STATA TECHNICAL BULLETIN

January 1992
STB-5

A publication to promote communication among Stata users

Subscriptions are available from Stata Corporation, email stata@stata.com, telephone 979-696-4600 or 800-STATAPC, fax 979-696-4601. Current subscription prices are posted at www.stata.com/bookstore/stb.html.

Previous Issues are available individually from StataCorp. See www.stata.com/bookstore/stbj.html for details.

Submissions to the STB, including submissions to the supporting files (programs, datasets, and help files), are on a nonexclusive, free-use basis. In particular, the author grants to StataCorp the nonexclusive right to copyright and distribute the material in accordance with the Copyright Statement below. The author also grants to StataCorp the right to freely use the ideas, including communication of the ideas to other parties, even if the material is never published in the STB. Submissions should be addressed to the Editor. Submission guidelines can be obtained from either the editor or StataCorp.

The *Stata Technical Bulletin* (ISSN 1097-8879) is published six times per year by Stata Corporation. Stata is a registered trademark of Stata Corporation.

Contents of this issue

an1.1 STB categories and insert codes

Inserts in the STB are presently categorized as follows:

General Categories:

an	announcements	*ip*	instruction on programming
cc	communications & letters	*os*	operating system, hardware, & interprogram communication
dm	data management		
dt	data sets	*qs*	questions and suggestions
gr	graphics	*tt*	teaching
in	instruction	*zz*	not elsewhere classified

Statistical Categories:

sbe	biostatistics & epidemiology	*srd*	robust methods & statistical diagnostics
sed	exploratory data analysis	*ssa*	survival analysis
sg	general statistics	*ssi*	simulation & random numbers
smv	multivariate analysis	*sss*	social science & psychometrics
snp	nonparametric methods	*sts*	time-series, econometrics
sqc	quality control	*sxd*	experimental design
sqv	analysis of qualitative variables	*szz*	not elsewhere classified

In addition, we have granted one other prefix, *crc*, to the manufacturers of Stata for their exclusive use.

an15.1 Regression with Graphics now available from CRC

Leonard Brown, CRC, 800-782-8272, FAX 310-393-7551

Regression with Graphics, by Lawrence Hamilton, is now available from CRC for $49.95, plus shipping. The book provides a unique treatment of regression by integrating graphical and regression methods for performing exploratory data analysis. Stata graphs and output are used throughout the book.

The Table of Contents printed in *an15* in STB-4 was based on a pre-publication manuscript. The following is the Table of Contents as it appears in the published book. Each chapter ends with a Conclusion, Exercises, and Notes (not shown).

Chapter 1: VARIABLE DISTRIBUTIONS—The Concord Water Study; Mean, Variance, and Standard Deviation; Normal Distributions; Median and Interquartile Range; Boxplots; Symmetry Plots; Quantile Plots; Quantile-Quantile Plots; Quantile-Normal Plots; Power Transformations; Selecting an Appropriate Power

Chapter 2: BIVARIATE REGRESSION ANALYSIS—The Basic Linear Model; Ordinary Least Squares; Scatterplots and Regression; Predicted Values and Residuals; R^2, Correlation, and Standardized Regression Coefficients; Reading Computer Output; Hypothesis Tests for Regression Coefficients; Confidence Intervals; Regression through the Origin; Problems with Regression; Residual Analysis; Power Transformations in Regression; Understanding Curvilinear Regression

Chapter 3: BASICS OF MULTIPLE REGRESSION—Multiple Regression Models; A Three-Variable Example; Partial Effects; Variable Selection; A Seven-Variable Example; Standardized Regression Coefficients; t-Tests and Confidence Intervals for Individual Coefficients; F-Tests for Sets of Coefficients; Multicollinearity; Search Strategies; Interaction Effects; Intercept Dummy Variables; Slope Dummy Variables; Oneway Analysis of Variance; Twoway Analysis of Variance

Chapter 4: REGRESSION CRITICISM—Assumptions of Ordinary Least Squares; Correlation and Scatterplot Matrices; Residual versus Predicted Y Plots; Autocorrelation; Nonnormality; Influence Analysis; More Case Statistics; Symptoms of Multicollinearity

Chapter 5: FITTING CURVES—Exploratory Band Regression; Regression with Transformed Variables; Curvilinear Regression Models; Choosing Transformations; Evaluating Consequences of Transformation; Conditional Effect Plots; Comparing Effects; Nonlinear Models; Estimating Nonlinear Models; Interpretation

Chapter 6: ROBUST REGRESSION—A Two-Variable Example; Goals of Robust Estimation; M-Estimation and Iteratively Reweighted Least Squares; Calculation by IRLS; Standard Errors and Tests for M-Estimates; Using Robust Estimation; A Robust Multiple Regression; Bounded-Influence Regression

Chapter 7: LOGIT REGRESSION—Limitations of Linear Regression; The Logit Regression Model; Estimation; Hypothesis Tests and Confidence Intervals; Interpretation; Statistical Problems; Influence Statistics for Logit Regression; Diagnostic Graphs

Chapter 8: PRINCIPAL COMPONENTS AND FACTOR ANALYSIS—Introduction to Components and Factor Analysis; A Principal Components Analysis; How Many Components?; Rotation; Factor Scores; Graphical Applications: Detecting Outliers and Clusters; Principal Factor Analysis; An Example of Principal Factor Analysis; Maximum-Likelihood Factor Analysis

Appendices: Population and Sampling Distributions; Computer-Intensive Methods; Matrix Algebra; Statistical Tables

dm3.1 Typesetting correction to automatic command logging for Stata

The `profile.add` program in `\dm3` of the STB-4 diskette contains code that is related to typesetting the STB. The program will not run correctly in its present form. The code should read as the `autolog` program shown on page 4 of STB-4. The corrected program is on the STB-5 diskette. We apologize to Mr. Judson for our error.

dm5 Creating a grouping variable for data sets

Marc Jacobs, Social Sciences, University of Utrecht, The Netherlands FAX (011)-31-30-53 4405

Sometimes I need a vector that runs from 1 to k, n times. Programs like GLIM (Generalized Linear Interactive Modelling) or ML3 (Multilevel analysis) make use of those vectors. If necessary it is possible to create such vectors. But both of the mentioned programs are not very easy to use. Data manipulating is not very simple. Everything that is possible I do in Stata. That is why I constructed two short programs. The first one, `bv.ado`, creates a vector, with total length `_N`, consisting of blocks running from 1 to k, as many times as is possible. The second one, `bvs.ado`, is similar, but constructs a vector, length `_N`, consisting of blocks of 1, 2, . . . , n.

Results would be like this:

```
. bv n 5        . bvs s 5
    n               s
    1               1
    2               1
    3               1
    4               1
    5               1
    1               2
    2               2
    3               2
    4               2
    5               2
    .               .
    .               .
    n               n
```

Both programs are found on the STB-5 diskette.

dm6 A utility to document beginning and ending variable dates

Sean Becketti, Federal Reserve Bank of Kansas City

`finddate` *varlist* [`if` *exp*] [`in` *range*] [`, date(`*datevars*`) nobs`]

lists the dates or observation numbers of the first and last nonmissing observations for each variable in *varlist*. `finddate` is useful for documenting data sets when they are constructed and for exploring unfamiliar data sets.

If the `date` option is specified, the first and last dates are listed; otherwise, the first and last observation numbers are listed. The data set is assumed to be sorted in the order of *datevars*.

If `nobs` is specified, the number of nonmissing observations, the total number of observations, and the number of gaps in the series are indicated. If there are no gaps, missing values at the beginning or ending of a series are ignored in determining the "total" number of observations.

Example

```
. use nfcbout, clear
(Nonfin. corp. liabilities)
. describe
Contains data from nfcbout.dta
  Obs:   158 (max= 66571)        Nonfin. corp.liabilities
 Vars:     5 (max=    99)
  1. year         int   %8.0g            Year
  2. quarter      int   %8.0g   quarter  Quarter
  3. corpeq       float %9.0g            Corporate equities
  4. bonds        float %9.0g            Corporate bonds, NF
  5. mortgage     float %9.0g            Mortgages, NF
Sorted by:  year  quarter

. finddate corpeq-mortgage
          First      Last
          Obs        Obs
------------------------------
corpeq    1          158
bonds     8          158
mortgage  1          158
```

```
. finddate corpeq-mortgage, date(year quarter) nobs
              First                 Last
          year    quarter     year    quarter
-------------------------------------------------------
corpeq    1952    Q1          1991    Q2  (158/158/0)
bonds     1953    Q4          1991    Q2  (151/151/0)
mortgage  1952    Q1          1991    Q2  (153/158/2)
```

ip1 Customizing a Stata menu system

Marc Jacobs, Social Sciences, University of Utrecht, The Netherlands FAX (011)-31-30-53 4405

t_menu is a program shell that allows the user to create a customized Stata menu system. The number of menu choices permitted is limited only by the length of the screen; however, more may be created if one wants to include deeper levels.

```
* 6 October (10) 1991 (12:33)
* (c) Marc Jacobs, Utrecht, The Netherlands ( Telefax:   31 30 53 44 05)
*                                           (  E-mail: CUSMAR@CC.RUU.NL)
*
* Menu maker in STATA
* Features 1: Is there a dataset in memory? Save it or drop it?
*          2: Test validity of choice, can be more sophisticated
* Problems 1: Pause in program? (via DOS shell it is possible)
*          2: Clear screen in Stata (now via the DOS shell, inelegant)
*          3: Displaying ASCII > 128 seems impossible
* Trying   1: Calling other programs or do-files
program define t_menu
        capture confirm existence %S_FN
        if (_rc~=6) {
                !cls
                mac def _bad 1
                while %_bad {
                        di in bl "WARNING! " in gr "Current data set in use is %S_FN"
                        di in gr "Do you want me to save it first?" _n
                        di in ye _col(15) "1" in bl " - SAVE " in gr "%S_FN"
                        di in ye _col(15) "2" in bl " - DROP " in gr "%S_FN"
                        di _newline(2) in gr "Your choice? " _request(_dr_sv)
                        if ("%_dr_sv" == "1") {
                                save, replace
                                capture drop _all
                                capture label drop _all
                                mac def _bad 0
                        }
                        else if ("%_dr_sv" == "2") {
                                capture drop _all
                                capture label drop _all
                                mac def _bad 0
                        }
                }
        }
        mac def _cont 1
        while %_cont {
                !cls                        /* schoon het scherm, via de shell */
                                          /* zet het woord menu in het midden neer */
                disp in ye _col(38) " MENU "
                disp _n(3)
                disp in ye _col(15) "1" in gr " - Prepare data set 1"
                disp in ye _col(15) "2" in gr " - Prepare data set 2"
                disp in ye _col(15) "3" in gr " - Prepare data set 3"
                disp in ye _col(15) "4" in gr " - Option 4"
                disp in ye _col(15) "5" in gr " - Option 5"
                disp in ye _col(15) "6" in gr " - Option 6"
                disp in ye _col(15) "7" in gr " - Option 7"
                disp in ye _col(15) "8" in gr " - Option 8"
                disp in ye _col(15) "9" in gr " - Just quit"
                disp _newline(1)
                disp in gr "make choice " _request(_choice)
                if "%_choice" == "1" {
                        disp in bl "Preparing data set 1 for analyses"
                        /* Stata commands */
                }
```

```
                else if "%_choice" == "2" {
                        disp in bl "Preparing data set 2 for analyses"
                        /* Stata commands */
                }
                else if "%_choice" == "3" {
                        disp in bl "Preparing data set 3 for analyses"
                        /* Stata commands */
                }
                else if "%_choice" == "4" {
                }
                else if "%_choice" == "5" {
                }
                else if "%_choice" == "6" {
                }
                else if "%_choice" == "7" {
                }
                else if "%_choice" == "8" {
                }
                else if "%_choice" == "9" {
                        disp in bl "Bye bye"
                        exit
                }
                else {
                        disp _newline(2)
                        disp in bl "WARNING!" in gr /*
                        */ " Choice not valid, choose number smaller then 9."
                        !pause
                }
        }
end
```

`t_menu` is found on the STB-5 diskette. To create your own customized menu, just edit `t_menu.ado` with any ASCII editor, and fill in the menu options with your own choices.

[I am interested in receiving example menu systems created with this shell.—Ed.]

gr9 | Partial residual graphs for linear regression

Joseph Hilbe, Editor, STB, FAX 602-860-1446

`partres` *depvar varlist* [`, lowess`]

produces partial residual graphs for each independent variable in *varlist*. The program implements the `lowess` option by invoking `ksm` (*gr6*, Royston 1991), so you must install *gr6* for this option to work. `lowess` graphs a lowess smoother over the partial residual graph, allowing the user to determine more easily the shape of partial residuals. With or without the lowess curve, the graphs aid in detecting nonlinearities as well as identifying cases with high-residual values.

