# TSP Handbook

by
Sergio L. Schmukler
and
Bronwyn H. Hall

to accompany

# Econometric Models and Economic Forecasts

by Robert S. Pindyck and Daniel L. Rubinfeld

Draft of August 1997

Please send comments or suggestions to bhhall@econ.berkeley.edu
and/or sergio@econ.berkeley.edu

# CONTENTS

# CHAPTER 1

## Introduction

This handbook's goal is to introduce you, the reader of Pindyck and Rubinfeld, *Econometric Models and Economic Forecasts,* Fourth Edition[1] (hereinafter P&R), to the use of TSP, a popular econometric software package. We assume you are a student enrolled in an econometrics course and have access to a DOS/Windows/OS2/Apple Macintosh personal computer or a Unix workstation that runs TSP.

To use this handbook effectively, you should already be familiar with your computer, its operating system, and an editor for plain ASCII (DOS) text. If you use TSP under Windows, you can use the visual interface utility TSP Through the Looking Glass (TLG) to prepare your TSP input, run TSP and look at the results. Although you may use TSP interactively, you will find an editor useful for class assignments and documenting your work. Specific suggestions about how to work with TSP on your computer are made later in this introduction.

This handbook is designed to be a self-sufficient introduction to using TSP. It is not necessary to have the *TSP User's Guide* and *Reference Manual* if you only want to do the computer exercises in P&R. However you will find them useful when you do your own econometric work, and for supplementary information about the program.

[1] Previous editions of the text can also be used, but be aware of some modifications.

Each chapter of this handbook corresponds to a chapter of P&R. Each chapter briefly discusses the statistical/econometric concepts being introduced, and then describes the new TSP commands used in the chapter. Next, it solves some examples from P&R using TSP. Lastly, it summarizes the set of TSP concepts introduced in the chapter and lists the batch programs generated for the examples.

This handbook starts with the most basic TSP commands, and introduces more complex statements as it goes. Explanations of particular commands are not repeated throughout the Handbook, so you may need to refer to earlier chapters to refresh your memory about particular commands. Description of the commands is not exhaustive. If you want to know about the complete set of options for each command, use the *TSP User's Guide* or *Reference Manual*.

We believe you will need to see more details at the beginning of this handbook, so the first chapters contain more econometric concepts and examples than later ones. We leave some P&R examples (for which the data sets are available) to the user. Many examples are not illustrated in this handbook simply because the data sets are not provided in P&R's datadisk. We also add examples not in P&R, to show the full range of TSP commands.

The TSP programs used in this handbook and the datadisk are available on the TSP web page:

**http://www.tspintl.com/examples/pr**

Note that some of the results illustrated here might differ in small ways from the ones reported in P&R. Slightly different data sets (from the ones used by P&R), or different estimation techniques cause these discrepancies.

## 1.1 How to Run TSP

How you run TSP on a computer depends on: (1) whether you want to use TSP in batch mode or interactive mode and (2) which computer(s) are available to you. Let's discuss the batch/interactive choice first (examples of these modes of use are given later in this chapter).

### 1.1.1 Interactive Use of TSP

When running TSP interactively: you type a command, TSP interprets it, and displays the output on the screen. The advantage of running interactively is that it is easy to get started, and to change what you are doing based on the results of previous commands.

In the DOS/Windows version of TSP, any commands you have previously executed can also be recalled (using the up-arrow or - key) and edited using the usual insert/delete and arrow keys. This feature is not currently available on Macintosh and Unix.

The disadvantage of using TSP interactively is that it is harder to reuse your work. For example, it would be difficult to rerun your commands with a revised data set. Although TSP can capture what you do in an interactive session (discussed in Chapter 4 of the *User's Guide*, and the REVIEW, TERM, OUTPUT entries in the *Reference Manual*), it is inherently more difficult to use an interactive session as a TSP program in future work, since most sessions will contain all your mistakes along with the final versions of the commands.

These considerations mean serious users of the program, and students who need to hand in their programs and output, will eventually want to switch to batch programming. Windows users will find it easiest simply to use TLG, which makes batch TSP programming very easy in the Windows environment.

### 1.1.2 Batch use of TSP

The term batch means all the commands you wish to execute are submitted in one file to TSP, "in a batch." In this mode, you use an editor to create a file that contains your TSP program, a series of simple commands composed of keywords and variable names. One example follows:

```
FREQ ; SMPL 66 75 ;
LOAD CONS YD ;
... series of numbers ......
MSD CONS YD ;
LOGC = LOG(CONS) ;
LOGY = LOG(YD) ;
OLSQ LOGC C LOGY ;
END ;
```

*Note: Every command in a batch file must end with a semicolon.*

TSP program files end with the extension "*tsp*". For example, you could call the program above "*simple.tsp*". After you start TSP (the procedure is described below), enter your program’s name (in this case, *simple.tsp*) when it asks for the name of a batch program to run. When TSP finishes running *simple.tsp*, there will be a new file on the computer called "*simple.out*" containing the results of the run. You can look at this output file with an editor, and print it. Each time you run *simple.tsp* a new version of *simple.out* is generated. So if you want to keep the current version you need to rename *simple.out* before rerunning *simple.tsp*.

The advantage of batch mode is that you can easily change commands and correct mistakes with your editor or TLG. The corrected file can then be submitted to TSP. This process can be repeated as many times as you like. The disadvantage is that if your changes depend on previous results, working in batch mode can take longer (because you have to keep running the whole program, instead of doing it piecemeal).

*Note: Interactive examples are only shown in Chapters 1 and 2. In the rest of the handbook, we use batch examples only.*

### 1.2 Note on computers

The choice to use either batch or interactive mode depends on the problem you are trying to solve and you will likely find uses for both. The exact way you run TSP will also depend on the computer you use. At the present time, the likely choices fall into three main classes.

#### 1.2.1 Personal computers running Windows 3.1, Windows 95, OS/2 or Macintosh OS

These operating systems have graphical user interfaces (GUIs). You typically start TSP by clicking on an icon (see the installation memo that came with your program). TSP begins with a display of the program's name. You will be prompted for the name of an input file, and given the option to run interactively:

```
Enter batch file name [or press Enter for
interactive]:
```

If you enter a batch file name (like *simple.tsp*), it will execute and then prompt you either to switch to interactive use, quit TSP, or switch to Windows to examine the output file. If you switch to Windows, TSP remains active, so you can rerun the input file after making changes. This is an extremely useful way to develop TSP programs, but it is only available under multi-tasking operating systems.

On Windows 3.1 and 95, you can use TLG. Instead of clicking on a TSP icon, you click the TLG icon. TLG opens and you can either edit an existing TSP input file or make a new one (on the drop-down file menu). When you have finished editing, just click on the TSP icon to run it.

### 1.2.2 Personal computers running MS/DOS

MS/DOS has a traditional line-oriented user interfaces where you type a command and it is immediately executed. Typically, you will see a prompt like:

```
C:\TSP44>
```

To start type TSP at this prompt. You will see the following prompt after an introductory screen:

```
Enter batch file name [or press Enter for
interactive]:
```

If you enter a batch file name, TSP reads in your batch file and executes it. If you press enter, TSP prints out some information (the address of TSP International, version number and date of TSP, etc.) and then prompts you to enter commands interactively. If you choose batch execution, an output file such as *simple.out* is automatically created, as described above.

### 1.2.3 Workstations running a version of Unix.

Although Unix workstations can have GUIs, TSP is usually run in the command shell like a DOS program. Depending on how TSP has been installed on your system, you will type something like "tsp" and receive the same prompts as in the DOS case.

***WARNING:*** *In all cases, and on all computers, whether the computer can find TSP when you type tsp or TSP, your input file when you type "simple.tsp," and for the data file or your output file (like simple.out) depends crucially on the directory structure, that is, it depends on how files are organized on your computer. It is impossible to guess how this works for every Handbook reader, so we strongly recommend that if you get error messages of the sort "File not found," etc., and you cannot figure out what to do, consult someone knowledgeable at your local installation for assistance. To help TSP locate files, wherever you can supply a filename to TSP, you can also supply the full directory information, as long as you enclose the whole filename in quotes. For example in DOS/Windows, you can use names like "us\patdata\patest.tsp"; in Unix, names like "/us/patdata/patest.tsp" or even "/US/patdata/patest.TSP" (keeping in mind that on Unix there is a difference between upper case and lower case letters in filenames).*

### 1.3 Introduction to Basic TSP commands

Now that you know how to start TSP on your computer, what can you do with it? TSP is both a powerful programming language for the advanced user and an easy-to-use regression package for the novice. The *Reference Manual* has over 100 commands, but you only need a few of these to make effective use of TSP. Here is a brief introduction to some of the most commonly used commands (from

Chapter 3 onwards, we present the commands in a more structured format):

| | |
|---|---|
| FREQ | sets the frequency of the data series. |
| SMPL | sets the range of the data series. |
| PRINT | prints the data. |
| READ | reads the data. |
| GRAPH | graphs the data. |
| MSD | computes simple statistics for the data |
| INPUT | reads an input (tsp) file and executes it. |

For interactive use only:

| | |
|---|---|
| HELP | a help facility. |
| SHOW | displays the program variables. |

The first example in Chapter 1 of P&R consists of 8 observations of data on the grade point average and family income of economics students. To run this example in TSP, you need to read in these data, and graph them with a best fit line as shown in Figure 1.1 of the text.

