0.09606	0.29476
X8	Age of head of household (in months)	1,624	250.80000	751.30000	446.46419	100.16314
X9	Number of household members	1,624	3.00000	19.00000	7.16071	2.67816
X10	Logarithm of household total expenditures per week per capita (1984 pesos; varies by round)	1,624	2.07000	6.07000	3.66057	0.55398
X11	Value of all assets (1984 pesos; average of rounds 1 and 4)	1,624	0	63,221.43	2,916.58	5,512.63
X12	Area owned per capita (average of four survey rounds, in hectares)	1,624	0	3.43000	0.28050	0.47643
X13	Municipal population density (persons per square kilometer)	1,624	51.00000	223.00000	150.05911	44.89573
X14	Years in school, head of household	1,624	0	14.00000	6.15335	2.68288

(continued)

Table 2—Continued

X15	Years in school, spouse of head of household	1,624	0	15.00000	5.58867	2.94384
D1	Fraction of household that are females less than or equal to 5 years of age	1,624	0	0.60000	0.13453	0.13368
D2	Fraction of household that are females greater than 5 years and less than or equal to 11 years of age	1,624	0	0.50000	0.09795	0.11121
D3	Fraction of household that are females greater than 11 years and less than or equal to 17 years of age	1,624	0	0.50000	0.06308	0.09115
D4	Fraction of household that are females greater than 17 years of age	1,624	0	0.50000	0.19045	0.07636
D5	Fraction of household that are males less than or equal to 5 years of age	1,624	0	0.67000	0.14748	0.13555
D6	Fraction of household that are males greater than 5 years and less than or equal to 11 years of age	1,624	0	0.43000	0.09853	0.10893
D7	Fraction of household that are males greater than 11 years and less than or equal to 17 years of age	1,624	0	0.44000	0.06481	0.09420
D8	Fraction of household that are males greater than 17 years of age	1,624	0	0.60000	0.20417	0.08034
RD1	Zero-one dummy for first round survey	1,624	0	1.00000	0.25000	0.43315
RD2	Zero-one dummy for second round survey	1,624	0	1.00000	0.25000	0.43315
RD3	Zero-one dummy for third round survey	1,624	0	1.00000	0.25000	0.43315

READING ASCII DATA INTO SOFTWARE

Both SPSS/PC+ and SAS PC have procedures within the main product for reading ASCII files. These procedures are very flexible; the procedures can handle free format (data with at least one space between each value) or fixed format (each variable is to be found within columns and on rows specified by the user). Both packages can, in addition, read data with multiple lines per observation (or case). Figures 1, 2, and 3 are sample programs for reading ASCII files, in GAUSS-386, SAS PC, and SPSS/PC+, respectively. The READFREE.SAS (Figure 2a) and READFREE.SPS (Figure 3a) programs demonstrate how to read data in free format. Similarly, READFIXD.SAS (Figure 2b) and READFIXD.SPS (Figure 3b) show how to read data in fixed format and on multiple lines.

GAUSS-386 (Figure 1) has a separate utility to read in ASCII data. This utility, called ATOG (ASCII to GAUSS), will convert a free-formatted (space delimited) ASCII file into a GAUSS-386 data file, *filename.DAT*, and a companion label file, *filename.DHT*. To use ATOG, you must construct a program as shown, which contains the name of the ASCII file, the name you wish to give the new GAUSS-386 data file, and a complete list of the variable names. The name for this program must end in the extension "CMD." For example, Figure 1 is saved under READASCI.CMD. (Note that comments are not permitted in ATOG programs.) To use this program from within the GAUSS-386 shell, at the command line prompt, >, type the following:

```
DOS ATOG filename.CMD
```

From DOS, simply type the following:

```
ATOG filename.CMD
```

Figure 1—Sample ATOG program for converting ASCII data to GAUSS-386 data

```
INPUT DATA.ASC;
OUTPUT DATA;
INVAR X1 X2 X3 X4 D1 D3 D2 D4 D5 D7 D6 D8 X5 X6 X7 X8 Y1 X9 X10 Y2
      X11 X12 X13 RD1 RD2 RD3 X14 X15;
OUTTYP D;
```

Figure 2—Sample programs for reading ASCII files, in SAS PC

2a—For free format

```

*****
*   PROGRAM:   READFREE.SAS   SOFTWARE: SAS PC 6.04   *
*   FILENAME   DESCRIPTION   *
*   INPUTS:    DATA.ASC     ASCII FILE       *
*   OUTPUTS:   DATA.SSD     SAS PC DATA SET      *
*   PURPOSE:   READ ASCII FILE INTO A SAS PC SYSTEM *
*   FILE. THIS PROGRAM ASSUMES THAT THE DATA *
*   ARE IN FREE FORMAT (VARIABLES ARE *
*   SEPARATED BY AT LEAST ONE SPACE). *
*****;

LIBNAME CDRV 'C:\DATA\';

DATA CDRV.DATA;
  INFILE 'C:\DATA\DATA.ASC';
  INPUT X1 X2 X3 X4 D1 D3 D2 D4 D5 D7 D6 D8 X5 X6 X7 X8 Y1 X9 X10 Y2
        X11 X12 X13 RD1 RD2 RD3 X14 X15;

  LABEL
  Y1 = 'HH CALORIE INTAKE, CAPITA, DAY (24-HOUR RECALL DATA)'
  Y2 = 'HH CALORIE INTAKE, CAPITA, DAY (FOOD EXPENDITURE DATA)'
  X1 = 'RETAIL PRICE OF SHELLED CORN,KG (1984 PESOS; AVERAGE,BARRIO)'
  X2 = 'RETAIL PRICE OF MILLED RICE,KG (1984 PESOS; AVERAGE,BARRIO)'
  X3 = 'CULTIVATED AREA PER CAPITA (AVERAGE OF FOUR SURVEY ROUNDS)'
  X4 = 'ZERO-ONE DUMMY FOR PRESENCE OF ELECTRICITY FOR HOUSE'
  X5 = 'ZERO-ONE DUMMY FOR QUALITY OF FLOORING MATERIALS FOR HOUSE'
  X6 = 'ZERO-ONE DUMMY FOR QUALITY OF ROOFING MATERIALS FOR HOUSE'
  X7 = 'ZERO-ONE DUMMY FOR QUALITY OF MATERIALS USED FOR HOUSE WALLS'
  X8 = 'AGE OF HEAD OF HOUSEHOLD (IN MONTHS)'
  X9 = 'NUMBER OF HOUSEHOLD MEMBERS'
  X10 = 'LOG OF HHOLD TOTAL EXPENDITURES,WK,CAP (1984 PESOS, ROUND)'
  X11 = 'VALUE OF ALL ASSETS (1984 PESOS; AVERAGE OF ROUNDS 1 AND 4)'
  X12 = 'OWNED AREA PER CAPITA (AVERAGE OF FOUR SURVEY ROUNDS)'
  X13 = 'MUNICIPAL POPULATION DENSITY (PERSONS PER SQUARE KILOMETER)'
  X14 = 'YEARS IN SCHOOL, HEAD OF HOUSEHOLD'
  X15 = 'YEARS IN SCHOOL, SPOUSE OF HEAD OF HOUSEHOLD'
  D1 = '% OF HH THAT ARE FEMALES < OR EQUAL TO 5 YRS OF AGE'
  D2 = '% OF HH THAT ARE FEMALES > 5 YRS AND < OR = TO 11 YRS OF AGE'
  D3 = '% OF HH THAT ARE FEMALES > 11 YRS AND < OR = 17 YRS OF AGE'
  D4 = '% OF HH THAT ARE FEMALES > 17 YRS OF AGE'
  D5 = '% OF HH THAT ARE MALES < OR EQUAL TO 5 YRS OF AGE'
  D6 = '% OF HH THAT ARE MALES > 5 YRS AND < OR = TO 11 YRS OF AGE'
  D7 = '% OF HH THAT ARE MALES > 11 YRS AND < OR = TO 17 YRS OF AGE'
  D8 = '% OF HH THAT ARE MALES > THAN 17 YRS OF AGE'
  RD1 = 'ZERO-ONE DUMMY FOR FIRST ROUND SURVEY'
  RD2 = 'ZERO-ONE DUMMY FOR SECOND ROUND SURVEY'
  RD3 = 'ZERO-ONE DUMMY FOR THIRD ROUND SURVEY';

RUN;

PROC PRINT DATA=CDRV.DATA(OBS=10);
  VAR Y1 X1 X2 X3 X4 X5;

RUN;

```

2b—For fixed format

```

*****
* PROGRAM:  READFIXD.SAS  SOFTWARE: SAS PC 6.04  *
*          FILENAME      DESCRIPTION          *
* INPUTS:   DATA.ASC    ASCII FILE          *
* OUTPUTS:  DATA.SSD    SAS PC DATA SET    *
* PURPOSE:  READ ASCII FILE INTO A SAS PC SYSTEM *
*          FILE.THIS PROGRAM ASSUMES THE DATA ARE *
*          IN FIXED FORMAT. VARIABLES TO BE READ IN *
*          MUST BE IDENTIFIED BY A NAME AND COLUMN *
*          LOCATION. SUBSEQUENT RECORDS ARE DENOTED *
*          BY A "/". *
*****;

LIBNAME CDRV 'C:\DATA\';
DATA CDRV.DATA;
  INFILE 'C:\DATA\DATA.ASC';
  INPUT
    X1 1-8 X2 10-17 X3 19-26 X4 28-35
    D1 37-44 D3 46-53 D2 55-62 D4 64-71
  /   D5 1-8 D7 10-17 D6 19-26 D8 28-35
    X5 37-44 X6 46-53 X7 55-62 X8 64-71
  /   Y1 1-8 X9 10-17 X10 19-26 Y2 28-35
    X11 37-44 X12 46-53 X13 55-59 RD1 61-68 RD2 70-77
  /   RD3 1-8 X14 10-17 X15 19-26;

LABEL
Y1 ='HH CALORIE INTAKE, CAPITA, DAY (24-HOUR RECALL DATA)'
Y2 ='HH CALORIE INTAKE, CAPITA, DAY (FOOD EXPENDITURE DATA)'
X1 ='RETAIL PRICE OF SHELLED CORN,KG (1984 PESOS; AVERAGE,BARRIO)'
X2 ='RETAIL PRICE OF MILLED RICE,KG (1984 PESOS; AVERAGE,BARRIO)'
X3 ='CULTIVATED AREA PER CAPITA (AVERAGE OF FOUR SURVEY ROUNDS)'
X4 ='ZERO-ONE DUMMY FOR PRESENCE OF ELECTRICITY FOR HOUSE'
X5 ='ZERO-ONE DUMMY FOR QUALITY OF FLOORING MATERIALS FOR HOUSE'
X6 ='ZERO-ONE DUMMY FOR QUALITY OF ROOFING MATERIALS FOR HOUSE'
X7 ='ZERO-ONE DUMMY FOR QUALITY OF MATERIALS USED FOR HOUSE WALLS'
X8 ='AGE OF HEAD OF HOUSEHOLD (IN MONTHS)'
X9 ='NUMBER OF HOUSEHOLD MEMBERS'
X10 ='LOG OF HHOLD TOTAL EXPENDITURES,WK,CAP (1984 PESOS, ROUND)'
X11 ='VALUE OF ALL ASSETS (1984 PESOS; AVERAGE OF ROUNDS 1 AND 4)'
X12 ='OWNED AREA PER CAPITA (AVERAGE OF FOUR SURVEY ROUNDS)'
X13 ='MUNICIPAL POPULATION DENSITY (PERSONS PER SQUARE KILOMETER)'
X14 ='YEARS IN SCHOOL, HEAD OF HOUSEHOLD'
X15 ='YEARS IN SCHOOL, SPOUSE OF HEAD OF HOUSEHOLD'
D1 ='% OF HH THAT ARE FEMALES < OR EQUAL TO 5 YRS OF AGE'
D2 ='% OF HH THAT ARE FEMALES > 5 YRS AND < OR = TO 11 YRS OF AGE'
D3 ='% OF HH THAT ARE FEMALES > 11 YRS AND < OR = 17 YRS OF AGE'
D4 ='% OF HH THAT ARE FEMALES > 17 YRS OF AGE'
D5 ='% OF HH THAT ARE MALES < OR EQUAL TO 5 YRS OF AGE'
D6 ='% OF HH THAT ARE MALES > 5 YRS AND < OR = TO 11 YRS OF AGE'
D7 ='% OF HH THAT ARE MALES > 11 YRS AND < OR = TO 17 YRS OF AGE'
D8 ='% OF HH THAT ARE MALES > THAN 17 YRS OF AGE'
RD1 ='ZERO-ONE DUMMY FOR FIRST ROUND SURVEY'
RD2 ='ZERO-ONE DUMMY FOR SECOND ROUND SURVEY'
RD3 ='ZERO-ONE DUMMY FOR THIRD ROUND SURVEY';

RUN;

PROC PRINT DATA=CDRV.DATA(OBS=10);
  VAR Y1 X1 X2 X3 X4 X5;
RUN;

```

Figure 3—Sample programs for reading ASCII files, in SPSS/PC+

3a—For free format

```

SET MORE = OFF.
SET LIS='READFREE.LIS'.
SET LOG='READFREE.LOG'.
*****
* PROGRAM:  READFREE.SPS  SOFTWARE: SPSS/PC+ 4.01  *
*          FILENAME      DESCRIPTION          *
* INPUTS:  DATA.ASC     ASCII FILE          *
* OUTPUTS: DATA.SYS     SPSS/PC+ SYSTEM FILE *
* PURPOSE: READ ASCII FILE INTO AN SPSS/PC+ SYSTEM *
*          FILE. THIS PROGRAM ASSUMES THAT THE DATA *
*          ARE IN FREE FORMAT (VARIABLES ARE          *
*          SEPARATED BY AT LEAST ONE SPACE).        *
*****

DATA LIST FREE FILE= 'DATA.ASC'
      /X1 X2 X3 X4 D1 D3 D2 'D4 D5 D7 D6 D8 X5 X6 X7 X8 Y1 X9 X10 Y2
      X11 X12 X13 RD1 RD2 RD3 X14 X15.

VARIABLE LABEL
Y1 'HH CALORIE INTAKE, CAPITA, DAY (24-HOUR RECALL DATA)'
Y2 'HH CALORIE INTAKE, CAPITA, DAY (FOOD EXPENDITURE DATA)'
X1 'RETAIL PRICE OF SHELLED CORN,KG (1984 PESOS; AVERAGE,BARRIO)'
X2 'RETAIL PRICE OF MILLED RICE,KG (1984 PESOS; AVERAGE,BARRIO)'
X3 'CULTIVATED AREA PER CAPITA (AVERAGE OF FOUR SURVEY ROUNDS)'
X4 'ZERO-ONE DUMMY FOR PRESENCE OF ELECTRICITY FOR HOUSE'
X5 'ZERO-ONE DUMMY FOR QUALITY OF FLOORING MATERIALS FOR HOUSE'
X6 'ZERO-ONE DUMMY FOR QUALITY OF ROOFING MATERIALS FOR HOUSE'
X7 'ZERO-ONE DUMMY FOR QUALITY OF MATERIALS USED FOR HOUSE WALLS'
X8 'AGE OF HEAD OF HOUSEHOLD (IN MONTHS)'
X9 'NUMBER OF HOUSEHOLD MEMBERS'
X10 'LOG OF HHOLD TOTAL EXPENDITURES,WK,CAP (1984 PESOS, ROUND)'
X11 'VALUE OF ALL ASSETS (1984 PESOS; AVERAGE OF ROUNDS 1 AND 4)'
X12 'OWNED AREA PER CAPITA (AVERAGE OF FOUR SURVEY ROUNDS)'
X13 'MUNICIPAL POPULATION DENSITY (PERSONS PER SQUARE KILOMETER)'
X14 'YEARS IN SCHOOL, HEAD OF HOUSEHOLD'
X15 'YEARS IN SCHOOL, SPOUSE OF HEAD OF HOUSEHOLD'
D1 '% OF HH THAT ARE FEMALES < OR EQUAL TO 5 YRS OF AGE'
D2 '% OF HH THAT ARE FEMALES > 5 YRS AND < OR = TO 11 YRS OF AGE'
D3 '% OF HH THAT ARE FEMALES > 11 YRS AND < OR = 17 YRS OF AGE'
D4 '% OF HH THAT ARE FEMALES > 17 YRS OF AGE'
D5 '% OF HH THAT ARE MALES < OR EQUAL TO 5 YRS OF AGE'
D6 '% OF HH THAT ARE MALES > 5 YRS AND < OR = TO 11 YRS OF AGE'
D7 '% OF HH THAT ARE MALES > 11 YRS AND < OR = TO 17 YRS OF AGE'
D8 '% OF HH THAT ARE MALES > THAN 17 YRS OF AGE'
RD1 'ZERO-ONE DUMMY FOR FIRST ROUND SURVEY'
RD2 'ZERO-ONE DUMMY FOR SECOND ROUND SURVEY'
RD3 'ZERO-ONE DUMMY FOR THIRD ROUND SURVEY'.

SAV OUT = 'DATA.SYS'.
N 10.
FORMATS ALL (F6.2).
LIST Y1 X1 X2 X3 X4 X5.
FINISH.

```

3b—For fixed format

```

SET MORE OFF.
SET LIS='READFIXD.LIS'.
SET LOG='READFIXD.LOG'.
*****
*   PROGRAM:   READFIXD.SPS   SOFTWARE: SPSS/PC+ 4.01   *
*   FILENAME   DESCRIPTION   *
*   INPUTS:    DATA.ASC     ASCII FILE         *
*   OUTPUTS:   DATA.SYS     SPSS/PC+ SYSTEM FILE    *
*   PURPOSE:   READ ASCII DATA FILE INTO SPSS/PC+   *
*             SYSTEM FILE. THIS PROGRAM ASSUMES THE  *
*             DATA ARE IN FIXED FORMAT. VARIABLES TO BE*
*             READ IN MUST BE IDENTIFIED BY A NAME   *
*             AND COLUMN LOCATION; SUBSEQUENT Records *
*             ARE DENOTED BY A "/".                 *
*****
DATA LIST FIXED FILE= 'DATA.ASC'
  /X1 1-8 X2 10-17 X3 19-26 X4 28-35 D1 37-44 D3 46-53 D2 55-62
    D4 64-71
  /D5 1-8 D7 10-17 D6 19-26 D8 28-35 X5 37-44 X6 46-53 X7 55-62
    X8 64-71
  /Y1 1-8 X9 10-17 X10 19-26 Y2 28-35 X11 37-44 X12 46-53 X13 55-59
    RD1 61-68 RD2 70-77
  /RD3 1-8 X14 10-17 X15 19-26.
VARIABLE LABEL
Y1 'HH CALORIE INTAKE, CAPITA, DAY (24-HOUR RECALL DATA)'
Y2 'HH CALORIE INTAKE, CAPITA, DAY (FOOD EXPENDITURE DATA)'
X1 'RETAIL PRICE OF SHELLED CORN,KG (1984 PESOS; AVERAGE,BARRIO)'
X2 'RETAIL PRICE OF MILLED RICE,KG (1984 PESOS; AVERAGE,BARRIO)'
X3 'CULTIVATED AREA PER CAPITA (AVERAGE OF FOUR SURVEY ROUNDS)'
X4 'ZERO-ONE DUMMY FOR PRESENCE OF ELECTRICITY FOR HOUSE'
X5 'ZERO-ONE DUMMY FOR QUALITY OF FLOORING MATERIALS FOR HOUSE'
X6 'ZERO-ONE DUMMY FOR QUALITY OF ROOFING MATERIALS FOR HOUSE'
X7 'ZERO-ONE DUMMY FOR QUALITY OF MATERIALS USED FOR HOUSE WALLS'
X8 'AGE OF HEAD OF HOUSEHOLD (IN MONTHS)'
X9 'NUMBER OF HOUSEHOLD MEMBERS'
X10 'LOG OF HHOLD TOTAL EXPENDITURES,WK,CAP (1984 PESOS, ROUND)'
X11 'VALUE OF ALL ASSETS (1984 PESOS; AVERAGE OF ROUNDS 1 AND 4)'
X12 'OWNED AREA PER CAPITA (AVERAGE OF FOUR SURVEY ROUNDS)'
X13 'MUNICIPAL POPULATION DENSITY (PERSONS PER SQUARE KILOMETER)'
X14 'YEARS IN SCHOOL, HEAD OF HOUSEHOLD'
X15 'YEARS IN SCHOOL, SPOUSE OF HEAD OF HOUSEHOLD'
D1 '% OF HH THAT ARE FEMALES < OR EQUAL TO 5 YRS OF AGE'
D2 '% OF HH THAT ARE FEMALES > 5 YRS AND < OR = TO 11 YRS OF AGE'
D3 '% OF HH THAT ARE FEMALES > 11 YRS AND < OR = 17 YRS OF AGE'
D4 '% OF HH THAT ARE FEMALES > 17 YRS OF AGE'
D5 '% OF HH THAT ARE MALES < OR EQUAL TO 5 YRS OF AGE'
D6 '% OF HH THAT ARE MALES > 5 YRS AND < OR = TO 11 YRS OF AGE'
D7 '% OF HH THAT ARE MALES > 11 YRS AND < OR = TO 17 YRS OF AGE'
D8 '% OF HH THAT ARE MALES > THAN 17 YRS OF AGE'
RD1 'ZERO-ONE DUMMY FOR FIRST ROUND SURVEY'
RD2 'ZERO-ONE DUMMY FOR SECOND ROUND SURVEY'
RD3 'ZERO-ONE DUMMY FOR THIRD ROUND SURVEY'.
SAV OUT = 'DATA.SYS'.
N 10.
FORMATS ALL (F6.2).
LIST Y1 X1 X2 X3 X4 X5.
FINISH.

```

WRITING DATA TO ASCII FORMAT FROM SOFTWARE DATA SET

Sometimes it is necessary to write data from a software data set to an ASCII (text) data set. Typically, this is done to transfer the data into a different software package. It is becoming more and more easy, however, to translate software data sets into a data set in different software without converting the data through ASCII (that is, without writing to ASCII from one software package and then reading the data into another software package). DBMSCopy is an example of software that performs such conversions directly. SAS PC and GAUSS-386 permit the user to write data out in free or fixed formats. SAS PC has programmed this option into its commands; GAUSS-386 requires the user to specify the format. SPSS/PC+ automatically writes the data out with each variable in a fixed position for each case, with each variable separated from other variables by at least one space. Thus, only one program for SPSS/PC+ (WRITASCI.SPS) has been included since this will create an ASCII file that is, in effect, both in a fixed and free format. There is more than one way to write a GAUSS program, but only one is included here. Figures 4, 5, and 6 are sample programs for writing data to an ASCII data set, for GAUSS-386, SAS PC, and SPSS/PC+, respectively.

Figure 4—Sample programs for writing data to an ASCII data set, in GAUSS-386

```

/*****
* PROGRAM: WRITASCI.G SOFTWARE: GAUSS-386 v3.0 *
* FILENAME DESCRIPTION *
* INPUTS: DATA.DAT *
* OUTPUTS: DATA2.ASC ASCII FILE *
* PURPOSE: CONVERT THE GAUSS-386 DATA SET DATA TO *
* THE ASCII FILE DATA2.ASC. *
*****/

FORMAT /RD 12,6;
OUTWIDTH 132;

OUTPUT FILE = DATA2.ASC RESET;
SCREEN OFF;

NAMES = GETNAME("DATA");
OPEN D = DATA.DAT;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
/* PRINT $NAMES';; WILL ADD THE VARIABLE NAMES TO THE BEGINNING
OF THE FILE. */
PRINT DATA;

OUTPUT FILE = DATA2.ASC OFF;

SYSTEM;

```

Figure 5—Sample programs for writing data to an ASCII data set, in SAS PC

5a—For free form

```

*****
* PROGRAM:   WRITFREE.SAS  SOFTWARE: SAS PC 6.04   *
*           FILENAME      DESCRIPTION             *
* INPUTS:    DATA.SSD    SAS PC FILE             *
* OUTPUTS:   DATA2.ASC   ASCII FILE             *
* PURPOSE:   WRITE AN ASCII FILE FROM A SAS PC SYSTEM *
*           FILE. THIS PROGRAM WRITES THE DATA IN FREE*
*           FORMAT (AT LEAST ONE SPACE BETWEEN   *
*           VARIABLES) TO A NEW ASCII FILE.      *
*****;

LIBNAME CDRV 'C:\DATA';

DATA _NULL_;
  SET CDRV.DATA;
  FILE 'C:\DATA\DATA2.ASC';
  PUT X1 X2 X3 X4 D1 D3 D2 D4 D5 D7 D6 D8 X5 X6 X7 X8 Y1 X9 X10 Y2
      X11 X12 X13 RD1 RD2 RD3 X14 X15;

RUN;

```

5b—For fixed form

```

*****
* PROGRAM:   WRITFIXD.SAS SOFTWARE: SAS PC 6.04   *
*           FILENAME      DESCRIPTION             *
* INPUTS:    DATA.SSD    SAS PC FILE             *
* OUTPUTS:   DATA2.ASC   ASCII FILE             *
* PURPOSE:   WRITE AN ASCII FILE FROM A SAS PC SYSTEM *
*           FILE. THIS PROGRAM WRITES THE DATA IN
*           FIXED FORMAT (EACH VARIABLE APPEARS IN
*           SAME COLUMN ON EACH CASE). THE '/'
*           INDICATES WRITE TO A NEW LINE.      *
*****;

LIBNAME CDRV 'C:\DATA';

DATA _NULL_;
  SET CDRV.DATA;
  FILE 'C:\DATA\DATA2.ASC';
  PUT
      X1 1-8 X2 10-17 X3 19-26 X4 28-35
      D1 37-44 D3 46-53 D2 55-62 D4 64-71
      / D5 1-8 D7 10-17 D6 19-26 D8 28-35
      X5 37-44 X6 46-53 X7 55-62 X8 64-71
      / Y1 1-8 X9 10-17 X10 19-26 Y2 28-35
      X11 37-44 X12 46-53 X13 55-59 RD1 61-68 RD2 70-77
      / RD3 1-8 X14 10-17 X15 19-26;

RUN;

```

Figure 6—Sample programs for writing data to an ASCII data set, in SPSS/PC+

```

SET MORE OFF.
SET LIS='WRITASCI.LIS'.
SET LOG='WRITASCI.LOG'.
*****
* PROGRAM: WRITASCI.SPS SOFTWARE: SPSS/PC+ 4.01 *
* FILENAME DESCRIPTION *
* INPUTS: DATA.SYS SPSS/PC+ FILE *
* OUTPUTS: DATA2.ASC ASCII FILE *
* PURPOSE: WRITE AN ASCII FILE FROM AN SPSS/PC+ *
* SYSTEM FILE. *
*****
* NOTE: THIS PROGRAM WRITES THE DATA IN FIXED FORMAT (EACH
* VARIABLE APPEARS IN THE SAME COLUMN FOR EACH CASE)
* SPECIFIED BY SPSS/PC+. SINCE SPSS/PC+ ADDS A SPACE BETWEEN
* EACH FIELD, THIS ASCII FILE IS ALSO IN FREE FORMAT.

GET FILE='DATA.SYS'.
SET RESULTS='DATA2.ASC'.
FORMATS ALL (F9.5).
WRITE /VARIABLES=X1 X2 X3 X4 D1 D3 D2 D4 D5 D7 D6 D8 X5 X6 X7 X8
Y1 X9 X10 Y2 X11 X12 X13 RD1 RD2 RD3 X14 X15.
FINISH.

```

EXERCISE: READING DATA AND REPLICATING TABLE 2

The first step in any econometric work is to ensure that the data has been correctly read and that the investigator can replicate any known sample statistics. This section provides three programs for reading the data and replicating the sample statistics presented in Table 2. It is assumed in each case that an appropriate program has been run to convert the file DATA.ASC to a data set corresponding to the software being used.

In the GAUSS-386 program (Figure 7), notice the use of the GETNAME command to create a vector of names associated with the columns of the data set. This vector is useful to have for reporting results that are associated with particular variable names. Note also the use of the VARINDXI option in the OPEN DATA command. VARINDXI creates indices of the form INAME that associate columns of the data set with the variable names. These are useful for designing data vectors and matrices. For example, if you wish to create a vector of data for the variable, Y1, you only need to use $Y1 = DATA[.,IY1]$.

For SPSS/PC+ (Figure 8), the descriptive statistics are reported in STATS.LIS (as specified). In SAS PC (Figure 9), the descriptive statistics are reported in STATS.LST (default), and in GAUSS-386, the descriptive statistics are reported in STATS.OUT (as specified).

Figure 8—Sample program for reading data and reporting descriptive statistics, in SAS PC

```

*****
* PROGRAM:   STATS.SAS      SOFTWARE: SAS PC 6.04   *
*           FILENAME      DESCRIPTION            *
* INPUTS:   DATA.SSD     TEST DATA SET         *
* PURPOSE:  COMPUTE SUMMARY STATISTICS TO COMPARE *
*           WITH TABLE 2.                        *
*****;

LIBNAME CDRV 'C:\DATA\';

PROC MEANS DATA=CDRV.DATA;
  VAR Y1 Y2
      X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15
      D1 D2 D3 D4 D5 D6 D7 D8
      RD1 RD2 RD3;
RUN;

```

Figure 9—Sample program for reading data and reporting descriptive statistics, in SPSS/PC

```

SET MORE OFF.
SET LIS = 'STATS.LIS'.
SET LOG = 'STATS.LOG'.
*****
* PROGRAM:   STATS.SPS     SOFTWARE: SPSS/PC+ 4.01 *
*           FILENAME      DESCRIPTION            *
* INPUTS:   DATA.SYS     TEST DATA SET         *
* PURPOSE:  COMPUTE SUMMARY STATISTICS TO COMPARE *
*           WITH TABLE 2.                        *
*****.

GET FILE = 'DATA.SYS'.
FORMATS ALL (F9.5).

DESC Y1 Y2
     X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15
     D1 D2 D3 D4 D5 D6 D7 D8
     RD1 RD2 RD3.
FINISH.

```

4 SPECIFICATION TESTS

TESTS FOR HETERO-SKEDASTICITY

The major consequence of heteroskedasticity (nonconstant variance of the stochastic disturbance term) is that it causes the OLS estimate of the stochastic error variance ($\hat{\sigma}^2$) to be biased, rendering hypothesis tests on coefficients invalid. Most tests for heteroskedasticity involve examining the regression residuals; the White test involves comparison of the OLS coefficient covariance matrix with a heteroskedasticity-consistent covariance matrix. The Goldfeld-Quandt, Breusch-Pagan, and White tests are described below. These tests are quite general. The White test is the most general in the sense that it requires no specification of a model of the heteroskedastic error-generating process. The Goldfeld-Quandt test requires only that the heteroskedasticity be related to one of the regressors; the Breusch-Pagan test requires that it be related to some set of regressors. If heteroskedasticity is detected, the usual practice is to specify a model by which the standard deviation of the stochastic disturbance can be estimated at each observation, then used in a "weighted least-squares" procedure. White's method produces an estimate of the variance-covariance matrix of coefficients that is consistent in the presence of heteroskedasticity so that tests on the OLS coefficients may be conducted. See the references for details.

The model is the usual one:

$$y = X\beta + \epsilon.$$

The hypothesis to be tested is as follows:

$$H_0: E[\epsilon_i^2] = \sigma^2 \text{ (constant variance—no heteroskedasticity);}$$

$$H_1: E[\epsilon_i^2] = \sigma_i^2 \text{ (heteroskedasticity).}$$

Goldfeld-Quandt Test

This older test is only applicable when there is a strong a priori reason to believe that the variance of the error term is explicitly related to one of the explanatory variables, say X_k . This test comprises the following steps:

- | | |
|--------|--|
| Step 1 | Reorder the data by magnitude of the observations on X_k , from smallest to largest. |
| Step 2 | Partition the ordered data set into three subsets, each of size $C = N/3$. Delete the middle subset, then denote the subset with small values of X_k as set 1 and the subset with large values of X_k as set 2. |

- Step 3 Perform OLS (using all of the regressors in X) on set 1 and set 2 separately and get the residual sum of squares (RSS) from each set.
- Step 4 If set 2 has the higher RSS, the estimated variance of the residuals is positively correlated with the size of X_k . Calculate $\hat{F} = RSS_2/RSS_1$. If Set 1 has the higher RSS (negative correlation between X and the estimated variance of the residuals), then calculate $\hat{F} = RSS_1/RSS_2$. The test statistic is

$$\hat{F} \sim F_{[(N-C-2K)/2, (N-C-2K)/2]}.$$

Compare to standard F-table; if $\hat{F} > F_{critical}$ at the desired level of significance, then reject H_0 of homoskedasticity.

The GAUSS-386 program (Figure 10) produces a Goldfeld-Quandt-statistic of 1.5164, with 541 numerator degrees of freedom and 542 denominator degrees of freedom. The P -value is 0.0000, indicating a strong rejection of the hypothesis of no heteroskedasticity. The SAS PC (Figure 11) and SPSS/PC+ (Figure 12) F -statistics differ slightly (although not enough to alter the conclusions, $\hat{F} = 1.4972$) because the programs select slightly different numbers of observations for the lower- and upper-thirds of the data set. The sample programs for this section use the same basic model that will be used in subsequent sections. It is assumed that the error variance is monotonically related to variable X_{10} .

NOTE: Some authors recommend using relatively large significance levels (say, 25 percent to 50 percent) for tests of heteroskedasticity such as the Goldfeld-Quandt test since its consequences are severe and consistent estimators are readily available.

Recommended References: Fomby, Hill, and Johnson (1984, 193–194); Goldfeld and Quandt (1965, 539–547); Greene (1990, 420); Griffiths, Hill, and Judge (1993, 498–499); Judge et al. (1984, 449); Kennedy (1985, 97; 1992, 118); Kmenta (1986, 292–294); Maddala (1988, 164).