Each variable is calculated separately and is not standardized. The formula for partial residuals is

$$r = X_{ij}\beta_{ij} + (y_i - \mu_i)$$

where $(y_i - \mu_i)$ are the values of the residuals produced as a result of linear regression of y on X_i. The partial residual graph is a scatterplot of the partial residuals on the y-axis versus the actual variable values on the x-axis. Unlike the `leverage` command, `partres` displays all independent variables automatically.

Using the `auto.dta` provided on one of the disks that came with Stata, I will graph `price` on `mpg` and `gratio` using both the `leverage` and `partres` commands.

```
. leverage price mpg gratio                (Figure 1)
. leverage price gratio mpg                (Figure 2)
. partres  price mpg gratio, lowess        (Figures 3 and 4)
```

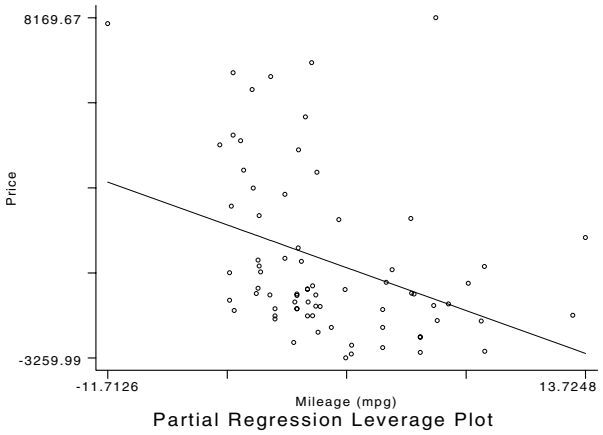

Figure 1

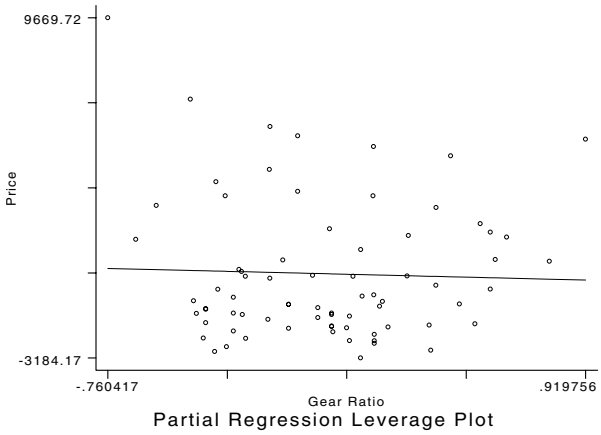

Figure 2

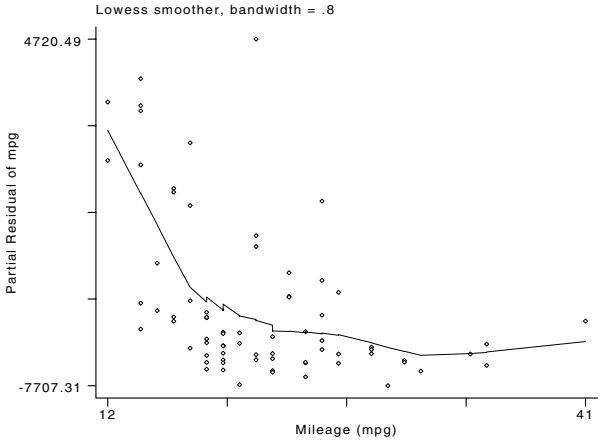

Figure 3

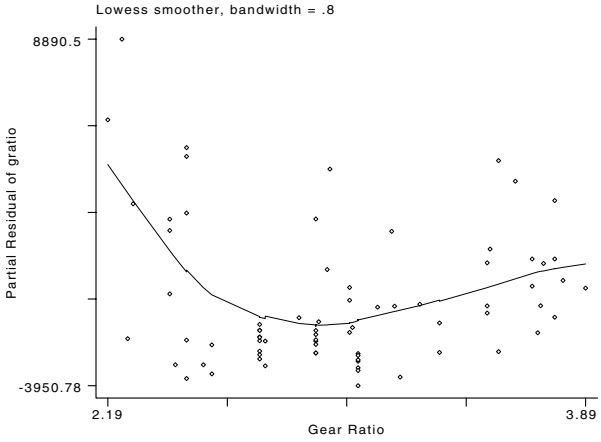

Figure 4

References

Royston, J. P. 1991. gr6: Lowess smoothing. *Stata Technical Bulletin* 3: 7–9.

gr10 Printing graphs and creating WordPerfect graph files

Thomas R. Saving & Jeff Montgomery, Dept. of Economics, Texas A&M University

The ability to print graphs and create WordPerfect graphs directly from Stata is something I found important enough to write ado commands to perform. The commands are `gphprt` and `gphwp`. `gphwp` creates the WordPerfect file and is printer independent, and `gphprt`—the print graph command—is printer specific.

`gphprt` takes any Stata `.gph` file, creates the appropriate print file, sends the file to your printer, and then erases the print file, leaving the original `.gph` file in place. Because I find the Stata default portrait-size graph too small, the command changes the default to landscape at 150% of the portrait size. This scale is the largest graph that can be printed on standard 11 x $8\frac{1}{2}$ paper. While the entire graph does fit on the page, Stata gives an erroneous error message that some of the graph may be chopped at this scale. The syntax for `gphprt` is

`gphprt` *graphfilename scale*

where the full path name of the graph file is required and scale is expressed as a percent of a full landscape page. If you want a smaller image, express your desired size as a percent of 100. For example, a scale of 75 will give you an image that has x- and y-axes 75% of the axes for the full-page graph. The file extension need not be included if you have used the Stata default extension `.gph`. The scale parameter need not be included if the default of full page landscape is satisfactory.

Example: `gphprt c:\stata\files\mygraph 75`

The `gphwp` command writes a HP Graphics Language (HP-GL) file from a Stata `.gph` file for importation into WordPerfect. It has been my experience that HP-GL files produce WordPerfect graphs that exactly duplicate the original. I should also note

that I have never been able to successfully import a Stata generated Postscript graphics file into WordPerfect. The syntax for gphwp is

gphwp *graphfilename*

where the full path name of the graph file is required. As in the graph print command, the file extension need not be included if you have used the Stata default extension .gph. The gphwp command creates a file in your WordPerfect file directory called graph.hgl. This file can be directly imported into WordPerfect.

```
gphwp c:\stata\files\mygraph
```

The two ado programs are

gphprt

```
program define gphprt
if "%_1"=="" {
     di in red "invalid syntax -- see help gphprt"
     exit 198
}
if "%_2"=="" {
     mac def scale=150
}
else {
     mac def scale=int(%_2*1.5)
}
! gphpen %_1 /n /ogphprt.ps /r%scale
! copy c:\stata\gphprt.ps lpt1
! erase c:\stata\gphprt.ps
end
```

gphwp

```
program define gphwp
if "%_*"=="" {
     di in red "invalid syntax -- see help gphwp"
     exit 198
}
! gphpen %_1 /dhp7475ls /oc:\wp51\files\graph.hpl
end
```

sbe4 Further aspects of RIA analysis

Paul J. Geiger, USC School of Medicine, pgeiger@vm.usc.edu

Statistical calculations for RIA (radioimmunoassay) using Stata with the logit-log method were described in *sbe3* (Geiger 1991). The logit-log method (Rodbard and Lewald 1970) is based on a transformation of data defined as $\text{logit}(Y) = \log_e(Y/(1-Y))$. The transformed variable must be Y=B/Bo, where B is CPM (counts per minute) of bound, labeled antigen (above nonspecific CPM) divided by Bo, the CPM in the absence of unlabeled antigen (above nonspecific CPM). The unlabeled antigen is that present in the unknown or sample being assayed. It is also present in the standards with which the standard curve is constructed. The logit-log transformation reduces the hyperbolic curve of CPM vs. dose to a straight line, logit(B/Bo) vs. $\log_{10}(dose)$. The hyperbolic curve can be seen by graphing cpm vs. stds_pg using the data supplied in *sbe3* (Geiger 1991). The logit-log method and its application have been extensively described (Chard 1990; Rodbard et al. 1987; and Tijssen 1985).

In fact this method works for 90%–95% of experimental cases. These cases might deal not only with RIA but also with any analytical system that provides a hyperbolic curve when response is plotted vs. dose. Two examples are EIA, enzyme immunoassay, and ELISA, enzyme-linked immunosorbent assay. Unfortunately, in about 5%–10% of assays, the logit-log method fails to provide an adequate description of the dose response curve. This performance failure can be found by plotting the residuals after fitting the regression line. An alternative is to observe an unacceptable coefficient of determination, R^2, that is, significantly less than 0.99–1.0 (ideal).

In the event of failure of the logit-log method, the more complicated four-parameter logistic model covers most of the rest of the cases. Further fixes are also possible as detailed by (Rodbard et al. 1987). In point of fact, the four-parameter logistic method is superior to the logit-log method, both theoretically and in practice, and should be regarded as the primary, not the secondary approach (Rodbard et al. 1987). This model is expressed by the following equation:

$$y = \frac{a-d}{1+(x/c)^b} + d$$

The value a is the expected response when dose=0 (mean of `b100` in *sbe3*) and d is the expected response for infinite dose (mean of `NSB` in *sbe3*). The value c is ED50, effective dose 50%, which can be estimated from the midpoint of the logit-log plot. It can be estimated more easily from the plot of CPM vs $\log_{10}(dose)$, midway between the upper and lower plateaus of the sigmoidal curve. The exponent b is the slope factor, corresponding to the slope of a logit-log plot, or the pseudo-Hill coefficient (Segel 1976 , 309–311; Atkinson et al. 1987, 141–148).

The nonlinear regression program, `nonlin.ado` (Danuso 1991; `nonlin.ado` is included in the \ sbe4 directory) is ideal for this type of application. It is a little slower than the ado-files supplied for the logit-log plot in *sbe3* because it is interactive and the above equation must be typed in and the parameters, `%b1`=a, `%b2`=b, `%b3`=c, `%b4`=d, assigned when prompted. If many experiments are to be analyzed, a macro program such as SmartKey or Superkey may be useful to enter the equation and shorten computation time.

For the present demonstration, the values to be typed into `nonlin.ado` are selected as above for a and d from the `ria.dct` (in the \`sbe3` directory on the STB-3 disk) file or the table in *sbe3*. The value 1 is estimated for b, and ED50 (c) can be chosen from the midpoint of the standards concentrations, 50 pg/ml. Of course, y is `cpm` and x is `stds_pg` from the same file. After the values are entered, the number of iterations is chosen and six or eight converged for the data in *sbe3*. The results are illustrated in the following table:

```
          =====NONLINEAR REGRESSION RESULTS=====

 File: RIAPAR4.DTA                          N. of iterations: 8
 Variable Y  : cpm
 Variables Xi: stds_pg
 Model: cpm=((1559.6904-87.57836)/(1+(stds_pg/38.066223)^ 1.0845367))+87.57836
 Data selection: if 1
Residual Statistics:
 Residual Average = -2.657e-06     Stand. Dev. = 20.344944
 Skewness         = -.33118934     Kurtosis    = 1.7274191
------------------------------------------------------------
  Variation       d.f.          SS                 MS
------------------------------------------------------------
  Model            4         23339744         5834936
  Residual         23        10761.835        467.90587
  Total            27        23350506         864833.55
  Corr Total       26        6347782.7        244145.49
------------------------------------------------------------
  R^ 2 = .9983

---------------------------------------------------------------------
       Parameter        Standard Error           t         Prob. t
---------------------------------------------------------------------
 b1         1559.6904        20.334942        76.700017   0
 b2         1.0845367        .05226691        20.749968   0
 b3         38.066223        1.4343017        26.5399     0
 b4         87.57836         22.027671        3.9758339   .0005973
---------------------------------------------------------------------

=== CORRELATION COEFFICIENT AMONG PARAMETERS ===
        |       a1       a2       a3       a4
--------+------------------------------------
      a1|   1.0000
      a2|  -0.8372   1.0000
      a3|  -0.3826  -0.0093   1.0000
      a4|  -0.6016  -0.8491  -0.4171   1.0000
```