### 1.4 Creating an Input File

To read in the data, create a simple input file that contains the data and the commands necessary for reading it. You can then input this file to TSP interactively, without having to retype the data every time you run the program. Here is the input file (called *ex1.tsp*):

```
FREQ N ;
SMPL 1 8 ;
?  Sample and Frequency are set; read in
data.
READ Y X ;
```

```
4 21 3 15 3.5 15 2 9 3 12 3.5 18 2.5 6 2.5
12;
```

The first statement, FREQ N, sets the frequency of the data at none. Other possible frequencies are: A for annual, Q for quarterly, M for monthly, or W for weekly. In this case, the data are not observations for calendar time, but for individuals, so N is used.

The next statement, SMPL 1 8, specifies the range of observations for our series. SMPL statements always contain a pair of numbers, the beginning and ending observation numbers of the data series to be used. In this case you specify observations 1 to 8, since there are data on 8 students.

Following the FREQ and SMPL statements, there is a "comment" statement, which is simply included to remind yourself what the program does. In TSP, anything that follows a ? on a line is printed out in the TSP program, but is not executed.

The READ statement is next. It assigns names X and Y to the numbers following it. You have two series, X and Y, and 8 observations per series, so 16 numbers follow. TSP gives an error message if the number of data points is not equal to the number of observations times the number of variables you try to read. Note the order of the data is Y(1), X(1), Y(2), X(2), and so forth. This is important.

As noted in 1.2, every statement in the program ends with a semicolon (;), including the list of 16 numbers. This is necessary when constructing a TSP input file for batch execution, since TSP uses the semicolons to tell when each statement ends. For example, the program

```
FREQ N ; SMPL 1 8 ;
```

```
READ Y X ;
4 21 3 15 3.5 15 2 9 3 12 3.5 18 2.5 6 2.5
12;
```

will execute exactly the same as the one shown above. TSP ignores the line boundaries when reading the input file.

## 1.5 Examples

### P&R Example 1.1 (Interactive Mode)

After you have made an input file containing the data (or have the data available in an external file such as a spreadsheet or a file produced by another program), you can run TSP interactively by using an INPUT (for tsp files) or READ (for other files) statement in your program to load the data. This example inputs data using the file you created above and then runs TSP interactively.[2] Start TSP interactively as described earlier and at the prompt "1" enter:

```
INPUT EX1
```

*Note when you use TSP interactively, no semicolons are needed and remember our warning about directory names on page 7. You may need to qualify EX1 in some way.*

TSP responds:

```
Do you want the output displayed at the
terminal (y/n)?
```

[2] For a sample batch program that does the same thing, see the end of this chapter. For an example that uses a spreadsheet file for input, see Chapter 3 onwards.

Type y to see the output on your screen (otherwise it is stored in an output file called *ex1.out*). The INPUT statement tells TSP to read and execute the *ex1.tsp* program, so the data on grade point average and family income will be read into TSP. To make sure this is the case, use the command:

```
SHOW SERIES
```

You should see the following table:

```
SERIES    Y    8 obs. from 1-8, no frequency
          X    8 obs. from 1-8, no frequency
```

This table summarizes the data you have read into the program; you may find this useful later when you have more complicated data files with many series and different sample sizes.

You can check your input file had the correct data by printing it out to the screen.

```
PRINT Y X
```

displays the series Y and X in a table, one column for Y and one for X.

Now type the command:

```
MSD Y X
```

You will see a table like this:

```
                     Univariate statistics
                     =====================

 Number of Observations: 8
```

```
       Mean          Std Dev    Minimum       Maximum
 Y    3.00000        0.65465    2.00000       4.00000
 X   13.50000        4.81070    6.00000      21.00000

       Sum           Variance   Skewness      Kurtosis
 Y   24.00000        0.42857    0.00000      -0.70000
 X  108.00000       23.14286    0.00000      -0.31111
```

You can use this table as another method of checking your data. Note that the range of the grade point average Y is [2.0,4.0], which is correct, and family income is [6,21]. The table also shows the mean and variance of each series, along with measures of the third moment (the skewness) and the fourth moment (kurtosis). These latter two measures are normalized so that they will be approximately zero if the data is normally distributed.

Finally, reproduce the best fit line displayed by P&R in Figure 1.2. To do this, obtain "least squares" estimates as suggested by P&R, using the OLSQ procedure in TSP:

```
OLSQ Y C X.
```

This statement estimates the equation $Y_i = \beta_1 + \beta_2*X_i +\varepsilon_i$. The first variable listed, $Y_i$, is the dependent variable (the variable on the left hand side) and the other variables are the independent variables. The variable C (for constant) is a special variable that for the estimate of the intercept $\beta_1$. See the output at the end of this chapter for the results. The details of this estimation will be discussed later in this handbook. For the moment, note that the best fitting line is *YFIT = 1.375 + .12037X.* You can make a graph with this best fit line on it with the following command:

```
GRAPH X Y @FIT
```

@FIT is a special variable available after every regression (least squares estimation) that contains the "fitted" values of the dependent variable. These values correspond to the best fit line through the data. The GRAPH statement plots X on the X-axis, and Y and @FIT on the Y-axis, so you see both the original data points and the line that OLSQ drew through them.

What if you don't remember the order of the arguments on the GRAPH statement later? At any time while you are running TSP, you can type a command like:

```
HELP GRAPH
```

and TSP will tell you the syntax for the command. Try HELP by itself to see what happens.

This completes our first try at running TSP. To quit the program, type:

```
EXIT      or   QUIT
```

**P&R Example 1.1 (Batch Mode)**

The sample output below is complete except for the output of the graph command, shown in the text (it is a graphical image and cannot be printed in the text output file). The output always begins with a header that tells you the version of TSP and the type of computer used. In subsequent sample outputs, we omit the header. After the header is a listing of the input program (helpful for checking commands). The title "EXECUTION" identifies the requested printed results. For example, this run requested the input data be printed (table of data), some univariate statistics be computed, and a simple regression be estimated (output entitled

"Equation 1"). Later you will learn the meaning of all the statistics printed with the regression.

At the end of the output, a printed table summarizes the number of errors and warning messages, and the memory used. In this case, we used very little of the available memory (2.1 megabytes out of 4.0 available, most of which was the program and not our data).

A batch program for the same interactive exercise follows. Its output can be found at the end of the chapter.

```
SMPL 1 8 ;
READ Y X ;
4 21 3 15 3.5 15 2 9 3 12 3.5 18 2.5 6 2.5 12
;
PRINT Y X ;
MSD Y X ;
OLSQ Y C X ;
GRAPH X Y @FIT ;
```

**TSP Concepts Introduced in this Chapter:**

FREQ<br>
SMPL<br>
INPUT<br>
READ<br>
SHOW SERIES<br>
MSD<br>
OLSQ<br>
GRAPH<br>
HELP<br>
QUIT<br>
END

## Program Output

### ***P&R Example 1.1***

```
                               TSP Version 4.3A
                           (12/05/95) DOS/Win      4MB
                      Copyright (C) 1995 TSP International
                               ALL RIGHTS RESERVED
                                01/22/96 3:51 PM
                In case of questions or problems, see your local TSP
               consultant or send a description of the problem and
               the associated TSP output to:
                                TSP International
                            P.O. Box 61015, Station A
                               Palo Alto, CA 94306
                                       USA

PROGRAM LINE
******************************************************************
 |      1  ?  P&R TSP Handbook Chapter 1 Example.
 |      2  smpl 1 8 ;
 |      3  read y x ;
 |      3  4 21
 |      3  3 15
 |      3  3.5 15
 |      3  2 9
 |      3  3 12
 |      3  3.5 18
 |      3  2.5 6
 |      3  2.5 12
 |      3  ;
 |      4  print y x ;
 |      5  msd y x ;
 |      6  olsq y c x ;
 |      7  graph x y @fit ;
 |      8  end ;
          EXECUTION

*********************************************************************
*

 Current sample:  1 to 8

                      Y             X
 1              4.00000      21.00000
 2              3.00000      15.00000
 3              3.50000      15.00000
 4              2.00000       9.00000
 5              3.00000      12.00000
 6              3.50000      18.00000
 7              2.50000       6.00000
 8              2.50000      12.00000


                              Univariate statistics
                              =====================

 Number of Observations: 8

              Mean       Std Dev       Minimum       Maximum
 Y          3.00000       0.65465       2.00000       4.00000
 X         13.50000       4.81070       6.00000      21.00000


              Sum      Variance      Skewness      Kurtosis
 Y         24.00000       0.42857       0.00000      -0.70000
 X        108.00000      23.14286       0.00000      -0.31111


                                      Equation   1
                                      ============

                       Method of estimation = Ordinary Least Squares


 Dependent variable: Y
 Current sample:  1 to 8
 Number of observations:  8

Mean of dependent variable = 3.00000      Adjusted R-squared =
.746142
Std. dev.of dependent var. =.654654  Durbin-Watson statistic =
3.23050
Sum of squared residuals =.652778   F-statistic zero slopes) =
21.5745
Variance of residuals = .108796  Schwarz Bayes. Info. Crit. =
-1.98610
Std. error of regression =.329843Log of likelihood function =
-1.32767
R-squared = .782407
```

```
            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C         1.37500       .368776       3.72856
 X         .120370       .025915       4.64483


**********************************************************************
*

 END OF OUTPUT.

 TOTAL NUMBER OF ERROR MESSAGES: 0

   MEMORY USAGE:    ITEM:    DATA ARRAY  TOTAL MEMORY
                   UNITS:  (4-BYTE WORDS) (MEGABYTES)
   MEMORY ALLOCATED         :    500000       4.0
   MEMORY ACTUALLY REQUIRED :       419       2.1
   CURRENT VARIABLE STORAGE :       324
```

# CHAPTER 2

## Statistical Review: Using the Random Number Generator and the Cumulative Distribution Function

Pindyck and Rubinfeld Chapter 2 reviews the important statistical ideas for the econometrics covered in the text. Many of these ideas can be illustrated or verified on the computer using a class of methods often referred to as simulation or Monte-Carlo simulation. Their basic idea is simple: by using a *random number generator* on the computer, one can simulate sets of random variables according to a specified probability distribution. These numbers are frequently termed *pseudo-random variables*, reflecting that they are not truly random, but have been computed using some numerical algorithm.

