TSP Handbook

by Bronwyn H. Hall
and Sergio L. Schmukler

to accompany

Econometric Models and Economic Forecasts

FOURTH EDITION

Pindyck and Rubinfeld

Econometric Models and Economic Forecasts

Fourth Edition

Robert S. Pindyck
Sloan School of Management,
Massachusetts Institute of Technology

Daniel L. Rubinfeld
University of California, Berkeley

Prepared by
Bronwyn H. Hall
University of California, Berkeley
and
Sergio L. Schmukler
University of California, Berkeley

Irwin
McGraw-Hill

Boston Burr Ridge, IL Dubuque, IA Madison, WI
New York San Francisco St. Louis
Bangkok Bogotá Caracas Lisbon London Madrid Mexico City
Milan New Delhi Seoul Singapore Sydney Taipei Toronto

Irwin/McGraw-Hill

A Division of The ***McGraw·Hill*** *Companies*

TSP Handbook to accompany
ECONOMETRIC MODELS AND ECONOMIC FORECASTS,
FOURTH EDITION

1 2 3 4 5 6 7 8 9 0 BKM/BKM 9 0 9 8 7

ISBN 0-07-025940-2

http://www.mhhe.com

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.

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

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

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)?
```

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.

[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.

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 β_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: