RATS 7.3 Supplement

Contents

1. Overview

This document describes improvements and new features added to RATS since the Version 7 *User's Guide* and *Reference Manual* were produced. These include features added in Versions 7.1, 7.2 and 7.3, and the **DSGE** instruction and Graph Style Sheets feature, which were included in Version 7.0, but were added after the Version 7 manuals were finalized. This first section provides a quick look at the key improvements in each release.

1.1 Version 7.3

Faster Computations

Thanks to further optimization efforts, RATS 7.3 does many computations even faster than Version 7.2, which boasted significant speed gains over previous versions (see page 4).

Data Wizard: Major Improvements

The *Data Wizard* is much improved. It now allows you to preview the contents of the data file, making it easier to set various options. Also, you can now set a target frequency and starting date that differ from those of the source file when you only want to read in a subset of the data, or want to compact or expand the data to a different frequency. Previously, this required setting the **CALENDAR** instruction manually—now you can do it directly through the Wizard.

Wizard for Recursive Least Squares

The *Statistics* menu now includes a wizard for doing recursive least squares.

Support for More Data File Formats, including Excel 2007

RATS 7.3 supports reading data from Excel® 2007 "xlsx" format spreadsheets, as well as Stata® data files and EViews® workbook files.

DISPLAY Instruction

DISPLAY now has an option for delimiting output with tabs, commas, or semicolons, in addition to the default behavior of separating terms with blank spaces. Also, you can now use **DISPLAY** to show the contents of a parameter set or an equation.

DLM Instruction

The `PRESAMPLE` option has a new choice (`DIFFUSE`) and the `ERGODIC` choice can now automatically handle models with a mix of stationary and non-stationary states. There are two new options: `MU`, for handling a shift in the observable equation, and `LIMIT`, allows you to limit the number of times certain matrices are recomputed, which can significantly reduce the computation time required for large, time-invariant state space models.

NPREG and DENSITY

Both now have SMOOTHING options which scale the default kernel width up or down.

PREG Instruction

Now offers pooled panel regression, via the METHOD=POOLED option.

Graphs

You can now include line breaks in headers, subheaders, and horizontal axis labels, by including the characters "\\" in the string at the point where you want the line break. This works on the HEADER, SUBHEADER, HLABEL, VLABEL, and FOOTER options on all of the relevant graphing instructions.

Also, **SPGRAPH**, **GRAPH**, **SCATTER**, **GCONTOUR** and **GBOX** now have ROW and COL options for positioning in arrays of graphs (i.e., **SPGRAPH**s) rather than using the parameters.

New Functions

RATS 7.3 includes several new built-in functions, mostly related to random draws and distributions. See the "New Functions" section (page 18) for details.

Reports

The new *Windows–Report Windows* operation (replacing the old *Restore Reports*) now applies to many more types of output, and makes it easier to pull up the desired report. See "Reports (7.3)" on page 6 for other related improvements.

Comment Blocks

Previously, the */ code indicating the end of a block of comments had to appear on a separate line. You can now place the */ anywhere on a line. For example:

```
/* This is a comment */
```

1.2 Version 7.2

Significant Speed Improvements (Windows Version)

Due to improved compiler optimization, the Windows version of RATS now does many computations nearly *twice* as fast as before. This can save you a great deal of time when estimating complex models or doing large simulation routines.

FRED Database Access (Pro Version Only)

The Professional version of RATS provides point-and-click access to the FRED® (Federal Reserve Economic Data) database provided by the St. Louis Federal Reserve Bank.

To access the database, just select the *FRED Browser* operation from the *Data* menu (requires active internet connection). This displays a list of the main database categories in a new window. Double-click on a category (or sub-category) to see a list of the available series in that category. While you can double-click on one of these series to view the data, or use the toolbar icons to generate graphs or compute statistics, we recommend that you download the series to your own computer first.

To do that, open a RATS format file using *New RATSData* or *Open RATSData* on the *File* menu, and then drag-and-drop the series you want from the FRED browser to the RATS format file. You can view, edit, or graph the data from the RATS file window, or use the RATS format *Data Wizard* to read the data into memory.

See **research.stlouisfed.org/fred2/** for more information on the database.

Census ARIMA-X12 Features (Pro Version Only)

Extensions to the **X11** and **BOXJENK** instructions in the Pro version allow you to implement most features of the Census Bureau's X12-ARIMA seasonal adjustment technique. See the section on **BOXJENK** later in this document and the expanded *X11 Supplement* PDF file for details.

New "REGArima" Options on BOXJENK

New options on **BOXJENK** allow estimating models with a primary focus on the regression model, rather than the time series properties of the residuals. See the **BOXJENK** section of this *Supplement* for details.

Date Labels on Excel Files

The new DATEFORMAT option on **DATA** and **STORE** allows RATS to process date labels in formats other than our standard "yyyy:m:d" or "yyyy:period" formats.

New Wizards

Version 7.2 added menu-driven Wizards for doing Unit Root Tests, Nonparametric Regressions, and Density Estimations.

1.3 Version 7.1

New Instruction

Version 7.1 added the **GBOX** instruction for drawing box (high-low-close) plots.

Interface Improvements

Context (right-click) menus have been added for most window types. A *View* menu was added, with operations for viewing data, and for menu-driven access to operations previously available only through toolbar buttons.

Improved SQL Support

The new `QUERY` option on **DATA** allows for handling longer SQL queries.

Other Improvements

Version 7.1 also included new options on various instructions, several new reserved variables, and expanding the Graph Style Sheet feature to support controlling the style used for the `SHADE` option on graphing instructions.

2. New Interface Features

The following improvements have been made to the RATS interface since the release of Version 7.0. The number in parentheses indicates the version in which a given feature first appeared.

New Wizards (7.2, 7.3)

Data Menu: FRED Browser (7.2)
Provides access to the FRED economic database. See page 4 for details.

Statistics Menu: Unit Root Test (7.2)
Provides access to seven different unit root testing procedures. You can use fields in the dialog box to filter the available choices based on various criteria including the null hypothesis used, the type of test, and whether or not the procedure allows for structural breaks.

Statistics Menu: Recursive Least Squares (7.3)
Provides an interface to the **RLS** instruction for recursive least squares.

Statistics Menu: Nonparametric Regression (7.2)
Provides an interface to the **NPREG** instruction for non-parametric regressions.

Statistics Menu: (Kernel) Density Estimation (7.2)
Provides an interface to the **DENSITY** instruction for estimating the density function of a series.

Reports (7.3)

The *Report Windows* operation replaces the old *Restore Report*. The new operation applies to many more of the reports generated automatically by RATS, including the tables of regression coefficients generated by estimation instructions. Also, rather than restoring reports in the order they were generated, you can now select the desired report from a list.

You can also now export the contents of a report window as a TeX table, in addition to the formats supported previously, including text formats, spreadsheets, HTML, and more. Finally, the new TITLE option on **REPORT** allows you to provide your own titles for user-generated reports.