Computations for the unknown samples are made by using `par4.ado`:

```
/* Do file for computing 4-parameter model answers to RIA
data analysis.  Please input b1 b2 b3 b4 from the nonlin output
by typing "do par4 b1 b2 b3 b4" */
. gen pg_ml = %_3* ((-1+(%_1-%_4)/(Scpm-%_4))^ (1/%_2))
. gen answer = pg_ml/Vol_ml
. format answer %8.2f
. noisily list smpl pg_ml answer
```

This short file computes the answers from the four-parameter equation rearranged to solve for x in pg/ml. This file is not interactive so the command '`do par4` a b c d' must be typed in carefully. The values a b c d from the four-parameter logistic equation are the estimated parameters, `b1 b2 b3 b4`, taken from the nonlinear regression table illustrated above. The following table allows comparison of the nonlinear, four-parameter logistic method with the results obtained using the logit-log method and the same data used in *sbe3*:

Sample ID no.	pg/ml four-param.	pg/ml logit-log	pg/ml Ref.[1] in sbe3
11772	1728.94	1767.13	
11772	1742.05	1781.31	1859.11
11772	1472.59	1544.25	
11772	1493.71	1567.16	1634.48
11773	1325.08	1332.42	
11773	1247.85	1300.03	1389.57
11774	1194.67	1193.03	
11774	1170.23	1167.37	1227.63
11774	1001.48	1032.11	
11774	1040.83	1074.86	1105.71
11775	1711.63	1748.40	
11775	1764.14	1805.22	1861.62
11775	1447.79	1517.34	
11775	1485.21	1557.94	1615.47
11776	1335.13	1343.17	1401.00
11776	1241.16	1292.76	1358.63
11777	1209.68	1209.31	1258.64
11777	998.92	1029.33	1080.38
11778	1468.81	1486.53	1553.45
11778	1443.71	1512.91	1589.77
11779	1487.33	1506.45	
11779	1506.08	1526.62	1585.31
11779	1354.05	1415.52	
11779	1303.08	1360.10	1458.23
11780	998.95	986.39	
11780	967.70	953.58	1004.17
11780	904.62	927.03	
11780	836.67	853.50	932.07

The power of the four-parameter logistic may be appreciated even more in that the zero and "infinite" doses actually need not be known. In fact in some cases they, or one of them, might be very difficult or impossible to obtain. Further, one of them might be lost in the experiment or the "infinite" dose might require too much of a very expensive antigen to estimate it. Without good values for both, analysis with the logit-log method is not possible. In this case, values for a and d are chosen from the highest and lowest values of the responses in the set of standards, provided enough standards have been included to indicate the high and low plateaus of the dose response curve. The other two parameters, b and c, are chosen as before. This approach was used with the data from *sbe3* and gave identical results after the same number of iterations.

Finally, the Danuso `nonlin.ado` program will show various residual plots and a graph of the hyperbolic curve fitted to the experimental values. Uncertainties are indicated by means of circles of varying sizes around the plotted points. The illustration from `nonlin.ado` shown here is the predicted line fitted to the experimental values (see Figure 1). If a plot of the sigmoidal curve is desired in order to view the regression fit in this form, one has only to plot *ycal* and `cpm` vs. $\log_{10}$(`stds_pg`) (see Figure 2).

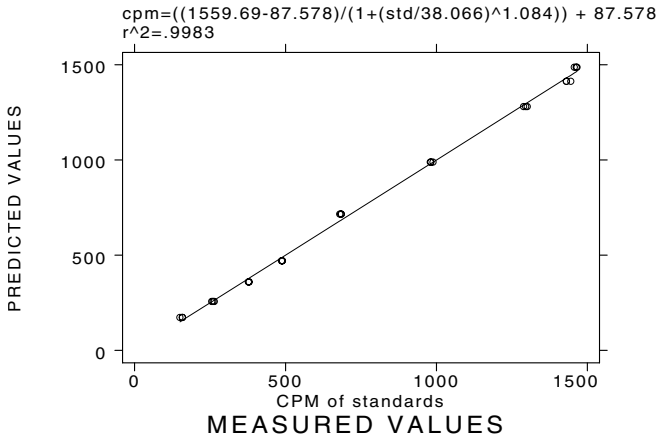

Figure 1

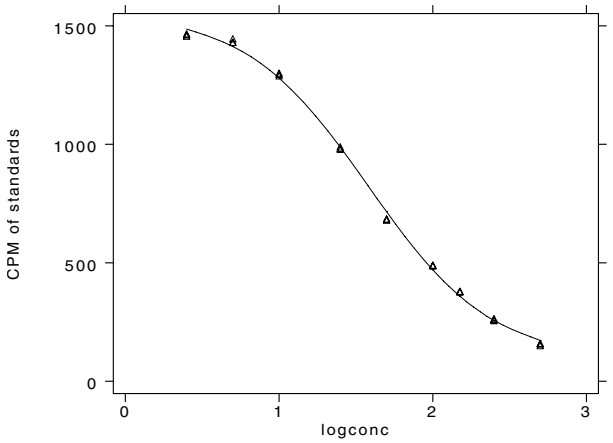

Figure 2

References

Atkinson, D. E. et al. 1987. *Dynamic Models in Biochemistry. A Workbook of Computer Simulations Using Electronic Spreadsheets.* Menlo Park, CA: Benjamin/Cummings.

Chard, T. 1990. An Introduction to Radioimmunoassay and Related Techniques. In *Laboratory Techniques in Biochemistry and Molecular Biology*, ed. R. H. Burdon and P. H. van Knippenberg, vol. 6, part 2. New York: Elsevier.

Danuso, F. 1991. sg1: Nonlinear regression command. *Stata Technical Bulletin* 1: 17–19.

Geiger, P. J. 1991. sbe3: Biomedical analysis with Stata: radioimmunoassay calculations. *Stata Technical Bulletin* 3: 12–15.

Rodbard, D. and J. E. Lewald. 1970. Computer analysis of radioligand assay and radioimmunoassay data. *Acta Endocr.* Suppl. 147: 79–103.

Rodbard, D., et al. 1987. Statistical Aspects of Radioimmunoassay. In *Radioimmunoassay in Basic and Clinical Pharmacology*, vol. 82 of *Handbook of Experimental Pharmacology*, ed. C. Patrono and B. A. Peskar, chapter 8. New York: Springer-Verlag.

Segel, I. H. 1976. *Biochemical Calculations.* 2d ed. New York: John Wiley & Sons.

Tijssen, T. 1985. Practice and Theory of Enzyme Immunoassays. In *Laboratory Techniques in Biochemistry and Molecular Biology*, ed. R. H. Burdon and P. H. van Knippenberg, vol. 15. New York: Elsevier.

sg3.7 Final summary of tests of normality

William Gould, CRC, FAX 310-393-7551

In this insert, I update the tables last presented in *sg3.4* to account for the final round of improvements by Royston in *sg3.5* and in private communications with me. This discussion has dragged on long enough that, before presenting the final results, it is worth summarizing what has happened.

CRC first introduced a homegrown test for normality that it dubbed `sktest`, the `sk` standing for the skewness and kurtosis on which the test was based. In keeping a promise I made, I compared our homegrown test to that of D'Agostino, et al. (1990) and to Bera-Jarque (Judge et al. 1985). The survivors from this comparison were the homegrown `sktest` and that of D'Agostino.

Royston in *sg3.1* retorted with strong evidence of problems with `sktest` and of lessor problems with D'Agostino and, in sg3.2, promoted the Shapiro–Wilk and Shapiro–Francia tests, dubbed `swilk` and `sfrancia`. In *sg3.4*, Rogers and I compared the (now) four tests and, agreeing with Royston, withdrew our `sktest`. We also discovered certain problems with `swilk` in dealing with aggregated data. Meanwhile, Royston in *sg3.5* went on to make an empirical correction to the D'Agostino test in the spirit of our `sktest` designed to fix the problems he had previously observed. Since then, in private communication, Royston has further improved the D'Agostino test and made improvements to `swilk`.

Thus, we are still left with four tests. In the results below, D'Agostino refers to the original D'Agostino test. `sktest` now refers to the D'Agostino test with the empirical corrections introduced by Royston. `swilk` is the Shapiro–Wilk test as most recently updated by Royston. `sfrancia` is the Shapiro–Francia test as submitted by Royston in *sg 3.2*. The results are

True Distribution	Test	1%	5%	10%	True Distribution	1%	5%	10%
Normal					Contaminated Normal			
	D'Agostino	.018	.059	.100		.965	.970	.973
	sktest	.011	.051	.104		.963	.968	.973
	swilk	.009	.051	.102		.961	.966	.971
	sfrancia	.010	.057	.108		.963	.970	.973
Uniform					Long-tail Normal			
	D'Agostino	.985	.997	.999		.081	.179	.263
	sktest	.975	.997	.999		.057	.165	.267
	swilk	.949	.997	.999		.068	.179	.269
	sfrancia	.767	.970	.993		.089	.229	.343
t(5)					t(20)			
	D'Agostino	.453	.595	.673		.069	.137	.197
	sktest	.406	.578	.676		.054	.127	.201
	swilk	.413	.558	.639		.043	.112	.174
	sfrancia	.466	.629	.712		.055	.142	.215
chi2(5)					chi2(10)			
	D'Agostino	.883	.977	.995		.606	.806	.895
	sktest	.837	.970	.995		.540	.784	.899
	swilk	.986	.997	1.000		.763	.903	.949
	sfrancia	.974	.996	.998		.711	.880	.933
grouped Normal					group t(5)			
	D'Agostino	.019	.057	.101		.444	.583	.661
	sktest	.011	.050	.103		.395	.566	.664
	swilk	.005	.024	.046		.352	.482	.547
	sfrancia	.003	.016	.033		.386	.528	.602

To refresh your memory, the numbers reported are the fraction of samples that are rejected at the indicated significance level. Tests were performed by drawing 10,000 samples, each of size 100, from the indicated distribution. Each sample was then run through each test and the test statistic recorded. (Thus, each test was run on exactly the same sample.)

I will leave the interpretation of the table to the reader except to note that all tests now perform well. In particular, `sktest` (D'Agostino with the Royston correction) performs quite well even on aggregated data and `swilk` now performs as least as satisfactorily as `sfrancia` on aggregated data. Final versions of all tests are provided on the STB diskette.

References

D'Agostino, R. B., A. Belanger and R. B. D'Agostino, Jr. 1990. A suggestion for using powerful and informative tests of normality. *The American Statistician* 44: 316–321.

——. 1991. sg3.3: Comment on tests of normality. *Stata Technical Bulletin* 3: 20.

Gould, W. W. 1991. sg3: Skewness and kurtosis tests of normality. *Stata Technical Bulletin* 1: 20–21.

Gould, W. W. and W. Rogers. 1991. sg3.4: Summary of tests for normality. *Stata Technical Bulletin* 3: 20–23.

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T. C. Lee. 1985. *Theory and Practice of Econometrics*. 2d ed. New York: John Wiley & Sons.

Royston, J. P. 1991a. sg3.1: Tests for departure from normality. *Stata Technical Bulletin* 2: 16–17.

——. 1991b. sg3.2: Shapiro–Wilk and Shapiro–Francia tests. *Stata Technical Bulletin* 3: 19.

——. 1991c. sg3.5: Comment on sg3.4 and an improved D'Agostino test. *Stata Technical Bulletin* 3: 23–24.

——. 1991d. sg3.6: A response to sg3.3: comment on tests of normality. *Stata Technical Bulletin* 4: 8–9.

sg5 | Correlation coefficients with significance levels

Sean Becketti, Federal Reserve Bank of Kansas City

`corrprob` *varname1 varname2* [`in` *range*] [`if` *exp*] [`,` *type*]

where *type* is one of

`all`	all correlation measures
`pearson`	Pearson's product-moment correlation
`spearman`	Spearman's rank-correlation
`kendall`	Kendall's τ_β

displays one or more correlation coefficients between *varname1* and *varname2* along with a normal-approximation to the test that the rank correlation is zero. If no *type* is specified, then `pearson` is assumed.

Examples

```
. use census
(Census data by state, 1980)
. describe
Contains data from census.dta
 Obs:    50 (max= 32249)            Census data by state, 1980
 Vars:    4 (max=    99)
 1. state      int    %8.0g   fips     State
 2. region     int    %8.0g   cenreg   Census region
 3. brate      float  %9.0g            Births per 100,000
 4. dvcrate    float  %9.0g            Divorces per 100,000
Sorted by:  region
. set rmsg on
r; t=0.00 10:27:52
. corrprob brate dvcrate
(nobs=50)
Pearson´s r    =  0.28
Prob z > |r|   =  0.05
r; t=1.20 10:27:54
. corrprob brate dvcrate, all
(nobs=50)
Pearson´s r    =  0.28
Prob z > |r|   =  0.05
Spearman´s r   =  0.41
Prob z > |r|   =  0.00
Kendall´s tau  =  0.29
Prob z > |tau| =  0.00
r; t=158.29 10:30:33
```

Note that calculation of Kendall's τ_β takes a long time.