Figure 10—Sample program for Goldfeld-Quandt test, in GAUSS-386

```

/*****
* PROGRAM:   GQTEST.G       SOFTWARE: GAUSS-386 V3.0 *
*           FILENAME       DESCRIPTION             *
* INPUTS:   DATA.DAT      GAUSS-386 DATA SET    *
* PURPOSE:  PERFORM THE GOLDFELD-QUANDT TEST.    *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = GQTEST.OUT RESET;

NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);

@----- ASSUME THAT HETEROSKEDASTICITY IS RELATED TO X10 -----@
@----- AND SORT ENTIRE DATA SET ACCORDINGLY -----@

DATA = SORTC(DATA,IX10);
Y = DATA[.,IY1];
X = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                        ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

@----- CHOOSE LOWER-THIRD AND UPPER-THIRD DATA SUBSETS -----@

NL = FLOOR(NCASE/3);
YL = Y[1:NL,.];
XL = X[1:NL,.];
NL = ROWS(XL);

NU = FLOOR(2*NCASE/3) + 1;
YU = Y[NU:NCASE,.];
XU = X[NU:NCASE,.];
NU = ROWS(XU);

@----- OLS REGRESSIONS ON DATA SUBSETS -----@

K = COLS(XL);
BL = INV(XL'XL)*XL'YL; @ BETAS @
E = YL - XL*BL; @ RESIDUALS @
RSSL = E'E; @ RESIDUAL SUM OF SQUARES @
SER = SQRT(INV(NL-K)*RSSL); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSSL/((NL-1)*(STDC(YL))^2); @ R-SQUARED @
COV = INV(NL-K)*RSSL*INV(XL'XL); @ COV MATRIX OF BETAS @
SE = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T = BL ./ SE; @ T-STATISTICS @
PT = 2*CDFTC(ABS(T),(NL-K)); @ P-VALUES @
PRN = BL ~ SE ~ T ~ PT; @ FOR PRINTING @

" ";
" ";
" ";
" OLS RESULTS FOR LOWER DATA SUBSET ";
" ";
" ";

```

(continued)

Figure 10—Continued

```

" NUMBER OF OBSERVATIONS =          ";; NL;
" ";
" STANDARD ERROR OF REGRESSION =     ";; SER;
" ";
" RESIDUAL SUM OF SQUARES =         ";; RSSL;
" ";
" R-SQUARED =                       ";; RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT ";;   PRN[1,.];
I       = 1;
DO WHILE I <= K-1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];;  FORMAT /M1 /RD 12,4; PRN[I+1,.];
  I       = I + 1;
ENDO;
" ";
"\f";

K       = COLS(XU);
BU      = INV(XU'XU)*XU'YU;           @ BETAS           @
E       = YU - XU*BU;               @ RESIDUALS       @
RSSU    = E'E;                      @ RESIDUAL SUM OF SQUARES @
SER     = SQRT(INV(NU-K)*RSSU);      @ STD ERROR OF REGRESSION @
RSQ     = 1 - RSSU/((NU-1)*(STDC(YU))^2); @ R-SQUARED       @
COV     = INV(NU-K)*RSSU*INV(XU'XU); @ COV MATRIX OF BETAS @
SE      = SQRT(DIAG(COV));          @ STD ERRS OF BETAS @
T       = BU ./ SE;                @ T-STATISTICS    @
PT      = 2*CDFTC(ABS(T), (NU-K));  @ P-VALUE        @
PRN     = BU ~ SE ~ T ~ PT;        @ FOR PRINTING   @

" ";
" ";
" ";
"   OLS RESULTS FOR UPPER DATA SUBSET ";
" ";
" ";
" NUMBER OF OBSERVATIONS =          ";; NU;
" ";
" STANDARD ERROR OF REGRESSION =     ";; SER;
" ";
" RESIDUAL SUM OF SQUARES =         ";; RSSU;
" ";
" R-SQUARED =                       ";; RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT ";;   PRN[1,.];
I       = 1;
DO WHILE I <= K-1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];;  FORMAT /M1 /RD 12,4; PRN[I+1,.];
  I       = I + 1;
ENDO;

```

(continued)

Figure 10—Continued

```

" ";
" ";

@-----          CALCULATION OF G/Q TEST STATISTIC          -----@

IF RSSL <= RSSU;
  F = RSSU/RSSL;
  NDF = NU-K;
  DDF = NL-K;
ELSE;
  F = RSSL/RSSU;
  NDF = NL-K;
  DDF = NU-K;
ENDIF;
PROB = CDFFC(F,NDF,DDF);

"  GOLDFELD/QUANDT RESULTS ";
" ";
" ";
"  NUMBER OF OBSERVATIONS IN LOWER DATA SET =";;  NL;
"  NUMBER OF OBSERVATIONS IN UPPER DATA SET =";;  NU;
" ";
"  RESIDUAL SUM OF SQUARES FOR LOWER REGRESSION =";;  RSSL;
"  RESIDUAL SUM OF SQUARES FOR UPPER REGRESSION =";;  RSSU;
" ";
"  G/Q F-STATISTIC = ";;  F; "          P-VALUE =";;  PROB;
" ";

"\f";

OUTPUT FILE = GQTEST.OUT OFF;
SYSTEM;

```

Figure 11—Sample program for Goldfeld-Quandt test, in SAS PC

```

*****
* PROGRAM:  GQTEST.SAS      SOFTWARE: SAS PC 6.04      *
*          FILENAME        DESCRIPTION              *
* INPUTS:   DATA.SSD      TEST DATA SET           *
* PURPOSE:  PERFORM GOLDFELD-QUANDT TEST.          *
*****;

LIBNAME CDRV 'C:\DATA\';

* WE SUSPECT THAT THE VARIANCE OF THE DISTURBANCE TERM IS RELATED TO X10;

* PROC RANK CREATES A NEW VARIABLE (RX10) WITH VALUES OF 0, 1, OR 2
* CORRESPONDING TO THREE EQUAL GROUPS;

PROC RANK DATA=CDRV.DATA OUT=DRANK GROUP=3;
  VAR X10;
  RANKS RX10;
RUN;

PROC REG DATA=DRANK;
  WHERE RX10 = 0;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

PROC REG DATA=DRANK;
  WHERE RX10 = 2;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

* TEST STATISTIC CALCULATION FROM OUTPUT;
* RSS1 = RESIDUAL SUM OF SQUARES FROM THE FIRST REGRESSION;
* RSS2 = RESIDUAL SUM OF SQUARES FROM THE SECOND REGRESSION;
* CONSTRUCT F = RSS1/RSS2 IF RSS1>RSS2, OR F = RSS2/RSS1 IF RSS2>RSS1;
* DEGREES OF FREEDOM = ((N-C-2*K)/2), ((N-C-2*K)/2);
* N=NUMBER OF OBSERVATIONS (1624), C = MIDDLE THIRD OF OBSERVATIONS
* DROPPED (538);
* K = NUMBER OF PARAMETERS IN MODEL (19). FOR THIS EXAMPLE, F=1.4972, AND THE
* NULL HYPOTHESIS OF NO HETEROSKEDASTICITY (WITH RESPECT TO X10) IS REJECTED;

```

Figure 12—Sample program for Goldfeld-Quandt test, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'GQTEST.LIS'.
SET LOG = 'GQTEST.LOG'.
*****
* PROGRAM:  GQTEST.SPS      SOFTWARE: SPSS/PC+ 4.01  *
*          FILENAME        DESCRIPTION            *
* INPUTS:  DATA.SYS      TEST DATA SET         *
* PURPOSE: PERFORM GOLDFELD-QUANDT TEST.         *
*****

GET FILE = 'DATA.SYS' .

* WE SUSPECT THAT THE VARIANCE OF THE DISTURBANCE TERM IS RELATED TO X10.

* RANK CREATES A NEW VARIABLE (RX10) WITH VALUES OF 1, 2, OR 3
* CORRESPONDING TO THREE EQUAL GROUPS.

RANK X10/NTILE (3) INTO RX10.

PROCESS IF ( RX10 = 1 ).
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                    D7 D8 RD1 RD2 RD3
                    /DEPENDENT=Y1
                    /METHOD=ENTER.

PROCESS IF ( RX10 = 3 ).
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                    D7 D8 RD1 RD2 RD3
                    /DEPENDENT=Y1
                    /METHOD=ENTER.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* RSS1 = RESIDUAL SUM OF SQUARES FROM THE FIRST REGRESSION.
* RSS2 = RESIDUAL SUM OF SQUARES FROM THE SECOND REGRESSION.
* CONSTRUCT F = RSS1/RSS2 IF RSS1>RSS2, OR F = RSS2/RSS1 IF RSS2>RSS1.
* DEGREES OF FREEDOM = ((N-C-2*K)/2), ((N-C-2*K)/2).
* N = NUMBER OF OBSERVATIONS (1624), C = MIDDLE THIRD OF OBSERVATIONS
* DROPPED (538).
* K = NUMBER OF PARAMETERS IN MODEL (19); FOR THIS EXAMPLE, F=1.4972, AND THE
* NULL HYPOTHESIS OF NO HETEROSKEDASTICITY (WITH RESPECT TO X10) IS REJECTED.
FINISH.

```

Breusch-Pagan Test This test assumes that the disturbance terms, ϵ_i , are normally and independently distributed. Moreover, the variances of ϵ_i are assumed to be of the form $\sigma^2 = f(Z\alpha)$, where Z is a set of p variables (these may be a subset of the X variables) thought to influence the heteroskedasticity (Z also includes a constant term) and α is a conformable vector of coefficients. This test does not depend on the functional form of f . The test evaluates whether the variables in Z have explanatory power for the variation in squared standardized residuals from the original model.

The model is the usual one:

$$y = X\beta + \epsilon.$$

The Breusch-Pagan test follows the following steps:

- | | |
|--------|--|
| Step 1 | Estimate the model by OLS and save the vector of residuals e . |
| Step 2 | Compute $\hat{\sigma}^2 = (1/N)\sum e_i^2$ and the $N \times 1$ vector v , where $v_i = e_i^2 / \hat{\sigma}^2$. |
| Step 3 | Specify the variables in Z , regress v on Z , and compute the explained sum of squares (ESS, sometimes called the regression or model sum of squares). |
| Step 4 | Calculate the statistic $Q = ESS/2$. Q is asymptotically chi-squared (χ^2) with $(p - 1)$ degrees of freedom. |
| Step 5 | Compare Q to $\chi_{critical}^2$ value at the desired level of significance. If $Q > \chi_{critical}^2$, then reject H_0 . |

In the sample programs (Figures 13 through 15), the same model is used as before and the variables in Z are selected to be identical with those in X . The Q value is 68.1561 (P -value = 0.0000) and again, the hypothesis of no heteroskedasticity is strongly rejected.

NOTE: As with the Goldfeld-Quandt test, some writers recommend using relatively large significance levels for the Breusch-Pagan test.

Recommended references: Breusch and Pagan (1979, 1287–1294); Fomby, Hill, and Johnson (1984, 195–196); Greene (1990, 421–422); Griffiths, Hill, and Judge (1993, 498–500); Judge et al. (1984, 446–447); Kennedy (1985, 97–98, 108; 1992, 118, 130–131); Kmenta (1986, 294–295); Maddala (1988, 164). *

Figure 13—Sample program for Breusch-Pagan test, in GAUSS-386

```

/*****
* PROGRAM:  BPTEST.G      SOFTWARE: GAUSS-386 V3.0  *
*          FILENAME      DESCRIPTION              *
* INPUTS:  DATA.DAT     TEST DATA SET          *
* PURPOSE: PERFORM BREUSCH-PAGAN TEST.          *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = BPTEST.OUT RESET;

NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);
Y = DATA[.,IY1];
X = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                        ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

K = COLS(X);

@-----@
                                OLS REGRESSION
                                @-----@
B = INV(X'X)*X'Y;                @ OLS BETAS                @
E = Y - X*B;                    @ OLS RESIDUALS        @
RSS = E'E;                      @ RESIDUAL SUM OF SQUARES @
SER = SQRT(INV(NCASE - K)*RSS);  @ S.E. OF REGRESSION   @
RSQ = 1 - RSS/((NCASE - 1)*(STDC(Y))^2); @ R-SQUARED           @
COV = INV(NCASE - K)*RSS*INV(X'X); @ VAR-COV MATRIX OF B @
SE = SQRT(DIAG(COV));           @ S.E. OF B ELEMENTS  @
T = B ./ SE;                   @ T-STATISTICS        @
PT = 2*CDFTC(ABS(T), (NCASE - K)); @ P-VALUES            @
PRN = B ~ SE ~ T ~ PT;          @ FOR PRINTING        @

@-----@
                                PRINT OLS RESULTS
                                @-----@
" ";
" ";
" ";
" OLS RESULTS";
" ";
" NUMBER OF OBSERVATIONS =      "; NCASE;
" STANDARD ERROR OF REGRESSION = "; SER;
" RESIDUAL SUM OF SQUARES =    "; RSS;
" R-SQUARED =                  "; RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT "; PRN[1,.];

I = 1;
DO WHILE I <= K-1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];  FORMAT /M1 /RD 12,4; PRN[I+1,.];
  I = I + 1;
ENDO;

"\f";

```

(continued)

Figure 13—Continued

```

@----- CONSTRUCTION OF STANDARDIZED SQUARED RESIDUALS -----@
G      = (E .^ 2)/(INV(NCASE)*E'E);

@----- CHOOSE REGRESSORS THAT EXPLAIN HETEROSKEDASTICITY -----@
@----- A COMMON CHOICE IS Z = X -----@

Z      = X;
K      = COLS(Z);
D      = INV(Z'Z)*Z'G;          @ B-P COEFFICIENTS          @
E      = G - Z*D;              @ RESIDUALS FROM AUX REG @
RSS    = E'E;                  @ B-P REG RSS          @
COV    = INV(NCASE - K)*RSS*INV(Z'Z); @ COV MATRIX FOR D COEFFS @
SE     = SQRT(DIAG(COV));      @ S.E. OF D ELEMENTS  @
T      = D ./ SE;              @ T-STATISTICS FOR D   @
PT     = 2*CDFTC(ABS(T), (NCASE - K)); @ P-VALUES             @
PRN    = D ~ SE ~ T ~ PT;      @ FOR PRINTING        @

GHAT   = Z*D;                  @ FITTED STANDARDIZED @
                               @ SQUARED RESIDUALS   @

ESS    = SUMC( (GHAT - MEANC(GHAT))^2 ); @ ESS FROM B-P REGRESSION @
Q      = ESS/2;                 @ B-P TEST STATISTIC   @
PCHI   = CDFCHIC(Q,K);          @ P-VALUE FOR Q        @

@----- PRINT B-P REGRESSION AND B-P TEST STATISTIC -----@
" ";
" ";
" ";
" AUXILIARY B-P REGRESSION RESULTS";
" ";
" NUMBER OF OBSERVATIONS =      ;;   NCASE;
" EXPLAINED SUM-OF-SQUARES =    ;;   ESS;
" ";
" VARIABLE          COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT ;;   PRN[1,.];

I      = 1;
DO WHILE I <= K-1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];;   FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;

" ";
" ";
" BREUSCH-PAGAN TEST STATISTIC:  Q = ;; Q;
" ";
" DEGREES OF FREEDOM = ;; K;
" ";
" P-VALUE =;; PCHI;

"\f";

OUTPUT FILE = BPTEST.OUT OFF;
SYSTEM;

```

Figure 14—Sample program for Breusch-Pagan test, in SAS PC

```

*****
* PROGRAM:  BPTTEST.SAS      SOFTWARE: SAS PC 6.04      *
* FILENAME  DESCRIPTION      *
* INPUTS:   DATA.SSD       TEST DATA SET           *
* PURPOSE:  PERFORM BREUSCH-PAGAN TEST.              *
*****;

LIBNAME CDRV 'C:\DATA\';

* VARIANCE OF DISTURBANCE TERM THOUGHT TO BE RELATED TO ALL
* EXPLANATORY VARIABLES;

PROC REG DATA=CDRV.DATA;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
  OUTPUT OUT=PRED R=RES;
RUN;

DATA E2;
  SET PRED;
  E2 = RES**2;
  CONSTANT = 1;
RUN;

PROC SUMMARY DATA=E2;
  VAR E2;
  ID CONSTANT;
  OUTPUT OUT=MEANE2 MEAN=MEANE2;
RUN;

DATA G;
  MERGE E2 MEANE2;
  BY CONSTANT;
  G = E2/MEANE2;
RUN;

PROC REG DATA=G;
  MODEL G=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

* TEST STATISTIC CALCULATION FROM OUTPUT;
* FROM THIS REGRESSION GET THE EXPLAINED SUM OF SQUARES (ESS);
* (SOMETIMES CALLED THE REGRESSION OR MODEL SUM OF SQUARES);
* THEN ESS/2 IS DISTRIBUTED CHI-SQUARED WITH P-1 DEGREES;
* OF FREEDOM, SO COMPARE TO CHI-SQUARED CRITICAL TABLE.;
* FOR THIS EXAMPLE, P = 19 AND THE CRITICAL VALUE = 28.869.;
* FOR THIS EXAMPLE, THE CHI-SQUARED TEST STATISTIC IS Q = 68.156.;
* REJECT NULL HYPOTHESIS OF NO HETEROSKEDASTICITY.;

```

Figure 15—Sample program for Breusch-Pagan test, in SPSS/PC+

```

SET MORE = OFF.
SET LIS = 'BPTEST.LIS'.
SET LOG = 'BPTEST.LOG'.
*****
*   PROGRAM:   BPTEST.SPS       SOFTWARE: SPSS/PC+ 4.01   *
*             FILENAME         DESCRIPTION              *
*   INPUTS:   DATA.SYS        TEST DATA SET           *
*   PURPOSE:  PERFORM BREUSCH-PAGAN TEST.              *
*****.

GET FILE = 'DATA.SYS' .

* VARIANCE OF DISTURBANCE TERM THOUGHT TO BE RELATED TO ALL
* EXPLANATORY VARIABLES.

REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                    D7 D8 RD1 RD2 RD3
                    /DEPENDENT=Y1
                    /METHOD=ENTER
                    /SAVE = RESID(RES) .

COMPUTE E2 = RES**2.
COMPUTE CONSTANT = 1.
SAVE OUT='E2.SYS' .

AGGREGATE OUTFILE='MEANE2.SYS'
  /BREAK=CONSTANT
  /MEANE2=MEAN(E2) .

JOIN MATCH FILE='E2.SYS'
  /TABLE='MEANE2.SYS'
  /BY CONSTANT.

COMPUTE G=(E2/MEANE2) .

REGRESSION VARIABLES = G X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                    D7 D8 RD1 RD2 RD3
                    /DEPENDENT=G
                    /METHOD=ENTER.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* FROM THIS REGRESSION GET THE EXPLAINED SUM OF SQUARES (ESS) (SOMETIMES
* CALLED THE REGRESSION OR MODEL SUM OF SQUARES).
* THEN ESS/2 IS DISTRIBUTED CHI-SQUARED WITH P-1 DEGREES.
* OF FREEDOM, SO COMPARE TO CHI-SQUARED CRITICAL TABLE.
* FOR THIS EXAMPLE, P = 19 AND THE CHI-SQUARED CRITICAL VALUE = 28.869.
* FOR THIS EXAMPLE, THE CHI-SQUARED TEST STATISTIC IS Q = 68.156.
* REJECT NULL HYPOTHESIS OF NO HETEROSKEDASTICITY.
FINISH.

```

The White Test The presence of heteroskedasticity makes the OLS variance-covariance matrix of coefficients inconsistent. White (1980) introduced an estimated variance-covariance matrix for the OLS coefficients that is consistent under heteroskedasticity. White also introduced a test statistic for heteroskedasticity based on the extent to which the OLS variance-

covariance matrix departs from White's heteroskedasticity-consistent covariance matrix.

One great advantage of White's procedure is that it produces an estimator for the variance-covariance matrix of coefficients that is consistent in the presence of heteroskedasticity, so that tests regarding the coefficients can be conducted without having to first correct for the heteroskedasticity. However, the White test may not be as powerful as some alternative tests that use more specific information about the form of the heteroskedasticity.

White's heteroskedasticity-consistent covariance matrix and the original form of his test are clearly laid out in several references, including those listed below. The test using the full set of explanatory variables is only presented in GAUSS-386 (Figure 16). This is because it is quite time-consuming to compute White's test manually in SPSS/PC+ and SAS PC for anything but a small set of explanatory variables (see Figures 17 and 18). While SAS PC has options for computing the heteroskedasticity-consistent covariance matrix and White's test automatically (ACOV SPEC), the SPEC algorithm appears to have a bug that a patch could not completely correct.

The GAUSS-386 program has two parts: first, the heteroskedasticity-consistent covariance matrix is computed, then the test for heteroskedasticity is conducted. Note that the investigator must be vigilant to avoid introducing redundancies among the constructed regressors for this test, especially if dummy variables are present.

The procedure for computing the test is as follows:

- | | |
|--------|--|
| Step 1 | Perform ordinary least squares on the model, save the residual vector e , and construct an $N \times 1$ vector of squared residuals, e^2 . |
| Step 2 | Compute the squares and cross-products of all regressors, deleting all redundancies. The obvious redundancies are those produced by the constant term and dummy variables. Your final set of regressors should include the original variables and all nonredundant squares and cross-products. |
| Step 3 | Regress the squared residuals, e^2 , on the regressors from step 2, using OLS. Retain the R^2 from this auxiliary regression. |
| Step 4 | Compute the test statistic, $W = N \times R^2$. |
| Step 5 | W will be asymptotically distributed χ^2 , with degrees of freedom equal to the number of regressors in step 3. If $W > \chi^2_{critical}$, the null hypothesis of no heteroskedasticity is rejected. |

The sample GAUSS-386 program produces the White heteroskedasticity-consistent covariance matrix. Notice that the square roots of its diagonal elements are quite different from the OLS standard

errors; it is expected that a formal test of the differences will find them significant. The test statistic, W , is 192.384 ($df = 183$). The null hypothesis of no heteroskedasticity is rejected.

The SAS PC and SPSS/PC+ programs for the reduced explanatory variable set produce a test statistic, $W = 24.3031$ ($df = 120$). Again, the null hypothesis of no heteroskedasticity is rejected.

Recommended references: Fomby, Hill, and Johnson (1984, 196); Greene (1990, 403-404); Kennedy (1985; 98, 108; 1992, 90, 118, 130-131); Kmenta (1986, 295-296); Maddala (1988, 162); Messer and White (1984, 181-184); White (1980, 817-838).

Figure 16—Sample program for White test, in GAUSS-386

```

/*****
* PROGRAM: WHITE.G SOFTWARE: GAUSS-386 V3.0 *
* FILENAME DESCRIPTION *
* CONSISTENT STANDARD ERRORS *
* INPUTS: DATA.DAT GAUSS-386 DATA SET *
* PURPOSE: CONSTRUCT ESTIMATES OF VARIANCE-COVARIANCE *
* MATRIX THAT ARE CONSISTENT IN PRESENCE OF *
* HETEROSKEDASTICITY AND DO WHITE TEST. *
* THIS PROGRAM RUNS ABOUT AN HOUR ON A 386- *
* 25 MHz MACHINE WITH 4 MB RAM. IT USES *
* EXTENDED MEMORY EXTENSIVELY, HENCE ITS *
* LONG RUN TIME. *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = WHITE.OUT RESET;
NAMES = GETNAME("DATA");
OPEN D = DATA VARINDEXI;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);

Y = DATA[.,IY1];
X = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

@----- OLS ESTIMATION -----@

K = COLS(X);

B = INV(X'X)*X'Y; @ BETAS @
E = Y - X*B; @ RESIDUALS @
RSS = E'E; @ RESIDUAL SUM OF SQUARES @
SER = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
OLSC = INV(NCASE-K)*RSS*INV(X'X); @ OLS COV MATRIX @
SE = SQRT(DIAG(OLSC)); @ STD ERRS OF BETAS @
T = B ./ SE; @ T-STATISTICS FOR BETAS @
PT = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES @
PRN = B ~ SE ~ T ~ PT; @ FOR PRINTING @

```

(continued)

Figure 16—Continued

```

" ";
" OLS RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS      =   ;;   NCASE;
" ";
" STANDARD ERROR OF REGRESSION =   ;;   SER;
" ";
" RESIDUAL SUM OF SQUARES     =   ;;   RSS;
" ";
" R-SQUARED                   =   ;;   RSQ;
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT ";;   PRN[1,.];

I      = 1;
DO WHILE I <= K - 1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];   FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";

@-----      SQUARE THE RESIDUAL FOR EACH OBSERVATION.      -----@
S      = E.^2;

@-----      HETEROSKEDASTICITY-CONSISTENT COVARIANCE MATRIX      -----@
HETCM  = ZEROS(K,K);

I      = 1;
DO WHILE I <= NCASE;

  HETCT  = S[I,.]*((X[I,.])'(X[I,.]));

  HETCM  = HETCM + HETCT;

  I      = I + 1;

ENDO;

HETC   = INV(X'X)*HETCM*INV(X'X);
HSE    = SQRT(DIAG(HETC));

" ";
" ";
" HETEROSKEDASTICITY-CONSISTENT STD ERRS"; HSE;

" ";
CLEAR HSE HETC HETCM HETCT PRN PT T SE OLSC RSQ SER RSS E B Y;

@-----      CONSTRUCT VARIABLES FOR WHITE-AUGMENTED      -----@
@-----      REGRESSION. NOTE THAT REDUNDANCIES      -----@
@-----      ARE AVOIDED BY FIRST CONSTRUCTING THE      -----@
@-----      SQUARES AND CROSS-PRODUCTS OF ALL NONDUMMY      -----@
@-----      VARIABLES, THEN CONCATENATING THE DUMMIES AND      -----@
@-----      THEIR INTERACTIONS WITH THE REGULAR REGRESSORS.      -----@

```

(continued)

Figure 16—Continued

```

X      = X[.,1:(K-3)];
K      = COLS(X);
AUGX   = X;
I      = 2;          OUTPUT FILE = WHITE.OUT OFF;
DO WHILE I <= K;
  AUGX  = AUGX ~ (X[.,I] .* X[.,I:K]);
  "LOOP =";; I;
  I     = I + 1;
ENDDO;          OUTPUT FILE = WHITE.OUT ON;
W      = AUGX ~ (DATA[.,IRD1] .* X)
        ~ (DATA[.,IRD2] .* X)
        ~ (DATA[.,IRD3] .* X);
CLEAR  AUGX X DATA;
K      = COLS(W);
D      = INV(W'W)*W'S;
ES     = S - W*D;
CLEAR  W;
RSSW   = ES'ES;
RSQW   = 1 - RSSW / ((NCASE-1) * (STDC(S))^2);
DF     = K - 1;
WTEST  = NCASE*RSQW;
PW     = CDFCHIC(WTEST,DF);
" ";
"WTEST =";; WTEST;; "   DF =";; DF;; "   P-VALUE =";; PW;
"\f";
OUTPUT FILE = WHITE.OUT OFF;
SYSTEM;

```

Figure 17—Sample program for White test, in SAS PC

```

*****
* PROGRAM:   WHITE.SAS      SOFTWARE: SAS PC 6.04   *
* FILENAME  DESCRIPTION    *
* INPUTS:   DATA.SSD      TEST DATA SET        *
* PURPOSE:  CONSTRUCT ESTIMATES OF VARIANCE-     *
*           COVARIANCE MATRIX THAT ARE CONSISTENT *
*           IN PRESENCE OF HETEROSKEDASTICITY, WITH *
*           A REDUCED SET OF EXPLANATORY VARIABLES *
*****;

LIBNAME CDRV 'C:\DATA\';
PROC REG DATA=CDRV.DATA;
  MODEL Y1=X1 X2 X9 X10;
  OUTPUT OUT=RDATA R=RES;
RUN;

DATA XRDATA;
  SET RDATA;
  RESSQ = RES**2;

* EACH VARIABLE SQUARED;
  ZX1 = X1**2;
  ZX2 = X2**2;
  ZX9 = X9**2;
  ZX10 = X10**2;

* INTERACTION WITH X1;
  X1X2 = X1*X2;
  X1X9 = X1*X9;
  X1X10 = X1*X10;

* INTERACTION WITH X2;
  X2X9 = X2*X9;
  X2X10 = X2*X10;

* INTERACTION WITH X9;
  X9X10 = X9*X10;
RUN;

PROC REG DATA=XRDATA;
  MODEL RESSQ=X1 X2 X9 X10
          ZX1 ZX2 ZX9 ZX10
          X1X2 X1X9 X1X10
          X2X9 X2X10
          X9X10 ;
RUN;

* TEST STATISTIC CALCULATION FROM OUTPUT;
* THE WALD TEST STATISTIC, W, EQUALS R-SQUARED FROM THE SECOND REGRESSION
* (WHICH CONTAINS THE TRANSFORMATIONS OF X1, X2, X9, AND X10)
* MULTIPLIED BY THE NUMBER OF OBSERVATIONS USED IN THE REGRESSION. W IS
* DISTRIBUTED AS CHI-SQUARED WITH K(K+1)/2 DEGREES OF FREEDOM (DF). IF W
* IS GREATER THAN THE CRITICAL CHI-SQUARED VALUE, THEN THE NULL HYPOTHESIS
* OF HOMOSKEDASTICITY IS REJECTED.;

* FOR THIS EXAMPLE N=1624, K=15, R-SQ= 0.01496, AND W=24.295. DF=120. THE
* NULL HYPOTHESIS OF NO HETEROSKEDASTICITY IS REJECTED.;

```

Figure 18—Sample program for White test, in SPSS/PC+

```

SET MORE=OFF.
SET LIS = 'WHITE.LIS'.
SET LOG = 'WHITE.LOG'.
*****
* PROGRAM: WHITE.SPS SOFTWARE: SPSS/PC+ 4.01 *
* FILENAME DESCRIPTION *
* INPUTS: DATA.SYS TEST DATA SET *
* PURPOSE: CONSTRUCT ESTIMATES OF VARIANCE- *
* COVARIANCE MATRIX THAT ARE CONSISTENT *
* IN PRESENCE OF HETEROSKEDASTICITY, WITH *
* A REDUCED SET OF EXPLANATORY VARIABLES *
*****.

GET FILE = 'DATA.SYS' .
REGRESSION VARIABLES =
    Y1 X1 X2 X9 X10
    /DEPENDENT=Y1
    /METHOD=ENTER
    /SAVE RESID(RES).

COMPUTE RESSQ = RES**2.

* EACH VARIABLE SQUARED.
COMPUTE ZX1 = X1**2.
COMPUTE ZX2 = X2**2.
COMPUTE ZX9 = X9**2.
COMPUTE ZX10 = X10**2.

* INTERACTION WITH X1.
COMPUTE X1X2 = X1*X2.
COMPUTE X1X9 = X1*X9.
COMPUTE X1X10 = X1*X10.

* INTERACTION WITH X2.
COMPUTE X2X9 = X2*X9.
COMPUTE X2X10 = X2*X10.

* INTERACTION WITH X9.
COMPUTE X9X10 = X9*X10.

REGRESSION VARIABLES =
    RESSQ X1 X2 X9 X10
    ZX1 ZX2 ZX9 ZX10
    X1X2 X1X9 X1X10
    X2X9 X2X10
    X9X10
    /DEPENDENT=RESSQ
    /METHOD=ENTER.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* THE WALD TEST STATISTIC, W, EQUALS R-SQUARED FROM THE SECOND REGRESSION
* (WHICH CONTAINS THE TRANSFORMATIONS OF X1, X2, X9, AND X10)
* MULTIPLIED BY THE NUMBER OF OBSERVATIONS USED IN THE REGRESSION. W IS
* DISTRIBUTED AS CHI-SQUARED WITH K(K+1)/2 DEGREES OF FREEDOM (DF). IF W
* IS GREATER THAN THE CRITICAL CHI-SQUARED VALUE, THEN THE NULL HYPOTHESIS
* OF HOMOSKEDASTICITY IS REJECTED.

* FOR THIS EXAMPLE N = 1624, K = 15, R - SQ = 0.01496, W = 24.295, AND DF = 120.
* THE NULL HYPOTHESIS OF NO HETEROSKEDASTICITY IS REJECTED.
FINISH.

```

NORMALITY OF RESIDUALS: THE JARQUE-BERA TEST

If the elements of the disturbance vector are not normally distributed, the OLS estimators for β are still best linear unbiased, but the usual t -tests and F -tests are no longer appropriate, and appropriate asymptotically justified tests should be used.

The Jarque-Bera test checks whether the skewness (symmetry) and kurtosis (fatness of tails) of the distribution of residuals matches the skewness and kurtosis expected under the null hypothesis that the disturbances are normally distributed. Skewness is measured by $\sqrt{\beta_1} = \mu_3/\mu_2^{3/2}$ and kurtosis is measured by $\beta_2 = \mu_4/\mu_2^2$, where estimates of the moments μ_r are given by $1/N \sum e_i^r$ ($r = 2, 3, 4$). Under the null hypothesis that the disturbances are normally distributed, $\beta_1 = 0$ and $\beta_2 = 3$. Thus, the null hypothesis is

$$H_0: \beta_1 = 0 \text{ and } \beta_2 = 3.$$

The alternative hypothesis is that the disturbances are not normal and belong to a class of distributions called the "Pearson family."

The test statistic is

$$\eta = N[(z_1/6) + (z_2 - 3)^2/24],$$

where z_1 and z_2 are the estimates of β_1 and β_2 , and N is the number of observations. η has a χ^2 distribution with 2 degrees of freedom. Note that $\eta = 0$ if $z_1 = 0$ and $z_2 = 3$.

Construction of the test proceeds by the following steps:

- | | |
|--------|--|
| Step 1 | Estimate the model by OLS and save the residual vector, e . |
| Step 2 | Calculate the sample estimates of the second, third, and fourth moments of the residuals about their mean (which is zero by construction): |

$$\hat{\mu}_r = (1/N) \sum e_i^r \quad (r = 2, 3, 4),$$

where μ_r is the r^{th} moment about the mean and the e_i 's are the OLS residuals. Denote these as $\hat{\mu}_2, \hat{\mu}_3, \hat{\mu}_4$, respectively.

- | | |
|--------|--|
| Step 3 | Calculate $z_1 = [\hat{\mu}_3/\hat{\mu}_2^{3/2}]^2$ and $z_2 = \hat{\mu}_4/\hat{\mu}_2^2$. |
| Step 4 | Calculate η and compare to the critical value at desired level of significance with two degrees of freedom. If $\eta > \chi_{critical}^2$ then reject the null hypothesis. This would imply that the disturbance terms are <i>not</i> normally distributed. |

For this model, the Jarque-Bera test statistic is 274.2360 (P -value = 0.0000) and the null hypothesis of normality of disturbance terms is rejected.

Figures 19, 20, and 21 are sample programs for the Jarque-Bera test, in GAUSS-386, SAS PC, and SPSS/PC+, respectively.

NOTE: For an additional normality test, see Shapiro and Wilks (1965) and Shapiro, Wilks, and Chen (1968).

Recommended references: Bowman and Shenton (1975, 243–250); Jarque and Bera (1981); Kennedy (1992, 79); Kmenta (1986, 260–267).

Figure 19—Sample program for Jarque-Bera test, in GAUSS-386

```

/*****
* PROGRAM:   JBTEST.G       SOFTWARE: GAUSS-386 V3.0 *
*           FILENAME       DESCRIPTION           *
* INPUTS:   DATA.DAT      GAUSS-386 DATA SET  *
* PURPOSE:  EXECUTE AND REPORT THE JARQUE-BERA TEST *
*           FOR NORMALITY OF DISTURBANCES.      *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = JBTEST.OUT RESET;
NAMES      = GETNAME("DATA");
OPEN D     = DATA VARINDXI;
NCASE      = ROWSF(D);
DATA       = READR(D,NCASE);
F          = CLOSE(D);

Y          = DATA[.,IY1];

X          = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES      = NAMES{IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                    ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3, .};

@-----@                               @-----@
K          = COLS(X);

B          = INV(X'X)*X'Y;                @ BETAS @
E          = Y - X*B;                    @ RESIDUALS @
RSS        = E'E;                        @ RESIDUAL SUM OF SQUARES @
SER        = SQRT(INV(NCASE-K)*RSS);      @ STD ERROR OF REGRESSION @
RSQ        = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV        = INV(NCASE-K)*RSS*INV(X'X);   @ OLS COVARIANCE MATRIX @
SE         = SQRT(DIAG(COV));             @ STD ERRS OF BETAS @
T          = B ./ SE;                    @ T-STATISTICS FOR BETAS @
PT         = 2*CDFTC(ABS(T), (NCASE-K));  @ P-VALUES @
PRN        = B ~ SE ~ T ~ PT;            @ FOR PRINTING @

" ";
" ";
" ";
" OLS RESULTS ";
" ";
" ";

```

(continued)

Figure 19—Continued

```

" NUMBER OF OBSERVATIONS      = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES     = ";; RSS;
" R-SQUARED                   = ";; RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT ";; PRN[1,.];

I      = 1;
DO WHILE I <= K -1;
FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

I      = I + 1;
ENDO;
" ";

@----- COMPUTATION OF SECOND, THIRD, AND FOURTH MOMENTS -----@
@----- OF OLS RESIDUALS -----@

E2     = E^2;
E3     = E^3;
E4     = E^4;

U2     = (SUMC(E2))/NCASE;
U3     = (SUMC(E3))/NCASE;
U4     = (SUMC(E4))/NCASE;

Z1     = (U3/(U2^(3/2)))^2;
Z2     = U4/(U2^2);

ETA    = NCASE*((Z1/6) + ((Z2-3)^2)/24));

PCHI   = CDFCHIC(ETA,2);

" ";
" JARQUE-BERA STATISTIC ETA =";; ETA;
" ";
" P-VALUE                =";; PCHI;

"\f";

OUTPUT FILE = JBTEST.OUT OFF;
SYSTEM;

```

Figure 20—Sample program for Jarque-Bera test, in SAS PC

```

*****
* PROGRAM:   JBTEST.SAS      SOFTWARE: SAS PC 6.04      *
*           FILENAME        DESCRIPTION              *
* INPUTS:   DATA.SSD      TEST DATA SET            *
* PURPOSE:  EXECUTE AND REPORT THE JARQUE-BERA TEST *
*           FOR NORMALITY OF DISTURBANCES.          *
*****;

LIBNAME CDRV 'C:\DATA\';
PROC REG DATA=CDRV.DATA;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
  OUTPUT OUT=JARQUE R=RES;
RUN;

DATA JARQUE2;
  SET JARQUE;
  E2=RES**2;
  E3=RES**3;
  E4=RES**4;
  CONST=1;
RUN;

PROC SUMMARY DATA=JARQUE2;
  VAR E2 E3 E4 CONST;
  OUTPUT OUT=RESSUM SUM=SUME2 SUME3 SUME4 NCASE;
RUN;

DATA CALC;
  SET RESSUM;
  MU2=SUME2/NCASE;
  MU3=SUME3/NCASE;
  MU4=SUME4/NCASE;
  Z1 = ((MU3)/(MU2**(3/2)))**2;
  Z2 = MU4/(MU2**2);
  ETA = NCASE*((Z1/6)+(((Z2-3)**2)/24));
RUN;

PROC PRINT DATA=CALC;
  VAR ETA;
RUN;
* TEST STATISTIC CALCULATION FROM OUTPUT.
* ETA IS THE TEST STATISTIC AND IS DISTRIBUTED AS CHI-SQUARED WITH TWO
* DEGREES OF FREEDOM. IF ETA IS GREATER THAN THE CRITICAL CHI-SQUARED
* VALUE, THEN REJECT THE NULL HYPOTHESIS OF NORMALLY DISTRIBUTED RESIDUALS.
* ETA IN THIS EXAMPLE IS 274.24, WHICH IS LARGER THAN THE CRITICAL CHI-
* SQUARED VALUE. NORMALITY IS REJECTED.;

```

Figure 21—Sample program for Jarque-Bera test, in SPSS/PC+

```

SET MORE = OFF.
SET LIS = 'JBTEST.LIS'.
SET LOG = 'JBTEST.LOG'.
*****
* PROGRAM:   JBTEST.SPS      SOFTWARE: SPSS/PC+ 4.01   *
*           FILENAME        DESCRIPTION              *
* INPUTS:   DATA.SYS      TEST DATA SET           *
* PURPOSE:  EXECUTE AND REPORT THE JARQUE-BERA TEST *
*           FOR NORMALITY OF DISTURBANCES.         *
*****

GET FILE = 'DATA.SYS' .
REGRESSION VARIABLES = Y1, X1 X2, X8 X9 X10 X13 X14 X15,
                    D1 D2 D3 D5 D6 D7 D8, RD1 RD2 RD3
                    /DEPENDENT=Y1
                    /METHOD=ENTER
                    /SAVE RESID(RES).