Manipulating these pseudo-random variables allows you to illustrate or verify statistical properties discussed in the text. Using different sample sizes (that is, generating a different number of draws of a particular random variable), you can examine the behavior of estimators and their variances as sample size increases. In this chapter, we introduce the use of TSP’s random number generator to simulate probability distributions and to compute various functions of basic random variables.

### 2.1 Generating Pseudo-Random Variables

TSP's pseudo-random number generator is called RANDOM and it can generate random numbers according to a wide variety of probability distributions: the normal, uniform, student's t, Cauchy, Laplace, Poisson, uniform, exponential, or an empirical distribution (EDF, based on actual data). Most of these distributions are for continuous random variables, with the exception of the Poisson (which generates nonnegative integers) and EDF (which places probability 1/N on each observation in the data).

Many other distributions can also be simulated using RANDOM and a bit of computation: for example, the TSP User's Guide contains a Chi-squared distribution example, and later in this chapter we demonstrate how to simulate the binomial distribution. Our first example, however, is the most familiar of the continuous probability distributions: the normal.

## 2.2 A Simple Example: the Normal Distribution

To see how simulation works, start TSP and choose the interactive option for execution. Enter the commands:

```
SMPL 1 10 ;
RANDOM X ;
HIST X ;
```

This produces a frequency distribution for the variable X. You can improve its appearance by using some options on the HIST command. MIN and MAX are the lower and upper bounds of the first and last cells. NBIN is the number of bins or cells.

```
HIST (MIN=-3, MAX=3, NBIN=20) X ;
```

What do these commands do? As you learned in Chapter 1, SMPL sets the number of observations to 10 and indexes them by the observation numbers 1 through 10. RANDOM generates a series X which has 10 observations, where each observation is a particular random draw from the probability distribution you specified. In this case, you did not specify any options on the RANDOM command, so the default probability distribution was used: a standard normal distribution with mean 0 and standard deviation and variance equal to 1. If X is generated by such a probability distribution, what do you expect its frequency distribution to look like? Does it?

Now increase the sample size and re-execute the 3 commands:

```
SMPL 1 100 ;
RANDOM X ;
HIST (MIN=-3, MAX=3, NBIN=20) X ;
```

How does this frequency distribution compare to the other one? Increase the sample size again, to 1000, and compare your result. The frequency distribution should get smoother and smoother, as you develop a better approximation to the normal distribution by taking larger and larger samples.

Our example generated random variables with the standard normal distribution (mean 0 and variance 1). RANDOM can also be used to generate normal random variables with arbitrary mean and variance. For example, suppose you wish to draw a series from a distribution with mean 5 and variance 25, use the command:

```
RANDOM (MEAN=5, STDEV=5) X55 ;
```

Note that the standard deviation (the square root of the variance) is supplied as an option rather than the variance itself.

### 2.3 Multivariate Normal Random Variables

To illustrate the concepts of covariance and independence, we need more than one random variable, that is, we need random variables with a multivariate distribution. TSP can generate independently distributed multivariate random variables for any distribution for which it knows the univariate distribution. For example:

```
SMPL 1 100 ;
RANDOM X Y ;
```

will generate two independent unit normal random variables, each with 100 observations. Because variables generated by separate RANDOM statements are usually independent, this is the same as the following series of commands:

```
SMPL 1 100 ;
RANDOM X ;
RANDOM Y ;
```

*(Advanced usage)* To generate dependent normal random variables, we must specify their covariance or correlation. For example, to generate 2 normal random variables with mean zero, variance one, and covariance (equal to the correlation in this case) 0.3, use the following commands:

```
READ (NROW=2, TYPE=SYM) COVX ;
1.0
0.3  1.0  ;
RANDOM (VCOV=COVX) X1 X2 ;
```

Note the way in which symmetric matrices are read into TSP: only the elements on and below the diagonal need to be supplied. (Why is the COVX matrix symmetric?)

## 2.4 Binary or Bernoulli random variables

Economists frequently use the simplest kind of discrete random variable: one that takes on the value 1 with probability p and the value 0 with probability 1-p. These variables describe things such as employment status (employed/unemployed), whether a particular purchase is made (buy/don't buy) and so forth. Although TSP does not provide the binary distribution with probability p as an option to RANDOM, you can easily generate one using the uniform distribution; for example, to generate a binary random variable with probability p = 0.5 (a fair coin toss), use:

```
RANDOM (UNIFORM) U ;
HEADS = U>0.5 ;
```

HEADS will be 1 for approximately half of the observations and 0 for the remainder. A useful fact about binary random variables is that:

$$E[X] = Prob\ [x=1] = p$$

where X is a realization of a binary random variable x. This means you can easily estimate p for a sample of binary random variables by taking their mean. You can use this in Problem 3 below (define a transformation of X that is 1 when X>30 and 0 otherwise; then estimate p for this new random variable).

## 2.5 Consistency (Advanced Material)

A consistent estimator is one that becomes arbitrarily close to the true value of the parameter as the sample size approaches infinity. Although you cannot use an infinite sample size on a finite computer, you can see how this property works using TSP. For example, consider the problem of measuring the true length of a table by taking many measurements and averaging them. Assume that the errors in each measurement are independently and normally distributed.

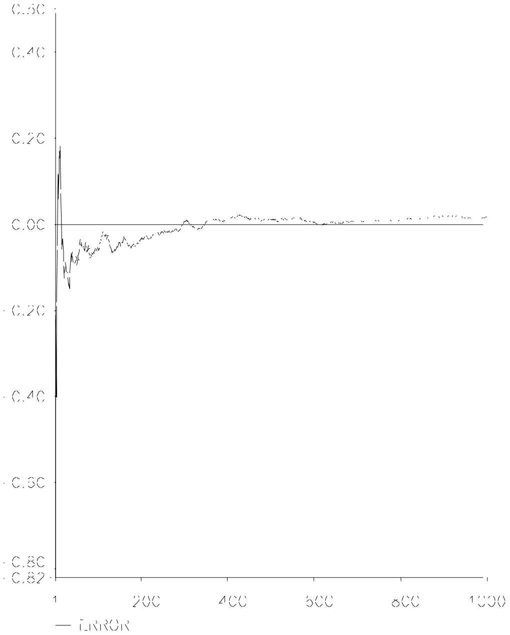
We can gain insight into the properties of the average as an estimator of the true length by running the following TSP program:

```
?  Generate 1000 random numbers with mean 72
?  inches and std. dev. of the error equal to
?  half an inch.
SMPL 1 1000 ;
RANDOM (MEAN=72, STDEV=0.5) LENGTH ;

?    Estimate   the   average   length   using
different ?  numbers of measurements.
?  First make a series with the obs. no. 1,2,
?  3,4,...., 1000.
TREND N ;

?  Then compute the sum of the lengths
?  recursively.
SUM = LENGTH ;  ? Initialize SUM.
SMPL 2 1000 ;
SUM = SUM(-1) + LENGTH ;

?  Compute the sample mean for 1 obs, 2 obs,
?  3 obs, and so forth.
SMPL 1 1000 ;
EST = SUM/N ;

?   Compute  the  error  in  each  estimate  and
plot
?  as a function of the sample size.
ERROR = EST-72 ;
```

```
PLOT   (ORIGIN,   TITLE="Convergence   of   the
Sample Mean") ERROR ;
```

This program is a bit more complicated than those used earlier, and requires a few comments. The sample average for N observations is the sum of the random variable over the N observations divided by N. The program computes this value for 1 observation, for 2 observations, for 3 observations, and so forth, up to 1000 observations on the length of the table. By plotting the error in the different estimates as a function of the sample size N, we can see how the estimator converges to its true value as N (the number of observations over which we average) gets large. This plot is shown below.

**Figure: Convergence of the Sample Mean**



## 2.6 Computing Probabilities: CDF (*Advanced*)

The Cumulative Distribution Function command (CDF) is very useful for performing hypothesis tests and doing statistical inference. CDF gives the probability associated with the argument for a wide range of probability distributions, such as the normal, t, F, chi-squared, and so forth. Using CDF is like looking up a statistic in the Statistical Tables in the back of P&R, but it is more accurate and convenient. CDF also covers additional distributions not included in the standard tables.