References

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1988. *Numerical Recipes in C*. Cambridge, MA: Cambridge University Press.

sg6	Regression switching models

Daniel Benjamin, Clemson University, and William Gould, CRC, FAX 310-393-7551

[The following exchange occurred by FAX on the Stata help line—Ed.]

Question

Consider the regression:

$$y = b_0 + b_1 x_1 + b_2 x_2 + e$$

I have a theory that implies both b_0 and b_2 depend on x_2. In particular for $x_2 > x_2^\star$, $b_0^\star > b_0 > 0$ and $b_2^\star < 0 < b_2$. The theory does not tell me the value of $x_2^\star$, so I propose to determine it empirically running a series of paired regressions of the form:

$$y = \begin{cases} b_0 + b_1 x_1 + b_2 x_2 + e, & \text{if } x_2 < x_2^\star \\ b_0^\star + b_1^\star x_1 + b_2^\star x_2 + e^\star & \text{otherwise} \end{cases}$$

The value of $x_2^\star$ that produces the lowest combined residual sum of squares for the two regressions will then be my value of x_2. Once this value is determined, I will utilize a regression that incorporates shift and slope dummies tailored to the estimated value of $x_2^\star$. I have tediously done this by hand, but now I want to automate this. How?

Answer

Let's begin by considering a different way to parameterize the model. In particular, we will rewrite the model as

$$y = f(x_2) + b_1 x_1 + g(x_2) x_2 + e$$

That is, the intercept is a function of x_2 as is the coefficient on x_2. In your formulation, $f(x_2) = b_0$ or $b_0^\star$ depending on whether $x_2 < x_2^\star$, and similarly $g(x_2) = b_2$ or $b_2^\star$. Let us introduce the notation $b_0^\star = b_0 + \Delta b_0$, so Δb_0 measures the difference between b_0 and $b_0^\star$, and $b_2^\star = b_2 + \Delta b_2$. Letting $\delta = 1$ if $x_2 > x_2^\star$ and 0 otherwise, we can rewrite the model:

$$\begin{aligned} y &= (b_0 + \Delta b_0 \delta) + b_1 x_1 + (b_2 + \Delta b_2 \delta) x_2 + e \\ &= b_0 + \Delta b_0 \delta + b_1 x_1 + b_2 x_2 + \Delta b_2 (\delta x_2) + e \end{aligned}$$

The coefficients to be estimated are b_0, Δb_0, b_1, b_2, and Δb_2. This is a preferable way to estimate the model because it constrains b_1 to be the same regardless of x_2 and it constrains the variance of the residuals to be the same.

The above model can be estimated by least squares *conditional on* δ (or, equivalently, conditional on the cutoff $x_2^\star$). If, however, you attempt to find the δ (cutoff $x_2^\star$) that minimizes the sum of squares, you can still use least squares to obtain the coefficients, but the standard errors will be wrong since they do not account for the fact that δ is an estimate. They are wrong and too small.

Before opting for the cutoff model, I would ask myself—does my theory really imply switching behavior? That is, are the coefficients really one value if $x_2 < x_2^\star$ and some other value otherwise, as opposed to a smooth function? For instance, another alternative would be $f(x_2) = f_0 + f_1 x_2$ and $g(x_2) = g_0 + g_1 x_2$. Then the model is

$$\begin{aligned} y &= (f_0 + f_1 x_2) + b_1 x_1 + (g_0 + g_1 x_2) x_2 + e \\ &= f_0 + b_1 x_1 + (f_1 + g_0) x_2 + g_1 x_2^2 + e \end{aligned}$$

The coefficients to be estimated are f_0, b_1, $(f_1 + g_0)$, and g_1. This equation can be estimated directly by least squares and the standard errors are all estimated correctly. In most cases, this second formulation is preferable. It is not preferable, of course, if your theory really implies switching behavior.

If you want to estimate the switching model, the following program should work:

```
program define gsearch
        * Assumptions:
        * Model is  y = f(x1,x2), where x1 and x2 are the variable names.
        * gsearch takes three parameters
        *  1.  lower bound for search
        *  2.  upper bound for search
        *  3.  increment
        * e.g.,
        *    gsearch 0 10 .1
        * will search X=0, .1, .2, ..., 10.
```

```
        mac def _X0 %_1
        mac def _X1 %_2
        mac def _Xd %_3
        mac def _minrss 1e+30
        mac def _X .
        capture drop delta deltax2
        quietly {
                while %_X0 <= %_X1 {
                        gen delta = x2 > %_X0
                        gen deltax2 = x2 * delta
                        regress y delta x1 x2 deltax2
                        di "for X= %_X0  rss = " _result(4)
                        if _result(4) < %_minrss {
                                mac def _minrss = _result(4)
                                mac def _X %_X0
                        }
                        drop delta deltax2
                        mac def _X0 = %_X0 + %_Xd
                }
        }
        mac def _X0 %_X
        di "Optimum value is X= %_X0"
        gen delta = x2 > %_X0
        gen deltax2 = x2 * delta
        di "Model is"
        regress y delta x1 x2 deltax2
        di "note:  standard errors conditional on estimated X"
end
```

Typing '`gsearch 1 2 .01`', for instance, will search for the optimal value of $x_2^\star$ between 1 and 2 by trying $x_2^\star = 1$, 1.01, 1.02, . . ., 1.98, 1.99, and 2. The best value found will be reported along with the corresponding regression.

smv2.1 | Minor change to single factor repeated measures ANOVA

A replacement `ranova.ado` program is included on the STB-5 disk. It fixes a problem that is unlikely to occur which concerns Stata automatically dropping programs that are not used and then reloading them if you execute the command later.

smv3 | Regression based dichotomous discriminant analysis

Joseph Hilbe, Editor, STB, FAX 602-860-1446

`discrim` *depvar varlist* [`if` *exp*] [`in` *range*] [`, anova detail graph`]

allows the user to perform a discriminant analysis on a Bernoulli-distributed response or grouping variable; that is, a response variate with values of either 0 or 1. It is not for use with a multinomial response variable. Moreover, since certain simulated matrix routines involve the creation of variables, the user may need to use the `set maxvar #` command prior to loading the data set upon which the discriminant analysis is to be performed.

`discrim` command options include an `anova` table of the discriminant scores by the response variable, a `detail` option that, for each observation, lists the values of the response, the predicted value, the logistic probability of group 1 membership, the discriminant index, and the discriminant score. A column is also provided that prints a star when an observation is misclassified. Finally, the `graph` option graphs the logistic probability versus the discriminant index score. Each misclassified observation is distinguished by a '+' mark. Negative index values represent group 1 predictions.

Discriminant analysis (DA) is primarily used to generate classification coefficients which one can use to classify additional extra-model cases. The discriminant function is a linear function of the variates that maximizes the ratio of the between-group and within-group sum of squares. In essence, it is an attempt to separate the groups in such a manner that they are as distinct as possible. The procedure is based on the assumptions that both group covariance matrices are nearly the same and that the independent variables are multivariately normal. Although DA is fairly robust against violations, it has been demonstrated that logistic regression (LR) does a far better job at classification when the above violations exist. Conversely, LR does not allow classification at all when there is perfect prediction. DA has no such limitation. As a side, when there are significantly fewer covariate patterns in the data set than observations, the LR program utilizing Hosmer and Lemeshow (LRHL) methodology generally yields superior correct classification rates than either DA or ordinary LR. LRHL is implemented in Stata by `logiodd2` (Hilbe 1991). Hence, DA is appropriate to use in cases where there is perfect prediction, that is, when LR cannot be used without modification of model variables, or when there is near equality of group covariance matrices and fairly normal variates.

The default output from the `discrim` command includes the following statistics: number of independent variables, observations in group 0 and in group 1, R^2, Mahalanobis distance, group centroids, grand centroid, eigenvalue, canonical correlation, η^2, Λ, χ^2 and significance, a confusion matrix, percentages of correctly predicted observations, model sensitivity, model specificity, false positive and false negative, and a table listing both the discriminant classification coefficients and the unstandardized canonical discriminant function coefficients.

Discriminant analysis is typically performed using linear algebra. Some matrix operations may be simulated using the Stata programming language; however, it clearly does not allow matrix inversion—a necessary prerequisite to determining discriminant functions. However, since discriminant and regression coefficients are proportional, it is possible to use regression, which involves a matrix inversion, as a base to determine discriminant coefficients. In other words, we can use the $(X'X)^{-1}$ matrix inversion routine in regression as a substitute for the inversion of the pooled within-groups covariance matrix required in normal discriminant analysis. Then all that needs to be accomplished is to determine the nature of the proportionality.

There are several references in the literature regarding the ability of regression to determine discriminant classification coefficients. Those that I became acquainted with, however, simply state that there is a constant *k by which one can multiply a regression coefficient to yield a corresponding discriminant function. However, *k changes for each separate analysis. Hence the actual proportion for a given operation is not inherently clear given only the regression coefficients. I shall outline the solution that I derived and upon which the `discrim` command is based. Feel free to optimize or alter it according to your requirements. I have tried to outline the steps in such a manner that it can be, if so desired, programmed into environments other than Stata.

Create a dummy variable (`_dummy`) with two values according to the following

$$c_0 = n_0/n \qquad \text{and} \qquad c_1 = -n_1/n$$

where n_0 is the number of observations in group 0 and n_1 is the number of observations in group 1. Assign the value of c_0 to `_dummy` if group 0 and c_1 to `_dummy` if group 1. Equal-sized groups will consist of `_dummy` having the value of .5 if group 0 and $-.5$ if group 1. Regress `_dummy` on the remaining independent variables. The statistic of interest is R^2. From it the multivariate statistic Mahalanobis distance may be calculated. The desired proportion is the result of dividing R^2 by Mahalanobis.

$$M = \frac{R^2}{1-R^2}\frac{n(n-2)}{n_0 n_1} \qquad \text{and} \qquad P = \frac{R^2}{M}$$

Divide each of the coefficients generated by the regression of `_dummy` on X by the above proportion P. The result is an array of discriminant classification coefficients. The constant is calculated separately and involves matrix operation simulation. Do not try to interpret the sign of the coefficients as you would regression coefficients. They may be arbitrary; the point of the discriminant analysis is foremost classification and prediction.

Calculation of the discriminant classification constant entails the summation of group variable means; for example, sum the mean of *var1* in group 0 with the mean of *var1* in group 1, etc. The result is a vector of group mean sums. Then matrix multiply this vector by the vector of discriminant coefficients. Finally multiply the sum by $-.5$. Given b_i^r as the dummy regression coefficients,

$$b_i^c = \frac{b_i^r}{P} \qquad S_i^m = \bar{X}_{0i} + \bar{X}_{1i} \qquad b_0^c = -\frac{1}{2}\sum b_i^c S_i^m$$

where b_i^c are the discriminant classification coefficients. Note that the usual adjustment made for unbalanced groups is not required at this point; adjustment was accommodated by the `_dummy` variable.

Unstandardized canonical discriminant function coefficients (UDF) are obtained by simply dividing each classification coefficient by minus the square root of the Mahalanobis distance. The constant is determined by multiplying each resultant UDF by the respective variable mean, summing and multiplying by -1.

$$b_i^u = -M^{-\frac{1}{2}} b_i^c \qquad b_0^u = -\sum b_i^u \bar{X}_i$$

Discriminant index values are determined by summing the variate fits, based on the classification coefficients, and adding the constant. The same procedure, but using UDF, applies to calculating the discriminant scores. Group *centroids* are simply the mean of the discriminant scores for each respective group.

$$D_i^x = b_0^c + b_1^c X_i + b_2^c X_i + \cdots + b_n^c X_n$$
$$D_i^s = b_0^u + b_1^u X_i + b_2^u X_i + \cdots + b_n^u X_n$$

The observation logistic probability for group 1 membership is calculated by $p_i^1 = 1/(1+e^{D_i^x})$.

Performing a one-way ANOVA of the discriminant scores on the response or grouping variable produces an ANOVA table that can be used for diagnostics. The *eigenvalue* is determined by dividing the between-groups Sum of Squares (SS_b) by the within-groups SS (SS_w). That is, $e = \frac{SS_b}{SS_w}$.