File—Preferences Operation (7.2)

Fields for setting the directories for Haver Analytics, CRSP, and Global Insight (Citibase) data files have been moved to a new "Data Sources" tab. This tab also includes a field for the "key" used to access the FRED® database (provided by the St. Louis Federal Reserve bank) via an internet connection. You should not change or delete this key code unless the St. Louis Federal Reserve changes the access keys at some point.

Contextual Pop-Up Menus (7.1)

Beginning with Version 7.1, right-clicking (Ctrl+clicking on the Macintosh) on a window or object in RATS will display a pop-up menu with operations that can be applied to that window or object. You can then simply click on the operation from the pop-up menu.

For example, clicking on a block of selected text displays a pop-up menu with the same operations available via the *Edit* menu, such as *Cut* and *Copy*. Clicking on a graph window displays a menu you can use to copy or print the graph, export the graph to a file, or switch between black and white or color representations.

View Menu (7.1)

RATS 7.1 added a new *View* menu, providing menu-driven access to operations previously only available via toolbar icons, as well as options for displaying lists of all symbols (variables) and all series in memory—operations which are also available on the *Wizards* and *Data* menus, respectively.

For example, if you use *View–Series Window* to display a list of series and select (highlight) one or more of the series in the window, you can use other operations on the *View* menu to generate several types of graphs, compute summary statistics, and more—all operations that were previously only available via the toolbar icons.

3. New Features for RATS Format Files (7.2)

Version 7.2 included the following improvements for working with RATS format data files. Except as noted, these features are available both in RATSDATA and (after doing *File–Open RATSDATA* or *File–New RATSDATA*) in RATS.

Context (Pop-up) Menus

Context (or pop-up) menus are now available for RATS format data file windows. For example, right-clicking on a series (or a set of selected series) in a series list window displays a pop-up menu with operations that can be applied to the series, like *Cut*, *Copy*, *Paste*, and *Export*.

New View Menu (RATSDATA) and Reset List... Operation

The *View* menu is new to RATSDATA, where it provides operations for graphing series or displaying a table of statistics for the selected series. These are equivalent to the toolbar icons that have been present in RATSDATA for some time.

The *View* menu also includes a new *Reset List...* operation that gives you greater control over the series list display. As before, you can limit the display to only those series whose names match a template, or whose comments contain specific text. You can now also filter the list by frequency and/or by starting and ending year. You can now sort the list by comments or by frequency, as well as by name.

The old *File—List* operation is still available, and is now equivalent to *View—Reset List*.

Full Undo/Redo Capabilities

You can now Undo and Redo any editing operation. For example, if you delete a series, you can undo that deletion using the *Edit—Undo* operation. Similarly, any edits made to the values of a data series can be undone (and redone, if desired).

Find Operation

The new *Find* operation on the *Edit* menu allows you to search for a particular value in the selected series.

Improved "Form Panel" Operation

The *Form Panel* operation in RATSDATA stacks a set of selected series into a panel series. In version 7.2 and later, you can control how series with differing starting and ending dates are handled: RATSDATA can generate an "unbalanced" panel series with NA's (missing values) for observations that are not available in a given individual, or it can generate a "balanced" panel, using the maximum range of observations common to all individuals (omitting observations that are not present in all individuals).

4. Changes to Existing Instructions

This section lists new options and other improvements made to specific instructions since the release of Version 7.0. The number in parentheses indicates the version in which a given feature first appeared.

ROW, COL Options Added to Graphing Instructions (7.3)

GRAPH, **SCATTER**, **GCONTOUR**, **GBOX**, and **SPGRAPH**, now have `ROW` and `COL` options for positioning in arrays of graphs (defined by **SPGRAPH**), as an alternative to using the *hfield* and *vfield* parameters.

WEIGHT Option Added to Many Instructions (7.1)

The `WEIGHT` option, previously available on the **DENSITY** instruction, has been added to the **CMOM**, **DDV**, **ESTIMATE**, **LDV**, **LINREG**, **MAXIMIZE**, **MCOV**, **NLLS**, **NPREG**, **RATIO**, **RLS**, **RREG**, **STATISTICS**, **STWISE**, **SUR**, **SWEEP**, **TABLE**, and **VCV** instructions:

weight=*series of weights for the data points* **[equal weights]**

This can be used if the input data points aren't weighted equally, due, for instance, to oversampling or importance sampling. The weights do not have to sum to one—the rescaling will be done automatically.

STARTUP Option on SSET, GSET, CSET (7.1)

SET, **GSET**, and **CSET** now have `STARTUP` options, similar to the option of the same name on instructions like **MAXIMIZE** and **NLLS**. The syntax is as follows:

startup=*FRML evaluated at period "start"*

The `FRML` provides an expression which is computed only for the first entry of the range, before function(T) is computed. This can be a `FRML` of any type.

BOXJENK (7.2)

Version 7.2 adds several new options for "RegARIMA" modelling.

gls/[nogls]

This is an alternative to the `REGRESSORS` option. As with `REGRESSORS`, you supply a list of explanatory variables on a supplementary card. However, the emphasis is different. With `GLS`, it's the mean equation represented by the explanatory variables which is the focus of the estimation; the ARIMA model is a "noise" term. The output is switched around so the explanatory variables are listed first. `GLS` forces the use of maximum likelihood and also includes the behavior of the `APPLYDIFFERENCES` option.

outlier=[none]/ao/ls/tc/standard/all
critical=*critical (t-statistic) value* **[based on # of observations]**

With any of the choices for `OUTLIER` other than `NONE`, **BOXJENK** does an automatic procedure for detecting and removing outliers. This can be used with or without the `GLS` option. If used without `GLS`, it operates like `GLS` with an empty set of base regressors, that is, it estimates dummy shifts to the mean of the dependent variable, using maximum likelihood. `CRITICAL` allows you to set the *t*-statistic value used for the automatic outlier detection threshold.

`AO` locates additive outliers. For an outlier at entry *t0*, the resulting dummy would be 1 only at *t0*. `LS` detects level shifts, generating a dummy with 1's starting at *t0* through the end of the sample. `TC` detects temporary changes. For a temporary change starting at *t0*, the dummy takes the value 1 at *t0*, then declines exponentially for data points beyond that. `OUTLIER=AO`, `OUTLIER=LS` and `OUTLIER=TC` select scans for only the indicated type of outlier. `OUTLIER=STANDARD` scans for `AO` and `LS`, `OUTLIER=ALL` does all three.

The following procedure is repeated until no further outliers are detected. Beginning with the last RegARIMA model (including previously accepted outliers), LM tests are performed for each of the requested types of outliers at all data points. If the largest *t*-stat exceeds the critical value, that shift dummy is added to the model, which is then re-estimated.

When there are no further outliers to be added, the list is then pruned by examining the *t*-stats from the full estimation using the same critical value.