COMPUTE E2 = RES**2.
COMPUTE E3 = RES**3.
COMPUTE E4 = RES**4.
COMPUTE CONST = 1.
AGGREGATE OUTFILE = *
                    /BREAK=CONST
                    /NCASE = NU(RES)
                    /SUME2 SUME3 SUME4 = SUM(E2 E3 E4).

COMPUTE MU2 = SUME2/NCASE.
COMPUTE MU3 = SUME3/NCASE.
COMPUTE MU4 = SUME4/NCASE.
COMPUTE Z1 = ((MU3)/(MU2**(3/2)))*2.
COMPUTE Z2 = MU4/MU2**2.
COMPUTE ETA = NCASE*((Z1/6)+(((Z2-3)**2)/24)).
LIST ETA.
* TEST STATISTIC CALCULATION FROM OUTPUT.
* ETA IS THE TEST STATISTIC AND IS DISTRIBUTED AS CHI-SQUARED WITH TWO
* DEGREES OF FREEDOM. IF ETA IS GREATER THAN THE CRITICAL CHI-SQUARED
* VALUE, THEN REJECT THE NULL HYPOTHESIS OF NORMALLY DISTRIBUTED RESIDUALS.
* ETA IN THIS EXAMPLE IS 274.24, WHICH IS LARGER THAN THE CRITICAL CHI-
* SQUARED VALUE. NORMALITY IS REJECTED.
FINISH.

```

ERRORS IN VARIABLES

A crucial assumption of the classical linear regression model is that the elements of the X matrix of regressors are nonstochastic. If any of the regressors are stochastic, then the problem of simultaneity bias or endogeneity may be faced. One common source of endogeneity is measurement error in the regressors.

There is little doubt that almost all observed variables are measured with error. While the emergence of extensive household surveys represents a wealth of information at the level of the household and individual, the possibility and consequences of measurement error in those data should be considered.

This discussion focuses on the simple linear regression model, that is, the model with a single regressor. The extension to the multiple regression context is straightforward and is illustrated in the sample programs (Figures 22 through 24).

Assume that

$$y_i = \alpha + \beta x_i + \epsilon_i \quad (1)$$

denotes the true model and that both x and y are measured with error. Let the errors be μ and v , respectively. Assume that the errors are normally distributed, with mean zero, and with constant variances so that

$$v_i \sim N(0, \sigma_v^2),$$

and

$$\mu_i \sim N(0, \sigma_\mu^2).$$

Moreover, assume that v and μ are uncorrelated with each other and are uncorrelated with all elements of x .

Now write

$$y_i^* = y_i + v_i,$$

and

$$x_i^* = x_i + \mu_i,$$

where an asterisk denotes an *observed* as opposed to a *true* value.

Rewriting equation 1 gives

$$y_i^* - v_i = \alpha + \beta (x_i^* - \mu_i) + \epsilon_i$$

or

$$y_i^* = \alpha + \beta x_i^* + w_i, \quad (2)$$

where

$$w_i = \epsilon_i + v_i - \beta \mu_i.$$

If x is measured with error, then the OLS assumption, $\text{cov}(w, x^*) = 0$, is violated because x^* and w both contain μ . In fact, the covariance between the stochastic regressors, x^* , and the error term is $-\beta \sigma_\mu^2$ (see Maddala 1988, 381 for details), and the estimated coefficient on β is biased toward zero.

In the multiple regression framework, the coefficient of the erroneously measured regressor is also biased toward zero. In addition, the coefficients on the remaining regressors are biased, but establishing the signs of the biases is more complicated.

The consequences of measurement error on y as opposed to x are very different. For example, if x is not measured with error, then measurement error in the dependent variable, y , is merely absorbed into the additive error term ($\epsilon + v$), which does not violate any of the assumptions of the classical OLS model.

Below, two tests that examine the importance of measurement error in regressors are discussed.

The Hausman Test The Hausman test takes advantage of the instrumental variables (IV) estimator, which (with appropriate instruments) is consistent in the presence of measurement error. Under the null hypothesis of no measurement error, the IV estimator is consistent but inefficient, while OLS is consistent and efficient. The essence of the Hausman test is to determine whether the difference between the OLS and IV estimators is statistically significant.

Now return to the multiple linear regression model,

$$y = X\beta + \epsilon,$$

and assume that the k^{th} variable in X is measured with error. As a consequence, *all* elements of the OLS estimator of β are biased.

The Hausman test is implemented by first constructing an IV estimator for the model. The existence of a matrix of L additional regressors that are highly correlated with X_k but uncorrelated with ϵ is assumed. A common method for constructing instruments is to regress the matrix X on a set of regressors Z that includes all variables in X *except* X_k and all of the additional regressors in L , so that Z has $(K + L - 1)$ regressors. The fitted value of X_k is then used as an instrument for X_k . The columns of X excluding X_k are simply replicated, but the k^{th} column is replaced by fitted values. Call this matrix \hat{X} . The instrumental variables estimator is then

$$\hat{\beta}_{IV} = (\hat{X}'\hat{X})^{-1}\hat{X}'y.$$

Let $V_{IV} = (\hat{X}'\hat{X})^{-1}$. Then a consistent estimator for the asymptotic variance-covariance matrix is

$$\hat{\sigma}^2 V_{IV},$$

where

$$\hat{\sigma}^2 = e'e / (N - K),$$

with

$$e = y - X\hat{\beta}_{IV}.$$

Notice that X is used here rather than \hat{X} . By comparison, the OLS estimator is $\hat{\beta}_0$, and V_0 is defined as $(X'X)^{-1}$.

The difference between the OLS and IV estimators is defined as

$$q = \hat{\beta}_0 - \hat{\beta}_{IV}.$$

Finally, the Hausman statistic is defined:

$$W = \{q_k' [V_{IV} - V_0]_k^{-1} q_k\} / \hat{\sigma}^2,$$

where σ^2 may be estimated either from the OLS residuals or from the IV residuals, and where q_k is the k^{th} element of q and $[V_{IV} - V_0]_k^{-1}$ is the k^{th} diagonal element of $[V_{IV} - V_0]^{-1}$. Many presentations of this test statistic do not indicate that it is constructed with the subvectors and submatrices designated by k . As Griffiths, Hill, and Judge (1993, 476) point out, those presentations assume that Z and X have no columns in common. When they do have columns in common, then the test statistic is constructed with the subvectors and submatrices that correspond with the columns of X not also in Z , namely the k^{th} column that has been replaced by fitted values.

W is asymptotically chi-square, with one degree of freedom. Sample values of W that exceed the selected critical value indicate significant differences between the OLS and IV estimators, hence indicate the presence of measurement error (or other source of endogeneity). Please refer to the references for cases in which more than one regressor is measured with error.

The Hausman test may be implemented in the following steps:

Step 1 Regress X on the set of instrumental variables Z and retain the fitted values:

$$\hat{X} = Z(Z'Z)^{-1}Z'X.$$

Step 2 Regress y on the set of instruments, \hat{X} , to give

$$\hat{\beta}_{IV} = (\hat{X}'\hat{X})^{-1}\hat{X}'y.$$

Step 3 Calculate the Hausman statistic as described above.

If the Hausman statistic is statistically significant, then reject the hypothesis of no endogeneity and use the instrumental variables estimates. Otherwise, the OLS estimates are suitable.

The Hausman-Wu Test An alternative approach to testing for endogeneity of a single variable in X is provided by the Hausman-Wu test:

Step 1 Regress X_k on the set of instrumental variables Z and retain the first-stage residuals:

$$e = X_k - Z(Z'Z)^{-1}Z'X_k.$$

Step 2 Add the vector of first-stage residuals to the original regression specification,

$$y = X\beta + e\gamma + \epsilon = W\delta + \epsilon,$$

$$\text{where } W = [X, e] \text{ and } \delta = \begin{bmatrix} \beta \\ \gamma \end{bmatrix}.$$

Step 3 Estimate this equation by OLS and check whether the estimated coefficient on u is zero. If it is statis-

tically significantly different from zero, then reject the hypothesis that X_k is not endogenous.

Notice that the β estimators obtained here are identical to the IV estimators obtained above. Notice also that, to obtain correct IV residuals and covariance matrix, the influence of e must be omitted from the calculation of s^2 . The correct covariance matrix is given by $s^2(W'W)^{-1}$.

Note that the classical distribution theory *does not* yield the result that the t -ratio on the coefficient of interest for the Wu test follows the t -distribution with the usual degrees of freedom. The t -ratio in this case is asymptotically normally distributed: a z -test (with a statement of asymptotic justification) is appropriate. If the same estimators of the error variance have been used to construct the Hausman statistic and the Hausman-Wu test, then the square of the t -ratio on the residual e identically equals the Hausman statistic.

Note that using SAS PC or SPSS/PC+ to perform a manual two-stage IV or Hausman-Wu estimation does *not automatically* produce the correct variance estimator.

In GAUSS-386, two sample programs, HAUSMAN.G and HAUSMNUWU.G (Figure 22), illustrate the procedures described above. In both cases, the programs test whether variable X_{10} is correlated with the stochastic disturbance terms. For SAS PC and SPSS/PC+, it is simpler to use the procedures as indicated in the sample programs, HAUSMNUWU.SAS and HAUSMNUWU.SPS. Notice that the coefficient estimates, standard errors, and t -ratios are identical for both types of programs ($t = 1.6238$) and that the Hausman statistic is equal to the square of the t -ratio on the residual u of the Hausman-Wu technique ($W = 2.637$). The null hypothesis of no endogeneity of X_{10} cannot be rejected at the 5 percent level.

Recommended references: Berndt (1991, 379–380); Greene (1990, 303); Griffiths, Hill, and Judge (1993, 458–476); Hausman (1978); Kennedy (1985, 71, 80, 119, 138, 187; 1992, 135, 148, 169–170); Kmenta (1986, 365); Maddala (1988, 435–441).

Figure 22—Sample programs for Hausman test and Hausman-Wu test, in GAUSS-386

22a—HAUSMAN.G program

```

/*****
* PROGRAM:   HAUSMAN.G       SOFTWARE: GAUSS-386 V3.0   *
* FILENAME  DESCRIPTION      *
* INPUTS:   DATA.DAT       GAUSS-386 DATA SET       *
* PURPOSE:  PERFORM OLS AND IV ESTIMATION, THEN      *
*           COMPARE THEM VIA THE HAUSMAN TEST TO     *
*           CHECK FOR EVIDENCE OF MEASUREMENT ERROR. *
*****/
FORMAT /M2 /RD 12,4;
OUTPUT FILE = HAUSMAN.OUT RESET;
NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);
Y = DATA[.,IY1];

XO = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX13 IX14 IX15
                        ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMESX = NAMES[IX1 IX2 IX8 IX9 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

X10 = DATA[.,IX10];

ZO = DATA[.,IX4 IX5 IX6 IX7 IX11 IX12 ];

NAMESZ = NAMES[IX4 IX5 IX6 IX7 IX11 IX12,.];

@----- OLS ESTIMATION -----@

X = XO ~ X10;
NAMESX = NAMESX | "X10";

K = COLS(X);

B = INV(X'X)*X'Y; @ BETAS @
E = Y - X*B; @ RESIDUALS @
RSS = E'E; @ RESIDUAL SUM OF SQUARES @
SOLS = INV(NCASE-K)*RSS; @ LS ERROR VARIANCE @
SER = SQRT(SOLS); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T = B ./ SE; @ T-STATISTICS FOR BETAS @
PT = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES @
PRN = B ~ SE ~ T ~ PT; @ FOR PRINTING @

BOLS = B;
COVOLS = COV;

" ";
" ";
" ";
" OLS RESULTS ";

```

(continued)

22a—Continued

```

" ";
" ";
" NUMBER OF OBSERVATIONS      = ";;  NCASE;
" STANDARD ERROR OF REGRESSION = ";;  SER;
" RESIDUAL SUM OF SQUARES     = ";;  RSS;
" R-SQUARED                   = ";;  RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT ";;  PRN[1, .];

I      = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAMESX[I, .];; FORMAT /M1 /RD 12,4; PRN[I+1, .];

  I      = I + 1;
ENDO;
"\f";

@-----          INSTRUMENTAL VARIABLES ESTIMATION          -----@
Z      = XO ~ ZO;                                @ NOTE THAT Z HAS ZO      @
                                                @ AND ALL X EXCEPT X10 @
K      = COLS(X);

PZX    = INV(X'Z*INV(Z'Z)*Z'X);                    @ X, Z PROJECTION INV    @
BIV    = PZX*X'Z*INV(Z'Z)*Z'Y;                    @ IV ESTIMATOR           @
E      = Y - X*BIV;                                @ RESIDUALS              @
RSS    = E'E;                                       @ RESIDUAL SUM OF SQUARES @
SIV    = INV(NCASE-K)*RSS;                          @ IV ERROR VARIANCE      @
SER    = SQRT(INV(NCASE-K)*RSS);                    @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSS/((NCASE-1)*(STDC(X10))^2); @ R-SQUARED              @
COV    = INV(NCASE-K)*RSS*PZX;                      @ IV COVARIANCE MATRIX   @
SE     = SQRT(DIAG(COV));                          @ STD ERRS OF BETAS      @
T      = BIV ./ SE;                                 @ T-STATISTICS FOR BETAS @
PT     = 2*CDENC(ABS(T));                          @ P-VALUES               @
PRN    = BIV ~ SE ~ T ~ PT;                        @ FOR PRINTING           @

" ";
" ";
" ";
"   INSTRUMENTAL VARIABLES RESULTS ";
" ";
" ";

" NUMBER OF OBSERVATIONS      = ";;  NCASE;
" STANDARD ERROR OF REGRESSION = ";;  SER;
" RESIDUAL SUM OF SQUARES     = ";;  RSS;
" R-SQUARED                   = ";;  RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      ASY Z -RATIO      P-VALUE";
" ";
"   INTERCEPT ";;  PRN[1, .];

I      = 1;

```

(continued)

22a—Continued

```

DO WHILE I <= K - 1;
  FORMAT /M1 /RD 12,8; $NAMESX[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;

@-----          CALCULATION OF HAUSMAN TEST STATISTIC          -----@

XXI      = INV(X'X);

Q        = BOLS[K,.] - BIV[K,.];

V        = SIV*( PZX[K,K] - XXI[K,K] );

W        = Q'INV(V)*Q;

DF       = 1;

PW       = CDFCHIC(W,DF);
" ";
" ";
" ";
"  HAUSMAN TEST STATISTIC:  W =";; W;
" ";
FORMAT /M1 /RD 3,0;
"  DEGREES OF FREEDOM:      =";; DF;
" ";
FORMAT /M1 /RD 12,4;
"  P-VALUE                  =";; PW;
" ";
" \f";

OUTPUT FILE = HAUSMAN.OUT OFF;
SYSTEM;

```

Figure 22b—HAUSMNWU.G program

```

/*****
* PROGRAM: HAUSMNWU.G SOFTWARE: GAUSS-386 V3.0 *
* FILENAME DESCRIPTION *
* INPUTS: DATA.DAT GAUSS-386 DATA SET *
* PURPOSE: PERFORM OLS AND IV ESTIMATION, THEN *
* CHECK FOR ENDOGENEITY VIA THE WU TEST. *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = HAUSMNWU.OUT RESET;

NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);

Y = DATA[.,IY1];

XO = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX13 IX14 IX15
ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMESXO = NAMES[IX1 IX2 IX8 IX9 IX13 IX14 IX15
ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

X10 = DATA[.,IX10];

ZO = DATA[.,IX4 IX5 IX6 IX7 IX11 IX12 ];

NAMESZO = NAMES[IX4 IX5 IX6 IX7 IX11 IX12,.];

@----- OLS ESTIMATION -----@

X = XO ~ X10;
NAMESX = NAMESXO | "X10";
K = COLS(X);

B = INV(X'X)*X'Y; @ BETAS @
E = Y - X*B; @ RESIDUALS @
RSS = E'E; @ RESIDUAL SUM OF SQUARES @
SER = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T = B ./ SE; @ T-STATISTICS FOR BETAS @
PT = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES @
PRN = B ~ SE ~ T ~ PT; @ FOR PRINTING @

BOLS = B;
COVOLS = COV;

" ";
" ";
" ";
" OLS RESULTS ";
" ";
" ";

```

(continued)

Figure 22b—Continued

```

" NUMBER OF OBSERVATIONS      = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES     = ";; RSS;
" R-SQUARED                   = ";; RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT ";; PRN[1, .];

I      = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAMESX[I, .];; FORMAT /M1 /RD 12,4; PRN[I+1, .];

  I      = I + 1;
ENDO;
"\f";

@----- TWO-STAGE LEAST-SQUARES CALCULATION OF -----@
@----- INSTRUMENTAL VARIABLES ESTIMATORS -----@

@----- FIRST STAGE -----@

Z      = XO ~ ZO;          @ NOTE THAT Z HAS ZO @
                                @ AND ALL X EXCEPT X10 @
K      = COLS(Z);
NAMESZ = NAMESXO | NAMESZO;

G      = INV(Z'Z)*Z'X10;    @ OLS OF X10 ON Z @
X10FIT = Z*G;              @ FITTED X10 @
U      = X10 - X10FIT;     @ RESIDUALS @
RSS    = U'U;              @ RESIDUAL SUM OF SQUARES @
SER    = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSS/((NCASE-1)*(STDC(X10))^2); @ R-SQUARED @
COV    = INV(NCASE-K)*RSS*INV(Z'Z); @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV));  @ STD ERRS OF BETAS @
T      = G ./ SE;         @ T-STATISTICS FOR BETAS @
PT     = 2*CDFTC(ABS(T), (NCASE-K)); @ P-VALUES @
PRN    = G ~ SE ~ T ~ PT; @ FOR PRINTING @

" ";
" ";
" ";
"   FIRST-STAGE RESULTS ";          @ LISTING FULL DETAIL HERE IS @
" ";                                @ OPTIONAL; ONE MAY WISH TO @
" ";                                @ EXAMINE THE QUALITY OF THE @
" ";                                @ FIRST STAGE. @

" NUMBER OF OBSERVATIONS      = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES     = ";; RSS;
" R-SQUARED                   = ";; RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT ";; PRN[1, .];

```

(continued)

Figure 22b—Continued

```

I      = 1;
DO WHILE I <= K - 1;
  FORMAT /M1 /RD 12,8; $NAMESZ[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
"\f";

@-----                SECOND STAGE ESTIMATION                -----@
@-----                REPLACE X10 BY X10FIT                    -----@

XH      = XO ~ X10FIT;
NAMESX  = NAMESXO | "X10FIT";

K      = COLS(XH);

B      = INV(XH'XH)*XH'Y;                                @ IV BETAS @
E      = Y - X*B;                                        @ NOTE THAT RESIDUALS @
                                                @ USE X NOT XH! @
RSS    = E'E;                                           @ RESIDUAL SUM OF SQUARES @
SER    = SQRT(INV(NCASE-K)*RSS);                         @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSS/((NCASE-1)*(STDC(Y))^2);               @ R-SQUARED @
COV    = INV(NCASE-K)*RSS*INV(XH'XH);                  @ IV COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV));                               @ STD ERRS OF BETAS @
T      = B ./ SE;                                       @ T-STATISTICS FOR BETAS @
PT     = 2*CDFNC(ABS(T));                               @ P-VALUES @
PRN    = B ~ SE ~ T ~ PT;                               @ FOR PRINTING @

BIV    = B;
SIV    = INV(NCASE-K)*RSS;

" ";
" ";
" ";
" INSTRUMENTAL VARIABLES RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS      = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES    = ";; RSS;
" R-SQUARED                  = ";; RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      ASY Z-RATIO      P-VALUE";
" ";
" INTERCEPT ";; PRN[1,.];

I      = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAMESX[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
"\f";

@-----                THE WU TEST                -----@
@-----                SPECIFY THE MATRIX OF REGRESSORS                -----@

```

(continued)

Figure 22b—Continued

```

@-----          TO INCLUDE THE RESIDUAL U FROM          -----@
@-----          THE FIRST STAGE OF THE IV PROCEDURE    -----@

XW      = XO ~ X10 ~ U;
NAMESW  = NAMESXO | "X10" | "U";

B        = INV(XW'XW)*XW'Y;          @ WU BETAS          @
K        = COLS(XW) - 1;

E        = Y - XW[.,1:K]*B[1:K,.];   @ RESIDUALS      @
                                           @ OMIT EFFECT OF U @

RSS      = E'E;                     @ RESIDUAL SUM OF SQUARES @

SER      = SQRT(INV(NCASE-K)*RSS);    @ STD ERROR OF REGRESSION @
RSQ      = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED      @
COV      = INV(NCASE-K)*RSS*INV(XW'XW); @ IV COVARIANCE MATRIX @
SE       = SQRT(DIAG(COV));          @ STD ERRS OF BETAS   @
T        = B ./ SE;                  @ T-STATISTICS FOR BETAS @
PT       = 2*CDFNC(ABS(T));          @ P-VALUES           @

PRN      = B ~ SE ~ T ~ PT;          @ FOR PRINTING      @

BIV      = B;
SIV      = INV(NCASE-K)*RSS;

" ";
" ";
" ";
" RESULTS FOR WU TEST REGRESSION";
" ";
" ";
" NUMBER OF OBSERVATIONS = ;; NCASE;
" STANDARD ERROR OF REGRESSION = ;; SER;
" RESIDUAL SUM OF SQUARES = ;; RSS;
" R-SQUARED = ;; RSQ;
" ";
" ";
" VARIABLE COEFF STD ERROR ASY Z-RATIO P-VALUE";
" ";
" INTERCEPT ;; PRN[1,.];

I        = 1;
DO WHILE I <= K;
  FORMAT /M1 /RD 12,8; $NAMESW[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";
"\f";

OUTPUT FILE = HAUSMNWU.OUT OFF;
SYSTEM;

```

Figure 23—Sample program for Hausman-Wu test, in SAS PC

```

*****
* PROGRAM:  HAUSMNU.SAS  SOFTWARE: SAS PC 6.04  *
* FILENAME  DESCRIPTION  *
* INPUTS:   DATA.SSD    TEST DATA SET      *
* PURPOSE:  PERFORM HAUSMAN-WU TEST.         *
*****;

LIBNAME CDRV 'C:\DATA';
* HAUSMAN TEST WHERE VARIABLE X10 IS SUSPECTED OF BEING ENDOGENOUS IN THE
* FOLLOWING MODEL.
* PROC REG DATA=CDRV.DATA;
* MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;

* VARIABLES X4, X5, X6, X7, X11, AND X12 ARE USED AS
* IDENTIFYING INSTRUMENTS FOR X10.;

* STEP 1: REGRESS X10 AGAINST EXOGENOUS EXPLANATORY VARIABLES (X1, X2, X8, X9,
* X13, X14, X15, D1, D2, D3, D5, D6, D7, D8, RD1, RD2, AND RD3) AND THE
* IDENTIFYING INSTRUMENTS (X4, X5, X6, X7, X11, AND X12) AND SAVE THE
* RESIDUALS OF X10 AS RX10.;

PROC REG DATA=CDRV.DATA;
  MODEL X10=X1 X2 X8 X9 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3
        X4 X5 X6 X7 X11 X12;
  OUTPUT OUT=HDATA1 R = RX10;
RUN;

* STEP 2: RUN ORIGINAL REGRESSION MODEL WITH BOTH X10 AND RX10 AS EXPLANATORY
* VARIABLES.;

PROC REG DATA=HDATA1 OUTEST=HBETA;
  MODEL Y1=X1 X2 X8 X9 X10 RX10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
  OUTPUT OUT=HDATA2 R=RY1;
RUN;

* NEED TO ADD CONSTANT TO BOTH DATA SETS TO MERGE BY;

DATA HDATA3;
  SET HDATA2;
  CONSTANT = 1;

DATA HBETA2;
  SET HBETA(RENAME=(X1=CX1 X2=CX2 X8=CX8 X9=CX9 X10=CX10 RX10=CRX10
                  X13=CX13 X14=CX14 X15=CX15 D1=CD1 D2=CD2 D3=CD3
                  D5=CD5 D6=CD6 D7=CD7 D8=CD8
                  RD1=CRD1 RD2=CRD2 RD3=CRD3 Y1=CY1));
  CONSTANT = 1;

* STEP 3: STEP 2 PRODUCES THE CORRECT INSTRUMENTAL VARIABLE (IV) ESTIMATES, BUT
* GENERATES RESIDUALS (AND THEREFORE TEST STATISTICS) THAT ARE BASED ON
* X10 AND RX10, WHEREAS THEY SHOULD ONLY BE BASED ON THE IV ESTIMATES AND X10.;
* SAVE THE IV COEFFICIENTS FROM STEP 2, AND GENERATE APPROPRIATE
* RESIDUALS (DROPPING R10X) AND SAVE CORRECT (E'E);

DATA HWRESOK;
  MERGE HDATA3 HBETA2;
  BY CONSTANT;

```

(continued)

Figure 23—Continued

```

RESOK = Y1 - (INTERCEP + CX1*X1 + CX2*X2 + CX8*X8 + CX9*X9 +
              CX10*X10 + CX13*X13 + CX14*X14 + CX15*X15 +
              CD1*D1 + CD2*D2 + CD3*D3 + CD5*D5 + CD6*D6 +
              CD7*D7 + CD8*D8 + CRD1*RD1 + CRD2*RD2 + CRD3*RD3);
RESOKSQ = RESOK ** 2;
RY1SQ = RY1 ** 2;
PROC SUMMARY DATA=HWRESOK;
  VAR RESOKSQ RY1SQ CONSTANT;
  OUTPUT OUT=SUMRES SUM=SRESOKSQ SRY1SQ N;

PROC PRINT DATA=SUMRES;

DATA RESULTS;
  SET SUMRES;
  SIGMAOK = SRESOKSQ / (N - 19);
  SIGMABAD = SRY1SQ / (N - 20);
  CORFACT = (SIGMABAD/SIGMAOK) ** 0.5;

PROC PRINT DATA=RESULTS;
  VAR CORFACT;

* STEP 4: MULTIPLY T'S FROM STEP 2 BY CORFACT TO GET APPROPRIATE T'S.;

* TEST STATISTIC CALCULATION FROM OUTPUT.
* IF THE CORRECTED T-STATISTIC ON RX10 IS GREATER THAN THE
* CRITICAL T-VALUE, THEN THE
* NULL HYPOTHESIS OF THE EXOGENEITY OF X10 IS REJECTED.
* FOR THIS EXAMPLE, THE UNCORRECTED T-RATIO ON RX10=1.6280. THE CORRECTION FACTOR
* IS 0.99744, SO THE CORRECT T-RATIO ON RX10 IS 1.6238.
* WE DO NOT REJECT THE NULL HYPOTHESIS OF NO ENDOGENEITY OF X10.;

```

Figure 24— Sample program for Hausman-Wu test, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'HAUSMNUW.LIS'.
SET LOG = 'HAUSMNUW.LOG'.
*****
* PROGRAM:   HAUSMNUW.SPS   SOFTWARE: SPSS/PC+ 4.01 *
* FILENAME  DESCRIPTION *
* INPUTS:   DATA.SYS     TEST DATA SET *
* PURPOSE:  PERFORM HAUSMAN-WU TEST. *
*****
GET FILE = 'DATA.SYS'.
* HAUSMAN TEST WHERE VARIABLE X10 IS SUSPECTED OF BEING ENDOGENOUS IN THE
  FOLLOWING MODEL.
* REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
  D7 D8 RD1 RD2 RD3
* /DEPENDENT=Y1
* /METHOD=ENTER.

* VARIABLES X4, X5, X6, X7, X11, AND X12 ARE USED AS
* IDENTIFYING INSTRUMENTS FOR X10.

* STEP 1: REGRESS X10 AGAINST EXOGENOUS EXPLANATORY VARIABLES (X1, X2, X8, X9,
* X13, X14, X15, D1, D2, D3, D5, D6, D7, D8, RD1, RD2, AND RD3) AND THE
* IDENTIFYING INSTRUMENTS (X4, X5, X6, X7, X11, AND X12) AND SAVE THE
* RESIDUALS OF X10 AS RX10.
REGRESSION VARIABLES = X10 X1 X2 X8 X9 X13 X14 X15 D1 D2 D3 D5 D6
  D7 D8 RD1 RD2 RD3
  X4 X5 X6 X7 X11 X12
  /DEPENDENT=X10
  /METHOD=ENTER
  /SAVE RESID(RX10).

* STEP 2: RUN ORIGINAL REGRESSION MODEL WITH BOTH X10 AND RX10 AS EXPLANATORY
* VARIABLES.
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 RX10 X13 X14 X15 D1 D2 D3 D5 D6
  D7 D8 RD1 RD2 RD3
  /DEPENDENT=Y1
  /METHOD=ENTER
  /SAVE RESID(RY1).
SAVE OUT='HDATA.SYS'.

*****
** VIEW THE OUTPUT FROM THIS REGRESSION AND USE THE ESTIMATED REGRESSION **
** COEFFICIENTS TO COMPUTE A PREDICTED VALUE FOR OKAY RESIDUAL. **
*****

GET FILE = 'HDATA.SYS'.
COMPUTE RESOK = Y1-(X1 * 35.595780 + X2 * 42.396332 +
  X8 * -.133989 + X9 * -26.758650 +
  X10 * 146.932871 +
  X13 * -1.464556 + X14 * -1.349472 +
  X15 * -6.187087 + D1 * -351.786732 +
  D2 * -129.845550 + D3 * 214.195837 +
  D5 * -320.026103 + D6 * 55.695210 +
  D7 * 510.051877 + D8 * 785.090139 +
  RD1 * 235.775053 + RD2 * 47.125762 +
  RD3 * -145.397285 + 1145.454039).

```

(continued)

Figure 24—Continued

```

COMPUTE RESOKSQ = RESOK ** 2.
COMPUTE RY1SQ = RY1 ** 2.
COMPUTE CONSTANT=1.

AGGREGATE OUTFILE=*
  /BREAK=CONSTANT
  /COUNT=N
  /SRESOKSQ SRY1SQ = SUM(RESOKSQ RY1SQ).

COMPUTE SIGMAOK = SRESOKSQ / (COUNT - 19).
COMPUTE SIGMABAD = SRY1SQ / (COUNT - 20).
COMPUTE CORFACT = (SIGMABAD/SIGMAOK) ** 0.5.

FORMATS ALL (F9.5).
LIST CORFACT.

* STEP 4: MULTIPLY T'S FROM STEP 2 BY CORFACT TO GET APPROPRIATE T'S.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* IF THE CORRECTED T-STATISTIC ON RX10 IS GREATER THAN THE
* CRITICAL T-VALUE, THEN THE
* NULL HYPOTHESIS OF THE EXOGENEITY OF X10 IS REJECTED.
* FOR THIS EXAMPLE, THE UNCORRECTED T-RATIO ON RX10=1.6280. THE CORRECTION FACTOR
* IS 0.99744, SO THE CORRECT T-RATIO ON RX10 IS 1.6238.
* WE DO NOT REJECT THE NULL HYPOTHESIS OF NO ENDOGENEITY OF X10.

FINISH.

```

The Levi Bounds (for Assessing the Presence of Measurement Error)

The Levi bounds may be calculated to indicate the presence of measurement error. It is well known that if only one regressor is measured with error, the OLS coefficient of that regressor is biased toward zero. If the roles of this regressor and the dependent variable are reversed in the regression, the coefficient on the artificial regressor is an estimator of the inverse of the coefficient on the original regressor. This estimator is also biased toward zero, but its inverse is biased away from zero. If the coefficient on the original regressor is taken as a lower bound for a consistent estimator and the inverse of the coefficient on the artificial regressor is taken as an upper bound for a consistent estimator, then it is expected that the size of this interval reflects the severity of the measurement error problem.

Levi's procedure is very simple to execute, but no formal statistical test is performed. Whether the interval between lower and upper bounds is "large" is a matter of judgment for the investigator. The steps below are presented in terms of a simple regression model; extension to the multiple regression model is straightforward.

- | | |
|--------|---|
| Step 1 | Estimate the regression, $Y_i = \alpha_1 + \beta_1 X_i + \epsilon_1$, and get $\hat{\beta}_1$. Call this $\hat{\beta}_L$. |
| Step 2 | Run the "reverse" regression, $X_i = \alpha_2 + \beta_2 Y_i + \epsilon_2$ and calculate $\hat{\beta}_U = (1/\hat{\beta}_2)$. |

Now, examine the interval

$$\hat{\beta}_L < \beta_1 < \hat{\beta}_U.$$

As Kmenta notes, if this interval is small, the effect of measurement error is likely to be bearable and OLS results are unlikely to be severely biased. Note that the above discussion assumes that β_1 is positive. If β_1 is negative, then the lower and upper bounds are reversed.

The sample programs (Figures 25 through 27) treat variable X_{10} as possibly susceptible to measurement error. The results are striking: $\hat{\beta}_L = 217$ and $\hat{\beta}_U = 5,732$ (5,747 in SAS PC and SPSS/PC+, due to rounding). This appears to be a very large interval, particularly in view of the statistical significance of this regressor. It is concluded that measurement error is a problem for X_{10} . These estimated coefficients translate into calorie-income elasticities of 0.1 and 2.0, respectively. From an economic viewpoint, this is a very large interval.

Recommended references: Kmenta (1986, 346–366); Levi (1977).