The eigenvalue is a means of evaluating the discriminating power of the model. An eigenvalue of near 0 indicates that the discriminant model has little discriminating power. Eigenvalues in excess of 0.40 are desirable.

Canonical correlation is similar to the eigenvalue, with the exception that its values are limited to 0.0–1.0. It is the same as the Pearson R^2 between the discriminant scores and the grouping variate. $cc = \frac{SS_b}{SS_t}$ where SS_t is the total SS of the model.

Wilk's Λ is a statistic that measures the degree of difference between group means. It can range from 0.0 to 1.0. Lower values indicate a model with better discriminating power. For example, a Λ of .20 means that the differences between the two groups account for some 80 percent of the variance in the independent variables. *Eta-squared* (η^2) is simply $1 - \Lambda$ or $\frac{SS_b}{SS_t}$. It indicates the ratio of total variance (SS_t) in the discriminant scores that can be explained by differences between the two groups. $\Lambda = \frac{SS_w}{SS_t}$

The significance of Λ is determined by creating a χ^2 variable from Λ with p degrees of freedom. It is a test that the group means are equal.

$$\chi^2 = -(n - \frac{p+2}{2} - 1)\ln(\Lambda)$$

where p is the number of predictor variables in the model.

The `anova` option provides an F statistic by which we can evaluate the equality of the means of the two groups. Bartlett's test for equal variances is also displayed. High values of χ^2 significance indicate that we cannot reject the assumed hypothesis that the variances are homogeneous. This is, of course, exactly what we desire in discriminant analysis.

Example

I shall model `foreign mpg price gratio` using the `auto.dta` data set as provided on one of the disks that came with Stata. First I shall employ `discrim`, then `logit` and finally `logiodd2`. `foreign` is the classification variable. This example demonstrates an occasion when discriminant analysis correctly classifies observations into groups with greater success than does logistic regression. Again, when both groups have fairly equal covariance matrices and the variables are multivariately normal, discriminant analysis will often outperform logistic regression. Unfortunately this is not often the case.

Classification outputs are provided using `discrim` with the optional classification graph (*Figure 1*), `logit`, and `logiodd2` with the code to produce a graph similar to that of `discrim` (*Figure 2*). The two logistic commands yield identical classification results since the number of covariate patterns is identical to the number of observations in the data set (Hilbe 1991). For this example, the `discrim` command yields a correct classification percentage of 89.18 % while logistic regression correctly classifies 86.49 %.

```
. use auto
(1978 Automobile Data)
. discrim foreign mpg price gratio, a d g

                Dichotomous Discriminant Analysis

Observations    = 74                      Obs Group 0 =    52
Indep variables = 3                       Obs Group 1 =    22

Centroid 0   =   -0.7596                  R-square    =   0.5836
Centroid 1   =    1.7954                  Mahalanobis =   6.5277
Grand Cntd   =    1.0358

Eigenvalue   =    1.4016                  Wilk´s Lambda =  0.4164
Canon. Corr. =    0.7639                  Chi-square    = 61.7673
Eta Squared  =    0.5836                  Sign Chi2     =  0.0000

                    ----- Predicted -----
Actual    |  Group 0       Group 1    |   Total
----------+---------------------------+--------
Group 0   |       48             4    |      52
Group 1   |        4            18    |      22
----------+---------------------------+--------
Total     |       52            22    |      74
----------+---------------------------+--------

        Correctly predicted =  89.19 %
        Model sensitivity   =  92.31 %
        Model specificity   =  81.82 %
```

```
                    False Positive (G1) =   7.69 %
                    False negative (G0) =  18.18 %

                         Discrim Function    Unstandardized
          Variable         Coefficients        Coefficients
          ------------------------------------------------
          mpg                 -0.0672              0.0263
          price               -0.0006              0.0002
          gratio              -8.5901              3.3622
          Constant            32.1269             -12.0566

                Discriminant Scores v Group Variable

                          Analysis of Variance
  Source           SS         df      MS         F     Prob > F
-------------------------------------------------------------------------
Between groups   100.914538      1   100.914538  100.91 0.0000
 Within groups   72.0000153     72   1.00000021
 ------------------------------------------------------------------
    Total           172.914553     73   2.36869251
Bartlett's test for equal variances:chi2(1)=0.3163 Prob>chi2=0.574

------------------------------------------------------------------
 PRED     = Predicted Group       DIFF     = Misclassification
 LnProb1  = Probability Gr 1      DscScore = Discriminant Score
                                  DscIndex = Discriminant Index
------------------------------------------------------------------

      foreign       PRED  DIFF    LnProb1   DscIndex   DscScore
  1.        0          1   *     0.9179    -2.4141     1.4628
  2.        0          0         0.0014     6.5752    -2.0556
  3.        0          0         0.1140     2.0501    -0.2845
  4.        0          0         0.0522     2.8998    -0.6171
  5.        0          0         0.0025     6.0056    -1.8327
 70.        1          1         0.9962    -5.5696     2.6978
 71.        1          1         0.9979    -6.1403     2.9212
 72.        1          1         0.9907    -4.6707     2.3460
 73.        1          1         0.9972    -5.8841     2.8209
 74.        1          1         0.7981    -1.3742     1.0558

. logit foreign mpg price gratio, nolog tab
Logit Estimates                           Number of obs =      74
                                          chi2(3)       =   52.45
Log Likelihood =-18.807429                Prob > chi2   =  0.0000
Variable |  Coefficient    Std. Error       t    Prob > |t|   Mean
---------+--------------------------------------------------------
 foreign |                                                 .2972973
---------+--------------------------------------------------------
     mpg |    .0488955      .1008566     0.485     0.629   21.2973
   price |    .0004591       .000185     2.482     0.015  6165.257
  gratio |    6.402314      1.606307     3.986     0.000  3.014865
   _cons |   -24.81679      5.552434    -4.470     0.000         1
---------+--------------------------------------------------------------

           Comparison of Outcomes and Probabilities
         Outcome |    Pr < .5    Pr >= .5 |      Total
        ---------+--------------------------+-----------
         Failure |         48           4 |         52
         Success |          6          16 |         22
        ---------+--------------------------+-----------
           Total |         54          20 |         74

. logiodd2 foreign mpg price gratio, i
                                    << partial output >>

Number of Predictors         =         3
Number of Non-Missing Obs    =        74
Number of Covariate Patterns =        74
Pearson X^ 2 Statistic       =        40.9854
  P>chi2(70)                 =         0.9978
Deviance                     =        37.6149
  P>chi2(70)                 =         0.9995
```

```
Additional diagnostic variables created...
    logindex =  Logit; Index value
    sepred   =  Standard error of index
    pred     =  Probability of success (1)

. gen a=1 if pred<.5 & foreign==0
(26 missing values generated)

. gen d=1 if pred>=.5 & foreign==1
(58 missing values generated)

. count if a==1
   48

. count if d==1
   16

. gen LP=pred if (pred>=.5 & foreign==1) | (pred<.5 & foreign==0)
(10 missing values generated)

. gen LM=pred if LP==.
(64 missing values generated)

. lab variable LP "Classified"

. lab variable LM "Misclassified"

. gr LP LM logindex, s(.p) xlab ylab border yline(.5)
```

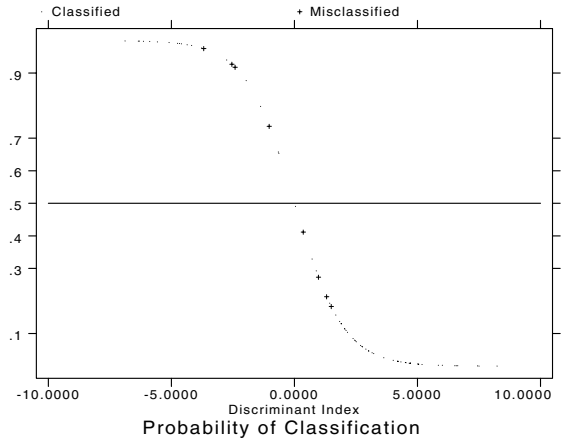

Figure 1

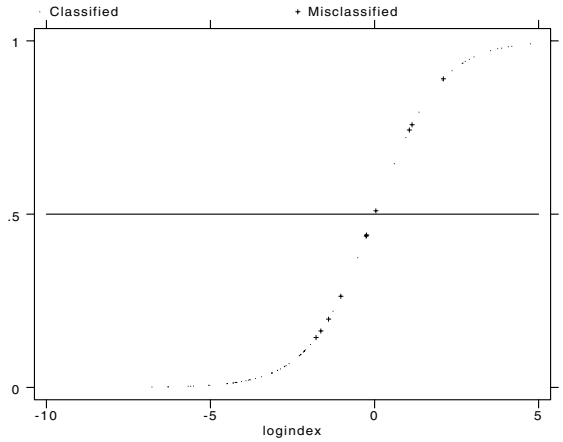

Figure 2

References

Afifi, A. A. and Clark, V. 1984. *Computer-aided Multivariate Analysis*. New York: Van Nostrand Reinhold Co.

Dillon, W. R. and M. Goldstein. 1984. *Multivariate Analysis*. New York: John Wiley & Sons.

Flury, B. and H. Riedwyl. 1985. T^2 Tests, the linear two-group discriminant function, and their computation by linear regression. *The American Statistician* 39 (1).

Hilbe, J. 1991. sqv1.3: An enhanced Stata logistic regression program. *Stata Technical Bulletin* 4: 16–18.

Klecka, W. R. 1980. *Discriminant Analysis*. Newbury Park, CA: Sage Publications.

Norusis, M. J. 1990. *SPSS/PC+ Advanced Statistics V4.0* Chicago: SPSS Inc.

sqv1.4	Typographical correction to enhanced logistic regression

The formula for `deltax` in *sqv1.3* (STB-4, p. 17) contains a typographical error. The listed 'q' should be changed to a '1'. The formula on the STB-4 disk is correct.

srd7	Adjusted summary statistics for logarithmic regressions

Richard Goldstein, Qualitas, Brighton, MA, EMAIL goldst@harvarda.bitnet

The syntax of `logsumm` is

`logsumm` *varlist*

Because of the types of calculations that must be made, `if` and `in` are *not* allowed; instead, drop cases that you don't want to use in the regression (or keep only those cases that you want).

Choice of functional form is one of the hardest, and least capable of automation, modeling decisions in regression analysis. Probably the most important criterion is the analyst's substantive, or theoretical, knowledge of the situation. However, this is rarely sufficient in itself. A number of tools have been devised to help analysts choose the appropriate functional form. This ado-file presents a number of those tools in one package for the special, but widely applicable, case of choosing between a linear and a log-linear form.

The general situation involves a choice among at least the following four forms:

1. Linear: $y = \beta_0 + \beta_1 X_1$
2. Semi-logarithmic: $\log(y) = \beta_0 + \beta_1 X_1$
3. Quadratic: $y = \beta_0 + \beta_1 X_1 + \beta_2({X_1}^2)$
4. Logarithmic: $\log(y) = \beta_0 + \beta_1 * \log(X_1)$

This particular ado-file is primarily aimed at helping users to distinguish between the first two of these forms, but can also be helpful regarding the other two (some additional comments appear below).

There are also other competitors, such as a log-transform of only (some of) the right-hand-side variables, or transforming the left-hand-side variable by taking its square root, or other fractional power, or by taking its inverse or inverse fractional power. I do not include the first alternative (log-transforming only the right-hand-side variables) because I have never found it useful in my own work. I do not include other possible transforms of the dependent variable because (1) I have found them less useful than the log-transform, and, (2) they require different forms of adjusted re-transformation to the original units (Miller 1984). It should prove easy, however, to modify this ado-file for any of the other dependent variable transformations.

Although there are many discussions of how to make such a choice in the statistical literatures of several disciplines, many users just compare the summary statistics from the two regressions. However, when the dependent variable in a linear regression is a logarithmic transform, the summary statistics are not comparable to the summary statistics from an untransformed regression. Maddala (1988, 177) puts it this way:

> When comparing the linear with the log-linear forms, we cannot compare the R^2's because R^2 is the ratio of explained variance to the total variance and the variances of y and $\log(y)$ are different. Comparing R^2's in this case is like comparing two individuals, A and B, where A eats 65% of a carrot cake and B eats 70% of a strawberry cake. The comparison does not make sense because there are two different cakes.

Kvålseth (1985, 280) is less entertaining but more straightforward when he says

> One of the most frequent mistakes occurs when comparing the fits of a linear and a nonlinear model by using the same R^2 expression but different variables: the original variable y and the fitted $\hat{y}$ for the linear model and transformed variables for the nonlinear model.