Note that the first step uses a "robust" estimate of the standard error of the residuals, based upon the median absolute value. There are several ways to compute maximum likelihood estimates; RATS uses Kalman filtering, Census X12-ARIMA uses optimal backcasting. The two lead to identical values for the likelihood function, identical values for the sum of squares of the residuals, but not to identical sets of estimated residuals. As a result, there can be slight differences between this robust estimate of the standard error. In some cases, they can be large enough to cause the two programs to differ on whether a marginal *t*-stat is above or below the limit. (X12-ARIMA tends to give a lower value for the standard error, and hence higher *t*-statistics). This tends to correct itself in the backwards pruning steps.

`adjust=`*`series of RegARIMA adjustments`* **`[not used]`**

This is a series which has the combined effects on the mean of all the regression coefficients, including input regressors and outliers, leaving out only the `CONSTANT` (if it's included in your original set of regressors). You can input this (or a transformation of it) into `X11` as a set of preliminary adjustment factors.

DATA

New choices for the `FORMAT` option:

`format=xlsx/dta/wf1` (7.3)

Version 7.3 adds three new choices for the `FORMAT` option for selecting the file format: `XLSX` for reading Excel® 2007 `.xlsx` format files, `DTA` for reading Stata® `.dta` format files, and `WF1` for reading EViews® `.wf1` workfiles.

New option:

`dateformat="`*`date format string`*`"` (7.2)

This can be used if dates on the file are text strings in a form other than year (delimiter) month (delimiter) day. In the date format string, use `y` for positions with the year, `m` for position with the month and `d` for positions with the day. Include the delimiters (if any) used on the file. Examples are `DATEFORMAT="mm/dd/yyyy"` and `DATEFORMAT="yyyymmdd"`.

Improved ODBC/SQL Support (7.1)

Previously, RATS could only handle a SQL query string of up to 255 characters when reading data using ODBC. With the `QUERY` option on **`DATA`**, you can supply much longer (virtually unlimited-length) SQL queries. You also have the option of reading the query from an external file. SQL support is also now available in the Mac version of RATS.

When using `FORMAT=ODBC`, you can now use *either* the `SQL` option or the `QUERY` option to provide your SQL query:

`sql="`*`SQL query string`*`"`

Use the `SQL` option to supply a relatively short (255 characters or fewer) SQL query, either as a literal string, or as a variable of type `STRING` defined earlier.

`query=input/`*`other unit`*

For a more complex SQL query, use the `QUERY` option. With `QUERY=INPUT`, RATS reads the SQL commands from the lines following the **`DATA`** instruction in the input window (or input file in batch mode). With `QUERY=`*`unit`*, RATS will read the query from the text file associated with the specified I/O unit (opened previously with an **`OPEN`** instruction). In either case, use a "`;`" symbol at the start of a new line to signal the end of the SQL string. See **`OPEN`** in the *Reference Manual* for details on I/O units.

As an example, the three sets of commands below all produce the same results:

```
cal(m) 1995:1
open odbc "Sales"
data(format=odbc,compact=sum,
sql="select date,sum(subtot) as sales from invoice order by date") $
   1995:1 2006:12
```

```
cal(m) 1995:1
open odbc "Sales"
data(format=odbc,compact=sum,query=input) 1995:1 2006:12
 select date,sum(subtot)
 as sales from invoice
 order by date
 ;

cal(m) 1995:1
open odbc "Sales"
open sqlfile "c:\rats\sqlquery.txt"
data(format=odbc,compact=sum,query=sqlfile) 1995:1 2006:12
```

where the file SQLQUERY.TXT contains the following lines:

```
 select date,sum(subtot)
 as sales from invoice
 order by date
 ;
```

DENSITY (7.2, 7.3)

Added the *(Kernel) Density Estimation* Wizard on the *Statistics* menu, which provides an interface to the **DENSITY** instruction. (7.2)

New SMOOTHING option:

smoothing=*smoothing scale factor* **[default is 1]** (7.3)
You can supply a real value (bigger than 0) to adjust the amount of smoothing. Use a value bigger than 1 for more smoothing than the default, values less than 1 for less smoothing.

DISPLAY (7.3)

DISPLAY has one new option:

delimited=[none]/tab/comma/semicolon
By default (with the NONE choice), output from a **DISPLAY** instruction is separated by blank spaces. You can use DELIMITED to generate tab, comma, or semicolon-delimited output instead. This works when outputting to the screen, but is most useful when using the UNIT option to output to a text file.

Also, you can now use **DISPLAY** to show the contents of a parameter set or equation.

DLM (7.3)

Changes to the `PRESAMPLE` option:

presample=ergodic/x1/x0/diffuse

`PRESAMPLE=ERGODIC` will now handle automatically models with a mix of stationary and non-stationary states. The new choice `DIFFUSE` is equivalent to the existing `EXACT` option.

DLM also now includes two new options: `MU` and `LIMIT`.

mu=*VECTOR or FRML[VECTOR]*
The `MU` option is for handling a shift in the observable: $y_t = mu_t + \mathbf{C}'\mathbf{X}_t + v_t$.

limit=*number of observations* **[all observations]**
If you set a value for the `LIMIT` option, RATS assumes that calculations for the Kalman gain and other matrices will converge to a limit after that number of observations. By using the final calculated matrices rather than recomputing, it can save a considerable amount of time in larger time-invariant state space models.

DSGE (7.2, 7.3)

DSGE has several new options:

etz=*vector[rectangular] with components*
analyze=[full]/output/input
form=[sims]/second/first
components=*VECT[RECT] of components*
controls=*# of controls* **[0]**
roots=*VECTOR of (absolute values) of the roots of the model*

Please see the **DSGE** section in this Supplement (page 24) for details.

GRTEXT (7.1)

GRTEXT has two new options:

direction=*compass heading in degrees (integer from 0 to 360)*
Used with the `X` and `Y` options, `DIRECTION` allows you to position the text by specifying a direction from the (x,y) point as a compass heading in degrees. For example, `DIRECTION=0` (or `DIRECTION=360`) will center the text at a point just above the (x,y) location; `DIRECTION=45` will display left-justified text, starting just above and to the right of the (x,y) location; `DIRECTION=270` will display right-justified text directly to the left of the (x,y) point.

transparent/[notransparent]
GRTEXT strings are normally displayed with an opaque white background, so any lines, patterns or symbols lying “under” the string will be obscured from

view. With TRANSPARENT, only the text itself will be opaque—all the white space within and between letters will be transparent, allowing any underlying graph elements to show through.

KALMAN (7.1)

KALMAN now has a DISCOUNT option. Like the option of the same name on **DLM**, this allows for multiplicative (rather than additive) changes to the variance of the states. The syntax is:

discount=*discount value* **[not used]**

The update for the covariance matrix takes the form: $\Sigma_{t|t-1} = \mathbf{A}_t \Sigma_{t-1|t-1} \mathbf{A}_t' \dfrac{1}{discount}$

NPREG (7.2, 7.3)

Added the *Nonparametric Regression* Wizard to the *Statistics* menu. This provides a point-and-click interface for the **NPREG** instruction. (added in 7.2)

New SMOOTHING option:

smoothing=*smoothing scale factor* **[default is 1]** (added in 7.3)

You can supply a real value (bigger than 0) to adjust the amount of smoothing. Use a value bigger than 1 for more smoothing than the default, values less than 1 for less smoothing.