Figure 25—Sample program for Levi bounds test, in GAUSS-386

```

/*****
* PROGRAM: LEVI.G SOFTWARE: GAUSS-386 V3.0 *
* FILENAME DESCRIPTION *
* INPUTS: DATA.DAT GAUSS-386 DATA SET *
* PURPOSE: CALCULATE LEVI BOUNDS FOR THE *
* COEFFICIENT OF A REGRESSOR THAT MAY BE *
* MEASURED WITH ERROR. *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = LEVI.OUT RESET;

NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);
Y = DATA[.,IY1];

X = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAME1 = NAMES{IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.};

@----- OLS ESTIMATION -----@

K = COLS(X);

B = INV(X'X)*X'Y; @ BETAS @
E = Y - X*B; @ RESIDUALS @
RSS = E'E; @ RESIDUAL SUM OF SQUARES @
SER = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T = B ./ SE; @ T-STATISTICS FOR BETAS @
PT = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES @
PRN = B ~ SE ~ T ~ PT; @ FOR PRINTING @

B1 = B[6,1]; @ COEFF OF INTEREST @

" ";
" ";
" ";
" OLS RESULTS FOR THE STANDARD MODEL ";
" ";
" ";
" NUMBER OF OBSERVATIONS = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES = ";; RSS;
" R-SQUARED = ";; RSQ;
" ";
" ";
" VARIABLE COEFF STD ERROR T-RATIO P-VALUE";
" ";
" INTERCEPT ";; PRN[1,.];

```

(continued)

Figure 25—Continued

```

I          = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAME1[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I          = I + 1;
ENDO;
" ";
"\f";

@-----
@-----          OLS ESTIMATION          -----@
@-----          WITH Y AND X10 REVERSED          -----@

X10        = DATA[.,IX10];
X          = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IY1 IX13 IX14 IX15
                    ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAME2      = NAMES{IX1 IX2 IX8 IX9 IY1 IX13 IX14 IX15
                    ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.};

K          = COLS(X);
B          = INV(X'X)*X'X10;                @ BETAS                @
E          = X10 - X*B;                    @ RESIDUALS                @
RSS        = E'E;                          @ RESIDUAL SUM OF SQUARES @
SER        = SQRT(INV(NCASE-K)*RSS);        @ STD ERROR OF REGRESSION @
RSQ        = 1 - RSS/((NCASE-1)*(STDC(X10))^2); @ R-SQUARED                @
COV        = INV(NCASE-K)*RSS*INV(X'X);    @ OLS COVARIANCE MATRIX  @
SE         = SQRT(DIAG(COV));              @ STD ERRS OF BETAS      @
T          = B ./ SE;                      @ T-STATISTICS FOR BETAS @
PT         = 2*CDFTC(ABS(T), (NCASE-K));    @ P-VALUES                @
PRN        = B ~ SE ~ T ~ PT;              @ FOR PRINTING           @

B2         = B[6,.];                       @ COEFF OF INTEREST      @

" ";
" ";
" ";
" OLS RESULTS FOR THE REGRESSION MODEL WITH Y AND X10 REVERSED";
" ";
" ";
" NUMBER OF OBSERVATIONS      = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES    = ";; RSS;
" R-SQUARED                  = ";; RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT ";; PRN[1,.];

I          = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAME2[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I          = I + 1;
ENDO;
" ";
" ";

```

(continued)

Figure 25—Continued

```

" ";
"   BOUNDS FOR THE COEFFICIENT ON X10";
" ";
"   LOWER BOUND:   B =";;   B1;
" ";
B2 = 1/B2;
"   UPPER BOUND:   B =";;   B2;

"\f";

OUTPUT FILE = LEVI.OUT OFF;
SYSTEM;

```

Figure 26—Sample program for Levi bounds test, in SAS PC

```

*****
*   PROGRAM:   LEVI.SAS       SOFTWARE: SAS PC 6.04   *
*   FILENAME:  DESCRIPTION   *
*   INPUTS:    DATA.SSD     TEST DATA SET        *
*   PURPOSE:   CALCULATE LEVI BOUNDS.              *
*****;

LIBNAME CDRV 'C:\DATA';

* X10 IS THE VARIABLE WE SUSPECT IS MEASURED WITH ERROR.
* STEP 1: RUN THE MODEL IN OLS.;

PROC REG DATA=CDRV.DATA;
  MODEL Y1=X10 X1 X2 X8 X9 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

* STEP 2: REVERSE Y1 AND X10 AND RUN THE MODEL IN OLS.;

PROC REG DATA=CDRV.DATA;
  MODEL X10=Y1 X1 X2 X8 X9 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

* INTERPRETATION OF OUTPUT.
* IF X10 IS MEASURED WITH RANDOM ERROR AND THE TRUE PARAMETER ON X10
* IS POSITIVE, THE ESTIMATED OLS COEFFICIENT ON X10 WILL BE BIASED TOWARDS ZERO.
* SINCE THE ESTIMATED OLS COEFFICIENT ON Y1 IN STEP 2 IS AN ESTIMATE OF
* THE RECIPROCAL OF THE PARAMETER ON X10 IN STEP 1, IT WILL BE BIASED AWAY
* FROM ZERO. IF THE INTERVAL BETWEEN THESE TWO OLS ESTIMATES IS NARROW (WITHIN
* PLAUSIBLE BEHAVIORAL BOUNDS), THEN THE MEASUREMENT ERROR ON X10 IS WITHIN
* ACCEPTABLE LIMITS.;

* FOR THIS EXAMPLE, THE LEVI BOUNDS ON X10'S OLS ESTIMATE ARE 216.97 AND
* 5747.126 AT THE MEAN OF X10 AND Y1. THESE TRANSLATE INTO ELASTICITIES OF
* APPROXIMATELY 0.1 AND 2.0, RESPECTIVELY. FROM AN ECONOMIC VIEWPOINT,
* THIS IS A VERY LARGE INTERVAL.;

```

Figure 27—Sample program for Levi bounds test, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'LEVI.LIS'.
SET LOG = 'LEVI.LOG'.
*****
* PROGRAM:  LEVI.SPS      SOFTWARE: SPSS/PC+ 4.01  *
*          FILENAME      DESCRIPTION          *
* INPUTS:   DATA.SYS    TEST DATA SET       *
* PURPOSE:  CALCULATES LEVI BOUNDS.         *
*****

GET FILE = 'DATA.SYS'.

* X10 IS THE VARIABLE WE SUSPECT IS MEASURED WITH ERROR.
* STEP 1: RUN THE MODEL IN OLS.

REGRESSION VARIABLES = Y1 X10 X1 X2 X8 X9 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                    RD1 RD2 RD3

  /DEPENDENT = Y1
  /METHOD = ENTER.

* STEP 2: REVERSE Y1 AND X10 AND RUN THE MODEL IN OLS.

REGRESSION VARIABLES = Y1 X10 X1 X2 X8 X9 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                    RD1 RD2 RD3

  /DEPENDENT = X10
  /METHOD = ENTER.

* INTERPRETATION OF OUTPUT.
* IF X10 IS MEASURED WITH RANDOM ERROR AND THE TRUE PARAMETER ON X10 IS
* POSITIVE, THE ESTIMATED OLS COEFFICIENT ON X10 WILL BE BIASED TOWARDS ZERO.
* SINCE THE ESTIMATED OLS COEFFICIENT ON Y1 IN STEP 2 IS AN ESTIMATE OF
* THE RECIPROCAL OF THE PARAMETER ON X10 IN STEP 1, IT WILL BE BIASED AWAY
* FROM ZERO. IF THE INTERVAL BETWEEN THESE TWO OLS ESTIMATES IS NARROW (WITHIN
* PLAUSIBLE BEHAVIORAL BOUNDS), THEN THE MEASUREMENT ERROR ON X10 IS WITHIN
* ACCEPTABLE LIMITS.
* FOR THIS EXAMPLE, THE LEVI BOUNDS ON X10'S OLS ESTIMATE ARE 216.97 AND
* 5747.126 AT THE MEAN OF X10 AND Y1. THESE TRANSLATE INTO ELASTICITIES
* OF APPROXIMATELY 0.1 AND 2.0, RESPECTIVELY. FROM AN ECONOMIC VIEWPOINT,
* THIS IS A VERY LARGE INTERVAL.
FINISH.

```

TESTS FOR NONNESTED HYPOTHESES

This class of tests is used to test the validity of one model for explaining y versus another model for explaining y when neither model can be obtained by imposing linear restrictions on the other model. These "model validity" tests are popular because they allow *all* competing models to be rejected if all are deficient (unlike "model selection" methods—such as high R^2 criteria, backwards elimination, or stepwise regression—in which one model will always be chosen).

The following models are nonnested models, because Z is not a subset of W , nor is W a subset of Z :

$$y = X\beta + Z\gamma + \epsilon_1; \quad (2)$$

$$y = X\beta + W\delta + \epsilon_2. \quad (3)$$

In these competing models that explain y , the explanatory variables are contained in X , Z , and W , which are of the dimension $N \times K_1$, $N \times K_2$, and $N \times K_3$, respectively. The coefficient vectors are conformable. It is important to note that tests of these models all assume that the stochastic disturbance terms satisfy the classical assumptions.

Two popular tests for nonnested models, the nonnested F -test and the nonnested J -test, are explained below.

Nonnested F -Test The strategy of this test is to artificially nest the two competing models in a more general model and then to test whether the restrictions that produce either original model (or both) are valid.

Step 1 Form the general model:

$$y = X\beta + Z\gamma + W\delta + \epsilon. \quad (4)$$

Step 2 Estimate the general model (4) using OLS.

Step 3 Use F -tests for incremental explanatory power to test the following three sets of hypotheses:

$$H_0: \gamma = 0,$$

$$H_1: \gamma \neq 0;$$

$$H_0: \delta = 0,$$

$$H_1: \delta \neq 0;$$

$$H_0: \gamma = \delta = 0,$$

$$H_1: \gamma \text{ and } \delta \text{ are not both } 0.$$

Note that the last hypothesis cannot be addressed using the F -tests for coefficients on Z and W : for the last hypothesis you need to construct an F -test for the joint incremental explanatory power of Z and W .

Step 4 If the estimates of γ or δ are not significantly different from zero, the model that includes the corresponding set of variables is rejected. If both sets of coefficients are significantly different from zero, then the general model (4) is preferred; if neither is significantly different from zero, then the restricted model,

$$y = X\beta + \epsilon, \quad (5)$$

may be adequate.

In the sample programs (Figures 28 through 30), X is taken to include a constant and variables $X1, X2, X8, X9, X10, X13, X14, X15, D1, D2, D3, D5, D6, D7, D8, RD1, RD2,$ and $RD3$. Then $Z = [X3, X7]$ and $W = [X6, X12]$.

For the sample data set, the F -statistic for the hypothesis that $\gamma = 0$ is 2.5261 (P -value = 0.0803): the variables Z should be retained in the model. The F -statistic for the hypothesis that $\delta = 0$ is 1.9970 (P -value = 0.1361): the variables W only have significant explanatory power at a significance level of, say 15 percent. Investigators who prefer to use smaller significance levels, say 10 percent or 5 percent, would fail to reject this null hypothesis and would choose model 2 over model 3 at this point (that is, include Z but not W). Finally, the F -statistic for the hypothesis $\gamma = \delta = 0$ is 2.2438 (P -value = 0.0622), and, at the 7 percent significance level, it is concluded that Z and W are jointly significant. The completely unrestricted model is most appropriate.

This test and several others in this manual are F -tests for linear restrictions on coefficients. Good general expositions of F -tests are given in Greene (1990, Chapter 7) and Kmenta (1986, Section 10-2). See **Testing for Structural Change** (p. 91) in this manual for a fuller exposition of an F -test.

Recommended references: Davidson and MacKinnon (1981, 781–793); Greene (1990, 231–234); Kennedy (1985, 70, 79–80, 85–87; 1992, 81, 87–88); Kmenta (1986, 595–600); MacKinnon (1983, 85–158); Maddala (1988, 443–446); McAleer and Pesaran (1986, 217–371).

Figure 28—Sample program for nonnested *F*-test, in GAUSS-386

```

/*****
* PROGRAM:   NNESTF.G       SOFTWARE: GAUSS-386 V3.0       *
*           FILENAME      DESCRIPTION                   *
* INPUTS:   DATA.DAT     GAUSS-386 DATA SET           *
* PURPOSE:  PERFORM NONNESTED F-TEST.                  *
*****/

OUTPUT FILE = NNESTF.OUT RESET;
FORMAT /M2 /RD 12,4;
NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);
K = COLSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);

@-----          SELECT VARIABLES THAT WILL BE USED          -----@

Y1 = DATA[.,IY1];
X0 = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                        ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];
Z = DATA[.,IX3 IX7];
W = DATA[.,IX6 IX12];

@-----          SELECT VARIABLE NAMES CORRESPONDING TO VARIABLES          -----@
@-----          USED IN ALTERNATIVE MODELS.  NAMES MUST BE LISTED          -----@
@-----          IN THE SAME ORDER AS THE VARIABLES APPEAR FOR X.          -----@

NAMESU = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3
              IX3 IX7 IX6 IX12,.];
NAMES1 = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3
              IX3 IX7,.];
NAMES2 = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3
              IX6 IX12,.];
NAMES3 = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

@ -----          MODEL U          -----@

@ -----          UNRESTRICTED MODEL THAT INCLUDES BOTH Z AND W          -----@

X = X0 ~ Z ~ W;

K0 = COLS(X);

B = INV(X'X)*X'Y1;          @ OLS ESTIMATION          @
E = Y1 - X*B;             @ RESIDUALS          @
RSSU = E'E;                @ UNRESTRICTED RSS          @
SER = SQRT(INV(NCASE-K0)*RSSU);          @ STD ERROR OF REGRESSION @
RSQ = 1 - RSSU/((NCASE-1)*(STDC(Y1))^2); @ R-SQUARED          @
COV = INV(NCASE-K0)*RSSU*INV(X'X);      @ VAR-COV MATRIX          @
SE = SQRT(DIAG(COV));          @ STD ERRS OF ESTIMATES @
T = B ./ SE;                   @ T-STATISTICS          @
PT = 2*CDFTC(ABS(T),(NCASE-K0));        @ P-VALUES          @
PRN = B ~ SE ~ T ~ PT;          @ FOR PRINTING          @

```

(continued)

Figure 28—Continued

```

@ ----- PRINT RESULTS -----@
" OLS RESULTS FOR UNRESTRICTED MODEL ";
" ";
" NUMBER OF OBSERVATIONS =      ;;  NCASE;
" STANDARD ERROR OF REGRESSION =  ;;  SER;
" RESIDUAL SUM OF SQUARES =     ;;  RSSU;
" R-SQUARED =                   ;;  RSQ;
" ";
" VARIABLE      COEFF.      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT  ;;  PRN[1, .];

I      =  1;
DO WHILE I <= K0-1;
  FORMAT /M1 /RD 12,8; $NAMESU[I, .];  FORMAT /M1 /RD 12,4;  PRN[I+1, .];

  I      =  I + 1;
ENDO;
" ";
"\f";

@ ----- MODEL 1 -----@
@ ----- RESTRICTED MODEL THAT EXCLUDES W -----@

X      =  X0 ~ Z;
K1     =  COLS(X);

B      =  INV(X'X)*X'Y1;           @ OLS ESTIMATION           @
E      =  Y1 - X*B;               @ RESIDUALS             @
RSSR1  =  E'E;                     @ RESTRICTED RSS 1     @
SER    =  SQRT(INV(NCASE-K1)*RSSR1); @ STD ERROR OF REGRESSION @
RSQ    =  1 - RSSR1/((NCASE-1)*(STDC(Y1))^2); @ R-SQUARED             @
COV    =  INV(NCASE-K1)*RSSR1*INV(X'X); @ VAR-COV MATRIX       @
SE     =  SQRT(DIAG(COV));         @ STD ERRS OF ESTIMATES @
T      =  B ./ SE;                 @ T-STATISTICS         @
PT     =  2*CDFTC(ABS(T), (NCASE-K1)); @ P-VALUES              @
PRN    =  B ~ SE ~ T ~ PT;         @ FOR PRINTING         @

@ ----- PRINT RESULTS -----@

" OLS RESULTS FOR RESTRICTED MODEL THAT EXCLUDES W";
" ";
" NUMBER OF OBSERVATIONS =      ;;  NCASE;
" STANDARD ERROR OF REGRESSION =  ;;  SER;
" RESIDUAL SUM OF SQUARES =     ;;  RSSR1;
" R-SQUARED =                   ;;  RSQ;
" ";
" VARIABLE      COEFF.      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT  ;;  PRN[1, .];

I      =  1;
DO WHILE I <= K1-1;
  FORMAT /M1 /RD 12,8; $NAMES1[I, .];  FORMAT /M1 /RD 12,4;  PRN[I+1, .];

  I      =  I + 1;

```

(continued)

Figure 28—Continued

```

ENDO;

" ";
"\f";

@ ----- MODEL 2 -----@
@ ----- RESTRICTED MODEL THAT EXCLUDES Z -----@

X      = X0 ~ W;
K2     = COLS(X);

B      = INV(X'X)*X'Y1;           @ OLS ESTIMATION           @
E      = Y1 - X*B;              @ RESIDUALS           @
RSSR2  = E'E;                   @ RESTRICTED RSS 2   @
SER    = SQRT(INV(NCASE-K2)*RSSR2); @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSSR2/((NCASE-1)*(STDC(Y1))^2); @ R-SQUARED           @
COV    = INV(NCASE-K2)*RSSR2*INV(X'X); @ VAR-COV MATRIX     @
SE     = SQRT(DIAG(COV));       @ STD ERRS OF ESTIMATES @
T      = B ./ SE;              @ T-STATISTICS       @
PT     = 2*CDFTC(ABS(T), (NCASE-K2)); @ P-VALUES           @
PRN    = B ~ SE ~ T ~ PT;      @ FOR PRINTING      @

@ ----- PRINT RESULTS -----@

" OLS RESULTS FOR RESTRICTED MODEL THAT EXCLUDES Z";
" ";
" NUMBER OF OBSERVATIONS =      ;; NCASE;
" STANDARD ERROR OF REGRESSION = ;; SER;
" RESIDUAL SUM OF SQUARES =     ;; RSSR2;
" R-SQUARED =                   ;; RSQ;
" ";
" VARIABLE      COEFF.      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT ;; PRN[1, .];

I      = 1;
DO WHILE I <= K2-1;
  FORMAT /M1 /RD 12,8; $NAMES2[I, .];;  FORMAT /M1 /RD 12,4; PRN[I+1, .];

  I      = I + 1;
ENDO;

" ";
"\f";

@ ----- MODEL 3 -----@
@ ----- RESTRICTED MODEL THAT EXCLUDES Z AND W -----@

X      = X0;
K3     = COLS(X);

B      = INV(X'X)*X'Y1;           @ OLS ESTIMATION           @
E      = Y1 - X*B;              @ RESIDUALS           @
RSSR3  = E'E;                   @ RESTRICTED RSS 3   @
SER    = SQRT(INV(NCASE-K2)*RSSR3); @ STD ERROR OF REGRESSION @

```

(continued)

Figure 28—Continued

```

RSQ  = 1 - RSSR3/((NCASE-1)*(STDC(Y1))^2); @ R-SQUARED @
COV  = INV(NCASE-K2)*RSSR3*INV(X'X); @ VAR-COV MATRIX @
SE   = SQRT(DIAG(COV)); @ STD ERRS OF ESTIMATES @
T    = B ./ SE; @ T-STATISTICS @
PT   = 2*CDFTC(ABS(T),(NCASE-K3)); @ P-VALUE @
PRN  = B ~ SE ~ T ~ PT; @ FOR PRINTING @

@ ----- PRINT RESULTS -----@

" OLS RESULTS FOR RESTRICTED MODEL THAT EXCLUDES Z AND W";
" ";
" NUMBER OF OBSERVATIONS = "; NCASE;
" STANDARD ERROR OF REGRESSION = "; SER;
" RESIDUAL SUM OF SQUARES = "; RSSR3;
" R-SQUARED = "; RSQ;
" ";
" VARIABLE COEFF. STD ERROR T-RATIO P-VALUE";
" ";
" INTERCEPT "; PRN[1, .];

I = 1;
DO WHILE I <= K3-1;
  FORMAT /M1 /RD 12,8; $NAMES3[I, .]; FORMAT /M1 /RD 12,4; PRN[I+1, .];

  I = I + 1;
ENDO;

" ";
"\f";

@----- F-TESTS FOR INCREMENTAL EXPLANATORY POWER -----@

F1 = ((RSSR1 - RSSU)/(K0 - K1)) / (RSSU/(NCASE - K0));
PROB1 = CDFFC(F1, (K0 - K1), (NCASE - K0));

F2 = ((RSSR2 - RSSU)/(K0 - K2)) / (RSSU/(NCASE - K0));
PROB2 = CDFFC(F2, (K0 - K2), (NCASE - K0));

F3 = ((RSSR3 - RSSU)/(K0 - K3)) / (RSSU/(NCASE - K0));
PROB3 = CDFFC(F3, (K0 - K3), (NCASE - K0));

" F-TESTS FOR INCREMENTAL EXPLANATORY POWER ";
" ";
" ";
" F-TEST FOR MODEL 1 vs MODEL U: F ="; F1;; " PROB ="; PROB1;
" ";
" ";
" F-TEST FOR MODEL 2 vs MODEL U: F ="; F2;; " PROB ="; PROB2;
" ";
" ";
" F-TEST FOR MODEL 3 vs MODEL U: F ="; F3;; " PROB ="; PROB3;
" ";
" ";
"\f";

OUTPUT FILE = NNESTF.OUT OFF;
SYSTEM;

```

Figure 29—Sample program for nonnested *F*-test, in SAS PC

```

*****
* PROGRAM:   NNESTF.SAS      SOFTWARE: SAS PC 6.04      *
* FILENAME  DESCRIPTION    *
* INPUTS:   DATA.SSD      TEST DATA SET            *
* PURPOSE:  PERFORM NONNESTED F-TEST.                *
*****;

LIBNAME CDRV 'C:\DATA\';

* ALL VARIABLES EXCEPT X3, X7, X6, AND X12 ARE COMMON TO ALL
* MODELS. SPECIFICATION 1 CONTAINS X3 AND X7.
* SPECIFICATION 2 CONTAINS X6 AND X12.;
* SPECIFICATION 3 DOES NOT CONTAIN X6, X12, X3, AND X7.;
PROC REG DATA=CDRV.DATA;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5
        D6 D7 D8 RD1 RD2 RD3 X3 X7 X6 X12;
  B1 : TEST X6=X12=0;
  B2 : TEST X3=X7=0;
  B3 : TEST X3=X7=X6=X12=0;
RUN;

* THE 'TEST' COMMANDS PRODUCE THE 3 F-STATISTICS DESCRIBED IN THE TEXT;
* F FROM TEST B1 = 1.9970
* F FROM TEST B2 = 2.5261
* F FROM TEST B3 = 2.2438;
* CONCLUSION: RETAIN ALL FOUR VARIABLES: X3, X7, X6, AND X12;

```

Figure 30—Sample program for nonnested F-test, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'NNESTF.LIS'.
SET LOG = 'NNESTF.LOG'.
*****
* PROGRAM:  NNESTF.SPS   SOFTWARE: SPSS/PC+ 4.01   *
*          FILENAME     DESCRIPTION           *
* INPUTS:   DATA.SYS   TEST DATA SET        *
* PURPOSE:  PERFORM NONNESTED F-TEST.        *
*****

GET FILE = 'DATA.SYS'.
* ALL VARIABLES EXCEPT X3, X7, X6 AND X12 ARE COMMON TO ALL MODELS.
* SPECIFICATION 1 CONTAINS X3 AND X7.
* SPECIFICATION 2 CONTAINS X6 AND X12.
* SPECIFICATION 3 DOES NOT CONTAIN X3, X7, X6, OR X12.
* SPECIFICATION 4 CONTAINS X3, X7, X6, AND X12.

* STEP 1: ESTIMATE SPECIFICATION 1.

REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                     D7 D8 RD1 RD2 RD3 X3 X7
                     /DEPENDENT=Y1
                     /METHOD=ENTER.

* STEP 2: ESTIMATE SPECIFICATION 2.

REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                     D7 D8 RD1 RD2 RD3 X6 X12
                     /DEPENDENT=Y1
                     /METHOD=ENTER.

* STEP 3: ESTIMATE SPECIFICATION 3 (COMPLETELY RESTRICTED MODEL).

REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                     D7 D8 RD1 RD2 RD3
                     /DEPENDENT=Y1
                     /METHOD=ENTER.

* STEP 4: ESTIMATE SPECIFICATION 4 (COMPLETELY UNRESTRICTED MODEL).

REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                     D7 D8 RD1 RD2 RD3 X3 X7 X6 X12
                     /DEPENDENT=Y1
                     /METHOD=ENTER.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* CALCULATE 3 F-STATISTICS:
* SPECIFICATION 1 VERSUS 4. * F = 1.9970.
* SPECIFICATION 2 VERSUS 4. * F = 2.5261.
* SPECIFICATION 3 VERSUS 4. * F = 2.2438.
* THESE ARE THE 3 F-STATISTICS DESCRIBED IN THE TEXT.
FINISH.

```

Nonnested J-Test The J -test, developed by R. Davidson and J. G. MacKinnon, can be used to test whether one of two models having different (but possibly overlapping) sets of regressors has greater explanatory power than the other. Once again, it is assumed that the stochastic disturbance terms satisfy the classical assumptions. Let the competing models be

$$y = X\beta + \epsilon_1, \text{ and} \quad (6)$$

$$y = Z\delta + \epsilon_2. \quad (7)$$

The J -test proceeds in the following steps:

Step 1 Estimate the second equation by OLS and calculate the fitted values of y , \hat{y} . Variation in \hat{y} reflects the linear influence on y of variation in the explanatory variables Z .

Step 2 Specify the augmented regression model,

$$y = X\beta + \hat{y}\lambda + \epsilon,$$

where λ is a scalar coefficient. Estimate this augmented model by OLS. If some of the explanatory variables in Z have significant explanatory power for y that is not captured by the regressors in X , then the estimate for λ will be statistically significant.

Step 3 The standard t -ratio produced by statistical packages is asymptotically distributed as standard normal and may be compared to standard normal critical values to test the following hypothesis (see Greene 1990, 231–233):

$$H_0: \lambda = 0$$

$$H_1: \lambda \neq 0.$$

If H_0 is rejected in favor of H_1 , then the second model has some explanatory power that is lacking in the first model.

Step 4 Reverse the roles of the two models and repeat the exercise.

Note that it is possible that, in both cases, the null hypothesis might be rejected. If both are rejected, then each model explains some variation that the other fails to explain; the investigator may consider some augmented model that includes regressors from both X and Z . If the null hypothesis is not rejected in both cases, then neither is preferred on the basis of this test. The investigator must use economic theory and/or other statistical results to choose.

The sample programs that illustrate this section (Figures 31 through 33) specify and test the following models:

$$y = X\beta + Z\gamma + \epsilon_1 \quad (8)$$

$$y = X\beta + W\delta + \epsilon_2. \quad (9)$$

These models are exactly the ones described in the preceding section, on the nonnested F -test.

In these results, the coefficient for $YHAT2$ (the fitted y values from model 7) in augmented specification 1 is 1.0372 with t -statistic = 2.2322 (P -value = 0.0258). This indicates that variables contained in W would contribute significant incremental explanatory power if included in model 6. By the same token the coefficient on $YHAT1$ in augmented specification 2 is 0.9560 with t -statistic = 1.8920 (P -value = 0.0586). This indicates that variables contained in Z would contribute significant incremental explanatory power if included in model 7. As expected, these results are qualitatively similar to those in the section on the nonnested F -test. Neither model dominates, and it appears that a model that includes variables from both specifications is called for. Notice that the t -statistics of the coefficients not associated with the fitted values in the augmented regressions are all quite small. This is because much of their explanatory power has been captured by the fitted y values and the fitted y values are collinear with the remaining variables. Figures 31 through 33 are sample programs for the nonnested J -test.

Recommended references: Davidson and MacKinnon (1981, 781–793); Greene (1990, 231–234); Judge et al. (1984, 884–885); Kennedy (1985, 70, 79–80, 85–87; 1992, 81, 87–88); Kmenta (1986, 595–600); Maddala (1988, 443–447); McAleer and Pesaran (1986).

Figure 31—Sample program for nonnested J-test, in GAUSS-386

```

/*****
* PROGRAM:   NNESTJ.G       SOFTWARE: GAUSS-386 V3.0   *
*           FILENAME      DESCRIPTION                *
* INPUTS:   DATA.DAT     GAUSS-386 DATA SET        *
* PURPOSE:  PERFORM NONNESTED J-TEST.                *
*****/
OUTPUT FILE = NNESTJ.OUT RESET;
FORMAT /M1 /RD 12,4;
NAMES      = GETNAME("DATA");
OPEN D     = DATA VARINDXI;
NCASE      = ROWSF(D);
DATA       = READR(D,NCASE);
F          = CLOSE(D);
Y          = DATA[.,IY1];
X0         = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES1     = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                  ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3 IX3 IX7,.];
NAMES2     = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                  ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3 IX6 IX12,.];

Z          = DATA[.,IX3 IX7];
W          = DATA[.,IX6 IX12];

@----- CALCULATE FITTED Ys FROM THE ALTERNATIVE MODELS -----@

X1         = X0 ~ Z;
B1         = INV(X1'X1)*X1'Y;
YHAT1     = X1*B1;
X2         = X0 ~ W;
B2         = INV(X2'X2)*X2'Y;
YHAT2     = X2*B2;

@----- AUGMENTED REGRESSION 1 -----@

X1         = X1 ~ YHAT2;
K1         = COLS(X1);
B1         = INV(X1'X1)*X1'Y;
E1         = Y - X1*B1;           @ RESIDUALS @
RSS1      = E1'E1;               @ RESID SUM SQUARES @
SER       = SQRT(INV(NCASE-K1)*RSS1); @ STD ERROR OF REGRESSION @
RSQ       = 1 - RSS1/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV       = INV(NCASE-K1)*RSS1*INV(X1'X1); @ VAR-COV MATRIX @
SE        = SQRT(DIAG(COV));     @ STD ERRS OF ESTIMATES @
T         = B1 ./ SE;           @ T-STATISTICS @
PT        = 2*CDFTC(ABS(T), (NCASE-K1)); @ P-VALUES @
PRN       = B1 ~ SE ~ T ~ PT;   @ FOR PRINTING @

" REGRESSION RESULTS FOR AUGMENTED SPECIFICATION 1 ";
" ";
" NUMBER OF OBSERVATIONS =          "; NCASE;
" STANDARD ERROR OF REGRESSION =    "; SER;
" RESIDUAL SUM OF SQUARES =         "; RSS1;
" R-SQUARED =                       "; RSQ;
" ";

```

(continued)

Figure 31—Continued

```

"  VARIABLE          COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"  INTERCEPT ";; PRN[1,.];

I      = 1;
DO WHILE I <= K1-2;
  FORMAT /M1 /RD 12,8; $NAMES1[I,.];;  FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;

"  YHAT2          ";; PRN[K1,.];

"\f";

@-----          AUGMENTED REGRESSION 2          -----@

X2      = X2 ~ YHAT1;
K2      = COLS(X2);
B2      = INV(X2'X2)*X2'Y;

E2      = Y - X2*B2;          @ RESIDUALS          @
RSS2    = E2'E2;          @ RESID SUM SQUARES      @
SER     = SQRT(INV(NCASE-K2)*RSS2);          @ STD ERROR OF REGRESSION @
RSQ     = 1 - RSS2/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED          @
COV     = INV(NCASE-K2)*RSS2*INV(X2'X2);          @ VAR-COV MATRIX      @
SE      = SQRT(DIAG(COV));          @ STD ERRS OF ESTIMATES @
T       = B2 ./ SE;          @ T-STATISTICS        @
PT      = 2*CDFTC(ABS(T),{NCASE-K2});          @ P-VALUES           @
PRN     = B2 ~ SE ~ T ~ PT;          @ FOR PRINTING       @

"  REGRESSION RESULTS FOR AUGMENTED SPECIFICATION 2 ";
" ";
"  NUMBER OF OBSERVATIONS =          ";; NCASE;
"  STANDARD ERROR OF REGRESSION =          ";; SER;
"  RESIDUAL SUM OF SQUARES =          ";; RSS2;
"  R-SQUARED =          ";; RSQ;
" ";
"  VARIABLE          COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"  INTERCEPT ";; PRN[1,.];

I      = 1;
DO WHILE I <= K2-2;
  FORMAT /M1 /RD 12,8; $NAMES2[I,.];;  FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;

"  YHAT1          ";; PRN[K2,.];

"\f";

OUTPUT FILE = NNESTJ.OUT OFF;
SYSTEM;

```

Figure 32—Sample program for nonnested J-test, in SAS PC

```

*****
* PROGRAM:   NNESTJ.SAS   SOFTWARE: SAS PC 6.04   *
*           FILENAME     DESCRIPTION           *
* INPUTS:   DATA.SSD    TEST DATA SET        *
* PURPOSE:  PERFORM NONNESTED J-TEST.         *
*****;

LIBNAME CDRV 'C:\DATA\';

* ALL VARIABLES EXCEPT X3, X7, X6, AND X12 ARE COMMON TO ALL
* MODELS. SPECIFICATION 1 CONTAINS X3, X7, AND
* SPECIFICATION 2 CONTAINS X6, X12.;

* TO TEST SPECIFICATION 1: FIRST ESTIMATE SPECIFICATION 2.;

PROC REG DATA=CDRV.DATA;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
        RD1 RD2 RD3 X6 X12;
  OUTPUT OUT=HAT2 P=YHAT2;
RUN;

* TO TEST SPECIFICATION 1: NEXT FORCE PREDICTED VALUE FROM SPECIFICATION 2
* INTO SPECIFICATION 1;

PROC REG DATA=HAT2;
  MODEL Y1=YHAT2 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
        RD1 RD2 RD3 X3 X7;
RUN;

* TO TEST SPECIFICATION 2: FIRST ESTIMATE SPECIFICATION 1;

PROC REG;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
        RD1 RD2 RD3 X3 X7;
  OUTPUT OUT=HAT1 P=YHAT1;
RUN;

* TO TEST SPECIFICATION 2: NEXT FORCE PREDICTED VALUE FROM SPECIFICATION 1
* INTO SPECIFICATION 2;

PROC REG DATA=HAT1;
  MODEL Y1=YHAT1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
        RD1 RD2 RD3 X6 X12;
RUN;

* TEST STATISTIC CALCULATION FROM OUTPUT;
* THE T-STATISTIC FOR YHAT2 IS 2.2322, AND THE T-STATISTIC FOR YHAT1 IS 1.8920;
* SEE TEXT FOR INTERPRETATION OF RESULTS;

```

Figure 33—Sample program for nonnested J-test, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'NNESTJ.LIS'.
SET LOG = 'NNESTJ.LOG'.
*****
*   PROGRAM:   NNESTJ.SPS   SOFTWARE: SPSS/PC+ 4.01   *
*             FILENAME     DESCRIPTION              *
*   INPUTS:   DATA.SYS   TEST DATA SET           *
*   PURPOSE:  PERFORM NONNESTED J-TEST.            *
*****.

GET FILE = 'DATA.SYS'.

* ALL VARIABLES EXCEPT X3, X7, X6, AND X12 ARE COMMON TO ALL
* MODELS. SPECIFICATION 1 CONTAINS X3, X7, AND
* SPECIFICATION 2 CONTAINS X6, X12.

* TO TEST SPECIFICATION 1: FIRST ESTIMATE SPECIFICATION 2.

REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                      D7 D8 RD1 RD2 RD3 X6 X12
                      /DEPENDENT=Y1
                      /METHOD=ENTER
                      /SAVE PRED(YHAT2).

* TO TEST SPECIFICATION 1: NEXT FORCE PREDICTED VALUE FROM SPECIFICATION 2
* INTO SPECIFICATION 1.

REGRESSION VARIABLES = Y1 YHAT2 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3
                      D5 D6 D7 D8 RD1 RD2 RD3 X3 X7
                      /DEPENDENT=Y1
                      /METHOD=ENTER.

* TO TEST SPECIFICATION 2: FIRST ESTIMATE SPECIFICATION 1.

REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5
                      D6 D7 D8 RD1 RD2 RD3 X3 X7
                      /DEPENDENT=Y1
                      /METHOD=ENTER
                      /SAVE PRED(YHAT1).

* TO TEST SPECIFICATION 2: NEXT FORCE PREDICTED VALUE FROM SPECIFICATION 1
* INTO SPECIFICATION 2.