Granger (1989, 131) says

> The R^2 values are of no importance ... [if] the form of the dependent variable is not the same for the two models.

Scott and Wild (1991, 127) say

> The use of R^2 is particularly inappropriate if the models are obtained by different transformations of the response scale.

Since many of the other summary statistics, including RMSE and the F statistic are problems for the same reason (different amount of variation in the dependent variable), this program provides these statistics also.

These summary statistics, as shown in the example below, are provided for five models: the raw variable model, the semi-log model (log of dependent variable), the adjusted output from the raw model (adjusted by taking the logs of the predicted values and the dependent variable and calculating the summary statistics), and two adjusted versions of the log model.

Note that this presentation is not meant to imply that you should choose between these functional forms based solely on these summary statistics. Lots of other things, including substantive knowledge (is a multiplicative or an additive scale preferable?), need to be taken into account. Another thing you may find helpful is a plot, created by specifying the `rescale` and `rlog graph` options, showing both the untransformed variable (using, say, the left scale) and the transformed variable (using, say, the right scale).

Two sets of adjusted statistics are provided: (1) "adj. exp" is an adjustment of the anti-log to take account of the changing skewness; (2) "exp" is just the anti-log. Many people re-transform the results from log-transformed equations by just using

the anti-log (exponential); however, if the log transformation is correct, then this gives you the median rather than the mean (regression normally gives you an expected, or conditional, mean value). To get the mean, you must adjust this by using the variance from the regression. See, for example, Miller (1984), Greene (1990, 168), Granger (1989, 132), or, any of the papers cited in `logdummy.hlp`. *[*`logdummy.hlp` *is found in the* `\srd8` *directory of the STB-5 disk—Ed.]*

The summary statistics from all five models, including from the two regressions that are shown anyway, appear together in a table. The summary statistics shown are R-squared, adjusted R-squared, the F value for the regression, the root mean squared error (RMSE) for the regression, and the coefficient of variation for the regression. Also, at the bottom of each regression output, I provide the Durbin–Watson statistic in unadjusted form; this is provided since often a log transform is used because of problems that will cause D–W to fail.

The program automatically transforms the dependent variable for you. Note that this ado-file does not in any way transform the right-hand-side, or independent, variables. Thus, if you think the real competition is between the log-transformed model and an untransformed model with a quadratic effect on the right-hand-side, then you will probably need to run this ado-file twice—once with the quadratic term included on the right, and once without it. As a side-benefit you might even find that the log-transformed model with a quadratic term is best! Similarly, if you want to compare a model that is transformed to logs on both the right and left sides, then again you should probably use this ado-file twice.

I also include two other procedures in the output: (1) a "test" of whether it is possible to reject either the linear or the log-transformed version; and, (2) a simple run of the `boxcoxg` transformation ado-file. *[See srd9 and/or the associated help file on the STB-5 disk for more information—Ed.]* The test is by R. Davidson and J. G. MacKinnon and is called the PE test. It is discussed in a number of texts as well as in the following two articles: Davidson and MacKinnon (1985) *[their data set is included in the* `\srd7` *directory of the STB-5 disk as* `cansim.dta`*—Ed.]* (this ado-file does not exactly match the results printed in the article, but I think the difference is just due to a typo in the printed data set); and Godfrey, McAleer, and McKenzie (1988). Two texts with good discussions are Greene (1990, 340–343) and Maddala (1988, 180). This test amounts to: (1) obtain the predicted values from the two regressions; (2) form the variables (prediction − transform of other predicted); (3) estimate each regression as before but include the relevant new variable from step 2 (i.e., include "prediction from log equation minus log of prediction from raw equation" in raw model, and vice versa for log model); (4) examine t-tests for this new right-hand-side variable: if t-test is statistically significant then you can reject that model as being insufficient (because you can improve it). The problem is that both models might be either significant or not significant leaving you with an unsolved problem.

The Godfrey et al., article (1988) compares a number of tests and finds that the PE test, included here, and the Ramsey RESET test, included in STB-2 (Goldstein 1991) as `ramsey` are among the best tests even when assumptions are violated.

There are other worthwhile things to do, at least two of which are possible in Stata. First, and very easy in Stata, is a graph showing both the transformed and untransformed dependent variable on one graph, with one y-axis in the untransformed scale and the other in the transformed scale. Two examples, one of made-up data and one of real data, show this. The other procedure requires the use of the bootstrap. *[Stata's ado-file for this—*`boot`*—is included in the* `\crc` *directory of the STB-5 disk—Ed.]* Use of the bootstrap to help choose between non-nested models is discussed in Efron (1984).

The `logdummy.ado` file (see *srd8*) canNOT be used at the end of a run using this file since the last regression actually estimated by this ado-file is for the `boxcoxg` run. Thus, to use `logdummy`, you must actually re-estimate the log-transformed regression. (See `logdummy.hlp`.)

Example using `nwk.dta` (Neter, Wasserman, and Kutner 1989, 150):

```
. use nwk
. logsumm plasma age
  Source |     SS         df       MS           Number of obs =      25
---------+------------------------------        F(  1,    23) =   70.21
   Model |  238.056198     1  238.056198        Prob > F      =  0.0000
Residual |  77.9830691    23  3.39056822        R-square      =  0.7532
---------+------------------------------        Adj R-square  =  0.7425
   Total |  316.039267    24  13.1683028        Root MSE      =  1.8413

Variable |  Coefficient    Std. Error       t    Prob > |t|        Mean
---------+--------------------------------------------------------------
  plasma |                                                        9.1112
---------+--------------------------------------------------------------
     age |       -2.182      .2604062    -8.379     0.000             2
   _cons |      13.4752      .6378622    21.126     0.000             1
---------+--------------------------------------------------------------
Durbin Watson Statistic = 1.6413435
```

```
  Source |       SS       df       MS          Number of obs =      25
---------+------------------------------       F(  1,    23) =  134.02
   Model |  2.77338628     1  2.77338628        Prob > F      =  0.0000
Residual |  .475948075    23  .020693395        R-square      =  0.8535
---------+------------------------------       Adj R-square  =  0.8472
   Total |  3.24933435    24  .135388931        Root MSE      =  .14385

Variable |  Coefficient    Std. Error       t    Prob > |t|        Mean
---------+--------------------------------------------------------------
 logdepv |                                                      2.141985
---------+--------------------------------------------------------------
     age |    -.2355159      .0203437   -11.577     0.000             2
   _cons |     2.613017      .0498318    52.437     0.000             1
---------+--------------------------------------------------------------
Durbin Watson Statistic = 1.7528526


Following are some summary statistics for each of the above two models.
3 of the 5 sets of statistics are ´adjusted´, the other two just repeat
what was shown above for ease of comparison.

The first column shows the unadjusted statistics for the linear model,
just as shown in the first regression above; the second column shows
summary statistics for the same model but this time adjusted by
transforming to logs; the third column repeats the unadjusted figures
from the transformed regression (the second regression above); this is
followed by two sets of adjusted statistics: (1) a less biased
re-transformation than the standard one (see the help file or the
STB article); (2) using the ´standard´, biased, re-transformation
by just exponentiating the predicted values from the log model.

             |             Adjusted               Better    Standard
             |    Raw        Raw        Log     Adj´d Log  Adj´d Log
-------------------------------------------------------------------
R-Square     |   0.7532     0.7981     0.8535     0.7911     0.7945
Adjusted R-SQ|   0.7425     0.7893     0.8472     0.7820     0.7856
F-Value      |  70.21      90.93     134.02      87.09      88.93
RMSE         |   1.8413     0.1689     0.1439     1.6944     1.7037
CV (*100)    |  20.21       7.88       6.72      18.59      20.00

Results of the MacKinnon-Davidson (PE) test:
The t-statistic (p-value) for test of linearity is      2.068   0.050
The t-statistic (p-value) for test of log-linearity is -1.114   0.277

Note that it is quite possible that BOTH the above tests might be
significant (non-significant)!!
This means that this test is indeterminate for this model;
in this case, the use of boxcoxg.ado may be particularly helpful;
regardless, you might also want to use ramsey.ado (STB-2).
If only one test is significant, then we reject the functional
form for which the test is significant and ´accept´ the other form.

Following is a crude look using boxcoxg; if this appears to be informative,
you might want to use boxcoxg again with a finer grid; see boxcoxg.hlp


lambda          SSE     Log-likelihood
-3.00         132.62       -61.0932
-2.50          89.93       -56.2381
-2.00          61.86       -51.5602
-1.50          44.01       -47.3064
-1.00          33.91       -44.0460
-0.50          30.56       -42.7460
 0.00          34.52       -44.2690
 0.50          48.37       -48.4862
 1.00          77.98       -54.4561
 1.50         135.17       -61.3314
 2.00         243.05       -68.6657
 2.50         446.86       -76.2781
 3.00         835.78       -84.1046
```

A number of variables are kept, but not saved in your data file. Here is the data after the above estimation, with automatic variable labels. You may want to use some of these; for example, comparing quantile graphs of the two different sets of residuals can be informative.

```
. describe, detail
Contains data from nwk.dta
  Obs:    25 (max= 28324)                 Neter, et al., 1989, p. 150
 Vars:    21 (max=   254)
Width:   108 (max=   510)
  1. age          float  %9.0g
  2. plasma       float  %9.0g
  3. logdepv      float  %9.0g             Log of Original D.V.
  4. yhatr        float  %9.0g             Pred. Values/Untransformed Reg.
  5. yhatl        float  %9.0g             Log of Pred. Values/Untransform
  6. _resr        double %10.0g           Residuals/Untransformed Reg.
  7. _DWr         double %10.0g           D-W/raw regression
  8. _SSEr        float  %9.0g             Log transformed SSE
  9. _SSTr        float  %9.0g             Log transformed SST
 10. yhat         double %10.0g           Pred.Values/Transformed Reg, Lo
 11. _res         double %10.0g           Residuals/Transformed Reg, Logs
 12. stdf         double %10.0g           Forecast Err/Transformed Reg, L
 13. yhata        float  %9.0g             Retransformed, Adj., Pred. Valu
 14. yhate        float  %9.0g             Retransformed, UNadj., Pred. Va
 15. _SSEa        float  %9.0g             Retransformed, Adjusted, SSE
 16. _SSEe        float  %9.0g             Retransformed, UNadjusted, SSE
 17. _SSTa        float  %9.0g             Retransformed, Adjusted, SST
 18. _SSTe        float  %9.0g             Retransformed, UNadjusted, SST
 19. _DW          double %10.0g           D-W from Transformed Reg.
 20. lidiff       float  %9.0g             Difference between Raw and Re-t
 21. lodiff       float  %9.0g             Difference between Log and Logg
Sorted by:
Note:  Data has changed since last save
```

Note that use of this ado-file does *not* match the results presented in Kvålseth's article (Kvålseth 1985); however, given that he has some strange definitions (e.g., RMSE is NOT the square root of MSE), I am not bothered.

References

Davidson, R. and J. G. MacKinnon. 1985. Testing linear and loglinear regressions against Box–Cox alternatives. *Canadian Journal of Economics* 18: 499–517.

Efron. B. 1984. Comparing non-nested linear models. *Journal of the American Statistical Association* 79: 791–803.

Godfrey, L. G., M. McAleer, and C. R. McKenzie. 1988. Variable addition and LaGrange multiplier tests for linear and logarithmic regression models. *The Review of Economics and Statistics* 70: 492–503.

Goldstein, R. 1991. srd5: Ramsey test for heteroscedasticity and omitted variables. *Stata Technical Bulletin* 2: 27.

Granger, C. W. J. 1989. *Forecasting in Business and Economics.* 2d ed. Boston: Academic Press.

Greene, W. H. 1990. *Econometric Analysis.* New York: Macmillan Publishing Company.

Kvålseth, T. O. 1985. Cautionary note about R^2. *The American Statistician* 39: 279–285.

Maddala, G. S. 1988. *Introduction to Econometrics.* New York: Macmillan Publishing Company.

Miller, D. M. 1984. Reducing transformation bias in curve fitting. *The American Statistician* 38: 124–126.

Neter, J., W. Wasserman, and M. H. Kutner. 1989. *Applied Linear Regression Models.* 2d ed. Homewood, IL: Richard D. Irwin.

Scott, A. and C. Wild. 1991. Transformations and R^2. *The American Statistician* 45: 127–129.

srd8 | Interpretations of dummy variables in regressions with log dependent variable

Richard Goldstein, Qualitas, Brighton, MA, EMAIL goldst@harvarda.bitnet

The syntax for `logdummy` is

`logdummy` *varlist*

No options are allowed or needed.