PREG (7.2)

Adds POOLED as an additional choice on the METHOD option, for doing pooled panel regression:

method=[fixedeffects]/randomeffects/fd/sur/between/pooled

REPORT (7.1, 7.3)

Reports generated by the **REPORT** instruction are now saved as a user-accessible variable of type REPORT. This allows users to work with multiple reports simultaneously, define reports as local variables in a procedure or function, and more. The key to this is the new USE option on **REPORT**:

use=*name of report*

USE allows you to define a new report object (when used with ACTION=DEFINE), or work with an existing report object (when used with the other choices for the ACTION option: MODFIY, FORMAT, SHOW, and SORT). If you omit the USE option, RATS uses the default internal report.

You can also define a variable of type REPORT using **DEFINE** or **LOCAL** instructions. **REPORT** also offers a TITLE option for supplying your own title for the output:

title="*title for REPORT***"**

SSTATS (7.2)

FRAC=*desired fractile* **[not used]**
Can be used to obtain median (FRAC=.50) or any other percentile.

STORE (7.2)

STORE has the same new DATEFORMAT option as described for the **DATA** instruction (page 11).

STWISE (7.1)

STWISE now has a GTOS choice for the METHOD option (for General TO Specific) which drops regressors, in order, starting from the end of the supplementary card list. This can be used to do automatic pruning of lags in an autoregression. The syntax is:

For example:

```
stwise(method=gtos) y
# constant x{1 to 12}
```

will remove regressors starting with lag 12 of X, then lag 11 of X, and so on, until all remaining regressors meet the minimum criterion value.

X11 (7.2)

Many changes were made to **X11** in the process of implementing the Census X12-ARIMA process. See the revised *X11 Supplement* PDF for full details on this instruction.

Most of the changes to this have been to the precision of the calculations. Many of the filters used in Census-X11 came from lookup tables, which in many cases rounded coefficients to just three or four significant digits. The new **X11** engine uses filters that are generated as needed to the full precision available.

New or changed options:

mode=[multiplicative]/additive/pseudoadditive/logadditive
While the old MULTIPLICATIVE option still works, the adjustment mode should now be selected using the MODE option.

prefactors=*preliminary adjustment factors* **[not used]**
X12-ARIMA emphasizes the use of preliminary adjustment factors to take care of various types of outliers, rather than using the internal outlier detection engine (which is still present). These are usually estimated using **BOXJENK**, and created using the ADJUST option on it.

For multiplicative and log-additive adjustments, these should be in the form of factors, that is, 1.0 means no adjustment. If you are doing a log additive adjustment starting with a **BOXJENK** model applied to the log of the series, you will have to transform the factors generated by **BOXJENK** from their additive form to the multiplicative form. The following is an example:

```
boxjenk(ar=%%autop,diffs=1,ma=%%autoq,sar=%%autops,$
  sdiffs=1,sma=%%autoqs,method=bfgs,outliers=standard,$
  adjustments=final) ldata
set prior = exp(final-final(2008:7))
x11(mode=logadd,prefactors=prior,print=full) u36cvs
```

The value of `FINAL` at the end of the data is subtracted before the *exp* is taken so the adjustment will leave the end of data value at its observed level. This is done because the level shift and temporary change dummies are defined from `t0` on, and so will give non-zero shift values to the end of the data, rather than the beginning. (The adjusted data will be the same either way; any printed output looks more natural with this correction).

extension=*series with out-of-sample forecasts of dependent var.*
leads=*number of periods of extension*

The `EXTENSION` series is used in computing some of centered moving averages within the X11 engine to reduce the bias in the end effects on the filters.

decimals=*number of decimals to show in output* **[depends upon data]**

Internal Regression Effects

These options control an internal regression of the irregular component on various dummy variables.

tradeday=[none]/apply

They allow you to apply a "trading day adjustment" for variation due to the number of Mondays, Tuesdays, etc., in a month. A typical series which would benefit from trading day adjustment is a total retail sales series, where there would be considerable predictable variation among the days of the week.

NONE	The trading day option is not applied.
APPLY	Computes the trading day factors and applies them to the final adjustment.

In previous versions of RATS, the following holidays were switch options. These are still supported, though it's recommended that any of these be done as preliminary factors instead. All of the holiday shifts are adjusted for long-run mean values, which prevents them from picking up a spurious trend effect for particular ranges of data.

easter=*number of days before Easter at which effect is felt*

It's assumed that the level of activity is different for this number of days before Easter. This generates a dummy which splits this among February, March and

April based upon the number of days falling in each month. The analogue to the old switch option is `EASTER=21`, though the calculation is now done differently.

laborday=*number of days before Labor Day for the effect*

It's assumed that the level of activity is different for this number of days prior to Labor Day. This generates a dummy which splits this between August and September based upon the number of days falling in each month. The value for this that's equivalent to the old `LABORDAY` switch is `LABORDAY=8`.

thanksgiving=*number of days before Thanksgiving for the effect*

The value which gives the old adjustment is –1, that is the day after Thanksgiving. The level of activity is assumed to be different from this point to December 24.

critical=*critical value (t-stat) for outlier detection*

[default depends upon number of data points, typically around 3.8]

The internal regression includes automatic detection and removal of additive outliers. This is to prevent contamination of the estimates of the calendar effects by very large outliers. The same procedure is followed in handling the regression effects within **X11** as it is in **BOXJENK**, though only additive outliers are examined.

5. New Functions

`%bicdf(x0,y0,rho)`	returns $P\left(x \leq x_0, y \leq y_0\right)$ for a bivariate standard Normal with correlation coefficient *rho*. (7.3)
`%cxdiag(cv)`	Creates a (complex) diagonal matrix from complex-valued $1 \times N$ rectangular or complex vector. (7.1)
`%dlmgfroma(A)`	Returns a matrix which transforms to stationarity a state-space model with transition matrix **A**. (7.2)
`%dlminit(A,SW,F,Z)`	Returns a full solution for initial conditions for a state space model with the given input matrices. It returns a `VECT[RECT]` with first component being the (finite) covariance matrix, the second the mean, and the third the diffuse covariance matrix. (7.3)
`%logdensitydiag(V,U)`	Diagonal multivariate log Normal density. Similar to `%LOGDENSITY` except that the covariance matrix is diagonal, and **V** is a vector with the diagonal elements. (7.3)
`%loggammadensity(x,a,b)`	Returns the log density function at *x* for a gamma distribution with shape parameter *a* and scale parameter *b*. (7.2)
`%logtdensitystd(V,U,nu)`	Standardized log multivariate *t* density. This is equivalent to: `%logtdensity(v*nu/(nu-2),u,nu)` (7.3)
`%mspexpand(pcde)`	Returns the full $N \times N$ transition matrix created from an $N-1 \times N$ matrix of the free parameters in a transition matrix. (7.2)
`%parmslabels(parmset)`	Returns the labels of the variables in the parameter set (see **`NONLIN`** in the Reference Manual for details on parameter sets). (7.3)
`%ranmvt(f,nu)`	Returns a random draw from a multivariate *t* distribution. (7.3)
`%ranTruncate(mu,sigma,lower,upper)`	This uses the rejection method to generate draws from a truncated Normal. (7.3)