We will illustrate the use of CDF by solving problem 2.15 in P&R, where you test whether the variation of rainfall is 20 percent per year. We begin by reading in the data; note the use of FREQ to specify annual frequency, and the SMPL in terms of years.

```
FREQ A ; SMPL 78 87 ;

LOAD RAINFALL ;        ? Read in the data.
51.06 30.06 31.81 74.46 32.41
35.48 30.42 33.09 30.39 41.08 ;
```

Next we compute the logarithm of the rainfall totals and find its mean and variance using MSD.

```
LOGR = LOG(RAINFALL);
MSD LOGR ;
```

The mean log of rainfall is 3.6 and the variance of the logs is .086, corresponding to a standard deviation of about 30 percent per year. Under the null hypothesis that the variance of the logarithms is .04, the quantity (10-1)$s^2$/.04 is distributed as a $\chi^2$ random variable with 9 degrees of freedom. To compute the test statistic, we need the sample variance of the log of rainfall, which is printed in the table produced by MSD. Results in this table are stored under the special names @MEAN, @STDDEV, @MIN, @MAX, @SUM, @VAR, and so forth; we retrieve the variance in @VAR for our computation. SET is a special command for operation on single-valued variables (scalar values), in contrast to GENR, which operates on series (vectors of values).

```
SET N1 = @NOB-1 ;     ? Degrees of freedom.
SET TEST = N1*@VAR/.04 ;  ? Test-statistic.
```

After computing TEST, we use CDF  to look up the probability that we would have observed a value as high as .086 or higher if the true variance were .04. This probability (sometimes called the p-value of the test) is .022, so we would accept the hypothesis at the one percent level, but reject it at the 5 percent level. Note that our alternative hypothesis is slightly different from the one implied by the text: our alternative is "greater than .04", whereas the text seems to mean "not equal to .04". Ours is a *one-tailed* test.

```
CDF (CHISQ, DF=N1) TEST ;  ? Test of var=.04.
```

These commands gave the following output in TSP:

```
CHISQ(9)  Test Statistic: 19.44560, Upper tail area: .02166
```

### 2.7 Confidence Intervals

To illustrate the use of CDF with the t-distribution, we construct a confidence interval for the estimate of average rainfall over a ten-year period, 1978 to 1987. In order to construct such a confidence interval, we need the 5 and 95 percent values for the t-distribution with appropriate degrees of freedom, 9 in this case (N-1). CDF produces this with the inverse probability option:

```
CDF (T, DF=9, INVERSE) .05 T05 ;
```

T05 is a scalar variable that is returned with the argument of the t-distribution at the 5 percent level. Note that we do not need to get the 95 percent level separately because the t is a symmetric distribution centered around a known constant (zero).

T05 is the value of the t-statistic for a standardized variable. To obtain the confidence interval bounds in units corresponding to the logarithm of rainfall, we use the equation given in P&R (eq. 2.15):

$$X \pm t s_x / \sqrt{N} = 3.62 \pm T05\, s_x / \sqrt{10}$$

$s_x$ is the standard deviation of the log of rainfall, which was also printed in the MSD table we obtained in section 5. Here is the completed program for computation of the confidence interval:

```
SET LRLO = @MEAN-T05*@STDDEV/SQRT(@NOB) ;
SET LRHI = @MEAN+T05*@STDDEV/SQRT(@NOB) ;
PRINT @MEAN LRLO LRHI ;
```

Note how we used @NOB, the number of observations in the current sample for N. Variables beginning with @ are special variables stored automatically by TSP.

To get the confidence interval in terms of the original rainfall units, we use some more SET statements:

```
SET RLO = EXP(LRLO) ;
SET RHI = EXP(LRHI) ;
SET RMEAN = EXP(@MEAN) ;
```

```
PRINT RMEAN RLO RHI ;
```

These commands produced the following results:

```
          RMEAN     RLO       RHI
Value     37.33     30.25     46.07
```

Assuming rainfall follows a log-normal distribution, the estimated geometric mean of rainfall is 37.33 inches per year, with a 90% confidence interval of (30.25,46.07) inches per year. Note that this confidence interval is not symmetric around the mean, although the confidence interval in terms of logs was. This is a property of the log-normal distribution.

**TSP Concepts Introduced in this Chapter:**

@ variables
CDF
Functions: LOG, SQRT, EXP
HIST
Lagged Variables
RANDOM
SET
TREND

## Sample Programs

NOTE: If you run these programs on your computer, you will get slightly different results each time, because TSP chooses a "seed" for its pseudo-random number generator that is a function of the current clock time. The SEED option in the RANDOM command can override this if you want to generate exactly reproducible results in each run.

**1. Simulating the Normal Distribution**

```
PROGRAM
 LINE
**********************************************************
********
 |      1  ?  P&R TSP Handbook Example 1 - Simulating the
Normal Distribution.
 |      1  ?  Note that moments are fairly inaccurate with
only 100 obs.
 |      2  random x ;
 |      3  hist (min=-3,max=3,nbin=20) x ;
 |      4  msd x ;
 |      5
 |      5  random (mean=5,stdev=5) x55 ;
 |      6  msd x55 ;
 |      7
 |      7  random x y ;
 |      8  msd(cova) x y ;
 |      9
 |      9  read (nrow=2,type=sym) covx ;
 |      9  1.0
 |      9  0.3 1.0 ;
 |     10  random (vcov=covx) x1 x2 ;
 |     11  msd (cova) x1 x2 ;
 |     12
 |     12  end ;
          EXECUTION

**********************************************************
*********************
 Current sample:  1 to 100

 HISTOGRAM OF X

 Number of observations: 99
                 MINIMUM
MAXIMUM
```

```
               0.00000
13.00000

|-+---------------------------------------+-|
          -3   | *
   | 0
      -2.70000   | ****
   | 1
      -2.40000   | *
   | 0
      -2.10000   | *******
   | 2
      -1.80000   | *******
   | 2
      -1.50000   | **********
   | 3
      -1.20000   | *************
   | 4
      -0.90000   | **************************
   | 8
      -0.60000   | **************************************
   | 12
      -0.30000   | ***********************************
   | 11
  -1.11022D-16   | *****************************
   | 9
       0.30000   | ***********************************
   | 11
       0.60000   | **************************************
   | 12
       0.90000   |
***************************************** | 13
       1.20000   | *************
   | 4
       1.50000   | *************
   | 4
       1.80000   | *******
   | 2
       2.10000   | ****
   | 1
```

```
       2.40000   | *
   | 0
       2.70000   | *
   | 0

|-+------------------------------------+-|
               0.00000
13.00000
                 MINIMUM
MAXIMUM

                              Univariate statistics
                              =====================

 Number of Observations: 100

                          Mean       Std Dev       Minimum
      Maximum
 X                     0.19561       0.97601      -2.45147
      3.04590


                           Sum      Variance      Skewness
     Kurtosis
 X                    19.56146       0.95260     -0.075501
      0.10297


                              Univariate statistics
                              =====================

 Number of Observations: 100

                          Mean       Std Dev       Minimum
      Maximum
 X55                   5.06724       4.85640      -8.08153
     18.71418
```

```
                              Sum      Variance      Skewness
    Kurtosis
X55                   506.72369      23.58462      -0.19168
     0.14630


                             Results of Covariance procedure
                             ===============================

Number of Observations: 100

                             Mean       Std Dev       Minimum
     Maximum
X                      0.077477       1.01590      -1.98632
     2.50249
Y                     0.0054340       1.01955      -2.35322
     2.82015


                              Sum      Variance      Skewness
    Kurtosis
X                       7.74769       1.03206      0.035257
    -0.49721
Y                       0.54340       1.03948       0.15182
    -0.21902

                                    Covariance Matrix

                                X             Y
X                       1.03206
Y                      0.020709       1.03948


                             Results of Covariance procedure
                             ===============================

Number of Observations: 100

                             Mean       Std Dev       Minimum
     Maximum
```

```
 X1                  0.013607       1.04325      -2.73527
      2.18510
 X2                  -0.12298       1.07034      -3.33170
      2.39302


                          Sum      Variance      Skewness
     Kurtosis
 X1                   1.36065       1.08837      -0.27341
     -0.16776
 X2                 -12.29841       1.14563     -0.050554
     0.082977

                               Covariance Matrix

                           X1            X2
 X1                   1.08837
 X2                   0.41875       1.14563


**********************************************************
*********************
```

**2. Estimating a Mean Recursively - Consistent Estimation.**

```
PROGRAM
 LINE
**********************************************************
*********
 |      1  options limwarn=25 ;
 |      2  ?
 |      2  ?   Sample Program for P&R Handbook Chapter 2.
 |      2  ?   Consistent Estimator of the Table Length
 |      2
 |      2  SMPL 1 1000 ;
 |      3
 |      3  ?  Generate 1000 random numbers with mean 72
 |      3  ?  inches and std. dev. of the error equal to
 |      3  ?  half an inch.
 |      3
 |      3  RANDOM (MEAN=72,STDEV=0.5) LENGTH ;
 |      4
 |      4  ?  Estimate the average length using different
 |      4  ?  numbers of measurements.
 |      4
 |      4  trend n ;                ? Make a variable
with the obs. no.
 |      5  sum = length ;
 |      6  smpl 2 1000 ;
 |      7  sum = sum(-1)+length ;   ? Recursively compute
the sum over n obs.
 |      8  smpl 1 1000 ;
 |      9  est = sum/n ;            ? Sample average
estimator
 |     10  error = est-72 ;         ? Error in the
estimator
 |     11
 |     11  plot (origin,title="Convergence of the Sample
Mean") error ;
 |     11  ?plot (title="Convergence of the Sample Mean")
est ;
 |     11
 |     11  stop ; end ;

          EXECUTION

*********************************************************
*******************

 Current sample:  1 to 1000

 Current sample:  2 to 1000
 NOTE:  Dynamic GENR for SUM


 Current sample:  1 to 1000
```

```
.........................  Plot output shown in Text
.......................

*********************************************************
*******************
```