When the dependent variable in a linear regression is a logarithmic transform, the interpretation of the right-hand-side variables is that they show the percentage change in the untransformed dependent variable per one-unit change in the right-hand-side variable, if the right-hand-side variable is in original units. If the variable has been transformed also, by taking logs, then its coefficient is interpreted as the percentage change in the untransformed dependent variable for a one percent change in the untransformed right-hand-side variable. This is fine for continuous variables, but is biased for dummy (categorical) variables (this provides the estimated median of the distribution rather than the mean). `logdummy` gives a simple change for dummy variables that is consistent and, though still biased, is very close to the unbiased result and is much easier to compute. For discussion, see the references below.

References

Giles, D. E. A. 1982. The interpretation of dummy variables in semilogarithmic equations. *Economic Letters* 10: 77–79.

Halvorsen, R. and R. Palmquist. 1980. The interpretation of dummy variables in semilogarithmic equations. *American Economic Review* 70: 474–475.

Kennedy, P. E. 1981. Estimation with correctly interpreted dummy variables in semilogarithmic equations. *American Economic Review* 71: 801.

srd9 Box–Cox statistics for help in choosing transformations

Richard Goldstein, Qualitas, Brighton, MA, EMAIL goldst@harvarda.bitnet

[It merely happened that Goldstein and Royston separately submitted inserts on the Box–Cox transform, so also see srd10. I have placed Goldstein first because he provides a more thorough explanation of the transform and its uses. Goldstein and Royston solve the problem differently; Goldstein provides a way to search powers while Royston provides the maximum-likelihood solution. Appropriately using either method should produce the same results.—Ed.]

The syntax for the `boxcoxg` command is

`boxcoxg` *lsl ul ss varlist* [`if` *exp*] [`in` *range*]

where *lsl* is the lower-search limit, *ul* is the upper-limit, and *ss* is the step-size.

In many cases, we cannot be sure that our regression model should include the dependent variable in its original form—we may want to, or need to, transform it. While it is sometimes possible to determine a transformation based on theoretical considerations, usually this is not the case. Many other methods have been used over the years, including plots of residuals from a regression, and plots of the raw data. For an excellent description of this last technique, now not very useful, see Hoerl (1954). Also, some people have used programs such as `ladder.ado` (see *sed2* in STB-2).

The problem with these techniques is that they are either too subjective (graphs) or they examine the wrong problem—what is the relation between the non-normality of a variable examined in isolation and the need to transform a variable when examined with other variables?

In a famous paper, Box and Cox (1964) suggested a numerical procedure for choosing a transformation of the dependent variable in a linear model (regression, anova). This paper limits the choice to the "power family," as follows:

$$y^{\lambda} = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(y) & \text{if } \lambda = 0 \end{cases}$$

This includes many types of transformations, but certainly not all that might be useful; further, this family is not useful when the dependent variable is a proportion. Of course, it is obvious that this family cannot be used when negative numbers are possible. Finally, note that this procedure is specifically aimed at transformations of the dependent variable only.

Other transformation families have been suggested and are discussed in Atkinson (1987, esp. chapter 7), and Draper and Smith (1981, 236–241). Transformations of variables on the right are discussed in Box and Tidwell (1962). Transformations of both sides simultaneously are discussed in Carroll and Ruppert (1988).

Within these limits, however, this seems to be a very useful procedure. The problem is that, as presented in the original paper, it is hard to compute. A number of people, however, have discovered computational shortcuts. These shortcuts do not necessarily give the same numerical answer as in the original paper, but they do appear to give the same qualitative answer; that is, the "solution" regarding what is the best transformation parameter (λ) is the same. Such shortcut formulas can be found in Atkinson (1987, 87; `boxcox2.dta` is Atkinson's example—see below); Draper and Smith (1981, 226 and example on page 228; data from this example appears in `drapsmth.dta`—see below); Maddala (1988, 178–179; no example provided); Neter, Wasserman, and Kutner (1989, 150; example on page 150 is `nwk.dta`—see below); Weisberg (1985, 148; example is provided as `weisberg.dta`—see below).

Weisberg also provides a formula for computing the log-likelihood—his are the ones I have used—and provides a formula for a confidence interval for λ (p. 151). His results are matched to the precision shown, as is the `nwk` example (see below). The conclusion is the same for the other examples, but the numbers are not the same. This is also true regarding other software: Dallal's ODDJOB (version 6.03), SHAZAM (6.2) and LIMDEP (6.0); note that both SHAZAM and LIMDEP use full MLE and their log-likelihoods agree to four decimal places.

The reason for the various transformations is to make the residual sum of squares and the log-likelihood comparable across transformations; they would not be without some form of transformation of the results or the data.

Note that sometimes the result of `boxcoxg` will be a missing value—this means that this particular transformation is not possible with the data provided.

`boxcoxg` can be used to search for the best transformation or to examine just one particular transformation. In either case, you *must* enter three numbers prior to entering the variable list. The first number is the minimum value of λ that you want to search over, the second is the maximum, and the third number is the step size. If you just want to see the results for one particular value of λ, enter the same number for the minimum and the maximum and enter any positive value for the step size (*do not* enter a step size of 0). If you want to search over the space 0 (log transform) to 1 (no transform) with a step size of .1, enter '`boxcoxg 0 1 .1` *varlist*'. Negative numbers are allowed for either the minimum or the maximum.

The data set is automatically cleared when the ado-file is finished so you can immediately re-enter `boxcoxg` with a finer search pattern if you so desire. For example, your first search might be from -3 to 3 with a step size of .25; you might then search from one step smaller than the minimum found to one step larger with a much smaller step size (e.g., $-.5$ 0 .01). Do not attempt to use a step size smaller than .01 as this is not allowed.

The only output is a three-column list; the first column is λ, the transformation parameter, the second is the error sum of squares, and the third is the log-likelihood. You choose the smallest SSE (largest log-likelihood) as the best transformation, or something close to this that makes substantive sense. The code could be easily modified to allow likelihood-ratio tests, using the CRC-provided ado-file `lrtest` (see *crc6* in STB-1) if the user cared to do so. The code could also be easily modified so that the results, either SSE or log-likelihood, could be graphed if so desired.

Examples

```
. use boxcox1
(Box & Cox, 1964, p. 220)
. boxcoxg -3 1 .2 survive treat poison          Results in article (p. 221)
lambda          SSE     Log-likelihood        SSE          Log-Likelihood
-3.00           3.61       -30.7812          2.0489            75.69
-2.80           2.96       -26.0329
-2.60           2.45       -21.5120
-2.40           2.05       -17.2496
-2.20           1.74       -13.2823
-2.00           1.50        -9.6518          0.6639           102.74
-1.80           1.31        -6.4038
-1.60           1.16        -3.5872          0.4625           111.43
-1.40           1.05        -1.2509          0.4007           114.86
-1.20           0.98         0.5594          0.3586           117.52
-1.00           0.93         1.8053          0.3331           119.29
-0.80           0.90         2.4591          0.3225           120.07
-0.60           0.90         2.5073          0.3258           119.82
-0.40           0.92         1.9508          0.3431           118.58
-0.20           0.97         0.8051          0.3752           116.44
 0.00           1.04        -0.9034          0.4239           113.51
 0.20           1.14        -3.1402
 0.40           1.28        -5.8669
 0.60           1.46        -9.0442
 0.80           1.69       -12.6346
 1.00           2.00       -16.6035          1.0509            91.72
. use boxcox2, replace
(Box & Cox, 1964,p.223 (At., p.82))
. boxcoxg -1 1 .2 cycles x1 x2 x3               Results in article (p. 224)
lambda          SSE     Log-likelihood        SSE          Log-Likelihood
-1.00     3995487.00      -205.2091          3.9955            25.79
-0.80     2139587.82      -196.7777          2.1396            34.22
-0.60     1103484.66      -187.8388          1.1035            43.16
-0.40      547841.84      -178.3855          0.5478            52.61
-0.20      292011.00      -169.8914          0.2920            61.11
 0.00      251900.41      -167.8966          0.2519            63.10
 0.20      411520.34      -174.5228          0.4115            56.48
 0.40      817792.18      -183.7939          0.8178            47.21
 0.60     1596824.29      -192.8276          1.5968            38.17
 0.80     2997794.61      -201.3307          2.9978            29.67
 1.00     5480980.89      -209.4767          5.4810            21.52
. use drapsmth, replace
(Draper & Smith, 1981, p. 228)
```

```
. boxcoxg -1 1 .05 W f p                          Results in book (p. 229)
lambda          SSE     Log-likelihood        SSE          Log-Likelihood
-1.00        2455.65       -89.7707                          -53.7
-0.95        2168.15       -88.3387
-0.90        1906.45       -86.8595
-0.85        1668.55       -85.3267
-0.80        1452.66       -83.7333                          -47.8
-0.75        1257.15       -82.0710
-0.70        1080.57       -80.3303
-0.65         921.60       -78.5003
-0.60         779.09       -76.5684                          -40.52
-0.55         651.99       -74.5203
-0.50         539.40       -72.3402
-0.45         440.52       -70.0115
-0.40         354.68       -67.5191                          -31.46
-0.35         281.31       -64.8537
-0.30         219.92       -62.0227
-0.25         170.16       -59.0722
-0.20         131.73       -56.1287                          -20.07
-0.15         104.47       -53.4619                          -17.40
-0.10          88.28       -51.5254                          -15.47
-0.05          83.17       -50.8402                          -14.78
 0.00          89.25       -51.6518                          -15.60
 0.05         106.73       -53.7080                          -17.65
 0.10         135.90       -56.4869                          -20.43
 0.15         177.18       -59.5373
 0.20         231.09       -62.5922                          -26.53
 0.25         298.27       -65.5269
 0.30         379.48       -68.2963
 0.35         475.62       -70.8932
 0.40         587.74       -73.3273                          -37.27
 0.45         717.02       -75.6137
 0.50         864.84       -77.7692
 0.55        1032.74       -79.8096
 0.60        1222.46       -81.7492                          -45.69
 0.65        1436.00       -83.6006
 0.70        1675.55       -85.3748
 0.75        1943.59       -87.0814
 0.80        2242.91       -88.7286                          -52.67
 0.85        2576.61       -90.3236
 0.90        2948.14       -91.8727
 0.95        3361.37       -93.3812
 1.00        3820.60       -94.8539                          -58.80

. use nwk, replace
(Neter, et al., 1989, p. 149)

. boxcoxg -1 1 .1 plasma age                      Results in book (p. 150)
lambda          SSE     Log-likelihood        SSE          Log-Likelihood
-1.00          33.91       -44.0460          33.9
-0.90          32.70       -43.5939          32.7
-0.80          31.76       -43.2294
-0.70          31.09       -42.9613          31.1
-0.60          30.69       -42.7979          30.7
-0.50          30.56       -42.7460          30.6
-0.40          30.72       -42.8109          30.7
-0.30          31.18       -42.9957          31.2
-0.20          31.95       -43.3016
-0.10          33.06       -43.7272          33.1
 0.00          34.52       -44.2690          34.5
 0.10          36.37       -44.9216          36.4
 0.20          38.64       -45.6779
 0.30          41.36       -46.5300          41.4
 0.40          44.59       -47.4690
 0.50          48.37       -48.4862          48.4
 0.60          52.76       -49.5727
 0.70          57.84       -50.7203          57.8
 0.80          63.67       -51.9213
 0.90          70.35       -53.1686          70.4
 1.00          77.98       -54.4561          78.0

. use weisberg, replace
(Weisberg, 1985, p. 149)
```

```
. boxcoxg -2 2 .5 area peri                    Results in book (p. 150)
lambda          SSE     Log-likelihood        SSE          Log-Likelihood
-2.00       90372.63       -142.6462
-1.50       19289.59       -123.3415
-1.00        4708.16       -105.7132
-0.50        1301.88        -89.6446
 0.00         377.30        -74.1632          377.3043        -74.16
 0.50         116.26        -59.4482          116.2636        -59.45
 1.00         217.96        -67.3039
 1.50        1168.32        -88.2915
 2.00        5493.07       -107.6405
```

References

Atkinson, A. C. 1987. *Plots, Transformations, and Regression* Oxford: Oxford University Press.

Box, G. E. P. and D. R. Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society*, Series B, 26: 211–252 (with discussion).

Box, G. E. P. and P. W. Tidwell. 1962. Transformations of the independent variables. *Technometrics* 4: 531–550.