`%reshape(A,n,m)`	Rearranges elements of **A** into an $N \times M$ matrix. Both input and output matrices are stored by columns (that is, consecutive entries go down a column). (7.2)
`%sumc(A)`	For $N \times M$ matrix **A**, returns the *M* vector with the column sums of **A**. (7.2)
`%sumr(A)`	For $N \times M$ matrix **A**, returns the *N* vector with the row sums of **A**. (7.2)
`%tcdf(x,nu)`	Returns the CDF for a *t* distribution. (7.3)

6. New Reserved Variables (7.1)

ESTIMATE now defines the following when applied to a VECM model (a model that includes an **ECT** term):

%VECMALPHA	Matrix of loadings on cointegrating relations in the VECM [Rectangular array]
%VECMPI	Matrix of coefficients on undifferenced lag in the VECM [Rectangular array]

PRJ now defines:

%PRJSTDERR	Standard Error of the fitted value produced by the XVECTOR or ATMEAN options [Real]

7. Graph Styles, GRPARM, and Importing Styles

About Graph Styles

The data in RATS graphs are presented using lines, fills or symbols, depending upon the type of graph selected via the `STYLE` and `OVERLAY` options. For each category (line, fill, symbol) RATS supports thirty user-definable styles for color graphs, and a corresponding thirty for black and white graphs.

The default styles used by RATS have been chosen to be fairly easily distinguishable roughly for style numbers 1 through 10. You can use *Graph Style Sheets*, as described in this section, to define and use your own style definitions.

Color and Black and White Styles

RATS normally displays graphs in color and uses the "color" styles. It will automatically switch to the corresponding black and white styles if you: print to a black and white printer; use the "color/b&w" toolbar button to switch a graph window to black and white mode; or use the `PATTERNS` option on the graphing instruction, which forces black and white "patterns" rather than colors).

You can create both color and black-and-white definitions for each style number in your style sheets.

Lines, Fills, and Symbols

For each style number, you can define separate styles for Lines, Fills, and Symbols. For lines, you can use a solid line or choose from six different dashed patterns. You can also set the color (for color styles) or level of gray (for black and white styles). Finally, you can also set the thickness (heaviness) of the line.

For fills, you can control the hatching pattern, and color or level of gray.

For symbols, you can chose the symbol (i.e. the shape) from twelve choices, the color or gray level of the symbol, and whether or not the symbol is filled in or drawn as an outline shape.

Defining Styles in a Graph Style Sheet File

You can redefine any of the styles. You do that by putting the new style definitions into a text file (which we refer to as a Graph Style Sheet file), and reading those definitions into RATS using **`OPEN`** and **`GRPARM`** instructions.

You can create the text file with RATS or any other text editor or word processor. Each line in this redefines the characteristics of one representation. You can define as many or as few of the styles as you want; if you don't redefine a style, it will just keep the previous settings. You're most likely to want to redefine the black and white styles, since those will be used in publications.

A line in the text file will take one of the following forms:

```
LINE_COLOR_NN=pattern,color,thickness
LINE_BW_NN=pattern,gray,thickness

FILL_COLOR_NN=pattern,color
FILL_BW_NN=pattern,gray

SYMBOL_COLOR_NN=pattern,color,filled
SYMBOL_BW_NN=pattern,gray,filled
```

The first part of the definition specifies the type of representation you are defining (`LINE`, `FILL`, or `SYMBOL`).

The second part (`COLOR` or `BW`) tells rats that it is a color or black and white style.

The third part (`NN`) is the style number that you're defining. It should be between 0 and 30. Styles 1 through 30 can be selected by the user (via the *representation* parameter on the supplementary card of the graphing command). Style 0 is reserved for shading performed via the `SHADE` option, so if you want to adjust the pattern or gray level used for shading, define style 0 as desired.

The arguments are as follows:

pattern is the pattern choice—dashing pattern for lines, hatching pattern for fills and symbol shape for symbols. The possible selections are shown on the next page.

color is represented as a 24 bit (six digit) hexadecimal number. The first two hexadecimal digits are the level of red (00=no red to FF=red fully on), the next two are the level of green and the final two the level of blue.

gray is a real number between 0 and 1 representing the degree of "grayness" (fraction of white). 0 means black, 1 means white. Note that it's much easier to distinguish the lighter end of this (near 1) than the darker end: 0 and .25 look very similar, .90 and .95 look quite different. The default values for the first four black and white fills are solid black, solid .90 gray, solid .50 gray and solid .80 gray.

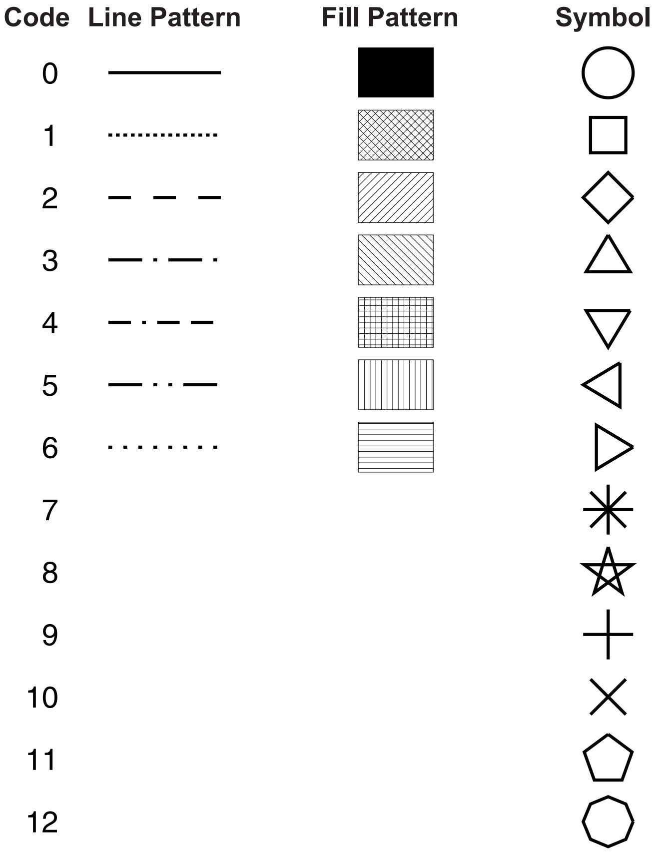
thickness is a real scale factor where 1.0 represents a standard line thickness. To make a line three times the standard thickness, use 3.0.

filled is 0 for not filled (outline only) and 1 for filled.

Pattern Definitions

The available line patterns, fill patterns, and symbol choices are show below. To select a particular line, fill, or symbol for a given style, use the number in the left-hand column as the value for the *pattern* parameter.

For example, the line "`SYMBOL_COLOR_2=1,FF0000,1`" defines color symbol style number two as a red, filled square (1 being the pattern code for a symbol).