**3. Testing the Log Variance of Rainfall.**

```
PROGRAM
 LINE
*********************************************************
*********
 |      1  options limwarn=25 ;
 |      2  ?
 |      2  ?   Sample Program for P&R Handbook Chapter 2.
 |      2  ?   Testing for the log variance of rainfall.
 |      2
 |      2  freq a ; smpl 78 87 ;
 |      4  load rainfall ;
 |      4  51.06 30.06 31.81 74.46 32.41
 |      4  35.48 30.42 33.09 30.39 41.08 ;
 |      5  logr = log(rainfall) ;
 |      6  msd logr ;
 |      7  set n1 = @nob-1 ;
 |      8  set test = n1*@var/.04 ;
 |      9  cdf(chisq,df=n1) test ;
 |     10  cdf(t,df=n1,inverse) .05 t05 ;
 |     11  cdf(t,df=n1,inverse) .95 t95 ;
 |     12  set lrlo = @mean-t05*@stddev/sqrt(@nob) ;
 |     13  set lrhi = @mean+t05*@stddev/sqrt(@nob) ;
 |     14  print @mean lrlo lrhi ;
 |     15  set rlo = exp(lrlo) ;
 |     16  set rhi = exp(lrhi) ;
 |     17  set rmean = exp(@mean) ;
 |     18  print rmean rlo rhi ;
 |     19  stop ; end ;
          EXECUTION
```

```
*********************************************************
*******************

 Current sample:  1978 to 1987

                              Univariate statistics
                              =====================

 Number of Observations: 10

                         Mean       Std Dev       Minimum
      Maximum
 LOGR                 3.61976       0.29398       3.40320
      4.31026


                          Sum      Variance      Skewness
     Kurtosis
 LOGR                36.19764      0.086425       1.77446
      2.79198

 CHISQ(9)  Test Statistic: 19.44560, Upper tail area:
.02166

                  @MEAN          LRLO          LRHI
 Value          3.61976       3.40946       3.83007


                  RMEAN           RLO           RHI
 Value         37.32877      30.24899      46.06556


*********************************************************
*******************
```

# CHAPTER 3

## The Two-Variable Regression Model

This chapter introduces two-variable linear regression models, which consist of a linear relationship between the independent variable, a constant and an independent variable. These models have the general specification:

$$Y_i = a + b*X_i + e_i. \qquad (1)$$

The two-variable model is the simplest econometric model that can be constructed. We solve Examples 3.2 and 3.3 to show how TSP estimates the parameters of equation (1) using ordinary least squares (OLS).

Using a typical TSP program, we show how to: size the output to fit a page or screen, transform or rename series, run a simple OLS regression, and interpret error and warning messages. We then use Examples 3.2 and 3.3 from P&R, to illustrate some TSP commands.

### 3.1 Formatting the Output

You can use the OPTIONS command to choose the format of your output. Most of the choices concentrate on printing formats. Although it may be most convenient to place OPTIONS at the beginning of your program, it can be placed anywhere. The exercises in this chapter show two alternatives, CRT and LIMWARN. CRT sizes the output for the screen, and LIMWARN limits the number of warning messages generated.

### 3.2 Reading the Data

You follow two steps to read the data. First, you specify the frequency and sample period. TSP uses the FREQ command to indicate frequency. FREQ has following options: annual (A), quarterly (Q), monthly (M), weekly (W), and cross-section or no-frequency (N). You use the SMPL command to specify the sample period.

In the second step, you actually enter the data set. This can be done in several different ways. One way is to use the INPUT command, as we did in Chapter 1 to read in a very simple data set. Another method is to use the READ command to enter data already stored in a file.[3] READ requires the name of the file containing the data along with its format and the list of variables in the file. An example of the most common use of READ (where the data is in a spreadsheet) follows.

```
READ(file='filename', format=lotus) x y z;
```

### 3.3 Printing or Writing the Data

The PRINT command is synonymous with the WRITE command. They are used to send variables to the screen, output file, or an external file. To only see the data on screen, you just need to include the list of variables. To store the variables in a file, you also need to specify the filename. The format of the command is:

```
WRITE (FILE='filename string') list of
vars.;.
```

[3] Other ways to read the data include IN (when the data is in a TSP databank) or FETCH (for micro TSP or EVIEWS series).

*Note: You can write a variety of file formats; see the Reference Manual for details.*

These commands are useful for checking that you loaded the correct data. TSP displays the list of series typed after the PRINT statement together with their respective dates. PRINT and WRITE are the inverse of READ.

### 3.4 Transforming the Data

Once the data is loaded, you can transform existing series into new series using the GENR command. Both mathematical and logical operations may be applied to the series. Lags and leads can be also be used as part of transformations. The name of the new series is written on the left hand side of the equation. The right hand side contains the expression to be calculated.

This chapter only uses GENR to rename the series. More sophisticated operations are performed in following chapters. As you can see from the examples below, you do not need to include the word GENR. Some examples of the GENR command are:

```
GENR Y=X; or Y=X;
```

makes series Y equal to series X.

```
GENR Y=X+Z; or Y=X+Z;
```

makes series Y equal to the sum of series X and Z.

```
GENR Y=LOG(X); or Y=LOG(X);
```

makes series Y equal to the logarithm of X.

### 3.5 Calculating a Linear Regression

To calculate linear regressions, TSP uses the OLSQ command. TSP estimates the parameters by ordinary least squares (OLS). In Chapter 1 we used this command to calculate simple linear regressions (one independent variable). But from Chapter 4 onwards we estimate multiple regressions (several independent variables). By changing its options, OLSQ also performs weighted least square regressions.

The most common structure of this command is:

```
OLSQ dep-var indep-var;
```

The inclusion of a constant is also recommended, so the letter C (or the word CONSTANT) should be part of the list of independent variables.

The OLSQ procedure generates statistics which are printed in the output. Two examples are included at the end of this chapter. OLSQ also stores most of the output results under special names for later use in the program. These variables are frequently used to form test statistics. Following chapters show the use of the stored variables.

### 3.6 Finishing the Execution

The STOP command tells TSP to stop running. When variables are marked for storage on output databanks, they are written to the bank before the program stops. STOP can be included anywhere in your program. It is useful for testing part of the program before running the whole thing. Executing only part of the program makes the output file smaller and saves computing time. When issued at the end of the program, before the END command, STOP is redundant since END implies STOP. The END command finishes the TSP program section and the TSP data section.

### 3.7 Interpreting Errors and Warning Messages

Often programs do not work perfectly the first time. At the bottom of the program TSP prints a summary of the warning and error messages printed throughout the output.

Warning messages alert you to changes from the expected execution of the program. A common example is the use of missing data. Some TSP commands ignore observations for which data is missing. TSP writes a warning message each time it drops observations because of missing data. The number of warning messages can be controlled with OPTIONS by making LIMWARN equal to the desired number.

There are different types of programming errors. The easiest ones are syntax errors. When a command is incorrectly typed or the format of the command is wrong, TSP prints an error message indicating the line where the error was found and does not execute that line. TSP continues running the program as long as the following commands do not depend on the one that contained the error. When the number of error messages exceeds 25, TSP stops.

The most difficult errors to find are the logical errors, since they are sometimes not detected by TSP. An example is a statement that is correct in terms of syntax, but badly specified in terms of econometrics. TSP only detects logical errors when it cannot execute a command.

## 3.8 Examples

### P&R Example 3.2:

This example studies the relationship between consumption and disposable income. The coefficient for the independent variable (disposable income) is expected to be positive. The example uses quarterly data from 1959:1 to 1995:2.

The use of CRT in the OPTIONS command sets the output on a 24 line by 80 character screen. LIMWARN=0 tells TSP to print no warning messages since we already know there is missing data, and TSP would otherwise print a set of uninformative messages. The two options can be grouped in the same line, as shown in the next example.

```
OPTIONS CRT;
OPTIONS LIMWARN=0;
```

The following two commands specify the sample structure.

```
FREQ Q;
SMPL 41:1 95:2;
```

The data is in the file EX43.XLS, which contains the variables GC and GYD:

```
READ(FILE='EX43.XLS', FORMAT=EXCEL) GC GYD;
```

We rename the variables to match the names used in P&R. The following two statements make the loaded series equal to new series. These two commands can be written using GENR, i.e. GENR CONSUM=GC and GENR YD=GYD, as done in the next example. Note that the use of the word GENR is redundant..

```
CONSUM=GC;
YD=GYD;
```

Even though the file contains data from the first quarter of 1941, we need to restrict the sample period to replicate the example in the book. We do that with the SMPL statement, which can be used anywhere in the program.

```
SMPL 59:1 95:2;
```

To calculate linear regressions with and without a constant we use:

```
OLSQ CONSUM C YD;
```

To stop the program execution we use two commands.

```
END; STOP;
```

**P&R Example 3.3:**

This example studies the relationship between the retail auto sales S, which is the dependent variable, and the level of aggregate wages W, which is the independent variable. The coefficient of the independent variable is expected to be positive, since higher wages tend to be positively correlated with higher auto sales. The example uses quarterly data, from the first quarter of 1946 through the second quarter of 1995. The following commands set the sample structure. The data is in the file EX33.XLS, an Excel file with two variables GCDAN and GWY.

```
OPTIONS CRT, LIMWARN=0;
```

```
FREQ Q;
SMPL 46:1 95:2;
READ(FILE='EX33.XLS',   FORMAT=EXCEL)   GCDAN
GWY;
GENR S=GCDAN;
GENR W=GWY;
SMPL 59:1 95:2;
OLSQ S C W;
OLSQ S W;
STOP;END;
```