Carroll, R. J. and D. Ruppert. 1988. *Transformations and Weighting in Regression.* London: Chapman and Hall.

Draper, N. and H. Smith. 1981. *Applied Regression Analysis.* 2d ed. New York: John Wiley & Sons.

Hoerl, A. E. Jr. 1954. Fitting curves to data. In *Chemical Business Handbook*, ed. J. H. Perry, 2055 to 2077. New York: McGraw–Hill Book Company.

LIMDEP software. Econometric Software, Inc., 43 Maple Ave., Bellport, N.Y. 11713. Tel. 516-286-7049. $350.

Maddala, G. S. 1988. *Introduction to Econometrics* New York: Macmillan.

Neter, J., W. Wasserman, and M. H. Kutner. 1989. *Applied Linear Regression Models.* 2d ed. Homewood, IL: Richard D. Irwin.

ODDJOB stand-alone FORTRAN program. G. Dallal, 53 Beltran St., Maldel, MA 02148. $10.

SHAZAM software. Department of Economics, University of British Columbia, Vancouver, British Columbia V6T 1Z1, Canada. Tel. 604-228-5062. $295.

Weisberg, S. 1985. *Applied Linear Regression.* 2d ed. New York: John Wiley & Sons.

srd10 | Maximum-likelihood estimation for Box–Cox power transformation

Patrick Royston, Royal Postgraduate Medical School, London, FAX (011)-44-81-740 3119

The syntax of `boxcoxr` is

`boxcoxr` *yvar* [*xvar(s)*] [`in` *range*] [`if` *exp*] [`, delta(#) zero(#) ci(#) iter(#)`
`lstart(#) ropt(`*regression_options*`) gen(`*newvar*`) anova symm quiet detail`]

The Box–Cox transform,

$$y^{(\lambda)} = \frac{y^{\lambda} - 1}{\lambda}$$

represents a family of popular transformations used in data analysis. For instance:

$$y = \begin{cases} y - 1 & \text{if } \lambda = 1 \\ \ln(y) & \text{if } \lambda = 0 \\ 1 - 1/y & \text{if } \lambda = -1 \end{cases}$$

The value of λ may be calculated from the data. `boxcoxr` finds the maximum-likelihood value of λ for the model

$$y_i^{(\lambda)} = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_k x_{ki} + \epsilon_i$$

where x_{1i}, ..., x_{ki} are the *xvars* in the syntax diagram, if any, and ϵ_i is assumed to be normally distributed and homoscedastic. Thus, typing '`boxcoxr` y' finds the Box–Cox transform for y; typing '`boxcoxr` y x' finds the Box–Cox transform for y in the model $y_i^{(\lambda)} = \beta_0 + \beta_1 x_i$.

The `ci(#)` option requests that a # confidence interval be calculated for λ; `ci(.95)` produces a 95% percent confidence interval. The default is to not calculate confidence intervals. Confidence intervals are calculated from the log-likelihood function and should strictly be called "support intervals."

The `gen(`*newvar*`)` option creates *newvar* containing the transformed values of *yvar*.

The `symm` option produces the mean-symmetry version of the Box–Cox transform:

$$\text{sign}(y-\bar{y})\frac{(|y-\bar{y}|+1)^{\lambda}-1}{\lambda}$$

This version can be used for *yvars* that are negative or have no natural meaning to the value recorded as zero.

`quiet` suppresses output. `detail` reports progress on the convergence of the iterative process as it happens and gives a plot of the log-likelihood function against λ (the "profile likelihood").

The remaining options control technical features of the program. `delta(#)` specifies a small increment for calculating derivatives of the log-likelihood function and is by default 0.01. `zero(#)` specifies a value for the derivative of the log-likelihood which is regarded as small enough to be considered zero to determine convergence and is by default 0.001. `iter(#)` is the maximum number of iterations to be permitted and defaults to 10. `lstart(#)` forces a specific starting value for λ; the default is 1.

Individual values of the log-likelihood function can be obtained by specifying `zero(0)` and the `lstart()` option. In this case, `boxcoxr` does not continue to find the maximum-likelihood value of λ, but simply reports the value of the log-likelihood function corresponding to `lstart()`.

If *xvar*(*s*) are specified, *yvar* is regressed on *xvars* with *regression_options* defined in `ropt()`. If `anova` is specified, analysis-of-variance rather than regression is done.

The program is iterative and is not guaranteed to converge. Convergence may be achieved in difficult cases by varying (typically increasing) the value of `delta` and/or by trying different starting values for λ. A cruder approximate maximum-likelihood estimate may be obtained by increasing the value of zero. Alternatively, the program may be used "manually" by specifying single values of λ and inspecting the log-likelihood values which result. The plot previously obtained when `detail` was specified may help here.

```
. boxcoxr mpg, ci(.95)
 Variable |    Obs      Lambda 95% confidence interval   LL(raw) LL(x^lambda)
 ---------+-------------------------------------------------------------------
      mpg |     74      -0.459    -1.210      0.287     -132.759     -125.303
. boxcoxr mpg price foreign, ci(.95) gen(xmpg)
 Variable |    Obs      Lambda 95% confidence interval   LL(raw) LL(x^lambda)
 ---------+-------------------------------------------------------------------
      mpg |     74      -0.879    -1.572     -0.179     -116.326     -102.608
```

In the last case, `xmpg` is created equal to the transformed values of `mpg`. We could not obtain the regression by typing '`regress xmpg price foreign`'.

References

Atkinson, A. C. 1987. *Plots, Transformations, and Regression* Oxford: Oxford University Press.

Box, G. E. P. and D. R. Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society*, Series B, 26: 211–252 (with discussion).

ssa2 Tabulating survival statistics

Wim L. J. van Putten, Daniel den Hoed Cancer Center, Rotterdam, The Netherlands

`survtab` *timevar deadvar* [`if` *exp*] [`in` *range*] [, `by(`*groupvar*`) at(`*t1*[,*t2*[,*t3*...]]`)`]

generates a survival table in a tabular format. Stata output in log-files generally forms the basis of a statistical report. It only takes a general text editor to delete or modify parts and add comments, chapters and discussions to the log file. In order to minimize the amount of editing, it is useful if the Stata output can be restricted to what is really needed. `survtab.ado` helps in this. `survtab.ado` is based on `survcurv.ado`, and makes use of `_crcsrvc.ado`, but with a slight modification to this program. The new version is called `_crcsrv2.ado`. The modification consists of the addition of a new variable `_atrisk`, the number at risk at time `t#`.

Example

```
. d surv dood hazrt
 45. surv         int    %5.0f             Survival time in days
 64. dood         int    %8.0g             Death indicator
 76. hazrt        float  %9.0g    hazrt    Risk group * RT
```

```
. tab hazrt dood
Risk group| Death indicator
       * RT|         0          1 |     Total
-----------+----------------------+----------
  Low  RT- |        61         13 |        74
  Low  RT+ |        72         15 |        87
  Med  RT- |        16         10 |        26
  Med  RT+ |        75         19 |        94
  High RT- |         8         11 |        19
  High RT+ |        34         25 |        59
-----------+----------------------+----------
      Total|       266         93 |       359

. survtab surv dood ,by(hazrt ) at(1827,3653)
        hazrt     _TIME   _atrisk     _surv     _stds
 31. Low  RT-      1827        43     0.857     0.042
 57. Low  RT-      3653        17     0.767     0.064
103. Low  RT+      1827        58     0.880     0.036
144. Low  RT+      3653        17     0.771     0.056
174. Med  RT-      1827        13     0.692     0.091
183. Med  RT-      3653         4     0.639     0.098
224. Med  RT+      1827        57     0.834     0.039
268. Med  RT+      3653        13     0.788     0.049
293. High RT-      1827         7     0.461     0.118
298. High RT-      3653         2     0.395     0.118
329. High RT+      1827        30     0.643     0.062
352. High RT+      3653         7     0.508     0.081
422.         .     1827        96     0.768     0.035
479.         .     3653        39     0.624     0.046
```

The column `hazrt` shows the classes of the *groupvar*; the column `_TIME` shows the time in days (1827=5 years, 3653=10 years.); the column `_atrisk` shows the number still at risk just after `_TIME`; the column `_surv` shows the survival probability at `_TIME` for the value of `hazrt`; and the column `_stds` shows the corresponding standard deviation.

sts1 | Autocorrelation and partial autocorrelation graphs

Sean Becketti, Federal Reserve Bank of Kansas City

The syntax diagrams for the commands are

`ac` *varname* [`if` *exp*] [`in` *range*] [`, nlags(#)`]

`pac` *varname* [`if` *exp*] [`in` *range*] [`, nlags(#)` [`no`]`constant`]

The Box–Jenkins approach to time-series models relies heavily on examining graphs of autocorrelations and partial autocorrelations to check for deviations from stationarity (diagnostics) and to infer an appropriate parameterization (identification). `ac` and `pac` produce these graphs along with standard-error bands. By default, the first twenty autocorrelations or partial autocorrelations are graphed, but this can be overridden. The standard error of the autocorrelations is estimated by Bartlett's approximation. The standard error of the partial autocorrelations is approximated by $1/\sqrt{n}$ where `n` is the number of observations.

The partial autocorrelations are estimated from a sequence of `nlag()` regressions. If `noconstant` is specified, the regressions will be estimated without a constant; otherwise, a constant will be included in the regressions.

Examples

To examine the autocorrelations and partial autocorrelations of real GNP, we could

```
. use gnp
. describe
Contains data from gnp.dta
  Obs:   178 (max= 32252)
 Vars:     3 (max=    99)
  1. year         int   %8.0g        Year
  2. quarter      int   %8.0g        Quarter
  3. gnp82        float %9.0g        Real GNP (1982 dollars)
Sorted by:  year  quarter
. generate lgnp82 = log(gnp82)
. label variable lgnp82 "Log of real GNP"
```

```
. gen dlgnp82 = lgnp82 - lgnp82[_n-1]
(1 missing value generated)
. ac lgnp82
```

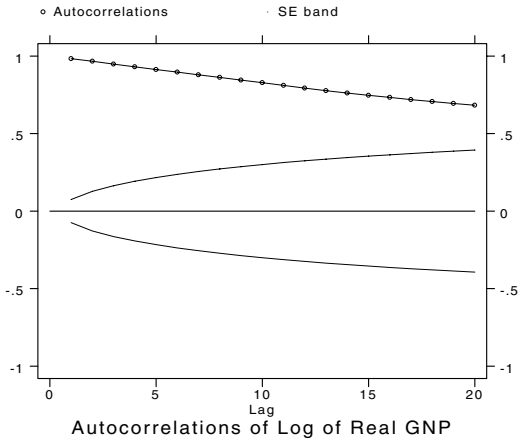

Figure 1

The autocorrelation plot shows clearly that the log level of real GNP is nonstationary.

```
. ac dlgnp82
. pac dlgnp82
```

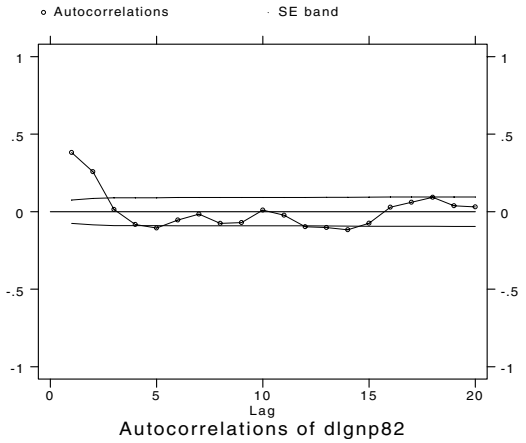

Figure 2

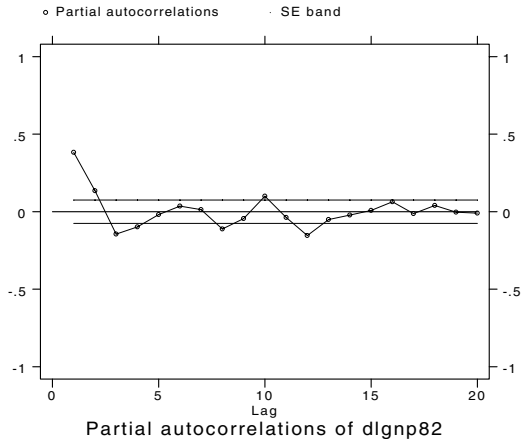

Figure 3

These graphs suggest that the growth rate (log difference) of real GNP is stationary.

References

Box, G. E. P., and G. M. Jenkins. 1976. *Time Series Analysis: forecasting and control.* revised ed. Oakland, CA: Holden–Day.