Example

If you create the file `thicklines.txt` with the following information:

```
LINE_BW_1=0,0.0,3.0
LINE_BW_2=0,0.9,3.0
LINE_BW_3=0,0.5,3.0
LINE_BW_4=0,0.8,3.0
```

then

```
open styles thicklines.txt
grparm(import=styles)
```

will redefine the black and white versions of the first four line styles to use solid lines with varying gray levels (rather than dash patterns) as the distinguishing feature. They are also thickened up by a factor of 3 from the standard line width.

DSGE — Dynamic Stochastic General Equilibrium Models

DSGE takes a dynamic model (possibly nonlinear) with expectational elements or unstable roots and solves it for a state space form. For a nonlinear model, this is done by linearizing about some expansion point, such as a steady state. **DSGE** first appeared in Version 7.0, but was added too late to be included in the *User's Guide* and *Reference Manual*.

dsge (**options**) *list of target series<<initial values*

Parameters

target series List of series which are the endogenous variables in the model. The number of series should match the number of equations in the model.

initial values (Optional) If desired, you can use the syntax "*series<<value*" to provide an (initial guess value for) the expansion point for any of the series in the list of targets.

Options

model=*MODEL to be solved*

This must take a particular form. See "Form for Model" later.

expand=[none]/linear/loglinear

This indicates the type of expansion required. The default (EXPAND=NONE) is used when the model is fully linear. EXPAND=LINEAR does a linear expansion and EXPAND=LOGLINEAR does a log-linear expansion.

initial=*VECTOR of initial guess values* **[0's or 1's]**
iters=*iteration limit for solution algorithm* **[50]**
cvcrit=*convergence criterion* **[.00001]**
steadystate=*VECTOR of final converged values* **[not used]**
solveby=[lu]/svd
trace/[notrace]

These apply if the model is non-linear and thus needs expansion. Most are used in solving for a steady state. If you want to input the expansion point (if, for instance, the model *has* no steady state), just use the INITIAL option or the *initial values* parameters, and use ITERS=0. The default starting values are zeros for all variables if EXPAND=LINEAR or all ones if EXPAND=LOGLINEAR. You can retrieve the final values with the STEADYSTATE option. The VECTORS for both INITIAL and STEADYSTATE are in the order listed in the *list of target series*.

The SOLVEBY option controls the method used for solving the Newton's method steps. SOLVEBY=LU uses the faster LU–decomposition, but that fails when the model has unit roots, in which case, you'll need to switch to the slower SOLVEBY=SVD (Singular Value Decomposition).

cutoff=*value or formula giving stable/unstable roots cutoff* **[1.0]**
See "Algorithm: Solving for State Space Representation" below.

a=*A matrix for state space model*
Z=*Z matrix for state space model*
F=*F matrix for state space model*
These are the output arrays from **DSGE** which describe the state space model which solves the (possibly expanded) model. This takes the form

(1) $\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{Z} + \mathbf{F}\mathbf{w}_t$

The first components of **X** are the target series in order. It may have more components than that to handle additional lags and auxiliary variables for the expectational terms. **w** has dimension equal to the number of non-identities in the model.

etz=*vector[rectangular] with components*
Use this option to have **DSGE** create a vector of rectangular arrays with components for computing effect of future exogenous shocks.

In Sims (2002), the general form of the solution to the (linearized) DSGE is

$$y(t) = \Theta_1 y(t-1) + \Theta_c + \Theta_0 z(t) + \Theta_y \sum_{s=1}^{\infty} \Theta_f^{s-1} \Theta_z E_t z\left(t+s\right)$$

You can retrieve Θ_1 using the A option, Θ_c with the Z option and Θ_0 with the F option. The final term drops out if the Z process is serially uncorrelated. If it isn't, or if you want to predict the effect of (known) future shocks to Z, you can use the ETZ option to obtain the three matrices need for that. After ETZ=THETA, THETA(1) has Θ_y, THETA(2) has Θ_f and THETA(3) has Θ_z. Note that the infinite sum in the final term will rarely simplify easily, so the sum will generally have to be approximated with a finite number of terms.

analyze=[full]/output/input
form=[sims]/second/first
components=*VECT[RECT] of components*
controls=*# of controls* **[0]**
ANALYZE=FULL gives the standard behavior of the **DSGE** instruction. The ANALYZE=OUTPUT option generates the components of the model for the form selected by the FORM option, storing the results in the variable provided on the COMPONENTS option. ANALYZE=INPUT takes as input the components provided via the COMPONENTS option and solves out the model. The CONTROLS option

controls the positioning of the states for the final set of series on the DSGE line. Instead of inserting any augmenting states (leads and extra lags) after the series, it inserts them between the non-controls and the controls.

roots=*VECTOR of (absolute values) of the roots of the model*
Saves the roots of the model to a vector.

Form for Model

The model is a set of FRML's which are to take the value zero at a solution. Expectational terms are handled by using leads of the series involved: a lead represents an expectation given information at time *t*. For instance, a standard condition for optimal consumption in a very simple model is

(2) $E_t\left(\beta R_t C_t / C_{t+1}\right) = 1$

This would be represented by

```
frml(identity) f1 = beta*r*c/c{-1}-1.0
```

where R and C are series and BETA is a real-valued parameter.

It's very important to declare formulas as identities if they are not subject to time *t* shocks. Each non-identity is assumed to be subject to a separate time *t* shock which will be one of the components of **w**.

You may need to make adjustments to your original model to put it into this particular form. We'll demonstrate several standard adjustments next.

Autoregressive Shocks

Convert the original equation into an identity by adding a new series which represents the shock, and add a new (non-identity) describing the AR for the shock. For instance:

```
frml(identity) eqn8 = yhat-(1-tau*ky-gy)*chat-tau*ky*ihat-gy*epsg
frml eqnepsg = epsg-rho_g*epsg{1}
```

Note that you now need to include EPSG among the target series, since the model's solution must include it. You might find it easier to handle *all* shocks this way, even if they aren't autocorrelated.

Redating Variables

Consider $C_t + K_{t+1} = fK_t + \varepsilon_t$, where ε_t is a time *t* shock to a simple production function. We can't directly translate this into a formula, because the K_{t+1} would be incorrectly interpreted as an expectation. Instead, we should redate the time associated with the capital variable, changing this to $C_t + K_t = fK_{t-1} + \varepsilon_t$.

This would be represented by the formula

```
frml f2 = f*k{1}-c-k
```

Note that this needs to be arranged so that the shock will have the desired sign.