**TSP Concepts Introduced in this Chapter:**

Error Messages
GENR
OLSQ
OPTIONS
PRINT
STOP
Warning Messages

## Sample Programs

### ***Example 3.2***

```
PROGRAM
 LINE
***********************************************************
********
 1  options crt;
 2  options limwarn=0;
 3  freq q;
 4  smpl 41:1 95:2;
 5  read(file='ex43.xls', format=excel) gc gyd;
 6  consum=gc;
 7  yd=gyd;
 8  smpl 59:1 95:2;
 9  olsq consum c yd;
 10  end; stop;
          EXECUTION

**********************************************************
*********************

 Current sample:  1941:1 to 1995:2

 Current sample:  1959:1 to 1995:2


                                      Equation   1
                                      ============

                       Method of estimation = Ordinary
Least Squares


 Dependent variable: CONSUM
 Current sample:  1959:1 to 1995:2
 Number of observations:  146

  Mean of dependent variable = 1757.38
Adjusted R-squared = .999436
 Std. dev. of dependent var. = 1386.97        Durbin-
Watson statistic = .370197
    Sum of squared residuals = 156151.      F-statistic
(zero slopes) = 257086.
       Variance of residuals = 1084.38     Schwarz Bayes.
Info. Crit. = 7.04324
    Std. error of regression = 32.9300    Log of
likelihood function = -716.338
                   R-squared = .999440

            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C         -27.5296      4.45192       -6.18376
 YD        .928698       .183162E-02   507.036


**********************************************************
*********************

 END OF OUTPUT.
```

## ***Example 3.3***

```
PROGRAM
 LINE
*********************************************************
 1  options crt, limwarn=0;
 2  freq q;
 3  smpl 46:1 95:2;
 4  read(file='ex33.xls', format=excel) gcdan gwy;
 5  genr s=gcdan;
 6  genr w=gwy;
 7  smpl 59:1 95:2;
 8  olsq s c w;
 9  olsq s w;
 10  stop; end;
          EXECUTION
 *********************************************************

 Current sample:  1946:1 to 1995:2

 Current sample:  1959:1 to 1995:2


                                      Equation   1
                                      ============

                       Method of estimation = Ordinary
Least Squares
```

```
 Dependent variable: S
 Current sample:  1959:1 to 1995:2
 Number of observations:  146

Mean of dependent variable = 50.1965
Adjusted R-squared = .904731
Std. dev. of dependent var. = 31.5969
Durbin-Watson statistic = .313547
Sum of squared residuals = 13696.2
F-statistic (zero slopes) = 1378.01
Variance of residuals = 95.1125
Schwarz Bayes. Info. Crit. = 4.60954
Std. error of regression = 9.75257
Log of likelihood function = -538.678
R-squared = .905388

            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C         9.47986       1.36181       6.96122
 W         .030774       .829015E-03   37.1216


                                      Equation   2
                                      ============

                       Method of estimation = Ordinary
Least Squares


 Dependent variable: S
 Current sample:  1959:1 to 1995:2
 Number of observations:  146

Mean of dependent variable = 50.1965
R-squared = .905388
Std. dev. of dependent var. = 31.5969
Adjusted R-squared = .905388
Sum of squared residuals = 18305.2
Durbin-Watson statistic = .235181
```

```
Variance of residuals = 126.243
Schwarz Bayes. Info. Crit. = 4.86547
Std. error of regression = 11.2358
Log of likelihood function = -559.852

            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 W         .035422       .566073E-03   62.5757

 *********************************************************

 END OF OUTPUT.
```

# CHAPTER 4

## The Multiple Regression Model

Chapter 4 of P&R generalizes the model presented in chapter 3 to include more than one independent variable in the regression. In our chapter 4, we illustrate how to estimate such a model and introduce another group of TSP commands. First, we show how to transform the series to generate other series using more sophisticated examples. Section 5 demonstrates the SET command with saved variables to calculate tests not reported in standard TSP outputs.

The models used in this chapter consist of a dependent variable regressed on a constant and a set of independent variables. They have the following specification:

$$Y_i = \alpha\, \beta_0 + \beta_1 {*} X_{1,i} + \ldots\ldots. + \beta_\kappa {*} X_{\kappa,i} + \varepsilon_i. \qquad (1)$$

where $X_{1,i},\ldots.,X_{\kappa,i}$ is the set of k independent variables.

We solve examples 4.1, 4.2, 4.3 and 4.5 to show how TSP computes a multiple regression.

### 4.1 Reading Multiple Files

Sometimes data is in more than one file. You can easily obtain data from multiple sources by issuing multiple READ commands. Each READ statement must be preceded by the proper FREQ and SMPL specification since each data set may contain a different structure in terms of its frequency and sample size. Example 4.1 shows how two data files are read in the same program.

### 4.2 Converting the Data Frequency

Sometimes the series have different frequencies. If you need all the data (for example to do OLS), you have to make the data homogeneous. Either the high-frequency data is converted into the low-frequency format or the low-frequency data is converted into the high frequency format. Before issuing the CONVERT command you must specify the desired frequency.

Data conversion can be done in various ways. When converting high-frequency data into the low-frequency format: data can be averaged over the period; the first, mid or last observation of the period can be taken; or the sum of observations can be used. When converting low-frequency data into the high-frequency format, TSP provides two options, a linear interpolation or a data duplication. Other types of transformations are possible, but are not TSP options. You have to do these “manually” using the GENR command.

The format of the convert command follows.

```
CONVERT(AVERAGE or FIRST or MID or LAST or
SUM, INTERPOL)seriesname;
```

### 4.3 Running Multiple Regressions

Multiple regressions are run in the same way as simple regressions. The only difference is in the number of independent variables. In simple regressions, we have one independent variable besides the constant. In multiple regressions, we can include as many variables as we want:

```
OLSQ dep-var list-of-indep-vars;
```

### 4.4 Using Program Variables (@)

After each regression, TSP saves the values of the variables under reserved names. These variables include scalars and series. For example, the estimated constant is a scalar saved under @COEF(1), while the fitted residuals are series saved under @RES. We used @FIT, the series of fitted values, in Chapter 2.

These scalars and coefficients can be used later in the program. For example, the scalars are frequently used to form tests. The series can be used in further regressions or can be plotted or printed, as done in Chapter 1.

Not all the commands generate program variables. The TSP *Reference Manual* lists each command's program variables.

## 4.5 SET Command

The SET command performs computations on scalar variables and single elements of time series or matrices. The format of the command consists of a new scalar's name, equal (=) to an expression or a program variable. Here are some examples:

```
SET a=3+4;
SET b=@COEFF(2);
SET R2=@RSQ;
```

This command is useful for computing test statistics, as shown in the next chapter.

## 4.6 Changing the Sample

TSP has several ways to change the sample of observations. Using SMPL is convenient when working with dated series; you can restrict the sample to certain dates. When you work with cross section data, you need to restrict the sample according to other criterion, since the data usually does not contain a particular order. For example, you may need to separate females and males, or industrial and agricultural firms. You will learn how to do this in later chapters using SELECT and SIMPLIF.

It is usually best to read your entire data file and then restrict the sample size in the program. You may want to be able to select different samples. For example, you could be interested in estimating for particular periods or performing tests comparing coefficients for different sample periods.

The format of SMPL follows.

```
SMPL beginning period   ending period;
```

You can insert the command SMPL at any time in the program. It is usually placed before commands like OLSQ. Remember that the sample is changed from the point you issued SMPL. If you need to retrieve the previous sample you need to reissue the command.

Examples of SMPL are:

| | |
|---|---|
| `SMPL 88 94;` | for annual data. |
| `SMPL 88:2 94:10;` | for monthly data. |
| `SMPL 5 75;` | for non-dated data, it selects the 5th. through the 75th observation. |
| `SMPL 2 2 7 10;` | it selects the 2th and 7-10th. observations. |

### 4.7 Examples

**P&R Example 4.1:**

This example estimates the relationship between quarterly personal consumption of new autos and quarterly personal income, 3-month Treasury bill rate and the quarterly Consumer Price Index. The new feature introduced is the data conversion. Consumption of new autos is a quarterly series, while the rest are monthly. The equation is estimated with a quarterly frequency, therefore the monthly data need to be converted to lower frequency.

```
FREQ Q;
SMPL 59:1 95:2;
READ(FILE='EX33.XLS',    FORMAT=EXCEL)    GCDAN
GWY;
S=GCDAN;
TREND TIME;
FREQ M;
SMPL 75:1 95:9;
READ(FILE='EX41.XLS',    FORMAT=EXCEL)    GMPY
FYGN3 PUNEW;
YP=GMPY;
R=FYGN3;
CPI=PUNEW;
```

The command CONVERT does the data transformation. We average monthly personal income, the T-bill rate and the consumer price index. We first indicate that the data will be converted to the quarterly form by issuing the command FREQ.

```
FREQ Q;
CONVERT(AVERAGE) YP;
CONVERT(AVERAGE) R;
CONVERT(AVERAGE) CPI;

SR=S/CPI;
YPR=YP/CPI;
RR=R/CPI;
SMPL 75:1 95:2;
OLSQ SR C YPR RR;
```

**P&R Example 4.2:**

This example studies the movement of monthly interest rates. The independent variables are the Federal Reserve Board index of industrial production, the nominal money supply and the producer price index. The example uses more complex series transformation than used in the previous chapter.

```
FREQ M;
SMPL 59:1 96:2;
READ(FILE='EX42.XLS', FORMAT=EXCEL) FYGN3 IP
FM2 PW;
R=FYGN3;
M2=FM2;
```

GM2 and GPW are the rates of growth of M2 and PW.

```
GM2=(M2-M2(-1))/M2(-1);
GPW=(PW-PW(-1))/PW(-1);

SMPL 60:1 95:8;
OLSQ R C IP GM2 GPW(-1);
```