REGRESSION VARIABLES = Y1 YHAT1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5
                      D6 D7 D8 RD1 RD2 RD3 X6 X12
                      /DEPENDENT=Y1
                      /METHOD=ENTER.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* THE T-STATISTIC FOR YHAT2 IS 2.2322 AND THE T-STATISTIC FOR YHAT1 IS 1.8920.
* SEE TEXT FOR INTERPRETATION OF RESULTS.
FINISH.

```

OMISSION OF VARIABLES: THE RAMSEY RESET TEST

This version of the Regression Specification Error Test (RESET) may be used to test for omission of relevant explanatory variables. When one or more relevant variables (either unobserved or unobservable) are omitted from a model, the error term of the incorrect model includes the influence of the omitted variables. If proxy variable(s), Z , can be constructed to stand in for the omitted variable(s), a specification error test may be formed by testing if Z has significant incremental explanatory power for y .

In this version of RESET, a proxy variable matrix Z is constructed from the second, third, and fourth moments of the fitted values of y from the original model.

Let the model of interest be

$$y = X\beta + \epsilon. \quad (10)$$

This model is "restricted" in the sense that it does not contain the proxy variables in matrix Z . The "augmented" model does contain them.

The RESET test is then conducted following the steps described below.

- | | |
|--------|---|
| Step 1 | Using OLS, estimate the restricted model (10). |
| Step 2 | Calculate fitted values: $\hat{y} = X\hat{\beta}$ |
| Step 3 | Form the proxy variables as powers of the fitted values: $\hat{y}^2, \hat{y}^3, \hat{y}^4$. |
| Step 4 | Estimate the augmented model by OLS: regress y on $X, \hat{y}^2, \hat{y}^3, \hat{y}^4$. |
| Step 5 | Using an F -test, check if the coefficients on the columns of the Z matrix are jointly significant. If so, the null hypothesis of no specification error is rejected. |

In the sample programs for the nonnested F -test and the nonnested J -test (previously discussed), we examined whether a model that contained variables $X3$ and $X7$ or variables $X6$ and $X12$ was to be preferred. Evidence was found that the preferred model would contain all four variables. In illustrating the RESET test, all of these variables will be *omitted* in forming the restricted model to check whether the RESET test detects this omission.

In fact, the F -test for incremental explanatory power yields an F -value of 0.6024 (P -value = 0.6135), and it is concluded that specification error is absent. The previous tests used $X3$ and $X7$, and $X6$ and $X12$, directly, but the RESET test uses no specific information about these variables. Thus, it is illustrated that the RESET test may not be powerful for detecting misspecification. If specific variables are to be tested to determine whether they should be included in a regression model, they should be tested explicitly rather than through a nonspecific test like RESET.

Figures 34 through 36 are sample programs for the Ramsey RESET Test.

NOTES:

1. Thursby (1979, 1981, 1982) discusses using RESET in conjunction with tests for other types of specification error.
 2. A method that has been shown by Monte Carlo studies to be preferable to using powers of \hat{y} is that of Thursby and Schmidt (1977). They used the second, third, and fourth powers of all explanatory variables to make up the proxy vector Z . However, with many explanatory variables, this may be unwieldy.
-
-

Recommended references: Griffiths, Hill, and Judge (1993, 498-499); Judge et al. (1984, 364); Kennedy (1985, 71, 81; 1992, 95, 102, 104); Kmenta (1986, 452-455); Maddala (1988, 162, 407); Ramsey (1969, 350-371); Thursby (1979, 222-225; 1981, 117-123; 1982, 314-321); Thursby and Schmidt (1977, 635-641).

Figure 34—Sample program for the Ramsey RESET Test, in GAUSS-386

```

/*****
* PROGRAM:   RESET.G       SOFTWARE: GAUSS-386 V3.0  *
*           FILENAME      DESCRIPTION             *
* INPUTS:   DATA.DAT     GAUSS-386 DATA SET     *
* PURPOSE:  PERFORM RAMSEY RESET TEST.           *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = RESET.OUT RESET;
NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);
Y = DATA[.,IY1];

X = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                        ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3, .];

@----- OLS ESTIMATION OF "RESTRICTED" MODEL -----@

KR = COLS(X);

B = INV(X'X)*X'Y; @ BETAS @
YHAT = X*B; @ FITTED VALUES @
E = Y - YHAT; @ RESIDUALS @
RSSR = E'E; @ RESTRICTED RSS @
SER = SQRT(INV(NCASE-KR)*RSSR); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSSR/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV = INV(NCASE-KR)*RSSR*INV(X'X); @ COV MATRIX OF BETAS @
SE = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T = B ./ SE; @ T-STATISTICS @
PT = 2*CDFTC(ABS(T),(NCASE-KR)); @ P-VALUES @
PRN = B ~ SE ~ T ~ PT; @ FOR PRINTING @

" ";
" ";
" ";
" OLS RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS = ";; NCASE;
" ";
" STANDARD ERROR OF REGRESSION = ";; SER;
" ";
" RESIDUAL SUM OF SQUARES = ";; RSSR;
" ";
" R-SQUARED = ";; RSQ;
" ";
" ";
" VARIABLE COEFF STD ERROR T-RATIO P-VALUE";
" ";
" INTERCEPT ";; PRN[1, .];

```

(continued)

Figure 34—Continued

```

I      = 1;
DO WHILE I <= KR-1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];;   FORMAT /M1 /RD 12,4;   PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";
"\f";

@-----          RESET VARIABLES          -----@

Y2      = YHAT^2;
Y3      = YHAT^3;
Y4      = YHAT^4;

@-----          OLS ESTIMATION OF "UNRESTRICTED" REGRESSION          -----@

X        = X ~ Y2 ~ Y3 ~ Y4;
KU       = COLS(X);
B        = INV(X'X)*X'Y;           @ BETAS           @
YHAT     = X*B;                   @ FITTED VALUES  @
E        = Y - YHAT;              @ RESIDUALS     @
RSSU     = E'E;                   @ UNRESTRICTED RSS @
SER      = SQRT(INV(NCASE-KU)*RSSU); @ STD ERROR OF REGRESSION @
RSQ      = 1 - RSSU/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED     @
COV      = INV(NCASE-KU)*RSSU*INV(X'X); @ COV MATRIX OF BETAS @
SE       = SQRT(DIAG(COV));       @ STD ERRS OF BETAS @
T        = B ./ SE;               @ T-STATISTICS   @
PT       = 2*CDFTC(ABS(T), (NCASE-KU)); @ P-VALUES      @
PRN      = B ~ SE ~ T ~ PT;       @ FOR PRINTING   @

" ";
" ";
" ";
"  OLS RESULTS FOR UNRESTRICTED REGRESSION ";
" ";
" ";
"  NUMBER OF OBSERVATIONS =      ";;   NCASE;
" ";
"  STANDARD ERROR OF REGRESSION = ";;   SER;
" ";
"  RESIDUAL SUM OF SQUARES =      ";;   RSSU;
" ";
"  R-SQUARED =                    ";;   RSQ;
" ";
" ";
"  VARIABLE          COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"  INTERCEPT ";;   PRN[1,.];

I      = 1;
DO WHILE I <= KU-4;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];;   FORMAT /M1 /RD 12,4;   PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";

```

(continued)

Figure 34—Continued

```

"          Y2 ";;      PRN[20,.];
"          Y3 ";;      PRN[21,.];
"          Y4 ";;      PRN[22,.];
" ";
" ";
@--- F-STAT FOR INCREMENTAL EXPLANATORY POWER OF RESET VARIABLES ---@
F      =  ( (RSSR-RSSU) / (KU - KR) ) / ( RSSU / (NCASE-KU) );
PROB   =  CDFFC(F, (KU-KR), (NCASE-KU));
" ";
"  RESET TEST STATISTIC  F =";;  F;;  "          PROB =";;  PROB;
"\f";
OUTPUT FILE = RESET.OUT OFF;
SYSTEM;

```

Figure 35—Sample program for the Ramsey RESET Test, in SAS PC

```

*****
* PROGRAM:  RESET.SAS      SOFTWARE: SAS PC 6.04      *
*          FILENAME      DESCRIPTION                *
* INPUTS:  DATA.SSD     TEST DATA SET             *
* PURPOSE:  PERFORM RAMSEY RESET TEST.              *
*****;

LIBNAME CDRV 'C:\DATA\';

PROC REG DATA=CDRV.DATA;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
  OUTPUT OUT=HAT P=YHAT;
RUN;

DATA YHATX;
  SET HAT;
  YHAT2=YHAT**2;
  YHAT3=YHAT**3;
  YHAT4=YHAT**4;
RUN;

PROC REG DATA=YHATX;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
        RD1 RD2 RD3 YHAT2 YHAT3 YHAT4;
  FTEST : TEST YHAT2, YHAT3, YHAT4;
RUN;

* TEST STATISTIC CALCULATION FROM OUTPUT.
* CALCULATE F = ((RSSR-RSSU)/D)/(RSSU/(N-K)).
* WHERE RSSR IS THE RESIDUAL SUM OF SQUARES FOR THE RESTRICTED EQUATION.
*       RSSU IS THE RESIDUAL SUM OF SQUARES FOR THE UNRESTRICTED EQUATION.
*       N IS THE NUMBER OF CASES (1624).
*       D IS THE NUMBER OF RESTRICTIONS (3).
*       K IS THE NUMBER OF PARAMETERS IN UNRESTRICTED REGRESSION (22).
* IF YHAT2, YHAT3, AND YHAT4 ARE NOT JOINTLY SIGNIFICANT THEN THE
* NULL HYPOTHESIS OF OMITTED VARIABLES IS REJECTED (FTEST=0.6024);

```

Figure 36—Sample program for the Ramsey RESET Test, in SPSS/PC+

```

SET MORE = OFF.
SET LIS = 'RESET.LIS'.
SET LOG = 'RESET.LOG'.
*****
*   PROGRAM:   RESET.SPS       SOFTWARE: SPSS/PC+ 4.01   *
*   FILENAME  DESCRIPTION      *
*   INPUTS:   DATA.SYS       TEST DATA SET          *
*   PURPOSE:  PERFORM RAMSEY RESET TEST.              *
*****

GET FILE = 'DATA.SYS' .

* RESTRICTED REGRESSION.
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6
                    D7 D8 RD1 RD2 RD3
                    /DEPENDENT=Y1
                    /METHOD=ENTER
                    /SAVE PRED(YHAT).

COMPUTE YHAT2 = YHAT**2.
COMPUTE YHAT3 = YHAT**3.
COMPUTE YHAT4 = YHAT**4.

* UNRESTRICTED REGRESSION.
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                    RD1 RD2 RD3 YHAT2 YHAT3 YHAT4
                    /CRITERIA=TOLERANCE(.0000001)
                    /DEPENDENT=Y1
                    /METHOD=ENTER.

* THE LOW TOLERANCE CRITERIA IS EMPLOYED TO FORCE YHAT2 AND YHAT3 INTO
* THE EQUATION. SPSS/PC+ WILL ISSUE A WARNING ABOUT THIS. SAS PC DOES NOT
* ISSUE A WARNING ABOUT THIS. THE F-TESTS FROM SPSS/PC+ AND SAS PC ARE IDENTICAL.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* CALCULATE  $F = ((RSSR - RSSU) / D) / (RSSU / (N - K))$ .
* WHERE RSSR IS THE RESIDUAL SUM OF SQUARES FOR THE RESTRICTED EQUATION.
*      RSSU IS THE RESIDUAL SUM OF SQUARES FOR THE UNRESTRICTED EQUATION.
*      N IS THE NUMBER OF CASES (1624).
*      D IS THE NUMBER OF RESTRICTIONS (3).
*      K IS THE NUMBER OF PARAMETERS IN UNRESTRICTED REGRESSION (22).
* IF YHAT2, YHAT3, AND YHAT4 ARE NOT JOINTLY SIGNIFICANT, THEN THE
* NULL HYPOTHESIS OF OMITTED VARIABLES IS REJECTED (FTEST=0.6024).
FINISH.

```

MULTI-COLLINEARITY DIAGNOSTICS

Multicollinearity exists when there is a linear relationship among some subset of regressors in a model. Multicollinearity exists in virtually every data set but is a problem only when the linear relationship among regressors is very strong. The main effects of high multicollinearity are that the variances of the estimated coefficients are inflated and the t -statistics are consequently small; and, in extreme cases, the coefficients may be very sensitive and unstable with respect to minor changes in model specification and data.

Since multicollinearity is essentially a matter of degree, attention has focused on descriptions of its extent and on assessments of the extent to which it inflates the variances of the coefficients. Two popular methods for assessing the strength of multicollinearity are discussed below.

Auxiliary Regressions

This is more useful than the popular method of simply looking at the correlation matrix of regressors, since the latter only reveals pair-wise relationships between variables. The auxiliary regression method makes use of the fact that the R^2 statistic is a measure of the extent to which one variable is a linear combination of a set of other variables. The strategy is to regress each continuous regressor, in turn, on all remaining regressors and to check the R^2 of each auxiliary regression. High R^2 values indicate the existence of strong linear dependencies. If only one linear relationship is very strong, then it provides an indication of which variable is suspect. However, if more than one linear dependency is strong, then the multicollinearity is more generally distributed among the regressors.

The steps for performing auxiliary regressions and interpreting their results are described below.

- | | |
|--------|---|
| Step 1 | Specify the first explanatory variable as the dependent variable and perform OLS, using the remainder of the explanatory variables (including a constant) as regressors. |
| Step 2 | Calculate R^2 for this regression. A high R^2 (one rule of thumb might be approximately 0.90 or above) indicates that the first explanatory variable is a strong linear function of the remaining explanatory variables. This general rule of thumb should be used as a benchmark, not as a strict bound. |
| Step 3 | Repeat steps 1 and 2 for each of the continuous explanatory variables in turn. |

For the eight continuous regressors in the standard model in the sample programs (Figures 37 through 39), the R^2 values for the auxiliary regressions range from 0.0895 to 0.8241. Therefore, it is concluded that multicollinearity is not severe.

Recommended references: Fomby, Hill, and Johnson (1984, 293-294); Greene (1990, 277-281); Griffiths, Hill, and Judge (1993, 436-437); Judge et al. (1984, 902-904); Kennedy (1985, 150, 153; 1992, 179-180, 183-184).

Figure 37—Sample program for performing auxiliary regressions, in GAUSS-386

```

/*****
* PROGRAM:  AUXREG.G      SOFTWARE: GAUSS-386 V3.0  *
*          FILENAME      DESCRIPTION              *
* INPUTS:   DATA.DAT    GAUSS-386 DATA SET      *
* PURPOSE:  EXECUTE AND REPORT AUXILIARY REGRESSIONS *
*          TO CHECK FOR MULTICOLLINEARITY.        *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = AUXREG.OUT RESET;
NAMES      = GETNAME("DATA");
OPEN D     = DATA.DAT VARINDXI;
NCASE      = ROWSF(D);
DATA       = READR(D,NCASE);
F          = CLOSE(D);

X          = DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

K          = COLS(X);

NAMES      = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

" ";
" ";
" ";
"  AUXILIARY      DEPENDENT      ";
"  REGRESSION     VARIABLE      R-SQUARED ";
" ";

I          = 1;
DO WHILE I <= K;

  XA       = X[.,I];

  IF I     == 1;
    XX     = X[.,2:K];
  ENDIF;

  IF I >= 2 AND I <= (K-1);
    XX     = X[.,1:(I-1)] ~ X[.,(I+1):K];
  ENDIF;

  IF I     == K;
    XX     = X[.,1:(K-1)];
  ENDIF;

  XX       = ONES(NCASE,1) ~ XX;

  @-----          OLS ESTIMATION OF AUXILIARY REGRESSION          -----@

  KA       = COLS(XX);

  B        = INV(XX'XX)*XX'XA;          @ BETAS          @
  E        = XA - XX*B;                @ RESIDUALS       @
  RSS      = E'E;                      @ RESIDUAL SUM OF SQUARES @

```

(continued)

Figure 37—Continued

```

SER      =  SQRT(INV(NCASE-KA)*RSS);           @ STD ERROR OF REGRESSION @
RSQ      =  1 - RSS/((NCASE-1)*(STDC(XA))^2); @ R-SQUARED           @
COV      =  INV(NCASE-K)*RSS*INV(XX'XX);     @ OLS COVARIANCE MATRIX @
SE       =  SQRT(DIAG(COV));                 @ STD ERRS OF BETAS    @
T        =  B ./ SE;                         @ T-STATISTICS FOR BETAS @
PT       =  CDFTC(ABS(T),(NCASE-K));         @ P-VALUES             @

PRN      =  B ~ SE ~ T ~ PT;                 @ FOR PRINTING        @

FORMAT /M2 /RD 8,0;  I;; FORMAT /M2 /RD 14,8; $NAMES[I,.];;
FORMAT /M2 /RD 12,4; RSQ;

I        =  I + 1;
ENDO;

"\f";

OUTPUT FILE = AUXREG.OUT OFF;
SYSTEM;

```

Figure 38—Sample program for performing auxiliary regressions, in SAS PC

```

*****
* PROGRAM:  AUXREG.SAS      SOFTWARE: SAS PC 6.04      *
*          FILENAME        DESCRIPTION                *
* INPUTS:   DATA.SSD      TEST DATA SET             *
* PURPOSE:  EXECUTE AND REPORT AUXILIARY REGRESSIONS *
*          TO CHECK FOR MULTICOLLINEARITY.           *
*****;

LIBNAME CDRV 'C:\DATA\';

* THE FOLLOWING IS THE MODEL TO BE ESTIMATED.;
PROC REG DATA=CDRV.DATA;
MM:  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;

* BELOW ARE THE EIGHT AUXILIARY REGRESSIONS FOR THE CONTINUOUS VARIABLES;

M1:  MODEL X1=X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
M2:  MODEL X2=X1 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
M3:  MODEL X8=X1 X2 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
M4:  MODEL X9=X1 X2 X8 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
M5:  MODEL X10=X1 X2 X8 X9 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
M6:  MODEL X13=X1 X2 X8 X9 X10 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
M7:  MODEL X14=X1 X2 X8 X9 X10 X13 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
M8:  MODEL X15=X1 X2 X8 X9 X10 X13 X14 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

* TEST STATISTIC CALCULATION FROM OUTPUT.
* OBSERVE R-SQUARED IN EACH REGRESSION. ONE RULE OF THUMB IS THAT AN
* R-SQUARED VALUE OF 0.9 OR HIGHER INDICATES SERIOUS COLLINEARITY. THIS
* IS A GENERAL RULE OF THUMB, NOT A STRICT BOUND. NONE OF THE EIGHT
* AUXILIARY REGRESSIONS IN THIS EXAMPLE HAS AN R-SQUARED VALUE ABOVE 0.9.;

```

Figure 39—Sample program for performing auxiliary regressions, in SPSS/PC+

```

SET MORE = OFF.
SET LIS='AUXREG.LIS'.
SET LOG='AUXREG.LOG'.
*****
*   PROGRAM:   AUXREG.SPS   SOFTWARE: SPSS/PC+ 4.01   *
*   FILENAME   DESCRIPTION                                     *
*   INPUTS:    MANUAL.SYS   TEST DATA SET             *
*   PURPOSE:   EXECUTE AND REPORT AUXILIARY REGRESSIONS *
*              TO CHECK FOR MULTICOLLINEARITY.         *
*****

GET FILE = 'DATA.SYS'.
* THE FOLLOWING IS THE MODEL TO BE ESTIMATED.
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=Y1
                      /METHOD=ENTER.
* BELOW ARE THE EIGHT AUXILIARY REGRESSIONS FOR THE CONTINUOUS VARIABLES.
REGRESSION VARIABLES = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=X1
                      /METHOD=ENTER.
REGRESSION VARIABLES = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=X2
                      /METHOD=ENTER.
REGRESSION VARIABLES = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=X8
                      /METHOD=ENTER.
REGRESSION VARIABLES = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=X9
                      /METHOD=ENTER.
REGRESSION VARIABLES = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=X10
                      /METHOD=ENTER.
REGRESSION VARIABLES = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=X13
                      /METHOD=ENTER.
REGRESSION VARIABLES = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=X14
                      /METHOD=ENTER.
REGRESSION VARIABLES = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                      RD1 RD2 RD3
                      /DEPENDENT=X15
                      /METHOD=ENTER.
* TEST STATISTIC CALCULATION FROM OUTPUT.
* OBSERVE R-SQUARED IN EACH REGRESSION. ONE RULE OF THUMB IS THAT AN
* R-SQUARED VALUE OF 0.9 OR HIGHER INDICATES SERIOUS COLLINEARITY. THIS
* IS A GENERAL RULE OF THUMB, NOT A STRICT BOUND. NONE OF THE EIGHT
* AUXILIARY REGRESSIONS IN THIS EXAMPLE HAS AN R-SQUARED VALUE ABOVE 0.9.
FINISH.

```

Condition Indices and the Condition Number

Strong multicollinearity among the regressors implies that at least one eigenvalue or characteristic root of the $(X'X)$ matrix is small. Condition indices are the square roots of the ratios of the largest eigenvalue of the standardized (XX) matrix to the remaining eigenvalues. The condition number is the largest of these values, that is, the square root of the ratio of the largest to the smallest eigenvalue. SAS PC and SPSS/PC+ both produce multicollinearity diagnostics based on condition indices as options of their regression routines. It is also easy to produce them in GAUSS-386. The steps described below may be followed in GAUSS-386.

- Step 1 Compute the square roots of the diagonal elements of (XX) . Use these to form a diagonal matrix (zeros except on the diagonal), then invert the diagonal matrix and call this result S .
- Step 2 Form the $K \times K$ matrix $Z = SXXS$.
- Step 3 Calculate the vector λ containing the K eigenvalues of Z ; identify the smallest one as λ_{\min} and the largest one as λ_{\max} .
- Step 4 Compute the vector of condition indices C as follows:

$$C = (\lambda_{\max}/\lambda)^{1/2}.$$

The largest of these indices is the condition number.

Extensive experimentation conducted by Belsley, Kuh, and Welsch (1980) suggests that condition indices in excess of 30 indicate the presence of multicollinearity; condition indices in excess of a few hundred indicate severe multicollinearity. In the sample programs, three condition indices are larger than 30 and one is greater than 100, which is consistent with the results of the auxiliary regressions—multicollinearity is moderate. Figures 40 through 42 are sample programs for determining the condition number.

NOTE: Belsley, Kuh, and Welsch (1980) present measures that describe the extent to which variances of estimated coefficients may be inflated because of the presence of multicollinearity; they also present measures to identify which regressors are most problematic. SPSS/PC+ and SAS PC have a preprogrammed option called Variance Decomposition Proportion, which helps to identify the variables that are involved in multicollinearity.

Recommended references: Belsley, Kuh, and Welsch (1980, chapter 3); Corlett (1990, 158-159); Greene (1990, 281); Johnston (1984, 249-250); Judge et al. (1984, 902, 914, 920); Kennedy (1985, 150, 153; 1992, 180, 183); Kmenta (1986, 439); Maddala (1988, 228).

Figure 40—Sample program for determining the condition number, in GAUSS-386

```

/*****
* PROGRAM:   CONDNUM.G       SOFTWARE: GAUSS-386 V3.0 *
*           FILENAME        DESCRIPTION           *
* INPUTS:    DATA.DAT      GAUSS-386 DATA SET   *
* PURPOSE:   COMPUTE REGRESSION RESULTS AND PRODUCE *
*           MULTICOLLINEARITY DIAGNOSTICS.       *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = CONDNUM.OUT ON;
NAMES      = GETNAME("DATA");
OPEN D     = DATA VARINDXI;
NCASE      = ROWSF(D);
DATA       = READR(D,NCASE);
F          = CLOSE(D);

Y          = DATA[.,IY1];

X          = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES      = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                    ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

@-----@                OLS ESTIMATION                @-----@

K          = COLS(X);

B          = INV(X'X)*X'Y;           @ BETAS           @
E          = Y - X*B;               @ RESIDUALS     @
RSS        = E'E;                   @ RESIDUAL SUM OF SQUARES @
SER        = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ        = 1 - RSS/((NCASE-1)*{STDC(Y)}^2); @ R-SQUARED     @
COV        = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE         = SQRT(DIAG(COV));        @ STD ERRS OF BETAS @
T          = B ./ SE;                @ T-STATISTICS FOR BETAS @
PT         = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES      @
PRN        = B ~ SE ~ T ~ PT;        @ FOR PRINTING  @

" ";
" ";
" ";
" OLS RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS = ;; NCASE;
" STANDARD ERROR OF REGRESSION = ;; SER;
" RESIDUAL SUM OF SQUARES = ;; RSS;
" R-SQUARED = ;; RSQ;
" ";
" ";
" VARIABLE COEFF STD ERROR T-RATIO P-VALUE";
" ";
" INTERCEPT ;; PRN[1,.];

I          = 1;
DO WHILE I <= K -1;

```

(continued)

Figure 40—Continued

```

FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

I      = I + 1;
ENDO;
" ";

@-----          FORM SCALED VERSION OF (X'X)          -----@

D      = SQRT(DIAG(X'X));
S      = INV(DIAGRV(EYE(K),D));
Z      = S*X'X*S;

@-----          COMPUTE EIGENVALUES OF Z          -----@

L      = EIGRS(Z);

LMIN   = MINC(L);
LMAX   = MAXC(L);

CONDINDX = SQRT(LMAX./L);
COND    = SQRT(LMAX/LMIN);

" ";
" ";
"  CONDITION INDICES  ";
" ";
CONDINDX;
" ";
" ";
"  CONDITION NUMBER:  C =";; COND;

"\f";

OUTPUT FILE = CONDDNUM.OUT OFF;
SYSTEM;

```

Figure 41—Sample program for determining the condition number, in SAS PC

```

*****
* PROGRAM:  CONDNUM.SAS   SOFTWARE: SAS PC 6.04   *
*          FILENAME     DESCRIPTION             *
* INPUTS:   DATA.SSD    TEST DATA SET         *
* PURPOSE:  COMPUTE REGRESSION RESULTS AND PRODUCE *
*          MULTICOLLINEARITY DIAGNOSTICS.       *
*****;

LIBNAME CDRV 'C:\DATA\';

* THE FOLLOWING IS THE MODEL TO BE ESTIMATED.;
PROC REG DATA=CDRV.DATA;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3
    / COLLIN;
RUN;

* TEST STATISTIC CALCULATION FROM OUTPUT.
* THE CONDITION INDEX IS AUTOMATICALLY CALCULATED IN SAS PC IF THE COLLIN
* OPTION IS USED. THREE OF THE CONDITION NUMBERS ARE LARGER
* THAN THE RULE-OF-THUMB CUTOFF OF 30, WITH ONE BEING LARGER THAN 100. THE VARIABLES
* MOST RESPONSIBLE FOR THE LARGE CONDITION NUMBERS SEEM TO BE X1 AND X2.;

```

Figure 42—Sample program for determining the condition number, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'CONDNUM.LIS'.
SET LOG = 'CONDNUM.LOG'.
*****
* PROGRAM:  CONDNUM.SPS   SOFTWARE: SPSS/PC+ 4.01   *
*          FILENAME     DESCRIPTION             *
* INPUTS:   DATA.SYS    TEST DATA SET         *
* PURPOSE:  COMPUTE REGRESSION RESULTS AND PRODUCE *
*          MULTICOLLINEARITY DIAGNOSTICS.       *
*****.

GET FILE = 'DATA.SYS' .
* THE FOLLOWING IS THE MODEL TO BE ESTIMATED.
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
  RD1 RD2 RD3
  /STATISTICS = COLLIN
  /DEPENDENT=Y1
  /METHOD=ENTER.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* THE CONDITION INDEX FOR EACH EIGENVALUE IS AUTOMATICALLY
* CALCULATED IN SPSS/PC+ IF THE COLLIN OPTION IS USED. THREE OF THE CONDITION
* NUMBERS ARE LARGER THAN THE RULE-OF-THUMB CUTOFF OF 30, WITH ONE BEING LARGER THAN 100.
* THE VARIABLES MOST RESPONSIBLE FOR THE LARGE CONDITION NUMBERS SEEM TO BE X1 AND X2.
FINISH.

```

TESTING FOR STRUCTURAL CHANGE

The Chow F -Test

The Chow F -test, more commonly known as the "Chow test," is a simple way to test if the underlying parameter values for a data set change across specified subsets of that data: across different time periods or household types, for example. The Chow test compares the RSS from a restricted model (that assumes that the parameters are constant across data subsets) with the RSS from an unrestricted model (that allows the parameters to vary across data subsets). The unrestricted RSS may be obtained by running separate regressions for the data subsets and summing the resulting RSSs or, alternatively, by running a single regression that includes a set of dummy and dummy-interaction variables that distinguish among the subsets of the data. Both methods are simple and they have identical results. Both are presented below, in GAUSS-386. In SAS PC and SPSS/PC+, only the second approach is presented. For the programs discussed here, the question of whether the data from "round 1" surveys are distinct from the data drawn from the other three rounds is investigated.

This example is slightly more complicated to program than typical examples of the Chow test because of the presence of two dummies to distinguish among the three rounds in the second data subset. In effect, distinct intercepts for all survey rounds are permitted, and this example only tests whether slope coefficients are distinct between round 1 and the other three rounds. The models used in this example are as follows:

- Round 1 model ($RD2 = RD3 = RD4 = 0$):

$$Y = \beta_0 + X\beta + \epsilon,$$

where X contains neither an intercept nor any "round" dummies.

- Rounds 2 through 4 model ($RD1 = 0$):

$$Y = \beta_0 + X\beta + \delta RD3 + \delta_4 RD4 + \epsilon,$$

where X is as described in the round 1 model, and $RD3$ and $RD4$ introduce intercept differentials for the third and fourth rounds.

Note that $RD4$ is not contained in the data set, but can be constructed from knowledge of $RD1$, $RD2$, and $RD3$.

- Restricted model (only intercepts allowed to vary):

$$Y = \beta_0 + X\beta + \delta_2 RD2 + \delta RD3 + \delta_4 RD4 + \epsilon.$$

First Approach: Estimating Separate Models for Two Data Subsets (GAUSS-386). In the first approach, the data are split into subsets and a separate model is estimated from each:

- | | |
|--------|--|
| Step 1 | Separate the data into two data subsets: one from the first round of the survey ($RD1 = 1$) and one from the other rounds ($RD1 = 0$). |
|--------|--|

- Step 2 Run three regressions:
- First:* Estimate the Round 1 model for the data set for which $RD1 = 1$ and retain the RSS. Call it RSS_1 .
- Second:* Estimate the Round 2 through 4 model for the data set for which $RD1 = 0$ and retain the RSS. Call it RSS_2 .
- Third:* Estimate the restricted model for the full data set and retain the RSS. Call it RSS_R for "restricted" RSS.

Step 3 The unrestricted RSS is $RSS_U = RSS_1 + RSS_2$.

Step 4 Form the test statistic

$$\hat{F} = [(RSS_U - RSS_R) / df_n] / [RSS_U / df_d].$$

Here, the numerator degrees of freedom is equal to the number of restrictions (the number of slope coefficients that are forced to be equal across the two models equals 15 in the sample programs) and the denominator degrees of freedom is equal to the degrees of freedom associated with the unrestricted model (sample size minus the total number of coefficients estimated in the unrestricted model[s]).

Second Approach: Dummy Variables (GAUSS-386, SAS PC, and SPSS/PC+ programs). In the second approach, dummy variables are used:

Step 1 Let $RD1$ be the dummy variable that identifies the first-round survey observations. Form the matrix of interaction variables $DX = RD1.*X$, where $.*$ is element-by-element multiplication of each row in X by corresponding elements of $RD1$ (15 rows in the sample programs).

Step 2 Estimate the unrestricted model by OLS:

$$y = \beta_0 + X\beta + DX\delta + \delta_2RD2 + \delta_3RD3 + \delta_4RD4 + \epsilon.$$

This is the unrestricted model, because the presence of the dummy interaction variables allows differential effects across subsamples for all slope coefficients.

Step 3 Estimate the restricted model by OLS:

$$y = \beta_0 + X\beta + \delta_2RD2 + \delta_3RD3 + \delta_4RD4 + \epsilon.$$

Comparing the restricted and unrestricted models, it is evident that the hypothesis to be tested is

$$H_0: \delta = 0, \text{ and}$$

$$H_1: \delta \neq 0.$$

Step 4 Compute the test statistic exactly as in step 4 above.

Both approaches to the test produce an F -statistic of 1.191 (df_1, df_2) = (15,190), hence the null hypothesis of equal slope coefficients in round 1 versus rounds 2 through 4 (no structural change) cannot be rejected.

The Chow test is applicable to a wide variety of hypotheses; this example shows only one case. Refer to the references for additional applications. Figures 43 through 45 are sample programs for the Chow test.

Recommended references: Chow (1960, 591-605); Fomby, Hill, and Johnson (1984, 197-199); Greene (1990, 218-222); Johnston (1984, 207-225); Kennedy (1985, 87-88, 186; 1992, 98, 108-109); Kmenta (1986, 420-422); Maddala (1988, 134).