Expectations at *t+k*, *t-k*

These can be handled as described in Sims (2002). Consider

$$(3)\quad m_t = p_t + y_t - \alpha(E_{t-1}p_{t+1} - E_{t-1}p_t)$$

Define $e_{1t} = E_t p_{t+1}, e_{2t} = E_t p_{t+2}$. Then (3) can be represented using the combination of three formulas:

```
frml(identity) md = m-p-y+alpha*(e2{1}-e1{1})
frml(identity) e1f = e1-p{-1}
frml(identity) e2f = e2-e1{-1}
```

Algorithm (Solving for Steady State)

All lags or leads (thus expectations) of each dependent variable are assumed to be represented by a single common value. If $\mathbf{x}$ represents the vector of steady state values, then the solution is the vector which solves $F(\mathbf{x}) = \mathbf{0}$. Newton's method updates using:

$$(4)\quad \mathbf{x}_{n+1} = \mathbf{x}_n - F'\left(\mathbf{x}_n\right)^{-1} F\left(\mathbf{x}_n\right)$$

until convergence. If the model is being subject to log-linear expansion, the update is

$$(5)\quad \log \mathbf{x}_{n+1} = \log \mathbf{x}_n - \left(diag\left(\mathbf{x}_n\right) F'\left(\mathbf{x}_n\right)\right)^{-1} F\left(x_n\right)$$

This is repeated until the maximum of the absolute values of the components of the adjustment vectors is less than the convergence criterion.

The default initial guess values are a vector of zeros if a linear expansion is used, and a vector of ones if a log-linear expansion is used. It is quite likely that you'll need to provide a better set of values than these.

Algorithm (Solving for State Space Representation)

This applies the algorithm described in Sims (2002), based upon the generalized Schur (QZ) decomposition. If the model is non-linear, this uses the linearized or log-linearized version. Derivatives are computed analytically. The `CUTOFF` option is used to control which roots will be considered stable (and solved backwards) and which will be unstable (and solved forwards).

Examples

Watson (1993) analyzes a special case of the model below (which some minor renaming of variables) whose equilibrium is described by the following equations:

$$(6)\quad \frac{\theta}{1-N_t} = C_t^{-\eta}\alpha\frac{Y_t}{N_t}$$

(7) $1 = \beta\gamma^{1-\eta} E_t R_{t+1} \left(C_t / C_{t+1} \right)^{\eta}$

(8) $\gamma R_t = (1-\alpha)\dfrac{Y_t}{K_{t-1}} + 1 - \delta$

(9) $C_t + I_t = Y_t$

(10) $\gamma K_t = I_t + (1-\delta) K_{t-1}$

(11) $Y_t = Z_t K_{t-1}^{1-\alpha} N_t^{\alpha}$

(12) $\log Z_t = (1-\rho)\log(\bar{Z}) + \rho \log(Z_{t-1}) + \varepsilon_t$

There is only one exogenous shock in this model (equation 12—the technology shock). This is a non-linear model, which will be analyzed using a log-linearization. Because *N* can't take the value 1, we can't use the default initial guess values for all variables, so the *initial values* parameters are used for some. SOLVEBY=SVD is included to deal with the unit root in the technology shock.

Several of the series used in the model are unobservable and so aren't in the data set. They're included in a DECLARE SERIES instruction so they can be used in defining the equations.

```
open data watson_jpe.rat
calendar(q) 1948
data(format=rats) 1948:1 1988:4 y c invst h

compute eta=1.0             Log utility
compute alpha=.58           Labor's share
compute nbar=.2             Steady state level of hours
compute gamma=1.004         Rate of technological progress
compute delta=.025          Depreciation rate for capital (quarterly)
compute rq=.065/4           Quarterly steady state interest rate
compute rho=1.00            AR coefficient in technical change
compute lrstd=.01           Long-run standard deviation of output
compute zbar=1.0            Mean of technology process
compute theta=3.29          Preference parameter for leisure
compute beta=gamma/(1+rq)

declare series n r z k
frml(identity) f1 = theta/(1-n)-c**(-eta)*alpha*y/n
frml(identity) f2 = 1-beta*gamma**(1-eta)*c/c{-1}*r{-1}
frml(identity) f3 = gamma*r-(1-alpha)*y/k{1}-1+delta
frml(identity) f4 = c+invst-y
frml(identity) f5 = gamma*k-invst-(1-delta)*k{1}
frml(identity) f6 = y-z*k{1}**(1-alpha)*n**alpha
frml           f7 = log(z)-(1-rho)*log(zbar)-rho*log(z{1})
```

```
group swmodel f1 f2 f3 f4 f5 f6 f7
dsge(model=swmodel,expand=loglinear,a=a,f=f,solveby=svd) $
   y c<<0.5 invst<<0.5 n<<nbar r<<1+rq k z

@dlmirf(a=a,f=f,labels=||"Technology"||,$
   vlab=||"Output","Consumption","Investment","Hours"||,$
   graph=byvar)
```

This is a simple Cass-Koopmans growth model. The model is deterministic and the conditions are linear. What **DSGE** does here is to suppress the unstable root. The state-space representation is simulated using **DLM** for several different sets of initial conditions. Because the model is linear, there's no need to compute an expansion point to get the state space representation; it's being done here to get the steady state so create the simulation scenarios.

```
declare series c lambda k
declare real u0 u1 f0 beta
frml(identity) f1 = u0-u1*c-lambda
frml(identity) f2 = f0*lambda-1.0/beta*lambda{1}
frml(identity) f3 = f0*k{1}-k-c{1}

compute beta=.95,f0=1.3,u0=1.0,u1=0.2

group casskoopmans f1 f2 f3
dsge(expand=linear,steadystate=ss,a=a,z=z,model=casskoopmans) $
  c k lambda
disp "Steady State"
disp "Consumption" @20 ss(1)
disp "Capital" @20 ss(2)
*
dlm(x0=||3.0,12.0,u0-u1*3.0||,a=a,z=z,presample=x1) 1 20 xstates
set c 1 20 = xstates(t)(1)
set k 1 20 = xstates(t)(2)
spgraph(vfields=2,footer="Initial consumption below steady state")
graph(hlabel="Consumption")
# c
graph(hlabel="Capital")
# k
spgraph(done)
*
dlm(x0=||6.0,12.0,u0-u1*6.0||,a=a,z=z,presample=x1) 1 20 xstates
set c 1 20 = xstates(t)(1)
set k 1 20 = xstates(t)(2)

spgraph(vfields=2,footer="Initial consumption above steady state")
graph(hlabel="Consumption")
# c
```

```
graph(hlabel="Capital")
# k
spgraph(done)
*
dlm(x0=||3.0,20.0,u0-u1*3.0||,a=a,z=z,presample=x1) 1 20 xstates
set c 1 20 = xstates(t)(1)
set k 1 20 = xstates(t)(2)
*
spgraph(vfields=2,footer="Initial capital above steady state")
graph(hlabel="Consumption")
# c
graph(hlabel="Capital")
# k
spgraph(done)
```

Bibliography

Sims, C.A. (2002). "Solving Linear Rational Expectations Models", *Computational Economics*, October 2002, Vol. 20, Nos. 1-2, , pp. 1-20.

Watson, M.W. (1993). "Measures of Fit for Calibrated Models", *Journal of Political Economy*, Vol. 101, No. 6, pp. 1011-1041.