**P&R Example 4.3:**

This example estimates three models, two consumption functions and a savings function. No new TSP concepts are introduced here.

```
FREQ Q;
SMPL 54:1 95:2;
READ(FILE='EX32.XLS', FORMAT=EXCEL) GC GYD;
CONSUM=GC;
Y=GYD;
S=Y-CONSUM;
OLSQ CONSUM C Y;
OLSQ CONSUM C Y CONSUM(-1);
OLSQ S C Y;
```

**P&R Example 4.5:**

This example estimates the monthly sales of durable goods. The independent variables are retail inventory of department stores in durable goods, the inventory sales ratios for all durable goods (retail stores), the open market rate on prime 6-month commercial paper, the average hourly gross earnings of workers and the consumer price index of durable goods. The independent variables enter in the estimated equation after some transformations are performed.

```
FREQ M;
SMPL 67:1 95:8;
READ(FILE='EX45.XLS', FORMAT=EXCEL) RTDR
IVRDR
FYCP LEH PUCD;
SD=RTDR;
DI=IVRDR;
IS=IVRDR/RTDR;
I=FYCP;
E=LEH;
P=PUCD;
```

In order to compute the standardized coefficients and their elasticities, we need to compute summary statistics of all the variables.

```
MSD CDI(-6) IS(-1) I(-1) E(-1) P(-1);
```

The OLSQ command yields the usual regression results.

```
OLSQ SD C DI(-6) IS(-1) I(-1) E(-1) P(-1);
```

The standardized coefficients can be obtained by multiplying the OLS coefficients (@COEF(#)) by the standard deviation of the independent variable (@STDDEV(#)) over the standard deviation of the dependent variable (@SDEV).

@COEF(#) contains the value of the specified estimated coefficient, and @STDEV(#) contains the standard deviation of the referred estimate.

```
SET B2STD=@COEF(2)*@STDDEV(2)/@SDEV;
SET B3STD=@COEF(3)*@STDDEV(3)/@SDEV;
SET B4STD=@COEF(4)*@STDDEV(4)/@SDEV;
SET B5STD=@COEF(5)*@STDDEV(5)/@SDEV;
SET B6STD=@COEF(6)*@STDDEV(6)/@SDEV;
PRINT B2STD B3STD B4STD B5STD B6STD;
```

The elasticities can be obtained by multiplying the OLS coefficients (@COEF(#)) by the mean of the independent variable (@MEAN(#)) over the mean of the dependent variable (@YMEAN).

```
SET B2EL=@COEF(2)*@MEAN(2)/@YMEAN;
SET B3EL=@COEF(3)*@MEAN(3)/@YMEAN;
SET B4EL=@COEF(4)*@MEAN(4)/@YMEAN;
SET B5EL=@COEF(5)*@MEAN(5)/@YMEAN;
SET B6EL=@COEF(6)*@MEAN(6)/@YMEAN;
PRINT B2EL B3EL B4EL B5EL B6EL;
```

*Note to advanced users: these two steps can be done more easily if you are familiar with the MAT command:*

```
MAT BSTD=@COEF % @STDDEV/@SDEV;
MAT BEL=@COEF % @MEAN/@YMEAN;
PRINT BSTD BEL;
```

To compute the partial correlation coefficient between SD and DI, first regress SD on all the independent variables except DI. Then, regress DI on all the other independent variables. Finally, use CORR to obtain the simple correlation between the residuals of the two regressions. Repeat these steps to calculate the other partial correlation coefficients.

```
OLSQ(SILENT) SD C IS(-1) I(-1) E(-1) P(-1);
Y=@RES;
OLSQ(SILENT) DI(-6) C IS(-1) I(-1) E(-1)
P(-1);
X=@RES;
CORR Y X;
.....
.....
OLSQ(SILENT) SD C DI(-6) IS(-1) I(-1) E(-1);
Y=@RES;
OLSQ(SILENT) P(-1) C DI(-6) IS(-1) I(-1)
E(-1);
X=@RES;
CORR Y X;
```

**TSP Concepts Introduced in this Chapter:**

@COEF
@CORR
@MEAN
@NCOEF
@RSQ
@SDEV
@SES
@STDDEV
@YMEAN
CONVERT
CORR
MSD

## Sample Program

### ***Example 4.1***

```
PROGRAM
 LINE
**********************************************************
********
 1  options crt;
 2  options limwarn=0;
 3  freq q;
 4  smpl 59:1 95:2;
 5  read(file='ex33.xls', format=excel) gcdan gwy;
 6  s=gcdan;
 7  trend time;
 8  freq m;
 9  smpl 75:1 95:9;
 10  read(file='ex41.xls', format=excel) gmpy fygn3 punew;
 11  yp=gmpy;
 12  r=fygn3;
 13  cpi=punew;
 14  freq q;
 15  convert(average) yp;
 16  convert(average) r;
 17  convert(average) cpi;
 18  sr=s/cpi;
 19  ypr=yp/cpi;
 20  rr=r/cpi;
 21  smpl 75:1 95:2;
 22  olsq sr c ypr rr;
 23  end; stop;
          EXECUTION

**********************************************************
********************

Current sample:  1959:1 to 1995:2

 Current sample:  1959:1 to 1995:6
 Current sample:  1975:1 to 1995:9

 Current sample:  1975:1 to 1995:3

 Current sample:  1975:1 to 1995:2


                                      Equation   1
                                      ============

                       Method of estimation = Ordinary
Least Squares


 Dependent variable: SR
 Current sample:  1975:1 to 1995:2
 Number of observations:  82

  Mean of dependent variable = .693843
Adjusted R-squared = .062125
 Std. dev. of dependent var. = .115099        Durbin-
Watson statistic = .390933
    Sum of squared residuals = .981551      F-statistic
(zero slopes) = 3.68271
       Variance of residuals = .012425    Schwarz Bayes.
Info. Crit. = -4.26412
    Std. error of regression = .111466     Log of
likelihood function = 65.0860
                   R-squared = .085282

            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C         .745708       .183197       4.07053
 YPR       .367410E-03   .470890E-02   .078024
 RR        -.810105      .523782       -1.54665


**********************************************************
********************
```

## ***Example 4.2***

```
PROGRAM
 LINE
**********************************************************
********
 1  options crt;
 2  options limwarn=0;
 3  freq m;
 4  smpl 59:1 96:2;
 5  read(file='ex42.xls', format=excel) fygn3 ip fm2 pw;
 6  r=fygn3;
 7  m2=fm2;
 8  gm2=(m2-m2(-1))/m2(-1);
 9  gpw=(pw-pw(-1))/pw(-1);
 10  smpl 60:1 95:8;
 11  olsq r c ip gm2 gpw(-1);
 12  end; stop;
          EXECUTION

**********************************************************
*********************

Current sample:  1959:1 to 1995:8

 Current sample:  1960:1 to 1995:8


                                      Equation   1
                                      ============


                       Method of estimation = Ordinary
Least Squares
```

```
 Dependent variable: R
 Current sample:  1960:1 to 1995:8
 Number of observations:  428

  Mean of dependent variable = 6.14576
Adjusted R-squared = .210816
 Std. dev. of dependent var. = 2.79281        Durbin-
Watson statistic = .183733
    Sum of squared residuals = 2609.93      F-statistic
(zero slopes) = 39.0218
       Variance of residuals = 6.15549     Schwarz Bayes.
Info. Crit. = 1.86458
    Std. error of regression = 2.48103    Log of
likelihood function = -994.208
                   R-squared = .216361

            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C         1.21408       .551692       2.20065
 IP        .048353       .550301E-02   8.78663
 GM2       140.326       36.0385       3.89378
 GPW(-1)   104.588       17.4422       5.99629


**********************************************************
*********************
```

## ***Example 4.3***

```
PROGRAM
 LINE
**********************************************************
********
```

```
 1  options crt;
 2  options limwarn=0;
 3  freq q;
 4  smpl 54:1 95:2;
 5  read(file='ex32.xls', format=excel) gc gyd;
 6  consum=gc;
 7  y=gyd;
 8  s=y-consum;
 9  olsq consum c y;
 10  olsq consum c y consum(-1);
 11  olsq s c y;
 12  end; stop;
          EXECUTION

**********************************************************
*********************

                                      Equation   1
                                      ============

                       Method of estimation = Ordinary
Least Squares


 Dependent variable: CONSUM
 Current sample:  1954:1 to 1995:2
 Number of observations:  166

  Mean of dependent variable = 1578.29
Adjusted R-squared = .999479
 Std. dev. of dependent var. = 1387.85        Durbin-
Watson statistic = .350343
    Sum of squared residuals = 164641.      F-statistic
(zero slopes) = 316407.
       Variance of residuals = 1003.91     Schwarz Bayes.
Info. Crit. = 6.96113
    Std. error of regression = 31.6845    Log of
likelihood function = -808.205
                   R-squared = .999482
```

```
            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C         -21.6063      3.75998       -5.74640
 Y         .926893       .164781E-02   562.501


                                      Equation   2
                                      ============

                       Method of estimation = Ordinary
Least Squares


 Dependent variable: CONSUM
 Current sample:  1954:2 to 1995:2
 Number of observations:  165

  Mean of dependent variable = 1586.42
 Std. dev. of dependent var. = 1388.10
    Sum of squared residuals = 18904.2
       Variance of residuals = 116.693
    Std. error of regression = 10.8024
                   R-squared = .999940
          Adjusted R-squared = .999939
     Durbin-Watson statistic = 1.80742
                  Durbin's h = 1.28750
      Durbin's h alternative = 1.23282
   F-statistic (zero slopes) = .135389E+07
  Schwarz Bayes. Info. Crit. = 4.83403
  Log of likelihood function = -625.273

              Estimated    Standard
 Variable    Coefficient     Error       t-statistic
 C           -.039664      1.42974       -.027742
 Y           .180490       .021153       8.53244
 CONSUM(-1)  .817037       .023145       35.3010


                                      Equation   3
                                      ============
                       Method of estimation = Ordinary
Least Squares


 Dependent variable: S
 Current sample:  1954:1 to 1995:2
 Number of observations:  166

  Mean of dependent variable = 147.795
Adjusted R-squared = .922621
 Std. dev. of dependent var. = 113.903        Durbin-
Watson statistic = .350343
    Sum of squared residuals = 164641.      F-statistic
(zero slopes) = 1968.37
       Variance of residuals = 1003.91     Schwarz Bayes.
Info. Crit. = 6.96113
    Std. error of regression = 31.6845    Log of
likelihood function = -808.205
                   R-squared = .923090

            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C         21.6063       3.75998       5.74640
 Y         .073107       .164781E-02   44.3663


**********************************************************
*********************
```