Figure 43—Sample program for Chow test, in GAUSS-386

```

/*****
* PROGRAM:   CHOW.G           SOFTWARE: GAUSS-386 V3.0   *
* FILENAME  DESCRIPTION      *
* INPUTS:   DATA.DAT        GAUSS-386 DATA SET        *
* PURPOSE:  ILLUSTRATE TWO APPROACHES TO CHOW TEST.    *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = CHOW.OUT RESET;

NAMES      = GETNAME("DATA");
OPEN D     = DATA VARINDEXI;
NCASE      = ROWSF(D);
DATA       = READR(D,NCASE);
F          = CLOSE(D);

Y          = DATA[.,IY1];

X          = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

RD1        = DATA[.,IRD1];

NAMES      = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                    ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3, .];

@----- FIRST APPROACH: RESTRICTED REGRESSION -----@
@----- AND TWO SUBSET REGRESSIONS FOR THE -----@
@----- UNRESTRICTED CASE -----@
@----- RESTRICTED REGRESSION -----@

```

(continued)

Figure 43—Continued

```

K      = COLS(X);
B      = INV(X'X)*X'Y;           @ BETAS           @
E      = Y - X*B;              @ RESIDUALS    @
RSSR   = E'E;                  @ RESTRICTED RSS @
SER    = SQRT(INV(NCASE-K)*RSSR); @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSSR/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED    @
COV    = INV(NCASE-K)*RSSR*INV(X'X); @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV));      @ STD ERRS OF BETAS @
T      = B ./ SE;             @ T-STATISTICS FOR BETAS @
PT     = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES     @

PRN    = B ~ SE ~ T ~ PT;      @ FOR PRINTING @

" ";
" ";
" ";
" RESTRICTED REGRESSION RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS      = ;; NCASE;
" STANDARD ERROR OF REGRESSION = ;; SER;
" RESIDUAL SUM OF SQUARES    = ;; RSSR;
" R-SQUARED                   = ;; RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT ";; PRN[1,.];

I      = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";
"\f";

@----- REGRESSION ON FIRST-ROUND (RD1 = 1) SUBSET -----@

Y1     = SELIF(Y,RD1);
X1     = SELIF(X,RD1);
N1     = ROWS(X1);
X1     = X1[.,1:(K-3)];
K      = COLS(X1);

B      = INV(X1'X1)*X1'Y1;      @ BETAS           @
E      = Y1 - X1*B;           @ RESIDUALS    @
RSS1   = E'E;                 @ UNRESTRICTED RSS1 @
SER    = SQRT(INV(N1-K)*RSS1); @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSS1/((N1-1)*(STDC(Y1))^2); @ R-SQUARED    @
COV    = INV(N1-K)*RSS1*INV(X1'X1); @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV));      @ STD ERRS OF BETAS @
T      = B ./ SE;             @ T-STATISTICS FOR BETAS @
PT     = 2*CDFTC(ABS(T),(N1-K)); @ P-VALUES     @
PRN    = B ~ SE ~ T ~ PT;      @ FOR PRINTING @

```

[continued]

Figure 43—Continued

```

" ";
" ";
" ";
" REGRESSION RESULTS FOR FIRST ROUND SUBSET (RD1 = 1) ";
" ";
" ";
" NUMBER OF OBSERVATIONS      =   ;;   N1;
" STANDARD ERROR OF REGRESSION =   ;;   SER;
" RESIDUAL SUM OF SQUARES     =   ;;   RSS1;
" R-SQUARED                   =   ;;   RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";

" INTERCEPT   ;;   PRN[1, .];

I      = 1;
DO WHILE I <= K - 1;
  FORMAT /M1 /RD 12,8; $NAMES[I, .];; FORMAT /M1 /RD 12,4; PRN[I+1, .];

  I      = I + 1;
ENDO;
" ";
"\f";

@-----          REGRESSION ON NON-FIRST-ROUND DATA          -----@

Y2      = DELIF(Y, RD1);
X2      = DELIF(X, RD1);
N2      = ROWS(X2);

X2      = X2[., 1:K K+2 K+3];

NAME2   = NAMES[1:(K-1) K+1 K+2, .];

K       = COLS(X2);
B       = INV(X2'X2)*X2'Y2;          @ BETAS          @
E       = Y2 - X2*B;                @ RESIDUALS      @
RSS2    = E'E;                       @ UNRESTRICTED RSS2 @
SER     = SQRT(INV(N2-K)*RSS2);       @ STD ERROR OF REGRESSION @
RSQ     = 1 - RSS2/((N2-1)*(STDC(Y2))^2); @ R-SQUARED      @
COV     = INV(N2-K)*RSS2*INV(X2'X2); @ OLS COVARIANCE MATRIX @
SE      = SQRT(DIAG(COV));           @ STD ERRS OF BETAS @
T       = B ./ SE;                   @ T-STATISTICS FOR BETAS @
PT      = 2*CDFTC(ABS(T), (N2-K));    @ P-VALUES      @
PRN     = B ~ SE ~ T ~ PT;           @ FOR PRINTING   @

" ";
" ";
" ";
" REGRESSION RESULTS FOR NON-FIRST-ROUND SUBSET ";
" ";
" ";
" NUMBER OF OBSERVATIONS      =   ;;   N2;
" STANDARD ERROR OF REGRESSION =   ;;   SER;

```

(continued)

Figure 43—Continued

```

" RESIDUAL SUM OF SQUARES      =   ;;   RSS2;
" R-SQUARED                    =   ;;   RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT   ;;   PRN[1,.];

I       = 1;
DO WHILE I <= K - 1;
  FORMAT /M1 /RD 12,8; $NAME2[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I       = I + 1;
ENDO;
" ";

RSSU    = RSS1 + RSS2;
DFN     = COLS(X) - 4;
DFD     = NCASE - (2*DFN + 4);

F       = ( (RSSR - RSSU)/DFN ) / (RSSU/DFD);

PROBF   = CDFFC(F,DFN,DFD);

" ";
" ";
" ";
"   RESULTS FOR SUBSET REGRESSION APPROACH";
" ";
" ";
"   CHOW TEST:  F =";; F;; "   P-VALUE =";; PROBF;
" ";
"   NUMERATOR DF =";; DFN;
"   DENOMINATOR DF =";; DFD;

"\f";

@-----          SECOND APPROACH:  RESTRICTED REGRESSION          -----@
@-----          AND DUMMY-VARIABLE REGRESSION                  -----@
@-----          FOR UNRESTRICTED CASE                          -----@

K       = COLS(X);

DX      = RD1 .* X[.,2:(K-3)];

NAMES   = NAMES
          | "DX1" | "DX2" | "DX8" | "DX9" | "DX10" | "DX13" |
          | "DX14" | "DX15" | "DD1" | "DD2" | "DD3" | "DD5" |
          | "DD6" | "DD7" | "DD8" ;

X       = X ~ DX;

K       = COLS(X);

@-----          UNRESTRICTED DUMMY-VARIABLE REGRESSION          -----@
B       = INV(X'X)*X'Y;                                     @ BETAS @

```

(continued)

Figure 43—Continued

```

E      = Y - X*B;                @ RESIDUALS                @
RSSU   = E'E;                    @ UNRESTRICTED RSS    @
SER    = SQRT(INV(NCASE-K)*RSSU); @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSSU/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED          @
COV    = INV(NCASE-K)*RSSU*INV(X'X); @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV));        @ STD ERRS OF BETAS   @
T      = B ./ SE;                @ T-STATISTICS FOR BETAS @
PT     = 2*CDFTC(ABS(T), (NCASE-K)); @ P-VALUES            @
PRN    = B ~ SE ~ T ~ PT;        @ FOR PRINTING        @

" ";
" ";
" ";
" UNRESTRICTED DUMMY-VARIABLE REGRESSION RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS      = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES    = ";; RSSU;
" R-SQUARED                   = ";; RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT ";; PRN[1,.];

I      = 1;
DO WHILE I <= K - 1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";

DFN     = (K-4)/2;
DFD     = NCASE - K;

F       = ( (RSSR - RSSU)/DFN ) / (RSSU/DFD);
PROBF   = CDFFC(F,DFN,DFD);

" ";
" ";
" ";
" RESULTS FOR DUMMY-VARIABLE APPROACH";
" ";
" ";
" CHOW TEST: F =";; F;; " P-VALUE =";; PROBF;
" ";
" ";
" NUMERATOR DF =";; DFN;
" DENOMINATOR DF =";; DFD;
" ";
" \f";

OUTPUT FILE = CHOW.OUT OFF;
SYSTEM;

```

Figure 44—Sample program for Chow test, in SAS PC

```

*****
* PROGRAM:   CHOW.SAS      SOFTWARE: SAS PC 6.04      *
* FILENAME  DESCRIPTION  *
* INPUTS:   DATA.SSD    TEST DATA SET          *
* PURPOSE:  ILLUSTRATE TWO APPROACHES TO CHOW TEST. *
*****;

* THE NULL HYPOTHESIS BEING TESTED IS THAT THE SLOPE COEFFICIENTS ON
* THE EXPLANATORY VARIABLES ARE IDENTICAL IN ROUND 1 VERSUS ROUNDS 2-4.
* THE INTERCEPT IS ALLOWED TO VARY BY ROUND, EVEN IN THE RESTRICTED MODEL.;

LIBNAME CDRV 'C:\DATA\';

DATA DAT2;
  SET CDRV.DATA;

  DX1 = RD1*X1;
  DX2 = RD1*X2;
  DX8 = RD1*X8;
  DX9 = RD1*X9;
  DX10= RD1*X10;
  DX13= RD1*X13;
  DX14= RD1*X14;
  DX15= RD1*X15;
  DD1 = RD1*D1;
  DD2 = RD1*D2;
  DD3 = RD1*D3;
  DD5 = RD1*D5;
  DD6 = RD1*D6;
  DD7 = RD1*D7;
  DD8 = RD1*D8;
RUN;

PROC REG DATA=DAT2;
  MODEL Y1= X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
         RD1 RD2 RD3
         DX1 DX2 DX8 DX9 DX10 DX13 DX14 DX15
         DD1 DD2 DD3 DD5 DD6 DD7 DD8;
  B1 : TEST DX1=DX2=DX8=DX9=DX10=DX13=DX14=DX15=
         DD1=DD2=DD3=DD5=DD6=DD7=DD8=0;
RUN;

* THE F-TEST STATISTIC IS CALCULATED FROM THE "B1: TEST" COMMAND;
* FOR THIS EXAMPLE, F-TEST = 1.1913 (DF=15, 1590). WE CANNOT REJECT THE NULL
* HYPOTHESIS THAT THE SLOPE COEFFICIENTS ARE IDENTICAL IN THE TWO TIME PERIODS.;

```

Figure 45—Sample program for Chow test, in SPSS/PC+

```

SET MORE = OFF.
SET LIS = 'CHOW.LIS'.
SET LOG = 'CHOW.LOG'.
*****
*   PROGRAM:   CHOW.SPS       SOFTWARE: SPSS/PC+ 4.01   *
*             FILENAME      DESCRIPTION              *
*   INPUTS:   DATA.SYS     TEST DATA SET           *
*   PURPOSE:  ILLUSTRATE TWO APPROACHES TO CHOW TEST. *
*****

* THE NULL HYPOTHESIS BEING TESTED IS THAT THE SLOPE COEFFICIENTS ON
* THE EXPLANATORY VARIABLES ARE IDENTICAL IN ROUND 1 VERSUS ROUNDS 2-4.
* THE INTERCEPT IS ALLOWED TO VARY BY ROUND, EVEN IN THE RESTRICTED MODEL.

GET FILE = 'DATA.SYS' .

COMPUTE DX1 = RD1*X1.
COMPUTE DX2 = RD1*X2.
COMPUTE DX8 = RD1*X8.
COMPUTE DX9 = RD1*X9.
COMPUTE DX10= RD1*X10.
COMPUTE DX13= RD1*X13.
COMPUTE DX14= RD1*X14.
COMPUTE DX15= RD1*X15.
COMPUTE DD1 = RD1*D1.
COMPUTE DD2 = RD1*D2.
COMPUTE DD3 = RD1*D3.
COMPUTE DD5 = RD1*D5.
COMPUTE DD6 = RD1*D6.
COMPUTE DD7 = RD1*D7.
COMPUTE DD8 = RD1*D8.

*UNRESTRICTED MODEL.
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                    RD1 RD2 RD3
                    DX1 DX2 DX8 DX9 DX10 DX13 DX14 DX15
                    DD1 DD2 DD3 DD5 DD6 DD7 DD8
                    /DEPENDENT=Y1
                    /METHOD=ENTER.

*RESTRICTED MODEL.
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                    RD1 RD2 RD3
                    /DEPENDENT=Y1
                    /METHOD=ENTER.

* TEST STATISTIC CALCULATION FROM OUTPUT.
* CALCULATE FTEST = ((RSSR-RSSU)/D)/(RSSU/(N-K)).
* WHERE RSSR IS THE RESIDUAL SUM OF SQUARES FOR THE RESTRICTED EQUATION.
*     RSSU IS THE RESIDUAL SUM OF SQUARES FOR THE UNRESTRICTED EQUATION.
*     N IS THE NUMBER OF CASES (FOR THIS EXAMPLE 1624).
*     D IS THE NUMBER OF RESTRICTIONS (FOR THIS EXAMPLE 15).
*     K IS THE NUMBER OF PARAMETERS IN THE UNRESTRICTED MODEL
*         (FOR THIS EXAMPLE, 34).
* FOR THIS EXAMPLE, FTEST=1.1913 (DF=15, 1590). WE CANNOT REJECT THE NULL
* HYPOTHESIS THAT THE SLOPE COEFFICIENTS ARE IDENTICAL IN THE TWO TIME PERIODS.
FINISH.

```

TESTING FOR NONLINEAR VARIABLES

The "linearity" assumption of the Classical Linear Regression Model refers to the assumption that the *parameters* enter the equation linearly. No such assumption is required concerning the manner in which the variables enter the equation. However, it is common to specify that the variables enter linearly. If this is inappropriate, then the consequences are similar to other forms of misspecification, such as the omission of relevant explanatory variables. In fact, if the Taylor theorem is used, inappropriate functional forms may be viewed as a special case of the omitted variables problem (Kmenta 1986, 449–451). Because of the similarity of the two problems, test results that indicate inappropriate functional form may actually be revealing an omitted variable problem. One test that is less susceptible to this problem is Utts' Rainbow test.

Utts' Rainbow Test

This test is related to the Chow test for structural stability, with the sample divided into two subsamples according to the observations' influence (or leverage) on the regression results. If observations with high leverage displace the regression results significantly, then it may be concluded that the specification of the regression function is inadequate. The test makes use of a measure of leverage that is also used to detect influential outliers in a regression.

The model is the standard one:

$$y = X\beta + \epsilon.$$

The test is based on the difference in the RSS from the restricted regression (same model applies to all observations) and the RSS from the unrestricted regression (on observations that have small leverage). The null hypothesis is that this difference is zero. Keep in mind that this test assumes that the stochastic disturbance terms satisfy the classical assumptions. If they do not, then the test is not valid. Here, proceed under the assumption that the classical assumptions are satisfied.

Step 1 Perform OLS on the full data set and retain the residual sum of squares RSS_R (restricted RSS).

Step 2 Compute the leverage measure for each observation in X :

$$h_{ii} = x_i(X'X)^{-1}x_i'$$

where x_i is the i^{th} row of X . Sort the leverage measures into ascending order and select the half that are smallest. Identify observations in X and Y that correspond with the small leverage measures.

Step 3 Perform OLS on the subsample selected in step 2, and retain the residual sum of squares RSS_U (unrestricted).

Step 4 Calculate the statistic U :

$$U = \frac{(RSS_R - RSS_U) / (N/2)}{RSS_U / [(N/2) - K]} \sim F_{(N/2, N/2 - K)},$$

where K = the number of estimated coefficients.

A rejection of the null hypothesis implies that the functional form is inadequate. For these sample programs, $U = 1.195$ (F -critical = 1, P -value = 0.0058), so that the null hypothesis is rejected. Recall, however, that this model omits $X3$, $X6$, $X7$, and $X12$, and that heteroskedasticity afflicts the disturbances. An improved test would be to include the additional variables known to be significant and to correct for heteroskedasticity before conducting the Rainbow test. Figures 46 through 48 are sample programs for Utts' Rainbow test.

Recommended references: Kennedy (1992, 104); Kmenta (1986, 454–455); Krämer et al. (1985, 120–121); Utts (1982, 2801–2815).

Figure 46—Sample program for Utts' Rainbow test, in GAUSS-386

```

/*****
* PROGRAM:   RAINBOW.G       SOFTWARE: GAUSS-386 V3.0   *
*           FILENAME       DESCRIPTION                *
* INPUTS:   DATA.DAT      GAUSS-386 DATA SET        *
* PURPOSE:  EXECUTE AND REPORT UTTS' RAINBOW TEST    *
*           FOR ADEQUACY OF FUNCTIONAL FORM.         *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = RAINBOW.OUT RESET;

NAMES      = GETNAME("DATA");
OPEN D     = DATA VARINDXI;
NCASE      = ROWSF(D);
DATA       = READR(D,NCASE);
F          = CLOSE(D);

Y          = DATA[.,IY1];

X          = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES      = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                    ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

@-----@
                                OLS ESTIMATION
                                -----@

K          = COLS(X);
B          = INV(X'X)*X'Y;          @ BETAS          @
E          = Y - X*B;              @ RESIDUALS      @
RSS        = E'E;                  @ RESIDUAL SUM OF SQUARES @
SER        = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ        = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED    @
COV        = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE         = SQRT(DIAG(COV));      @ STD ERRS OF BETAS   @
T          = B ./ SE;              @ T-STATISTICS FOR BETAS @
PT         = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES          @
PRN        = B ~ SE ~ T ~ PT;      @ FOR PRINTING      @

" ";
" ";
" ";
" OLS RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS      = ;; NCASE;
" STANDARD ERROR OF REGRESSION = ;; SER;
" RESIDUAL SUM OF SQUARES    = ;; RSS;
" R-SQUARED                   = ;; RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT ;; PRN[1,.];

I          = 1;
DO WHILE I <= K -1;

```

(continued)

Figure 46—Continued

```

FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";
"\f";
@-----          CONSTRUCT VECTOR OF LEVERAGE MEASURES.          -----@
@-----          THE MATRIX X CONTAINS "OBSERVATION              -----@
@-----          NUMBER" IN THE FIRST COLUMN AND THE              -----@
@-----          CORRESPONDING LEVERAGE MEASURE IN THE            -----@
@-----          SECOND COLUMN.                                     -----@

N      = NCASE;
I      = 1;

XXI    = INV(X'X);
H      = ZEROS(N,2);

DO WHILE I <= N;                                @ LOOP OVER WHOLE SAMPLE @

  Z      = X[I,.];
  HII    = Z*XXI*Z';                              @ Ith LEVERAGE MEASURE @
  H[I,1] = I;
  H[I,2] = HII;

  I      = I + 1;

ENDO;                                            @ END OF LOOP @

@-----          SORT H BY THE MAGNITUDE OF THE LEVERAGE          -----@

H      = SORTC(H,2);
M      = H[1:N/2,.];                              @ SELECT LOWER HALF OF H @
M      = SORTC(M,1);                              @ AND SORT BY OBSERVATION @

@-----          CHOOSE ELEMENTS OF X AND Y THAT CORRESPOND        -----@
@-----          TO THE OBSERVATIONS IDENTIFIED IN M                -----@

YS     = Y[M[.,1],.];
XS     = X[M[.,1],.];

@-----          OLS ON SUBSET OF DATA HAVING SMALL              -----@
@-----          LEVERAGE VALUES                                  -----@

NS     = ROWS(XS);
KS     = COLS(XS);
BS     = INV(XS'XS)*XS'YS;
E      = YS - XS*BS;                              @ RESIDUALS @
RSSS   = E'E;                                     @ RESIDUAL SUM OF SQUARES @
SER    = SQRT(INV(NS-KS)*RSS);                     @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSSS/((NS-1)*(STDC(YS))^2);           @ R-SQUARED @
COV    = INV(NS-KS)*RSSS*INV(XS'XS);              @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV));                          @ STD ERRS OF BETAS @
T      = B ./ SE;                                  @ T-STATISTICS FOR BETAS @
PT     = 2*CDFTC(ABS(T),(NS-KS));                  @ P-VALUES @

PRN    = B ~ SE ~ T ~ PT;                          @ FOR PRINTING @

```

(continued)

Figure 46—Continued

```

" ";
" ";
" ";
" OLS RESULTS FOR SUBSAMPLE WITH SMALL LEVERAGE VALUES";
" ";
" ";
" NUMBER OF OBSERVATIONS      =   ;;   NS;
" STANDARD ERROR OF REGRESSION =   ;;   SER;
" RESIDUAL SUM OF SQUARES     =   ;;   RSSS;
" R-SQUARED                   =   ;;   RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT   ;;   PRN[1, .];

I      = 1;
DO WHILE I <= KS - 1;
  FORMAT /M1 /RD 12,8; $NAMES[I, .];; FORMAT /M1 /RD 12,4; PRN[I+1, .];

  I      = I + 1;
ENDO;

" ";

@-----      CALCULATION OF THE RAINBOW TEST STATISTIC      -----@
DFN      = N/2;                                @ NUMERATOR D.F.      @
DFD      = N/2 - K;                            @ DENOMINATOR D.F.     @

U        = ( (RSS - RSSS) / DFN ) / ( RSSS / DFD ) ;

PU       = CDFFC(U, DFN, DFD);

" ";
" ";
" ";
" RAINBOW TEST STATISTIC:  U =";; U;
" ";
" ";
" NUMERATOR D.F.      =";; DFN;
" DENOMINATOR D.F.   =";; DFD;
" ";
" P-VALUE            =";; PU;

" \E";

OUTPUT FILE = RAINBOW.OUT OFF;
SYSTEM;

```

Figure 47—Sample program for Utts' Rainbow test, in SAS PC

```

*****
* PROGRAM:  RAINBOW.SAS   SOFTWARE: SAS PC 6.04   *
*          FILENAME     DESCRIPTION             *
* INPUTS:   DATA.SSD   TEST DATA SET         *
* PURPOSE:  EXECUTE AND REPORT UTTS' RAINBOW TEST *
*          FOR ADEQUACY OF FUNCTIONAL FORM.     *
*****;

LIBNAME CDRV 'C:\DATA\';

* MODEL WITH ALL OBSERVATIONS (MODEL 1).;

PROC REG DATA=CDRV.DATA;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
  OUTPUT OUT=HDATA H=LEV;
RUN;

* MODEL WITH HALF OF THE OBSERVATIONS (812) THAT HAVE
* THE LEAST LEVERAGE (MODEL 2).;

PROC RANK DATA=HDATA OUT=RHDATA GROUP=2;
  VAR LEV;
  RANKS RLEV;
RUN;

PROC REG DATA=RHDATA;
  WHERE RLEV=0;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

* RETAIN THE RESPECTIVE RESIDUAL SUM OF SQUARES (RSS) VALUES.
* TEST STATISTIC CALCULATION FROM OUTPUT.
* HERE, THE UTTS TEST STATISTIC, U, IS CALCULATED AS:
*  $[(RSS\ MODEL\ R - RSS\ MODEL\ U)/(1624-812)]/[RSS\ MODEL\ U/(812-19)]=1.195.$ 
* U IS DISTRIBUTED AS AN F STATISTIC WITH N/2, (N/2)-K DEGREES OF FREEDOM.
* THE NULL HYPOTHESIS IS REJECTED (F CRITICAL = 1).
* SEE TEXT FOR INTERPRETATION.;

```

Figure 48—Sample program for Utts' Rainbow test, in SPSS/PC+

```

SET MORE = OFF.
SET LIS = 'RAINBOW.LIS'.
SET LOG = 'RAINBOW.LOG'.
*****
* PROGRAM:   RAINBOW.SPS   SOFTWARE: SPSS/PC+ 4.01   *
*           FILENAME     DESCRIPTION                *
* INPUTS:   DATA.SYS    TEST DATA SET            *
* PURPOSE:  EXECUTE AND REPORT UTTS' RAINBOW TEST  *
*           FOR ADEQUACY OF FUNCTIONAL FORM.      *
*****

GET FILE = 'DATA.SYS' .

* MODEL WITH ALL OBSERVATIONS (RESTRICTED MODEL).
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
                    RD1 RD2 RD3
                    /DEPENDENT=Y1
                    /METHOD=ENTER
                    /SAVE LEVER(LEV).
* THE LEV VARIABLE INDICATES THE INFLUENCE EACH OBSERVATION HAS ON THE
* COEFFICIENT ESTIMATES.
RANK LEV /NTILE (2).

* MODEL WITH HALF OF THE OBSERVATIONS (812) THAT HAVE
* THE LEAST LEVERAGE (UNRESTRICTED MODEL).

PROCESS IF (NLEV = 1).
REGRESSION VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15, D1 D2 D3 D5 D6 D7 D8
                    RD1 RD2 RD3
                    /DEPENDENT=Y1
                    /METHOD=ENTER.

* RETAIN THE RESPECTIVE RESIDUAL SUM OF SQUARES (RSS) VALUES.
* TEST STATISTIC CALCULATION FROM OUTPUT.
* HERE, THE UTTS TEST STATISTIC, U, IS CALCULATED AS:
*  $[(RSS\ MODEL_R - RSS\ MODEL_U) / (1624 - 812)] / [RSS\ MODEL_U / (812 - 19)] = 1.195.$ 
* U IS DISTRIBUTED AS AN F STATISTIC WITH N/2, (N/2)-K DEGREES OF FREEDOM.
* THE NULL HYPOTHESIS IS REJECTED (F CRITICAL = 1).
* SEE TEXT FOR INTERPRETATION.
FINISH.

```

Linear Splines This technique is useful for approximating a curvilinear regression without specifying the mathematical form of the curvature. A linear spline is a continuous piecewise-linear function, that is, one in which the adjacent line segments meet at the interval boundaries (or "knots"). As with other models that incorporate break points, the number and location of the intervals may be difficult to specify a priori. Attention should be paid to theoretical considerations, although a grid data search may also be employed, as in the example below. The linear spline is most appropriately used where the regression model is expected to be linear, but to have structural breaks at specific values of an explanatory variable. In the standard regression model the coefficients of the regression are restricted to be equal across spline segments. The standard version of this model is

$$y = X\beta + Z\gamma + \epsilon.$$

However, it is expected that the response of y to changes in Z is distinct for three distinct regions of Z . In the example at hand, y is household calorie intake per day and Z is total weekly household expenditures. X contains all of the remaining regressors. The relationship between caloric intake and total expenditures might be expected to be different for low-expenditure, medium-expenditure, and high-expenditure families, but the precise dividing lines between low, medium, and high may not be known. The spline program will help to determine this. Note that this model has two knots; it is possible to develop models that have more, but the tensions among good fit, theory, and parsimonious parameterization should be kept in mind.

It is useful to begin by considering this model as a dummy-variable model with $D_1 = 1$ for medium-expenditure households, zero otherwise; and $D_2 = 1$ for high-expenditure households, zero otherwise. Then the model is

$$y = X\beta + D_1\gamma_1 + D_1Z\gamma_1 + D_2\gamma_2 + D_2Z\gamma_2 + \epsilon.$$

The dummy variable model does not guarantee that the piecewise segments join at the knots. Let the first knot be at L , so that low-expenditure households have income $Z \leq L$. The second knot is at H , so that low- and medium-expenditure households have $Z \leq H$. Then continuity at the knots is ensured if the model is specified as

$$y = X\beta + D_1(Z - L)\gamma_1 + D_2(Z - H)\gamma_2 + \epsilon.$$

One way to proceed is to program the computer to do a grid search over L and H , performing OLS for each (L, H) pair and checking for the pair that minimizes the RSS. These sample programs illustrate this approach. Whether the spline function leads to a significant improvement in RSS may be tested with a standard F -test (note that this is a simple application of the Chow test for structural stability). In this version of the F -test, the numerator degrees of freedom is equal to the number of knots specified and the denominator degrees of freedom is

equal to the sample size less the total number of coefficients estimated in the spline function model. An alternative approach to spline modeling is given in Johnston 1984, 392–394.

The sample programs determine that the knot dividing low- and medium-expenditure households is at a log-expenditure level of approximately $Z = 2.45$ and that the knot dividing medium- and high-expenditure households is at a log-expenditure level of approximately $Z = 4.45$. The F -test (performed only in GAUSS-386) for the restricted (linear) model versus the unrestricted model (spline) yields $F = 5.3889$ (P -value = 0.0047), and the linear model is rejected in favor of the spline function.

NOTE: Since SPSS/PC+ for DOS does not include looping or macro capabilities (although SPSS/PC+ for Windows does allow loops), the spline program is not feasible. To accomplish the grid-search procedure, the SPSS/PC+ program would include thousands of lines, with the same batch of 15 to 20 lines repeated hundreds of times.

The spline program in SAS PC is feasible but a little clumsy. The program relies heavily on the macro facility included in SAS PC. This makes it difficult to understand. Basically, the macro feature allows the user to define his/her own procedure (in this case, SPLINE) and then run this new procedure with user-defined parameters (START1, STOP1, STOP2, INCRM, and DENOM).

In GAUSS-386, the spline program is more straightforward. Techniques used in this program are not unusual for GAUSS code; most GAUSS programmers could easily understand the program.

Notice that the sample programs (Figures 49 and 50) carry out an extensive grid search over a finely divided grid. This is not necessary: experimentation with large grid steps may enable the investigator to quickly narrow down the regions in which the knots lie; then a finer search may pinpoint them. Note also that the loops begin the grid search for the upper point (H or CUTOFF2) a specific distance above the lower point (L or CUTOFF1) to avoid overlapping regions for low- and high-expenditure households.

Recommended references: Greene (1990, 248-251); Johnston (1984, 392-396); Kmenta (1986, 569); Stewart and Wallis (1981, 202-204); Suits, Mason, and Chan (1978, 132-133).

Figure 49—Sample spline program, in GAUSS-386

```

/*****
* PROGRAM:   SPLINE.G       SOFTWARE: GAUSS-386 V3.0   *
*           FILENAME      DESCRIPTION                *
* INPUTS:   DATA.DAT     GAUSS-386 DATA SET        *
* PURPOSE:  USE SPLINE FUNCTION TO CHECK FOR NON-   *
*           LINEARITY WITH RESPECT TO VARIABLE X10.  *
*****/

@-NOTE:  RUN TIME IS ABOUT 7 MINUTES ON 486DX2-66.-@

FORMAT /M2 /RD 12,4;

OUTPUT FILE = SPLINE.OUT RESET;

NAMES     = GETNAME("DATA");
OPEN D    = DATA VARINDEXI;
NCASE     = ROWSF(D);
DATA      = READR(D,NCASE);
F         = CLOSE(D);

Y         = DATA[.,IY1];

Z1        = DATA[.,IX10];

X         = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9      IX13 IX14 IX15
                               ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3]
          ~ Z1;

NAMES     = NAMES[IX1 IX2 IX8 IX9      IX13 IX14 IX15
                  ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3 IX10,.];

@-----          OLS ESTIMATION          -----@

K         = COLS(X);
B         = INV(X'X)*X'Y;           @ BETAS           @
E         = Y - X*B;               @ RESIDUALS       @
RSS       = E'E;                   @ RESIDUAL SUM OF SQUARES @
SER       = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ       = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED     @
COV       = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE        = SQRT(DIAG(COV));        @ STD ERRS OF BETAS   @
T         = B ./ SE;                @ T-STATISTICS FOR BETAS @
PT        = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES           @
PRN       = B ~ SE ~ T ~ PT;        @ FOR PRINTING       @

" ";
" ";
" ";
" OLS RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS      = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES    = ";; RSS;
" R-SQUARED                   = ";; RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";

```

(continued)

Figure 49—Continued

```

" ";
" INTERCEPT ";; PRN[1,.];

I      = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";
"\f";

@-----
          LOOPS FOR SPLINE FUNCTION          -----@
@-----
          L-LOOP IS OUTER LOOP (FOR LOWER KNOT AT L) -----@
@-----
          H-LOOP IS INNER LOOP (FOR UPPER KNOT AT H) -----@

OUTPUT FILE = SPLINE.OUT OFF;

RSSR      = RSS;          @ RSS FOR ORIGINAL LINEAR MODEL @
                        @ THE "RESTRICTED" MODEL           @
RSSMIN     = RSS;
L          = 2.20;        @ OUTER LOOP TAKES L FROM 2.20 @
DO WHILE L <= 4.25 ;     @ TO 4.25 @
  H        = L + 0.5;    @ INNER LOOP TAKES H FROM L+0.5 @
  DO WHILE H <= 5.25;   @ TO 5.25 @

  D1      = DUMMYDN(Z1,L,2);
  D2      = DUMMYDN(Z1,H,2);

  XS      = X ~ D1.*(Z1 - L*ONES(NCASE,1)) ~ D2.*(Z1 - H*ONES(NCASE,1));

  BS      = INV(XS'XS)*XS'Y;

  ES      = Y - XS*BS;

  RSS     = ES'ES;

  IF RSS < RSSMIN;
    RSSMIN = RSS;          @ KEEP MINIMUM RSS @
    LOPT   = L;           @ L ASSOCIATED WITH MIN RSS @
    HOPT   = H;           @ H ASSOCIATED WITH MIN RSS @
  ENDIF;

@-----
          SHOW PROGRESS OF ITERATIONS ON SCREEN          -----@

FORMAT /M1 /RD 5,2; "L ="; L;; "H ="; H;;
FORMAT /M1 /RD 12,0; "RSSMIN ="; RSSMIN;; "RSSR ="; RSSR;

H      = H + 0.1;

ENDO;

L      = L + 0.1;

ENDO;
OUTPUT FILE = SPLINE.OUT ON;

```

(continued)

Figure 49—Continued

```

@----- OLS REGRESSION FOR SELECTED SPLINE FUNCTION -----@
NAMES = NAMES | "Z2" | "Z3";

D1 = DUMMYDN(Z1, LOPT, 2);
D2 = DUMMYDN(Z1, HOPT, 2);

Z2 = D1.*(Z1 - LOPT*ONES(NCASE, 1));
Z3 = D2.*(Z1 - HOPT*ONES(NCASE, 1));

X = X ~ Z2 ~ Z3;

K = COLS(X);

B = INV(X'X)*X'Y; @ BETAS @
E = Y - X*B; @ RESIDUALS @
RSSU = E'E; @ RESIDUAL SUM OF SQUARES @
SER = SQRT(INV(NCASE-K)*RSSU); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSSU/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV = INV(NCASE-K)*RSSU*INV(X'X); @ OLS COVARIANCE MATRIX @
SE = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T = B ./ SE; @ T-STATISTICS FOR BETAS @
PT = 2*CDFTC(ABS(T), (NCASE-K)); @ P-VALUES @

PRN = B ~ SE ~ T ~ PT; @ FOR PRINTING @

@----- PRINT RESULTS FOR SELECTED SPLINE FUNCTION -----@

FORMAT /M1 /RD 12,4;

" ";
" ";
" ";
" RESULTS FOR SELECTED SPLINE FUNCTION ";
" ";
" ";
" KNOTS ARE LOCATED AT:";
" ";
" L = ;; LOPT;
" H = ;; HOPT;
" ";
" NUMBER OF OBSERVATIONS = ;; NCASE;
" STANDARD ERROR OF REGRESSION = ;; SER;
" RESIDUAL SUM OF SQUARES = ;; RSSU;
" R-SQUARED = ;; RSQ;
" ";
" ";
" VARIABLE COEFF STD ERROR T-RATIO P-VALUE";
" ";
" INTERCEPT ;; PRN[1,.];

I = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I = I + 1;
ENDO;
" ";

```

(continued)

Figure 49—Continued

```

@----- F-TEST WHETHER (RSSR - RSSU) IS SIGNIFICANT -----@
DFN      = 2;                                @ NUMERATOR DF = # BREAKS @
                                                @ IN SPLINE @

DFD      = NCASE - K;

F        = ( (RSSR - RSSU) / DFN ) / (RSSU / DFD );
PF       = CDFFC(F,DFN,DFD);

" ";
" ";
" ";
" F-TEST FOR RESTRICTING TO LINEAR MODEL: F =";; F;
" ";
" NUMERATOR DF =";; DFN;
" DENOMINATOR DF =";; DFD;
" ";
" P-VALUE =";; PF;

"\f";

OUTPUT FILE = SPLINE.OUT OFF;
SYSTEM;

```

Figure 50—Sample spline program, in SAS PC

```

*****
* PROGRAM:   SPLINE.SAS      SOFTWARE: SAS PC 6.04      *
*           FILENAME      DESCRIPTION                *
* INPUTS:   DATA.SSD      TEST DATA SET            *
* OUTPUTS:  SPLOUT.SSD     RESULTS OF REGRESSIONS     *
* PURPOSE:  USE SPLINE FUNCTION TO CHECK FOR NON-    *
*           LINEARITY WITH RESPECT TO VARIABLE X10.   *
*****;
* NOTE: RUN TIME IS ABOUT 30 MINUTES ON 486DX2-66;
LIBNAME CDRV 'C:\DATA\';
* NONLINEARITIES ARE SUSPECTED ALONG THE DIMENSION OF THE LOG OF
* TOTAL EXPENDITURE PER CAPITA (X10). X10 WILL BE SPLIT INTO
* THREE SECTIONS.

* THE FOLLOWING PROC SUMMARY AND DATA STEPS MERGE THE MINIMUM AND
* MAXIMUM OF X10 ONTO EACH OBSERVATION IN THE ORIGINAL DATA SET.;

DATA DATAX;
  SET CDRV.DATA;
  CONSTANT=1;

PROC SUMMARY DATA=DATAX;
  VAR X10;
  ID CONSTANT;
  OUTPUT OUT=MINMAX MIN=MINX10 MAX=MAXX10;

DATA SDATA;
  MERGE DATAX MINMAX(DROP=_TYPE_ _FREQ_);
  BY CONSTANT;

* THE FOLLOWING DATA STEP WILL CREATE A TEMPORARY BINARY DATA FILE TO STORE
* A MODEL NAME AND ROOT MEAN SQUARE ERROR (RMSE) FOR EACH REGRESSION. THIS
* STEP IS JUST CREATING A FIRST DUMMY RECORD.;

FILENAME OUTPUT 'C:\DATA\SPLINE.BIN';

DATA _NULL_;
  _MODEL_ = 'DUMMY';
  _RMSE_ = .;
  FILE OUTPUT RECFM=N;
  PUT
    _MODEL_ $8.
    _RMSE_ RB4. ;

* THE FOLLOWING STATEMENT BEGINS THE DEFINITION OF THE SAS PC MACRO.;
%MACRO SPLINE;

* START, STOP, AND INCRM MUST BE INTEGERS.;
* THEREFORE, THE VALUES ARE DIVIDED BY DENOM IN THE DATA SET;

%DO PNT1 = &START1 %TO &STOP1 %BY &INCRM;
  %DO PNT2 = &PNT1 + &INCRM2 %TO &STOP2 %BY &INCRM;

* X10 IS THE VARIABLE ACROSS WHICH WE SUSPECT NONLINEARITY OF THE
* REGRESSION LINE.;
DATA SPLINE;
  SET SDATA (KEEP=Y1 X1 X2 X8 X9 X10 X13 X14 X15
            D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3 MINX10 MAXX10);

```

(continued)