Variables Defined

%CONVERGED	1 or 0. Takes the value 1 if the solution for the steady state converged, 0 otherwise.
%CVCRIT	Final convergence criterion (if steady state solution is needed)

DUMMY — Generating Dummy and Related Variables

DUMMY provides an easy way to generate standard dummies and other constructed variables.

dummy (**options**) *series start end*

Parameters

series	series to define
start end	range to set

Options

ao=*period for additive outlier* **[not used]**
AO=t_0 defines a dummy which is zero except for 1 at t_0.

ls=*period for level shift dummy* **[not used]**
LS=t_0 defines a dummy which is zero for $t < t_0$ and 1 for $t \geq t_0$

tc=*period for temporary change dummy* **[not used]**
TC=t_0 defines a dummy which is zero for $t < t_0$, 1 for $t = t_0$ and declines exponentially from t_0 until *end*

from=*starting period for 1's*
to=*ending period of 1's*
Used alone or together, defines a shift dummy which is 0 outside the FROM, TO range, and 1 inside it.

GBOX — Graphing Box Plots

GBOX produces box plots (also known as "box-and-whisker" plots) for one or more series. Box plots provide a simple graphical representation of some of the basic statistical properties of one or more series, including the median, the interquartile range, the maximum and minimum values, and significant outliers. **GBOX** was introduced in Version 7.1.

```
gbox( options )   number   hfield   vfield
# series   start   end      (one card for each series)
```

Parameters

number	Number of series to graph. The maximum permitted is twenty.
hfield vfield	When using **SPGRAPH** to put multiple graphs on a single page, these allow you to put a graph in a specific field. By default, the fields are filled by column, starting at the top left (field 1,1).

Supplementary Cards

Use one supplementary card for each series you want to plot.

series	the series to be plot.
start end	(Optional) the range to use in generating the box plot. If you have not set a **SMPL**, this defaults to the defined range of *series*. *start* and *end* can be different for each series in the graph.

Options

Most of the **GBOX** options are the same as those available on the **GRAPH** instruction. These common options are listed briefly below—see **GRAPH** in the *Reference Manual* for details on these. The options that are unique to **GBOX** are described in more detail below.

[axis]/noaxis	Draw horizontal axis if Y=0 is within bounds
extend/[noextend]	Extend horizontal grid lines across graph
footer=*footer label*	Adds a footer label below graph
frame=[full]/half/none/bottom	Controls frame around the graph
header=*string*	Adds a header to the top of the graph
hlabel=*label*	Adds a label to the horizontal axis
log=*value*	Base for log scale graphs
max=*value*	Value for upper boundary of graph

min=*value*	Value for lower boundary of graph
picture=*pict. clause*	Picture clause for axis label numbers
scale=[left]/right/ both/none	Placement of vertical scale
smpl=*series or frml*	Series/formula indicating entries to be graphed
subhead=*string*	Subheader string for graph
vgrid=*vector*	Values for grid lines across from vertical axis.
vlabel=*label*	Label for the vertical axis
vticks=*number*	Maximum number of vertical ticks
window=*string*	Title for graph window

Options Specific to GBOX

labels=*VECTOR[STRINGS] of labels for the plots* **[series names]**

By default, **GBOX** labels the x-axis with the names of the series being plotted. If you prefer, you can provide your own labels using the LABELS option. The first box plot will be labeled with the first string in the vector, the second plot with the second string, and so on.

group=*SERIES or FRML with distinct values for each group*

You can use this option to have **GBOX** divide the series being graphed into groups, or subsamples, based on the values of this series or formula. For example, if you supply a 0/1 dummy variable, **GBOX** will do two plots for each series—one plot using observations where the GROUP series contains zeros, and another for observations where the GROUP series contains ones.

Elements of a Box Plot

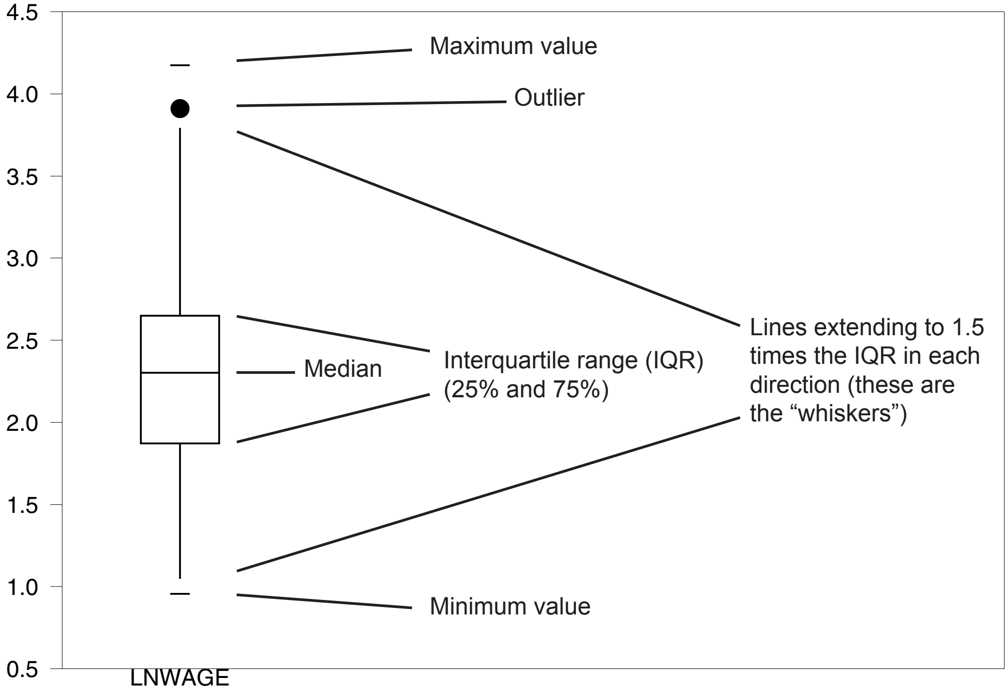
Box plots include the following elements:

- A line representing the median value
- A box representing the interquartile range (IQR). The top and bottom lines of the box correspond to the 75th and 25th percentiles, respectively.
- Vertical lines, or “whiskers” indicating 1.5 times the IQR in either direction from the 75th and 25th percentiles. This is about 2.7 standard deviations on either side of the median for a Normal series.
- Short horizontal lines representing the maximum and minimum values.
- Dots representing outliers (if any). Outliers are defined as values that fall outside 1.5 times the IQR in either direction.

Examples

Here is a sample box plot showing the various elements. This is generated using data taken from Sections 13.3-13.6 of *Econometric Methods*, by Johnston and DiNardo, 4th Ed. (1996, McGraw-Hill/Irwin).

```
open data cps88.asc
data(format=prn,org=columns) 1 1000 age exp2 grade ind1 $
   married lnwage occ1 partt potexp union weight high
gbox
# lnwage
```



This draws a separate box plot for each year's worth of data in the quarterly series `WASH`.

```
open data washpower.dat
calendar(q) 1980
data(format=free,org=columns) 1980:1 1986:4 wash
gbox(group=%year(t),$
labels=||"1980","1981","1982","1983","1984","1985","1986"||)
# wash
```