## ***Example 4.5***

```
PROGRAM
 LINE
**********************************************************
********
 1  options crt;
```

```
options limwarn=0;
freq m;
smpl 67:1 95:8;
read(file='ex45.xls', format=excel) rtdr ivrdr fycp leh pucd;
sd=rtdr;
di=ivrdr;
is=ivrdr/rtdr;
i=fycp;
e=leh;
p=pucd;
msd di(-6) is(-1) i(-1) e(-1) p(-1);
olsq sd c di(-6) is(-1) i(-1) e(-1) p(-1);
set b2std=@coef(2)*@stddev(1)/@sdev;
set b3std=@coef(3)*@stddev(2)/@sdev;
set b4std=@coef(4)*@stddev(3)/@sdev;
set b5std=@coef(5)*@stddev(4)/@sdev;
set b6std=@coef(6)*@stddev(5)/@sdev;
print b2std b3std b4std b5std b6std;
set b2el=@coef(2)*@mean(1)/@ymean;
set b3el=@coef(3)*@mean(2)/@ymean;
set b4el=@coef(4)*@mean(3)/@ymean;
set b5el=@coef(5)*@mean(4)/@ymean;
set b6el=@coef(6)*@mean(5)/@ymean;
print b2el b3el b4el b5el b6el;
olsq(silent) sd c is(-1) i(-1) e(-1) p(-1);
y=@res;
olsq(silent) di(-6) c is(-1) i(-1) e(-1) p(-1);
x=@res;
msd(corr,noprint) y x;
print @corr;
olsq(silent) sd c di(-6) i(-1) e(-1) p(-1);
y=@res;
olsq(silent) is(-1) c di(-6) i(-1) e(-1) p(-1);
x=@res;
msd(corr,noprint) y x;
print @corr;
olsq(silent) sd c di(-6) is(-1)  e(-1) p(-1);
y=@res;
olsq(silent) i(-1) c di(-6) is(-1) e(-1) p(-1);
```

```
 41  x=@res;
 42  msd(corr,noprint) y x;
 43  print @corr;
 44  olsq(silent) sd c di(-6) is(-1) i(-1)  p(-1);
 45  y=@res;
 46  olsq(silent) e(-1) c di(-6) is(-1) i(-1) p(-1);
 47  x=@res;
 48  msd(corr,noprint) y x;
 49  print @corr;
 50  olsq(silent) sd c di(-6) is(-1) i(-1) e(-1);
 51  y=@res;
 52  olsq(silent) p(-1) c di(-6) is(-1) i(-1) e(-1);
 53  x=@res;
 54  msd(corr,noprint) y x;
 55  print @corr;
 56  end; stop;
          EXECUTION

**********************************************************
*********************

 Current sample:  1967:1 to 1995:8

                             Univariate statistics
                             =====================

 Number of Observations: 338

                         Mean        Std Dev        Minimum
      Maximum
 DI(-6)           65514.46450   40703.67079   13882.00000
 153826.00000
 IS(-1)               2.01019       0.15083       1.65037
      2.41678
 I(-1)                7.58018       2.83619       3.19000
     16.66000
 E(-1)                6.92885       2.77122       2.67000
     11.49000
 P(-1)               83.58314      29.66518      39.30000
    128.20000
```

```
                          Sum      Variance      Skewness
    Kurtosis
 DI(-6)           2.21439D+07   1.65679D+09       0.40226
    -1.20323
 IS(-1)             679.44277      0.022749    -0.0066421
    -0.35853
 I(-1)             2562.10000       8.04399       0.99357
     0.98168
 E(-1)             2341.95001       7.67964     -0.025275
    -1.40695
 P(-1)            28251.09999     880.02307      -0.14651
    -1.55929


                                      Equation   1
                                      ============

                       Method of estimation = Ordinary
Least Squares


 Dependent variable: SD
 Current sample:  1967:7 to 1995:8
 Number of observations:  338

  Mean of dependent variable = 33305.0
 Std. dev. of dependent var. = 20183.4
    Sum of squared residuals = .139439E+10
       Variance of residuals = .419997E+07
    Std. error of regression = 2049.38
                   R-squared = .989843
          Adjusted R-squared = .989690
     Durbin-Watson statistic = .642636
   F-statistic (zero slopes) = 6470.97
  Schwarz Bayes. Info. Crit. = 15.3360
  Log of likelihood function = -3053.92
```

```
            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C          22632.7       1775.64       12.7462
 DI(-6)     .414706       .018106       22.9048
 IS(-1)     -12715.9      1022.70       -12.4336
 I(-1)      -120.464      53.7737       -2.24020
 E(-1)      1529.95       726.531       2.10583
 P(-1)      -7.45720      50.6962       -.147096

                  B2STD         B3STD         B4STD
  B5STD         B6STD
 Value          0.83633     -0.095025     -0.016928
0.21006     -0.010960


                   B2EL          B3EL          B4EL
   B5EL          B6EL
 Value          0.81577      -0.76749     -0.027417
0.31829     -0.018715


            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C          20627.3       2852.81       7.23053
 IS(-1)     -14408.2      1632.43       -8.82621
 I(-1)      -431.474      83.2458       -5.18313
 E(-1)      14244.6       723.598       19.6859
 P(-1)      -642.826      66.8247       -9.61959

            Estimated    Standard
 Variable  Coefficient     Error       t-statistic
 C          -3819.37      5370.20       -.711215
 IS(-1)     -5149.21      3082.49       -1.67047
 I(-1)      -652.619      158.777       -4.11029
 E(-1)      31343.5       1373.05       22.8276
 P(-1)      -1585.76      126.462       -12.5394

                                      @CORR

                 1             2
```

```
       1        1.0000
       2       0.78200       1.00000


           Estimated    Standard
Variable  Coefficient     Error        t-statistic
C         1616.18       657.419       2.45837
DI(-6)    .435227       .021795       19.9689
I(-1)     -452.024      56.4469       -8.00796
E(-1)     -431.481      857.293       -.503306
P(-1)     114.751       60.1197       1.90871

           Estimated    Standard
Variable  Coefficient     Error        t-statistic
C         1.65278       .029141       56.7161
DI(-6)    -.161387E-05  .966117E-06   -1.67047
I(-1)     .026075       .250211E-02   10.4210
E(-1)     .154250       .038001       4.05910
P(-1)     -.961068E-02  .266492E-02   -3.60637

                                     @CORR

                1             2
       1       1.00000
       2      -0.56365        1.0000


           Estimated    Standard
Variable  Coefficient     Error        t-statistic
C         24338.8       1613.69       15.0827
DI(-6)    .423618       .017769       23.8398
IS(-1)    -13852.0      893.435       -15.5042
E(-1)     1753.56       723.971       2.42214
P(-1)     -36.9917      49.2464       -.751155

           Estimated    Standard
Variable  Coefficient     Error        t-statistic
C         -14.1623      1.63464       -8.66384
DI(-6)    -.739856E-04  .180001E-04   -4.11029
IS(-1)    9.43139       .905038       10.4210
```

```
E(-1)     -1.85624      .733373       -2.53110
P(-1)     .245173       .049886       4.91467

                                     @CORR

                1             2
       1       1.00000
       2      -0.12203       1.00000


           Estimated    Standard
Variable  Coefficient     Error       t-statistic
C         21948.5       1754.64       12.5088
DI(-6)    .444487       .011363       39.1155
IS(-1)    -12248.3      1003.44       -12.2063
I(-1)     -136.021      53.5377       -2.54067
P(-1)     94.3796       15.2919       6.17188

           Estimated    Standard
Variable  Coefficient     Error       t-statistic
C         -.447226      .131669       -3.39660
DI(-6)    .194655E-04   .852717E-06   22.8276
IS(-1)    .305644       .075298       4.05910
I(-1)     -.010169      .401749E-02   -2.53110
P(-1)     .066562       .114751E-02   58.0059

                                     @CORR

                1             2
       1       1.00000
       2       0.11481        1.0000


           Estimated    Standard
Variable  Coefficient     Error       t-statistic
C         22573.9       1727.53       13.0672
DI(-6)    .416214       .014900       27.9334
IS(-1)    -12686.7      1001.82       -12.6636
I(-1)     -122.521      51.8472       -2.36312
E(-1)     1428.00       217.707       6.55929
```

```
           Estimated    Standard
Variable  Coefficient     Error       t-statistic
C         7.88408       1.87011       4.21584
DI(-6)    -.202261E-03  .161300E-04   -12.5394
IS(-1)    -3.91113      1.08451       -3.60637
I(-1)     .275843       .056126       4.91467
E(-1)     13.6706       .235675       58.0059

                                     @CORR

                 1             2
       1       1.00000
       2    -0.0080727       1.00000
```