Figure 50—Continued

```

* THE FOLLOWING USES MACRO VARIABLES TO CREATE THE TWO CUTOFFS.;
CUTOFF1 = &PNT1./&DENOM.;
CUTOFF2 = &PNT2./&DENOM.;
* THE FOLLOWING CREATES Z1, Z2, Z3 AS EXPLAINED IN TEXT.;
IF (X10 LT MINX10) THEN Z1=0;
IF (X10 GE MINX10 AND X10 LT CUTOFF1) THEN
  Z1=X10-MINX10;
IF (X10 GE CUTOFF1) THEN
  Z1=&PNT1./&DENOM.-MINX10;
IF (X10 LT CUTOFF1) THEN Z2=0;
IF (X10 GE CUTOFF1 AND X10 LT CUTOFF2)
  THEN Z2=X10-CUTOFF1;
IF (X10 GE CUTOFF2) THEN
  Z2=CUTOFF2-CUTOFF1;
IF (X10 LT CUTOFF2) THEN Z3=0;
IF (X10 GE CUTOFF2 AND X10 LT MAXX10)
  THEN Z3=X10-CUTOFF2;
IF (X10 GE MAXX10 ) THEN Z3=MAXX10-CUTOFF2;

* THE FOLLOWING REGRESSION SAVES THE RMSE AND A MODEL LABEL TO THE BINARY ;
* OUTPUT FILE C:\DATA\SPLINE.BIN.;
PROC REG DATA=SPLINE
  OUTEST=SPLEST NOPRINT;
P&PNT1.P&PNT2.: MODEL Y1=
  X1 X2 X8 X9 Z1 Z2 Z3 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
DATA _NULL_;
  SET SPLEST;
  FILE OUTPUT RECFM=N MOD;
  PUT
    _MODEL_ $8.
    _RMSE_ RB4. ;

* THE FOLLOWING PROVIDES OUTPUT TO THE SCREEN TO MONITOR THE PROGRESS OF THE
* PROGRAM.;
DATA _NULL_;
  FILE 'CON';
  CUTOFF1 = &PNT1./&DENOM.;
  CUTOFF2 = &PNT2./&DENOM.;
  PUT " CUTOFF1 = " CUTOFF1 " CUTOFF2 = " CUTOFF2;
%END;
%END;
RUN ;

%MEND SPLINE;
* THE USER MUST PROVIDE THE SEARCH RANGE FOR CUTOFF1 AND CUTOFF2 AND THE
* INCREMENTS USED TO DETERMINE THE PRECISION OF THE SEARCH. IN SAS, MACRO
* PARAMETERS MUST BE INTEGERS. THEREFORE, WE USE START1, STOP1, STOP2,
* INCRM, AND INCRM2 TO DEFINE PNT1 AND PNT2. THEN, WE DIVIDE THESE INTEGERS
* BY DENOM TO DERIVE CUTOFF1 AND CUTOFF2. THE VALUES OF CUTOFF1 AND CUTOFF2
* ARE IN THE SAME UNITS AS THE VARIABLE OF INTEREST (X10). PNT1 VARIES
* FROM START1 TO STOP1, INCREASING BY INCRM FOR EACH REGRESSION (THIS
* CORRESPONDS TO CUTOFF1 VARYING FROM START1/DENOM TO STOP1/DENOM). FOR
* EACH PNT1 VALUE, PNT2 RANGES FROM PNT1 + INCRM2 TO STOP2, ALSO
* INCREASING BY INCRM FOR EACH REGRESSION.
* FOR OUR EXAMPLE, CUTOFF1 RANGES FROM 2.2 TO 4.25 AT INCREMENTS OF 0.01,
* AND CUTOFF2 RANGES FROM CUTOFF1+0.5 TO 5.25 AT INCREMENTS OF 0.01.
* PRIOR TO THIS DETAILED SEARCH, AN INITIAL ROUGH SEARCH COULD BE

```

(continued)

Figure 50—Continued

```

* CONDUCTED WITH LARGER GRID STEPS BY INCREASING THE INCRM.
* FOR INSTANCE, IF INCRM = 25, THE CUTOFFS WILL CHANGE WITH INCREMENTS OF
* 0.25. THE LARGER INCRM VALUE WILL RESULT IN A SUBSTANTIALLY REDUCED
* EXECUTION TIME.;

%LET START1 = 220;
%LET STOP1 = 425;
%LET STOP2 = 525;
%LET INCRM = 10;
%LET INCRM2 = 50;
%LET DENOM = 100;
%SPLINE;

* THE FOLLOWING DATA AND PROC STATEMENTS READ IN THE RESULTS FROM EACH
* REGRESSION AND PROVIDE COMPLETE DESCRIPTIVE STATISTICS.;

DATA CDRV.SPLOUT;
  INFILE OUTPUT RECFM=N;
  INPUT
    _MODEL_ $8.
    _RMSE_ RB4. ;

PROC UNIVARIATE DATA=CDRV.SPLOUT;
  VAR _RMSE_;
  ID _MODEL_;

* INTERPRETING OUTPUT;
* THE MODEL WITH THE OPTIMAL CUTOFFS IS INDICATED AS THE MODEL WITH THE
* MINIMUM RMSE (ROOT MEAN SQUARE ERROR) THAT CORRESPONDS TO THE
* MINIMUM RESIDUAL SUM OF SQUARES.

* PROC UNIVARIATE LISTING DISPLAYS THIS MINIMUM AND THE
* ACCOMPANYING MODEL LABEL (ID) UNDER THE "EXTREMES" HEADING.;
* IN THIS EXAMPLE, THE OPTIMAL CUTOFFS (OR KNOTS) ARE 2.45 AND 4.45
* FOR THE SEARCH INCREMENTS OF 0.25. WITH A SEARCH USING INCREMENTS
* OF 0.1, THE CUTOFFS ARE 2.50 AND 4.50;

```

5 DEFICIENT DATA PROBLEMS

INFLUENTIAL OBSERVATIONS

When an observation has an unusually large or small value for the dependent variable or for a regressor, that observation can substantially influence a regression. It is helpful to be able to detect and identify such observations in order to check whether they are erroneous values. The basic idea in detection is to examine how the omission of a suspect observation affects the overall regression fit and the parameter estimates. If the effect is "large," then the relevant data point is considered to be an "influential observation."

DFFITS The DFFITS statistic is a standardized measure of the effect of dropping the i^{th} observation on the fitted value of the dependent variable. DFFITS is calculated as

$$DFFITS_i = [\hat{y}_i - \hat{y}(i)] / (s_i * h_{ii}^{1/2}),$$

where

- \hat{y}_i = i^{th} OLS fitted value of the dependent variable, y_i ;
- $\hat{y}(i)$ = fitted value of y_i , after deleting the i^{th} observation and reestimating the parameters;
- s_i = standard error of the residuals, with i^{th} observation deleted;
- h_{ii} = i^{th} diagonal of the projection matrix, $x_i(X'X)^{-1}x_i'$;
- x_i' = i^{th} row of X , the $N \times K$ matrix of explanatory variables;
- N = number of observations; and
- K = number of explanatory variables, including the constant term.

Formal critical values for DFFITS statistics have not been developed, but some rules of thumb have been suggested. One such rule states that an absolute $DFFITS_i$ value greater than $2(K/N)^{1/2}$ (DFFITS can be both positive or negative) indicates an influential observation (Krasker, Kuh, and Welsch 1983). An alternative rule suggests 0.34 as a useful cutoff (Welsch 1980).

DFFITS statistics are calculated following the steps described below.

- Step 1 Perform OLS on the full data set and retain the fitted- y values.
- Step 2 Loop through the data, at the i^{th} loop deleting the i^{th} observation, and perform steps 3 through 5.

- Step 3 With the data set reduced by one observation, calculate the OLS coefficients and the standard error of the residuals.
- Step 4 using the X_i values, calculate the fitted value of y_i , $\hat{y}(i)$.
- Step 5 Calculate the DFFITS statistic according to the formula above.

The sample programs for DFFITS calculation, Figures 51 through 53, estimate the model that has been used in all other sample programs. Using the $2(K/N)^{1/2}$ cutoff, all three sample programs find a total of 100 (out of 1,624) influential observations. The largest 10 are printed.

Note that SPSS/PC+ and SAS PC label these calculations differently in their preprogrammed options. In SAS PC, the option called DFFITS produces what is described in the text as DFFITS_{*i*}. SPSS/PC+, however, calculates the same statistic for each observation, but the procedure that generated the numbers is called SDFIT.

NOTE: DFBETAS is another procedure in SPSS/PC+ that assesses the sensitivity of regression estimates to the deletion of the i^{th} data point.

Recommended references: Kennedy (1992, 284, 285); Kmenta (1986, 424–426); Krasker, Kuh, and Welsch (1983); Maddala (1988, 417–418); Welsch (1980).

Figure 51—Sample program for DFFITS calculation, in GAUSS-386

```

/*****
* PROGRAM:   DFFITS.G       SOFTWARE: GAUSS-386 V3.0 *
*           FILENAME      DESCRIPTION           *
* INPUTS:   DATA.DAT     GAUSS-386 DATA SET  *
* PURPOSE:  CALCULATE DFFITS STATISTICS FOR ALL *
*           OBSERVATIONS AND REPORT THOSE THAT ARE *
*           LARGE. THE RUNNING TIME FOR THIS *
*           PROGRAM IS ABOUT 3 HOURS WITH THE *
*           FULL DATA SET. USE A SUBSET OF THE *
*           DATA FOR FASTER TURNAROUND TIME. *
*****/

* NOTE: RUN TIME IS ABOUT 110 MINUTES ON 486DX2-66; *

FORMAT /M2 /RD 12,4;
OUTPUT FILE = DFFITS.OUT RESET;

NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);
DATA = READR(D,NCASE);
F = CLOSE(D);
Y = DATA[.,IY1];

X = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                        ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
              ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

@----- OLS ESTIMATION -----@

K = COLS(X);

B = INV(X'X)*X'Y; @ BETAS @
YHAT = X*B; @ FITTED VALUES @
E = Y - YHAT; @ RESIDUALS @
RSS = E'E; @ RESIDUAL SUM OF SQUARES @
SER = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T = B ./ SE; @ T-STATISTICS FOR BETAS @
PT = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES @
PRN = B ~ SE ~ T ~ PT; @ FOR PRINTING @

" ";
" ";
" ";
" OLS RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS = ";; NCASE;
" STANDARD ERROR OF REGRESSION = ";; SER;
" RESIDUAL SUM OF SQUARES = ";; RSS;
" R-SQUARED = ";; RSQ;
" ";

```

(continued)

Figure 51—Continued

```

" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT ";;   PRN[1,.];

I       = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I       = I + 1;
ENDO;
" ";
"\f";
@-----@
@-----@   CONSTRUCT VECTOR OF DFFITS STATISTICS.   @-----@
@-----@   THE MATRIX DFFITS CONTAINS "OBSERVATION   @-----@
@-----@   NUMBER" IN THE FIRST COLUMN AND THE       @-----@
@-----@   CORRESPONDING DFFITS STATISTIC IN THE     @-----@
@-----@   SECOND COLUMN.                             @-----@

N       = NCASE;
I       = 1;
YO      = Y;
XO      = X;
COUNT = SEQA(1,1,NCASE);

CLEAR   DATA COV SE T PT PRN Y X;

XXI     = INV(XO'XO);
H       = ZEROS(N,2);

OUTPUT FILE = DFFITS.OUT OFF;

DO WHILE I <= N;                                @ LOOP OVER WHOLE SAMPLE @

  YI     = YO[I,.];
  XI     = XO[I,.];
  Y      = SELIF(YO,COUNT[.,1] .NE I);
  X      = SELIF(XO,COUNT[.,1] .NE I);
  G      = INV(X'X)*X'Y;

  YHATI  = XO[I,]*G;

  E      = Y - X*G;
  SERI   = SQRT(INV(NCASE - K - 1)*E'E);

  HAT    = XO[I,]*XXI*XO[I,]';

  DFFITS = (YHAT[I,] - YHATI) / (SERI*SQRT(HAT));

  H[I,1] = I;
  H[I,2] = DFFITS;

  "LOOP I = ";; I;

  I      = I + 1;

ENDO;                                           @ END OF LOOP @

```

(continued)

Figure 51— Continued

```

OUTPUT FILE = DFFITS.OUT ON;

@----- SELECT |DFFIT| VALUES GREATER THAN 2*SQRT(K/N) -----@

CUT      = 2*SQRT(K/N);
H        = ABS(H);
H        = SELIF(H,H[,2] .> CUT);
ND       = ROWS(H);
H        = REV(SORTC(H,2));

" ";
" ";
" ";
" TEN LARGEST ABS(DFFITS) GREATER THAN 2*SQRT(K/N)";
" ";
" ";
FORMAT /M1 /RD 8,0;
ND;; "OBSERVATIONS HAVE ABSOLUTE VALUES > 2*SQRT(K/N)";
FORMAT /M1 /RD 12,4;
" ";
" ";
" OBSERVATION DFFITS STATISTIC";
" ";

I        = 1;

DO WHILE I <= 10;
  FORMAT /M1 /RD 12,0; H[I,1];; FORMAT /M1 /RD 12,4; H[I,2];

  I      = I + 1;
ENDO;

"\f";

OUTPUT FILE = DFFITS.OUT OFF;
SYSTEM;

```

Figure 52—Sample program for DFFITS calculation, in SAS PC

```

*****
* PROGRAM:   DFFITS.SAS      SOFTWARE: SAS PC 6.04      *
*           FILENAME      DESCRIPTION                *
* INPUTS:   DATA.SSD      SAS PC DATA SET          *
* PURPOSE:  CALCULATE DFFITS STATISTICS FOR ALL     *
*           OBSERVATIONS AND REPORT THE 10 LARGEST. *
*****;

LIBNAME CDRV 'C:\DATA\';

PROC REG DATA = CDRV.DATA;
  MODEL Y1 = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
          RD1 RD2 RD3;
  OUTPUT OUT = HAT1 DFFITS = ODFFIT;
RUN;

* LIST THE 10 LARGEST VALUES OF ODFFIT.;

DATA HAT2;
  SET HAT1;
  AODFFIT=1/ABS(ODFFIT);
RUN;

PROC RANK DATA=HAT2 OUT=RHAT;
  VAR AODFFIT;
  RANKS RAODFFIT;
RUN;

PROC SORT DATA=RHAT;
  BY RAODFFIT;
RUN;

PROC PRINT DATA = RHAT (OBS = 10);
  VAR RAODFFIT ODFFIT Y1 X1 X2 X8 X9 X10 RD1 RD2 RD3;
RUN;

* NOTE THAT SAS PC EMPLOYS A DIFFERENT CALCULATION FOR THE PROCEDURE IT
* LABELS AS DFFITS THAN DOES SPSS/PC+.
* SAS PC DFFITS = SPSS/PC+ SDFIT (STANDARDIZED VERSION OF WHAT SPSS/PC+ LABELS AS
* DFFITS).;

```

Figure 53—Sample program for DFFITS calculation, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'DFFITS.LIS'.
SET LOG = 'DFFITS.LOG'.
*****
* PROGRAM:   DFFITS.SPS      SOFTWARE: SPSS/PC+ 4.01   *
*           FILENAME      DESCRIPTION                *
* INPUTS:   DATA.SYS      SPSS/PC+ DATA SET        *
* PURPOSE:  CALCULATE DFFITS STATISTICS FOR ALL     *
*           OBSERVATIONS AND REPORT THE 10 LARGEST.  *
*****.

GET FILE = 'DATA.SYS'.

REG VAR=Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3
  /DEP=Y1
  /METHOD=ENTER
  /SAVE SDFIT(ODFFIT).
* LIST THE 10 LARGEST VALUES OF ODDFIT.
COMPUTE ABSDFIT=ABS(ODFFIT).
RANK ABSDFIT.
SORT RABSDFIT (D).
N 10.
LIST RABSDFIT ODDFIT Y1 X1 X2 X8 X9.

* NOTE THAT SPSS/PC+ EMPLOYS A DIFFERENT CALCULATION FOR THE PROCEDURE IT
* LABELS AS DFFITS THAN DOES SAS.
* SPSS/PC+ SDFIT (STANDARDIZED VERSION OF WHAT SPSS/PC+ LABELS AS DFFITS) =
* SAS PC DFFITS.
FINISH.

```

Bounded Influence Estimation

As noted by Maddala (1988), the conventional approach to outliers based on least squares residuals is to delete observations with large residuals and reestimate the equation. Given that the OLS residuals do not provide any readily useful information as to the importance of a given observation for overall results, a number of alternative procedures for dealing with outliers have been developed. The Bounded Influence Estimation (BIE) of Welsch (1980) is designed to evaluate the influence of individual observations, and to weight influential observations by a weight that is inversely related to the measure of influence. Thus, highly influential observations are not deleted (reducing the degrees of freedom and throwing out potentially useful information), but their influence is reduced. The measure of influence used is the DFFITS measure discussed in the previous section.

The simple one-step BIE developed by Welsch is defined as the value of β that minimizes:

$$\sum (w_i [y_i - \beta x_i])^2,$$

where

$$w_i = 1 \quad \text{if } |DFFITS| \leq 2*[K/N]^{1/2}$$

and

$$w_i = \frac{2*(K/N)^{1/2}}{|DFFITS|} \quad \text{if } |DFFITS| > 2*[K/N]^{1/2}.$$

Thus, for noninfluential observations, $w_i = 1$. If $w_i = 1$ for all i , then this is the OLS estimator. Essentially, this technique places observations into two distinct regimes, on the basis of their DFFITS values, and then observations are weighted accordingly. However, the BIE technique should not be used as a substitute for a careful examination of the data-generating process. It may be the case that the influential observations are only exceptional because the model is inappropriate or because observations are inappropriately pooled.

The sample programs (Figures 54 through 56) are extensions of the DFFITS programs presented in the DFFITS section. The regression results do not change very much when the BIE technique is employed (for example, the OLS coefficient on X_{10} is 216.97, and the BIE estimator for X_{10} is 217.994).

Recommended references: Kennedy (1992, 282, 284–285); Maddala (1988, 418); Welsch (1980).

Figure 54—Sample program for estimating bounded influence, in GAUSS-386

```

/*****
* PROGRAM: BIE.G SOFTWARE: GAUSS-386 V3.0 *
* FILENAME DESCRIPTION *
* INPUTS: DATA.DAT GAUSS-386 DATA SET *
* PURPOSE: BOUNDED INFLUENCE ESTIMATION. *
* RUNNING TIME FOR THIS PROGRAM IS *
* APPROXIMATELY 3 HOURS WITH THE FULL *
* DATA SET. *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = BIE.OUT ON;

NAMES = GETNAME("DATA");
OPEN D = DATA VARINDXI;
NCASE = ROWSF(D);

DATA = READR(D,NCASE);
F = CLOSE(D);
Y = DATA[.,IY1];

X = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAMES = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

@----- OLS ESTIMATION -----@

K = COLS(X);

B = INV(X'X)*X'Y; @ BETAS @
YHAT = X*B; @ FITTED VALUES @
E = Y - YHAT; @ RESIDUALS @
RSS = E'E; @ RESIDUAL SUM OF SQUARES @
SER = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T = B ./ SE; @ T-STATISTICS FOR BETAS @
PT = CDFTC(ABS(T),(NCASE-K)); @ P-VALUES @
PRN = B ~ SE ~ T ~ PT; @ FOR PRINTING @

" ";
" ";
" ";
" OLS RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS = ;; NCASE;
" STANDARD ERROR OF REGRESSION = ;; SER;
" RESIDUAL SUM OF SQUARES = ;; RSS;
" R-SQUARED = ;; RSQ;
" ";
" ";
" VARIABLE COEFF STD ERROR T-RATIO P-VALUE";

```

(continued)

Figure 54—Continued

```

" ";
" INTERCEPT ";; PRN[1,.];

I      = 1;
DO WHILE I <= K - 1;
FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

I      = I + 1;
ENDO;
" ";
"\f";
@----- CONSTRUCT VECTOR OF DFFITS STATISTICS. -----@
@----- THE MATRIX DFFITS CONTAINS "OBSERVATION -----@
@----- NUMBER" IN THE FIRST COLUMN AND THE -----@
@----- CORRESPONDING DFFITS STATISTIC IN THE -----@
@----- SECOND COLUMN. -----@

N      = NCASE;
I      = 1;
YO     = Y;
XO     = X;
COUNT = SEQA(1,1,NCASE);

CLEAR DATA COV SE T PT PRN Y X;

XXI    = INV(XO'XO);
H      = ZEROS(N,2);

OUTPUT FILE = BIE.OUT OFF;

DO WHILE I <= N;                                @ LOOP OVER WHOLE SAMPLE @

YI     = YO[I,.];
XI     = XO[I,.];

Y      = SELIF(YO,COUNT[.,1] .NE I);
X      = SELIF(XO,COUNT[.,1] .NE I);

G      = INV(X'X)*X'Y;
YHATI  = XO[I,]*G;

E      = Y - X*G;
SERI   = SQRT(INV(NCASE - K - 1)*E'E);

HAT    = XO[I,]*XXI*XO[I,]';

DFFITS = (YHAT[I,] - YHATI) / (SERI*SQRT(HAT));

H[I,1] = I;
H[I,2] = DFFITS;

"LOOP I = ";; I;; H[I,.];

I      = I + 1;

ENDO;                                           @ END OF LOOP @
OUTPUT FILE = BIE.OUT ON;

```

(continued)

Figure 54—Continued

```

@----- SELECT |DFFIT| VALUES GREATER THAN 2*SQRT(K/N); CALC WEIGHTS -----@
CUT      = 2*SQRT(K/N);
H        = ABS(H);
A        = H;
A        = SELIF(A,A[.,2] .> CUT);
ND       = ROWS(A);
A        = REV(SORTC(A,2));

" ";
" ";
" ";
" TEN LARGEST ABS(DFFITS) GREATER THAN 2*SQRT(K/N)";
" ";
FORMAT /M1 /RD 8,0;
ND;; "OBSERVATIONS EXCEED 2*SQRT(K/N)";
" ";
FORMAT /M1 /RD 12,4;
" OBSERVATION DFFITS STATISTIC";
" ";

I        = 1;

DO WHILE I <= 10;
FORMAT /M1 /RD 12,0; A[I,1];; FORMAT /M1 /RD 12,4; A[I,2];

I        = I + 1;

ENDO;
"\f";

@----- CREATE WEIGHTS ACCORDING TO SIZE OF DFFITS -----@

W        = H[.,2];
I        = 1;

DO WHILE I <= NCASE;
IF W[I,1] <= CUT;
W[I,1]   = 1.00;
ELSE;
W[I,1]   = CUT/W[I,1];
ENDIF;

I        = I + 1;

ENDO;

@----- WEIGHT THE VARIABLES AND ESTIMATE THE REGRESSION -----@

W        = SQRT(W);
Y        = W .* YO;
X        = W .* XO;

```

(continued)

Figure 54—Continued

```

B      = INV(X'X)*X'Y;           @ BETAS           @
E      = Y - X*B;              @ RESIDUALS      @
RSS    = E'E;                  @ RESIDUAL SUM OF SQUARES @
SER    = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED      @
COV    = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV));      @ STD ERRS OF BETAS @
T      = B ./ SE;              @ T-STATISTICS FOR BETAS @
PT     = CDFTC(ABS(T), (NCASE-K)); @ P-VALUES       @

PRN    = B ~ SE ~ T ~ PT;      @ FOR PRINTING   @

" ";
" ";
" ";
" BOUNDED INFLUENCE ESTIMATION RESULTS ";
" ";
" ";
" NUMBER OF OBSERVATIONS      = ";;  NCASE;
" STANDARD ERROR OF REGRESSION = ";;  SER;
" RESIDUAL SUM OF SQUARES    = ";;  RSS;
" R-SQUARED                   = ";;  RSQ;
" ";
" ";
" VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
" INTERCEPT ";;  PRN[1,.];

I      = 1;
DO WHILE I <= K -1;
FORMAT /M1 /RD 12,8; $NAMES[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

I      = I + 1;
ENDO;
" ";

"\f";

OUTPUT FILE = BIE.OUT OFF;
SYSTEM;

```

Figure 55—Sample program for estimating bounded influence, in SAS PC

```

*****
* PROGRAM:   BIE.SAS           SOFTWARE: SAS PC 6.04   *
*           FILENAME         DESCRIPTION             *
* INPUTS:   DATA.SSD        SAS PC DATA SET       *
* PURPOSE:  BOUNDED INFLUENCE ESTIMATION.          *
*****;

LIBNAME CDRV 'C:\DATA';

*STEP 1: RUN REGRESSION USING COMPLETE DATA SET (MATRIXL). SAVE DFFITS STATISTIC IN
        VARIABLE DFT, AND WRITE TO FILE, INFL.;

PROC REG DATA = CDRV.DATA;
  MODEL Y1 = X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
          RD1 RD2 RD3;
  OUTPUT OUT = HAT DFFITS = ODFFIT;
RUN;

* STEP 2: MERGE DATA SETS MATRIX1 AND INFL. THE VARIABLE, DFT, IS REPEATED FOR EACH
        OBSERVATION IN MATRIX1. EVALUATE CONDITION GIVEN BY EQN ** ABOVE, AND
        CREATE NEW (WEIGHT) VARIABLE, W.;

DATA DFDATA;
  MERGE CDRV.DATA HAT;
  CUTOFF = 2 * ((19 / 1624) ** .5);
  IF ABS(ODFFIT) LE CUTOFF THEN
    W = 1;
  ELSE
    W = CUTOFF/ABS(ODFFIT);
RUN;

* STEP 3: RUN NEW (WEIGHTED LEAST SQUARES) REGRESSION.;

PROC REG DATA = DFDATA;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
  WEIGHT W;
RUN;

* FOR THIS EXAMPLE, NOTICE THAT THE COEFFICIENTS CHANGE SLIGHTLY
* BECAUSE OF THIS PROCEDURE. FOR EXAMPLE, IN THE OLS MODEL THE COEFFICIENT OF
* X10 IS 216.97, BUT IN THE BOUNDED INFLUENCE ESTIMATES THE COEFFICIENT
* OF X10 IS 217.994.;

```

Figure 56—Sample program for estimating bounded influence, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'BIE.LIS'.
SET LOG = 'BIE.LOG'.
*****
* PROGRAM:   BIE.SPS      SOFTWARE: SPSS/PC+ 4.01   *
*           FILENAME     DESCRIPTION              *
* INPUTS:   DATA.SYS   TEST DATA SET           *
* PURPOSE:  BOUNDED INFLUENCE ESTIMATION.        *
*****.

GET FILE = 'DATA.SYS'.

* STEP 1: RUN REGRESSION USING COMPLETE DATA SET (MATRIX1).
*         SAVE DFFIT IN VARIABLE DFT.

REG VAR=Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3
  /DEP=Y1
  /METHOD=ENTER X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
    RD1 RD2 RD3
  /SAVE SDFIT(ODFFIT).

* STEP 2: UTILIZE A SIMPLE IF STATEMENT TO CREATE NEW VARIABLE, W, TO BE
*         USED IN A WEIGHTED LEAST SQUARES.

COMPUTE CUTOFF = 2 * ((19 / 1624) ** .5).
COMPUTE W = CUTOFF/ABS(ODFFIT).
IF (ABS(ODFFIT) LE CUTOFF) W = 1.

* STEP 3: RUN A SECONDARY REGRESSION, UTILIZING THE NEWLY CONSTRUCTED
*         VARIABLE, W, AS A WEIGHT.

REGRESSION
  /VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15 D1 D2 D3 D5 D6 D7 D8
    RD1 RD2 RD3
  /REGWGT = W
  /DEPENDENT = Y1
  /METHOD=ENTER.

* FOR THIS EXAMPLE, NOTICE THAT THE COEFFICIENTS CHANGE SLIGHTLY
* BECAUSE OF THIS PROCEDURE. FOR EXAMPLE, IN THE OLS MODEL THE COEFFICIENT OF
* X10 IS 216.97, BUT IN THE BOUNDED INFLUENCE ESTIMATES THE COEFFICIENT
* OF X10 IS 217.994.

FINISH.

```

MISSING DATA An obvious problem for estimation occurs when a data set is incomplete, such as when a survey respondent only partially completes a questionnaire. The easiest solution is to simply drop the observations that are incomplete. If the quality of information for a particular observation is very poor, this may be the only reasonable solution. However, given the often high cost of gathering data and the fact that discarding data reduces the precision of estimators, this solution is often resisted. An alternative, if relatively few pieces of information are missing, is to try to fill in the blanks. As alternatives to dropping observations, the following two procedures are easily implemented:

- *Zero-Order Regressions.* If both regressors and dependent variables have missing values, these regressions—in which the missing data are replaced by sample means—may be used.
- *First-Order Regressions.* If only the regressors have missing values, these regressions—in which the missing values are first estimated by considering the relationships among all of the regressors—may be used.

Simple Zero-Order Regressions (Mean Substitution)

Let X denote an $N \times K$ matrix of regressors. Assume that a single column of X , X_k , has a number of missing observations.

Let Y denote an $N \times 1$ dependent variable. Assume that Y also has a number of missing observations; they need not be the same observations as those missing from X_k .

The strategy is to simply replace the missing observations of X_k and Y by their mean values for the complete observations. Greene (1990, 285–189) summarizes known results for this strategy and concludes that using mean Y values of complete observations to impute values for missing Y s is a poor strategy that is unlikely to yield any gain to the researcher. Greene also points out (footnote 16, page 287) that replacing missing X -values by their means does not yield unbiased results, as suggested by Kmenta (1986). Therefore the zero-order regression strategy is not pursued any further.

Recommended references: Greene (1990, 285–289); Kmenta (1986, 379–387).

First-Order Regressions (Incidental Equations)

In contrast to zero-order regressions, the incidental equations method may enable the researcher to exploit information contained in correlations among X s to impute some missing values of a regressor. In the sample programs, every twentieth observation on X_k (= X10) (beginning with number 20) is coded as missing.

Step 1

Using only those observations with complete data, regress X_k on the variables in X for which no observations are missing (all variables except X_k). Let this matrix be Z . Retain the estimated coefficients from regressing X_k on Z .

- Step 2 Compute fitted values for the missing values of X_k , using the estimated regression coefficients and the relevant observations on Z . So, if the seventh observation in X_k is missing, use the seventh observation on Z together with the coefficients from Step 1 to fit $X_{k,7}$.
- Step 3 Substitute these fitted values for the missing observations in X_k . Now proceed with your intended regression.

Figures 57 through 59 are sample programs for calculating first-order regressions when data are missing.

NOTES:

1. Maddala (1977) suggests that if the correlations among the regressors in an equation are moderately high, this first-order method is preferable to the zero-order method.
2. Kmenta (1986) argues that the first-order method implicitly defines a system of simultaneous equations (because X_k is a dependent as well as an independent variable) and, therefore, this method may be theoretically unsound. In addition, Kmenta warns against the introduction of measurement error to X_k through this type of interpolation.
3. All three programs produce estimates that are similar to estimates for no missing values. For instance, the estimated coefficient on X_{10} with missing values (5 percent of observations on X_{10} are coded as missing) is 217.19 as opposed to 216.97 with no missing values on X_{10} .

Recommended references: Afifi and Elashoff (1966, 1967, 1969); Greene (1990, 285–289); Haitovsky (1968, 67–82); Kmenta (1986, 379–388); Maddala (1977, 201–207).

Figure 57—Sample program for calculating first-order regressions when data are missing, in GAUSS-386

```

/*****
* PROGRAM:  MISSINGF.G   SOFTWARE: GAUSS-386 V3.0  *
*          FILENAME     DESCRIPTION              *
* INPUTS:   DATA.DAT   GAUSS-386 DATA SET      *
* PURPOSE:  CALCULATES FIRST-ORDER REGRESSIONS   *
*           (FITTED VALUE SUBSTITUTION) WHEN SOME *
*           VALUES OF X10 ARE MISSING.          *
*****/

FORMAT /M2 /RD 12,4;
OUTPUT FILE = MISSINGF.OUT RESET;

NAMES   = GETNAME("DATA");
OPEN D  = DATA VARINDXI;
NCASE   = ROWSF(D);
DATA    = READR(D,NCASE);
F       = CLOSE(D);
Y       = DATA[.,IY1];

X       = ONES(NCASE,1) ~ DATA[.,IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                          ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3];

NAME    = NAMES[IX1 IX2 IX8 IX9 IX10 IX13 IX14 IX15
                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

@----- SET EVERY TWENTIETH OBSERVATION ON X10 EQUAL TO -999 -----@

NMI     = FLOOR(NCASE/20);
NM      = SEQA(20,20,NMI);
X[NM,6] = -999*ONES(NMI,1);

" ";
"   THE VARIABLE X10 IS MISSING VALUES AT OBSERVATIONS:";
" ";
FORMAT /M1 /RD 8,0;
NM';
FORMAT /M1 /RD 12,4;
"\f";

@----- CALCULATE FITTED VALUES FOR THE MISSING OBSERVATIONS -----@
@----- VIA AN INCIDENTAL REGRESSION AND REPLACE THE -----@
@----- MISSING OBSERVATIONS WITH THE FITTED ONES. -----@

XNM     = SELIF(X[.,6],X[.,6] .NE -999);
YNM     = SELIF(Y[.,1],X[.,6] .NE -999);
XR      = SELIF(X[.,.],X[.,6] .NE -999);

Z       = X[.,1 2 3 4 5 7 8 9 10 11 12 13 14 15 16 17 18 19];
ZNM     = SELIF(Z,X[.,6] .NE -999);
NAMESI  = NAMES[IX1 IX2 IX8 IX9 IX13 IX14 IX15
                ID1 ID2 ID3 ID5 ID6 ID7 ID8 IRD1 IRD2 IRD3,.];

K       = COLS(ZNM);
NI      = ROWS(ZNM);

G       = INV(ZNM'ZNM)*ZNM'XNM;           @ GAMMAS: INCIDENTAL EQ @

```

(continued)

Figure 57—Continued

```

XFIT   = Z*G;                               @ FITTED VALUES: ALL OBS @
E      = XNM - ZNM*G;                       @ RESIDUALS @
RSS    = E'E;                               @ RESIDUAL SUM OF SQUARES @
SER    = SQRT(INV(NI-K)*RSS);                @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSS/((NI-1)*(STDC(XNM))^2);     @ R-SQUARED @
COV    = INV(NI-K)*RSS*INV(ZNM'ZNM);        @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV));                   @ STD ERRS OF GAMMAS @
T      = G ./ SE;                           @ T-STATISTICS FOR GAMMAS @
PT     = 2*CDFTC(ABS(T), (NI-K));           @ P-VALUES @
PRN    = G ~ SE ~ T ~ PT;                   @ FOR PRINTING @

" ";
" ";
" ";
" INCIDENTAL EQUATION REGRESSION RESULTS ";
" ";
" ";
" NON-MISSING OBSERVATIONS = ;; NI;
" STANDARD ERROR OF REGRESSION = ;; SER;
" RESIDUAL SUM OF SQUARES = ;; RSS;
" R-SQUARED = ;; RSQ;
" ";
" ";
" VARIABLE COEFF STD ERROR T-RATIO P-VALUE";
" ";
" INTERCEPT ;; PRN[1,.];

I      = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAMESI[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";
"\f";

@----- SELECT FITTED VALUES THAT CORRESPOND TO MISSING OBS -----@

I      = 1;
DO WHILE I <= NMI;
  X[NM[I,.],6] = XFIT[NM[I,.],1];
  I      = I + 1;
ENDO;

" ";
" ";
" THE MISSING VALUES OF X10 HAVE BEEN REPLACED BY FITTED VALUES";
" ";
" ";
" OBSERVATION FITTED VALUE";
" ";
" ";
I      = 1;
DO WHILE I <= NMI;
  FORMAT /M1 /RD 8,0; NM[I,.];; FORMAT /M1 /RD 12,4; XFIT[I,.];
  I      = I + 1;
ENDO;

```

(continued)

Figure 57—Continued

```

"\f";

@-----          OLS ESTIMATION OF "FIRST-ORDER" MODEL          -----@

K      = COLS(X);
B      = INV(X'X)*X'Y;          @ BETAS          @
E      = Y - X*B;          @ RESIDUALS          @
RSS    = E'E;          @ RESIDUAL SUM OF SQUARES @
SER    = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSS/((NCASE-1)*(STDC(Y))^2); @ R-SQUARED @
COV    = INV(NCASE-K)*RSS*INV(X'X); @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T      = B ./ SE;          @ T-STATISTICS FOR BETAS @
PT     = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES @
PRN    = B ~ SE ~ T ~ PT; @ FOR PRINTING @

" ";
" ";
" ";
"  FIRST-ORDER REGRESSION RESULTS ";
" ";
" ";
"  NUMBER OF OBSERVATIONS      = ;;  NCASE;
"  STANDARD ERROR OF REGRESSION = ;;  SER;
"  RESIDUAL SUM OF SQUARES     = ;;  RSS;
"  R-SQUARED                   = ;;  RSQ;
" ";
" ";
"  VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"  INTERCEPT ;;  PRN[1,.];

I      = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAME[I,.];; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";
"\f";

@-----          OLS ESTIMATION OF INTENDED MODEL          -----@
@-----          WHEN OBSERVATIONS WITH MISSING VALUES          -----@
@-----          HAVE BEEN DELETED          -----@

K      = COLS(XR);
NCASE  = ROWS(XR);
B      = INV(XR'XR)*XR'YNM;          @ BETAS          @
E      = YNM - XR*B;          @ RESIDUALS          @
RSS    = E'E;          @ RESIDUAL SUM OF SQUARES @
SER    = SQRT(INV(NCASE-K)*RSS); @ STD ERROR OF REGRESSION @
RSQ    = 1 - RSS/((NCASE-1)*(STDC(YNM))^2); @ R-SQUARED @
COV    = INV(NCASE-K)*RSS*INV(XR'XR); @ OLS COVARIANCE MATRIX @
SE     = SQRT(DIAG(COV)); @ STD ERRS OF BETAS @
T      = B ./ SE;          @ T-STATISTICS FOR BETAS @
PT     = 2*CDFTC(ABS(T),(NCASE-K)); @ P-VALUES @
PRN    = B ~ SE ~ T ~ PT; @ FOR PRINTING @

```

(continued)

Figure 57—Continued

```

" ";
" ";
" ";
"   OLS REGRESSION RESULTS: OBSERVATIONS WITH MISSING VALUES DELETED";
" ";
" ";
" NUMBER OF OBSERVATIONS      =   ;;   NCASE;
" STANDARD ERROR OF REGRESSION =   ;;   SER;
" RESIDUAL SUM OF SQUARES    =   ;;   RSS;
" R-SQUARED                   =   ;;   RSQ;
" ";
" ";
"   VARIABLE      COEFF      STD ERROR      T-RATIO      P-VALUE";
" ";
"   INTERCEPT   ;;      PRN[1,.];

I      = 1;
DO WHILE I <= K -1;
  FORMAT /M1 /RD 12,8; $NAME[I,.]; FORMAT /M1 /RD 12,4; PRN[I+1,.];

  I      = I + 1;
ENDO;
" ";
"\f";

OUTPUT FILE = MISSINGF.OUT OFF;
SYSTEM;

```

Figure 58—Sample program for calculating first-order regressions when data are missing, in SAS PC

```

*****
* PROGRAM:  MISSINGF.SAS  SOFTWARE: SAS PC 6.04      *
*          FILENAME      DESCRIPTION                *
* INPUTS:                                     *
* PURPOSE:  CALCULATES FIRST-ORDER REGRESSIONS     *
*           (FITTED VALUE SUBSTITUTION) WHEN SOME  *
*           VALUES OF X10 ARE MISSING.            *
*****;
LIBNAME CDRV 'C:\DATA';
* SINCE THE DATA SET HAS NO MISSING VALUES, CREATE A DATA SET
* WITH EVERY 20TH VALUE OF X10 MISSING.;
DATA MISSING;
  SET CDRV.DATA;
  IF MOD(_N_,20) = 0 THEN X10 = .;
RUN;

* STEP 1: IF X10 IS THE EXPLANATORY VARIABLE WITH THE MISSING VALUES TO BE
  REPLACED, RUN REGRESSION WITH X10 AS DEPENDENT VARIABLE.;
PROC REG DATA=MISSING OUTEST=COEFF;
  MODEL X10=X1 X2 X8 X9 X13 X14 X15
        D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

DATA CDATA;
  SET MISSING;
  CONSTANT=1;
RUN;

DATA CCOEF;
  SET COEFF(RENAME=(X1=CX1 X2=CX2 X8=CX8 X9=CX9 X10=DX10
                  X13=CX13 X14=CX14 X15=CX15
                  D1=CD1 D2=CD2 D3=CD3 D5=CD5 D6=CD6 D7=CD7 D8=CD8
                  RD1=CRD1 RD2=CRD2 RD3=CRD3));
  CONSTANT=1;
RUN;

DATA FIRST;
  MERGE CDATA CCOEF;
  BY CONSTANT;
  X10FIRST = CX1*X1 + CX2 *X2 + CX8 *X8 + CX9 *X9 +
            CX13*X13 + CX14*X14 + CX15*X15 + CD1 *D1 +
            CD2 *D2 + CD3 *D3 + CD5 *D5 + CD6 *D6 +
            CD7 *D7 + CD8 *D8 + CRD1*RD1 + CRD2*RD2 +
            CRD3*RD3 + INTERCEP;
  IF X10=. THEN X10 = X10FIRST;
RUN;

* STEP 2: RUN REGRESSION (WITH Y1 AS DEPENDENT VARIABLE) USING COMPLETE SET
  OF OBSERVATIONS ON X10, WHERE MISSING VALUES IN X10 HAVE BEEN
  SUBSTITUTED BY PREDICTED VALUES OF X10 (X10FIRST).;
PROC REG DATA=FIRST;
  MODEL Y1=X1 X2 X8 X9 X10 X13 X14 X15
        D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3;
RUN;

* FOR THIS EXAMPLE NOTE THAT THIS PROCEDURE FOR FILLING IN MISSING OBSERVATIONS
* ON X10 GIVES AN OLS ESTIMATE OF 217.19 AS COMPARED TO AN OLS ESTIMATE OF
* 216.97 WITH NO MISSING DATA.;

```

Figure 59—Sample program for calculating first-order regressions when data are missing, in SPSS/PC+

```

SET MORE OFF.
SET LIS = 'MISSINGF.LIS'.
SET LOG = 'MISSINGF.LOG'.
*****
* PROGRAM:  MISSINGF.SPS  SOFTWARE: SPSS/PC+ 4.01  *
*          FILENAME      DESCRIPTION          *
* INPUTS:  DATA.SYS     TEST DATA SET      *
* PURPOSE: CALCULATES FIRST-ORDER REGRESSIONS *
*          (FITTED VALUE SUBSTITUTION) WHEN SOME *
*          VALUES OF X10 ARE MISSING.        *
*****

* SINCE THE DATA SET HAS NO MISSING VALUES, CREATE A DATA SET
* WITH EVERY 20TH VALUE OF X10 MISSING.
GET FILE = 'DATA.SYS'.
  IF (TRUNC($CASENUM/20)*20 = $CASENUM) X10 = -999.
  MISSING VALUE X10 (-999).
SAVE FILE='MISSING.SYS'.

* STEP 1: IF X1 IS THE EXPLANATORY VARIABLE WITH THE MISSING VALUES TO BE
* REPLACED, RUN REGRESSION WITH X1 AS DEPENDENT VARIABLE.
REGRESSION
  /VARIABLES X1 X2 X8 X9 X10 X13 X14 X15
            D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3
  /DEPENDENT = X10
  /METHOD = ENTER.

*****
** VIEW THE OUTPUT FROM THIS REGRESSION AND USE THE BETA COEFFICIENTS TO **
** COMPUTE A PREDICTED VALUE FOR X1 **
*****

GET FILE = 'MISSING.SYS'.

COMPUTE X10FIRST= X1 * .098245 + X2 * -.266151 + X8 * .0008157970 +
                 X9 * -.031469 + X13 * .0001182776 + X14 * .048470 +
                 X15 * .040088 + D1 * -.964098 + D2 * -1.065891 +
                 D3 * -.098893 + D5 * -.969640 + D6 * -.774009 +
                 D7 * -.234392 + D8 * -.086512 + RD1 * .040301 +
                 RD2 * -.197204 + RD3 * .030409 + 4.610311.

IF (MISSING(X10)) X10 = X10FIRST.

* STEP 2: RUN REGRESSION (WITH Y1 AS DEPENDENT VARIABLE) USING COMPLETE SET
* OF OBSERVATIONS ON X1, WHERE MISSING VALUES IN X1 HAVE BEEN
* SUBSTITUTED BY PREDICTED VALUES OF X1 (X1FIRST).
REGRESSION
  /VARIABLES = Y1 X1 X2 X8 X9 X10 X13 X14 X15
            D1 D2 D3 D5 D6 D7 D8 RD1 RD2 RD3
  /DEPENDENT = Y1
  /METHOD = ENTER.

* FOR THIS EXAMPLE NOTE THAT THIS PROCEDURE FOR FILLING IN MISSING OBSERVATIONS
* ON X10 GIVES AN OLS ESTIMATE OF 217.19 AS COMPARED TO AN OLS ESTIMATE OF
* 216.97 WITH NO MISSING DATA.
FINISH.

```

APPENDIX 1: SPSS/PC+ ENVIRONMENT AND COMMANDS

ENVIRONMENT • Starting an SPSS/PC+ Interactive Session

To start an SPSS/PC+ interactive session, from the DOS prompt, type `SPSSPC`. This takes you into the Menu and Help System, which is one of three ways you can enter commands in SPSS/PC+. You may also enter commands directly from the SPSS/PC+ prompt "SPSS/PC:" or you may use the REVIEW text editor built into SPSS/PC+ or any other text editor to submit batches of commands. Other text editors that can be used are DOS EDLIN, NORTON EDITOR, or WordPerfect (saving as text).

• The Menu and Help System

The top panel shows a Menu window on the left side and a Help window on the right side. By using the arrow keys you may move up and down the menu and into lower-level menus indicated by "▶". The bottom panel is the scratch pad (filename = SCRATCH.PAD) where REVIEW works as a text editor.

• The Menu and Help System: Entering Commands.

You enter commands by selecting from the menu and pasting it onto the scratch pad. For a guide on MENU commands and the REVIEW function keys, press  and select "Review Help."

• The Menu and Help System: Clearing and Calling up the Menu

The menu may be cleared and called up at any time by pressing  . Once the menu is cleared, the window displays the listing file (see SPSS/PC+ Default Files, SPSS.LIS, SPSS.LOG, and SCRATCH.PAD, below).

• The Menu and Help System: Moving Between Windows

Once the menu is cleared, you may use  to move between windows.

- **The Menu and Help System: Editing Different Files**

You may also edit different files on either window by pressing **F3** and selecting "Edit new file."

- **The Menu and Help System: To Run a Command or Batch of Commands**

Select command(s) from the menu and paste them on the scratch pad or clear the menu and type in the command directly on the scratch pad. Position the cursor on the first command you wish to execute, press **F10**, and select "Run from cursor." Your command is saved automatically under SCRATCH.PAD.

- **The REVIEW Text Editor**

To use the REVIEW text editor from DOS, type in

```
SPSSPC/RE 'filename.ext'
```

where *filename.ext* is the file you wish to edit or create. To use it from the SPSS.PC+ prompt, type in

```
REVIEW 'filename.ext'
```

From within the editor, you may call up (and clear) the Menu and Help System at any time by pressing **ALT M**.

- **The REVIEW Text Editor: Running a Command or Batch of Commands**

You may write a batch of commands directly on to a file. Save the file, position the cursor on the first command you wish to run, press **F10**, and select "Run from cursor." The commands read or executed by SPSS/PC+ will be saved under SPSS.LOG. (See SPSS/PC+ Default Files: SPSS.LIS, SPSS.LOG, SCRATCH.PAD, below.)

- **Entering Commands Interactively**

You may enter a command directly by typing in the command at the SPSS/PC+ prompt and pressing **ENTER**. You may also submit an entire batch file from the SPSS/PC+ prompt by typing

```
INCLUDE 'filename.ext'
```

- **Customizing the Work Environment**

To change the starting environment, the SET commands in the automatic profile "SPSSPROF.INI" must be changed. To change the work environment at any time from the Menu and Help System to the SPSS/PC+ prompt, press **F10** and select "Exit to prompt." From the

Menu and Help System to the REVIEW editor, using the scratch pad, press  . From the scratch pad to the Menu and Help System, press . From the SPSS/PC+ prompt to the Menu and Help System, type REVIEW.

- **Ending Your Interactive Session**

From the SPSS/PC+ prompt, type BYE or FIN. From within the REVIEW editor, type BYE or FIN, press , and select "Run from cursor."

- **The SPSS/PC+ Default Files: SPSS.LIS, SPSS.LOG, SCRATCH.PAD**

SPSS.LIS contains your display output, SPSS.LOG contains a log of your commands, and SCRATCH.PAD contains the commands typed or pasted into it. The default files—SPSS.LIS, SPSS.LOG, SCRATCH.PAD—are reinitialized at the beginning of each new session. You need to rename these files to save their contents.

- **Submitting an SPSS Command File from DOS**

To submit an SPSS command file from DOS, type the following:

```
d:\subdir>SPSSPC filename.SPS
```

- **DOS Interface**

DOS commands can be run from within SPSS/PC+. Type DOS command and press . This executes the DOS command without directly entering the DOS shell. To get into DOS, type DOS. Use "EXIT" at the DOS prompt to get back into SPSS/PC+.

- **Interrupting a Sequence of Commands**

To interrupt a sequence of commands, press the  and  keys simultaneously or the  and  keys simultaneously.

SPSS/PC+ COMMANDS

The following section describes a small subset of SPSS/PC+ commands that are essential for understanding the programs in this volume. There are three main categories of SPSS/PC+ commands: data definition and manipulation, procedure, and operation.

- **Data Definition and Manipulation**

The commands for data definition and manipulation are as follows:

GET FILE 'filename.SYS'.	Retrieves SPSS system file <i>filename.SYS</i> into the active file.
SAVE FILE 'filename.SYS'.	Saves active file as SPSS system file <i>filename.SYS</i> .
DATA LIST FILE 'filename.ASC' /var1 var2 var3.	Reads ASCII file <i>filename.ASC</i> into the active file.
WRITE.	Writes active file into an ASCII file (default filename is SPSS.PRC).
COMPUTE var4 = var1/var2.	Calculates a new variable <i>var4</i> , which is the ratio of <i>var1</i> and <i>var2</i> .
RECODE var3 (1 = 2)/var1 (9=sysmis).	Changes all code in <i>var3</i> with a value of 1 to 2 and, in <i>var1</i> , from 9 to system-missing.
PROCESS IF (var3 = 2).	Temporarily selects cases where <i>var3</i> is equal to 2 for the subsequent procedure.
SELECT IF (var3 = 2).	Permanently selects cases where <i>var3</i> is equal to 2 for all subsequent procedures.

- **Procedure**

The commands for procedures are as follows:

RANK var2.	Creates a new variable called <i>Rvar2</i> , which assigns ranks to <i>var2</i> .
REGRESSION VARIABLES=var1 var2 var3 /DEPENDENT=var1 /METHOD=ENTER var2 var3.	Runs a regression with <i>var1</i> as the dependent variable and <i>var2</i> and <i>var3</i> as the independent variables.

LIST <i>var1</i> .	Lists the value of <i>var1</i> for all cases.
AGGREGATE FILE='temp.SYS' /BREAK <i>dummy</i> . /ncase=NU(<i>var3</i>).	Creates a new system file <i>temp.SYS</i> , which contains as many cases as there are values of <i>dummy</i> . Each case includes two variables, <i>dummy</i> and <i>ncase</i> (which is the unweighted number of cases in the break group).
JOIN MATCH /FILE = * /TABLE = ' <i>temp.SYS</i> ' /BY <i>dummy</i> .	Merges the <i>ncase</i> variable in <i>temp.SYS</i> created in the previous step with a corresponding value of <i>dummy</i> to the current active file.
● Operation	
The commands for operations are as follows:	
SET MORE OFF.	Causes the output to scroll continuously without pause to give the MORE prompt when the screen fills.
SET LIS ' <i>filename.LIS</i> '.	Sends output to <i>filename.LIS</i> instead of the default SPSS.LIS.
SET LOG ' <i>filename.LOG</i> '.	Sends all commands into <i>filename.LOG</i> instead of the default SPSS.LOG.
* <i>comment line</i> .	Allows the user to insert comments into the program.

APPENDIX 2: SAS PC ENVIRONMENT AND COMMANDS

ENVIRONMENT • Starting a SAS PC Interactive Session

To start a SAS PC interactive session, from the DOS prompt, type SAS.

• Ending an Interactive Session

To end an interactive session, type BYE or END from any ==> command line.

• Screen Panels or Windows

The three screen panels or windows are as follows:

PROGRAM EDITOR	From here, you can submit and edit commands interactively or in batches.
LOG	This is a record of all SAS PC commands issued in the session; you should save this at the end of the session.
OUTPUT	This is a record of all results generated in the session; you should save this at the end of the session.

• Moving Between Windows

To move between windows, use the function keys. Press

 to move to the LOG window,
 to move to the OUTPUT window, and
 to move back to the PROGRAM EDITOR window.

• Entering a Command Interactively

To enter a command interactively, move from the ==> command line to line 1 of the editor, type in your command, and then submit

your command by pressing . Groups of commands can also be submitted in this way.

- **Recalling a Group of Commands from the Memory Buffer**

To recall a group of commands from the memory buffer, press .

- **Returning to the Command Line**

To return to the command line, `===>`, press .

- **Clearing a Window**

To clear a window, type `CLEAR` at the window's `===>` prompt.

- **Expanding a Window**

To expand the window you are working in, type `ZOOM` at the `===>` command line.

- **The Program Editor**

You may edit commands by altering the 00000 lines on the left of the screen, as follows:

00d000	Deletes that line and renumbers the lines. (The <i>d</i> can be in any position.)
0dd000 to 0dd000	Deletes the blocked-off lines.
00ib00	Inserts a blank line before the current line.
00ia00	Inserts a blank line after the current line.
0cc000 to 0cc000	Blocks lines for copying.

A complete list of line commands is available on pages 339–340 of the *SAS Language Guide for Personal Computers* (SAS Institute, Inc. 1988).

NOTE: You can use SAS PC perfectly well without being very proficient with this editor by editing program command files with WordPerfect (saving as text), or NORTON EDITOR, and then submitting as batch files (see Editing Input (Command) and Output (Log and Results) Files, below).

- **Editing Input (Command) and Output (Log and Results) Files**

Files containing SAS PC commands can be brought into the PROGRAM EDITOR window with the INCLUDE command by typing the following at the ===> prompt:

```
INCLUDE 'filename.SAS'
```

The command file can be created with any word processor/editor. After bringing the file into the program editor with INCLUDE, it can be submitted by pressing .

- **Saving Output or Log Windows**

To save output in the output or log windows, type the following at the ===> prompt of the output or log windows:

```
FILE 'filename.out'
```

- **Submitting a SAS PC Command File from DOS**

Either of the following statements will execute the SAS PC commands in the file *filename.SAS*:

```
d:\subdir> SAS filename
```

or

```
d:\subdir> SAS filename.SAS
```

In addition, SAS PC will create a *filename.LOG* file and a *filename.LST* file for you.

- **DOS Interface**

DOS commands can be run from within SAS PC by typing x at the ===> prompt. For example, the following will execute the DOS command without directly entering the DOS shell:

```
X 'dos command'
```

Typing only x at the ===> prompt puts you into DOS. Typing EXIT at the DOS prompt will put you back in SAS PC.

- **Interrupting a Sequence of Commands**

To interrupt a sequence of commands, press   and you will be asked if you want any of the submitted commands to be terminated. There is a default NOREPLACE option that does not allow a system file overwrite if there is a syntax error in the program, or if you terminate with  . However, whenever the program runs and

you have the same name on the DATA and SET lines, your SAS PC data set will be automatically overwritten.

- **SAS PC COMMANDS**

SAS PC commands fall into three categories (see inside cover of *SAS Language Guide for Personal Computers* [SAS Institute, Inc. 1988]): statements used in DATA steps, statements used in PROC steps, and statements used anywhere. Nearly all of the SAS PC commands in the following sections must be issued from the 0000n lines in the PROGRAM EDITOR (or from a command file), but *not* the ==> line.

- **The DATA Step**

Most operations that alter or create a SAS PC data set (similar to an SPSS/PC+ system file) are carried out within a DATA step. Examples: merging, creating new variables, and selecting subsamples.

```
DATA;
  SAS statements
RUN;
```

- **The PROC Commands**

Most SAS PC data transformation procedures and all statistical procedures are carried out with PROC commands, for example, REGRESSION, FREQUENCIES, and MEANS. SAS PC procedures often include options such as BY, CLASS, WEIGHT, and others.

```
PROC procedure;
  SAS statements
RUN;
```

- **The RUN Statement**

A SAS PC command, or set of SAS PC commands, will be executed when followed by

```
RUN;
```

- **Labeling DOS Subdirectories: The LIBNAME Command**

The statement

```
LIBNAME label 'D:\subdir\';
```

tells SAS PC that, for the rest of the session, label points to the subdirectory D:\subdir\. For example,

```
LIBNAME elast 'D:\calinc\';
```

labels the CALINC subdirectory on the D drive as ELAST. Note that up to eight characters can be used to label the subdirectory.

- **SAS PC Data Set Name Conventions**

There are two types of SAS PC data sets: permanent and temporary. The SAS internal and DOS names of these data sets are as follows:

	Permanent	Temporary
SAS internal name	<i>label.filename</i>	<i>filename</i>
DOS internal name	<i>D:\subdir\filename.SSD</i>	<i>C:\SAS\SASWORK\filename.SSD</i>

For example, the set of commands

```
LIBNAME elast 'D:\calinc\';
DATA elast.hhcal;
    sascommands
RUN;
```

labels the CALINC subdirectory as ELAST, and constructs a permanent SAS PC data set ELAST.HHCAL, which, after the SAS PC session has ended, will be found in the CALINC subdirectory as HHCAL.SSD.

The set of commands

```
DATA hhcal;
    sascommands
RUN;
```

will construct a temporary SAS PC data set, HHCAL, which will temporarily reside in C:\SAS\SASWORK as HHCAL.SSD, but will be deleted automatically when the session ends.

- **Constructing a SAS PC Data Set from a Text (ASCII) File**

To construct a SAS PC data set from a text (ASCII) file, use the INPUT command:

```
DATA irrig1;
    INFILE 'c:\DATA\MANUAL.ASC';
    INPUT v1 v2 v3 v4 v5;
RUN;
```

The five variables in text file WATER.DAT (and named by the user as v1 to v5) will be read into a temporary SAS PC data set called IRRIG1.

- **Constructing a SAS PC Data Set from Another SAS PC Data Set**

To construct a SAS PC data set from another SAS PC data set, use the SET command:

```
LIBNAME water 'D:\ghana\';
DATA water.irrig3;
SET water.irrig2;
sascommands
RUN;
```

This example brings in the permanent SAS PC data set WATER.IRRIG2—located in subdirectory D:\GHANA, which is labeled as WATER (using LIBNAME)—to serve as the starting point for the creation of a new permanent SAS PC data file. The new permanent file is called WATER.IRRIG3 and is located in D:\GHANA as IRRIG3.SSD.

- **Writing to ASCII**

To write to ASCII, use the PUT and FILE commands:

```
DATA instrum;
SET elast.jh1508;
FILE 'instrum.dat';
PUT v1 v2 v3 v4 v5;
RUN;
```

This example will construct a temporary SAS PC data file, INSTRUM, for the purpose of writing variables v1 to v5 from the permanent SAS PC data set, ELAST.JH1508, to a text file, INSTRUM.DAT.

- **Customizing the Work Environment**

The OPTIONS command is a stand-alone command used to customize the whole work environment. This command can appear anywhere outside of a DATA step. The following example sets output page length to 66 lines.

```
OPTIONS PAGESIZE=66;
```

- **Inserting Comments in the Program**

To insert comments, use the following format:

```
* comment;
```

SAS PC comment lines can appear anywhere outside of a DATA step.

- **The AUTOEXEC.SAS File**

This text file resides in C:\SAS and is similar to the AUTOEXEC.BAT file in C:\ in that it allows you to execute some of the commands every time you enter SAS PC. For example, you may find it convenient for your AUTOEXEC.SAS file to include your most common LIBNAME commands; in this way, you do not have to enter them at each session.

APPENDIX 3: GAUSS-386 ENVIRONMENT AND COMMANDS

ENVIRONMENT • Batch Programs

To submit a batch program, type

```
GAUSS386 filename
```

at the DOS prompt.

New programs may be constructed, or old ones edited, by then typing the following at the ">>" GAUSS-386 prompt:

```
EDIT filename.ext
```

From edit mode, the program can be saved and run by pressing .

The program can be filed and not run simply by pressing .

After running, the program may be brought into edit mode again by pressing .

• GAUSS-386 MATRIX NOTATION

To create the three-by-three matrix

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

you could type the following:

```
X = {1 2 3, 4 5 6, 7 8 9};
```

The same statement without any commas creates a row vector of dimensions nine columns-by-one row:

```
Y = {1 2 3 4 5 6 7 8 9};
```

Similarly, the same statement with commas between each pair of numbers creates a column vector of one column-by-nine rows:

```
Z = {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

GAUSS-386 COMMANDS • Creating a matrix

The following multiplication will create a matrix that is nine columns by nine rows:

```
A = Y * Z;
```

The following command will horizontally concatenate two matrices side-by-side to create a matrix that is three rows by six columns:

```
XC = X~X;
```

The next command vertically concatenates two matrices, one on top of the other to create a matrix that is six rows by three columns:

```
XR = X|X;
```

The following command selects just the first column of a matrix to create a column vector.

```
X1 = X[., 1];
```

The period (.) tells GAUSS-386 to include all elements in this row or column. To transpose a matrix, use the following command:

```
XTRAN = X';
```

GAUSS-386 includes numerous functions to facilitate matrix manipulation.

<u>Command</u>	<u>Function</u>
ABS	Returns the absolute value of the argument
CDFCHIC	Computes the complement of CDF of the chi-squared distribution
CDFNC	Computes the complement (1-CDF) of the normal distribution (upper tail)
CDFTC	Computes the complement of CDF of the t-distribution
CLOSE	Closes an open data set (.DAT file)
COLS	Returns the number of columns in a matrix
DELIF	Deletes rows from a matrix using a logical expression
DIAG	Extracts the diagonal of a matrix
EIGRS	Computes eigenvalues of a real, general matrix
GETNAME	Returns the column vector of variables' names in a data set

INV	Inverts a matrix
MAXC	Returns the largest element in each column of a matrix
MEANC	Computes the sample mean of each column of a matrix
MINC	Returns the smallest element in each column of a matrix
ONES	Creates a matrix of ones
OPEN	Opens an existing data set
READR	Reads rows from an open data set
ROWS	Returns the number of rows in a matrix
ROWSE	Returns the number of rows in an open GAUSS-386 data set
SEQA	Creates a sequence of numbers
SELIF	Selects rows from a matrix, using a logical expression
SORTC	Quick-sorts rows of a matrix on the basis of a numeric key
SQRT	Computes the square root of each element
STDC	Computes the standard deviation of the columns of a matrix
SUMC	Computes the sum of each column of a matrix
ZEROS	Creates a matrix of zeros

Example: OLS Estimation

Given an $N \times 1$ vector y , an $N \times K$ matrix of regressors, X , the $K \times 1$ vector of unknown parameters, β , may be estimated by ordinary least squares, using standard matrix manipulation (that is, $(X'X)^{-1}(X'Y)$). The GAUSS-386 code to do this is

```
BETA = INV (X'X) * X'Y;
```

where syntax to the right of the equality sign is actual programming syntax, and that to the left consists of user-assigned names.

- **Creating comment statements**

Comment statements begin with a `/*` or `@` and end with a `*/` or `@`, respectively:

```
@This is a comment@  
/*This is another comment*/;
```

MODULES Although all GAUSS-386 procedures may be programmed by the user, optional modules are available that provide access to relatively complicated linear and nonlinear models, for example, the Quantal Regression module facilitates estimation of logit, probit, and tobit models. In addition, the OPTMUM module provides an extensive array of nonlinear optimization routines, allowing the user to choose between Steepest Descent; Newton Raphson; Berndt, Hall, Hall, and Hausman (BHHH); and other quasi-Newton methods.

BIBLIOGRAPHY

- Afifi, A. A., and R. M. Elashoff. 1966. Missing observations in multivariate statistics I. *Journal of the American Statistical Association* 61 (September): 595.
- _____. 1967. Missing observations in multivariate statistics II. Point estimation in simple linear regression. *Journal of the American Statistical Association* 62 (March): 10-29.
- _____. 1969. Missing observations in multivariate statistics III, IV. *Journal of the American Statistical Association* 64 (March): 337-358.
- Allen, A. T., and B. C. Kalt, eds. 1985. *SAS language guide for personal computer*, Version 6 ed. Cary, N.C., U.S.A.: SAS Institute, Inc.
- Aptech Systems, Inc. 1992. *GAUSS version 3.0*. Two volumes. Maple Valley, Wash., U.S.A.
- Behrman, J. R., and A. Deolalikar. 1987. Will developing country nutrition improve with income? A case study for rural South India. *Journal of Political Economy* 95 (3): 492-507.
- Belsley, D. A., E. Kuh, and R. E. Welsch. 1980. *Regression diagnostics: Identifying influential data and sources of collinearity*. New York: John Wiley.
- Berndt, E. R. 1991. *The practice of econometrics: Classic and contemporary*. Reading, Mass., U.S.A.: Addison-Wesley.
- Berndt, E., B. Hall, R. Hall, and J. Hausman. 1974. Estimation and inference in nonlinear structure models. *Annals of Economic and Social Measurement* 3/4: 653-665.
- Bouis, H., and L. Haddad. 1990. *Effects of agricultural commercialization on land tenure, household resource allocation, and nutrition in the Philippines*. Research Report 79. Washington, D.C.: International Food Policy Research Institute.
- _____. 1992. Are estimates of calorie-income elasticities too high? A recalibration of the plausible range. *Journal of Development Economics* 39 (2): 333-364.

- Bowman, K. O., and L. R. Shenton. 1975. Omnibus test contours for departures from normality based on $\sqrt{b_1}$ and b_2 . *Biometrika* 62: 243–250.
- Breusch, T. S., and A. R. Pagan. 1979. A simple test for heteroskedasticity and random coefficient variation. *Econometrica* 47 (5): 1287–1294.
- Brown, R. L., J. Durbin, and J. M. Evans. 1975. Techniques for testing the constancy of regression relationships over time. *Journal of the Royal Statistical Society B37*: 149–163.
- Chow, G. C. 1960. Tests of equality between sets of coefficients in two linear regressions. *Econometrica* 28: 591–605.
- Corlett, W. 1990. Multicollinearity. In *The new Palgrave: Econometrics*, ed. J. Eatwell, M. Milgate, and P. Neyman. London: Macmillan.
- Davidson, R., and J. G. MacKinnon. 1981. Several tests for model specification in the presence of alternative hypotheses. *Econometrica* 49 (3): 781–793.
- Fomby, T. B., R. C. Hill, and S. R. Johnson. 1984. *Advanced econometric methods*. New York: Springer-Verlag.
- Goldfeld, S. M., and R. F. Quandt. 1965. Some tests for homoskedasticity. *Journal of the American Statistical Association* 60: 539–547.
- Greene, W. H. 1990. *Econometric analysis*. New York: Macmillan.
- Griffiths, W. E., R. C. Hill, and G. G. Judge. 1993. *Learning and practicing econometrics*. New York: John Wiley & Sons.
- Haitovsky, Y. 1968. Missing data in regression analysis. *Journal of the Royal Statistical Society Series B*: 67–82.
- Hausman, J. A. 1978. Specification tests in econometrics. *Econometrica* 46: 1251–1271.
- Honda, Y. 1982. On tests of equality between sets of coefficients in two linear regressions when disturbance variances are unequal. *Manchester School* 50: 116–125.
- Jarque, C. M., and A. K. Bera. 1981. An efficient large-sample test for normality of observations and regression residuals. Australian National University, Canberra. Unpublished manuscript.
- Johnston, J. 1972. *Econometric methods*, 2nd ed. New York: McGraw-Hill.

- _____. 1984. *Econometric methods*, 3rd ed. New York: McGraw-Hill.
- Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T. C. Lee. 1985. *The theory and practice of econometrics*, 2nd ed. New York: John Wiley.
- Kennedy, P. 1985. *A guide to econometrics*, 2nd ed. Boston, Mass., U.S.A.: MIT Press.
- _____. 1992. *A guide to econometrics*, 3rd edition. Boston, Mass., U.S.A.: MIT Press.
- Kmenta, J. K. 1971. *Elements of econometrics*. New York: Macmillan.
- _____. 1986. *Elements of econometrics*, 2nd ed. New York: Macmillan.
- Krämer, W., H. Sonnberger, J. Maurer, and P. Havlik. 1985. Diagnostic checking in practice. *Review of Economics and Statistics* 67 (1): 118–123.
- Krasker, W. S., E. Kuh, and R. E. Welsch. 1983. Estimation for dirty data and flawed models. In *Handbook of econometrics*, vol. 2, chapter 11, ed. Z. Griliches and M. D. Intrilligator. Amsterdam: North Holland.
- Levi, M. D. 1977. Measurement errors and bounded OLS estimates. *Journal of Econometrics* 6: 165–177.
- MacKinnon, J. G. 1983. Model specification tests against nonnested alternatives. *Econometric Reviews* 2 (1): 85–110.
- Maddala, G. S. 1977. *Econometrics*. New York: McGraw-Hill.
- _____. 1988. *Introduction to econometrics*. New York: Macmillan.
- _____. 1989. *Microeconomics: Theory and application*. New York: McGraw-Hill.
- McAleer, M., and M. H. Pesaran. 1986. Statistical inference in nonnested econometric models. *Applied Mathematics and Computation* 20: 271–311.
- Messer, K., and H. White. 1984. A note on computing the heteroskedasticity-consistent covariance matrix using instrumental variable techniques. *Oxford Bulletin of Economics and Statistics* 46 (2): 181–184.
- Norusis, M. J. 1990. *SPSS/PC+ 4.0 base manual for the IBM PC/XT/AT and PS/2*. Chicago, Ill., U.S.A.: SPSS, Inc.

- Ramsey, J. B. 1969. Tests for specification error in classical linear least squares regression analysis. *Journal of the Royal Statistical Society B31*: 250–271.
- SAS Institute, Inc. 1988. *SAS language guide for personal computers*, release 6.03 ed. Cary, N.C., U.S.A.
- Shapiro, S. S., and M. B. Wilks. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52: 591–611.
- Shapiro, S. S., M. B. Wilks, and H. J. Chen. 1968. A comparative study of various tests for normality. *Journal of the American Statistical Association* 63: 1343–1372.
- Stewart, M. B., and K. F. Wallis. 1981. *Introductory econometrics*. New York: Halsted Press.
- Suits, D. B., A. Mason, and L. Chan. 1978. Spline functions fitted by standard regression methods. *Review of Economics and Statistics* 60 (1): 132–139.
- Thursby, J. G. 1979. Alternative specification error tests: A comparative study. *Journal of the American Statistical Association* 74 (March): 222–225.
- _____. 1981. A test strategy of discriminating between autocorrelation and misspecification in regression analyses. *Review of Economics and Statistics* 63 (1): 117–123.
- _____. 1982. Misspecification, heteroskedasticity, and the Chow and Goldfeld-Quandt tests. *Review of Economics and Statistics* 64 (2): 314–321.
- Thursby, J. G., and P. Schmidt. 1977. Some properties of tests for specification error in a linear regression model. *Journal of the American Statistical Association* 72 (September): 635–641.
- Utts, J. M. 1982. The rainbow test for lack of fit in regression. *Communications in Statistics—Theory and Methods* 11: 2801–2815.
- Welsch, R. E. 1980. Regression sensitivity analysis and bounded influence estimation. In *Evaluation of econometric models*, ed. J. Kmenta and J. B. Ramsey, 153–167. New York: Academic Press.
- White, H. 1980. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica* 48 (4): 817–838.