



P-STAT[®]

An Introductory

Guide for

Windows Users



P-STAT: Introductory Guide For Windows Users

June 2013

This publication corresponds to **P-STAT Version 3 Revision 5**, June 2013. This publication is designed to provide a general introduction to the P-STAT system as implemented under PC/Windows.

Please direct any questions to:

P-STAT, Inc.
230 Lambertville-Hopewell Rd.
Hopewell, New Jersey 08525-2809
U.S.A.

Telephone: 609-466-9200

Fax: 609-466-1688

Internet: support@pstat.com

Web Page URL: <http://www.pstat.com>

All rights reserved. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system without the prior written permission of P-STAT, Inc.

P-STAT is a registered trademark of P-STAT, Inc. Windows is a registered trademark of Microsoft Corp. Copyright © 2006-2013 P-STAT, Inc. Printed in the US. Published by P-STAT, Inc.

CONTENTS

Features and Statistics

P-STAT'S CAPABILITIES	1.1
USER FEATURES	1.3
LEARNING TO USE P-STAT	1.4
P-STAT DOCUMENTATION	1.6
INSTALLATION	1.7
MIGRATE	1.9
SUMMARY	1. 10

Getting Started

GETTING STARTED	2.1
OVERVIEW OF THE MAIN MENU	2.1
ORGANIZATION OF INFORMATION	2.4
CREATING A P-STAT SYSTEM FILE	2.7
MORE FEATURES	2.13
SUMMARY	2. 15

A Single Project

EXPORTING THE DATA FROM EXCEL	3.1
PREPARING P-STAT	3.1
IMPORTING THE TAB DELIMITED FILE INTO P-STAT	3.2
EXECUTING THE TURF COMMAND	3.2
PRINTING THE RESULTS	3.8
OTHER TURF IDENTIFIERS	3.11
HELP OPTIONS	3.12
OTHER DATA INPUT	3.13

The Editor

THE EDITOR'S MAIN MENU	4.2
AN EXAMPLE WITH A COMMAND FILE	4.4
MORE COMMAND EDITOR FEATURES	4.8

A Larger Project

A WEIGHTED ANALYSIS	5.1
COMBINATIONS	5.10
EFFICIENT PROCESSING	5.12

FIGURES

The P-STAT Main Window	1.5
The Information (Data) for the File	2.5
Elements of a P-STAT System File	2.5
Displaying Autosave and External File names	2.12
Turf Report	3.7
Turf: Reach.results Output File	3.8
REACH.STATS : All possible statistics	3.10
REACH.STATS Controls the Rows. SHOW Controls Columns	3.10
An Empty Editor File	4.1
Import an Existing Editor File	4.5
An Editor File	4.7
The COUNTS Command	5.3
Generate Weight: the Command	5.8
Weighted TURF Report	5.10
A File of Commands	5.13
Macro with Subfiles	5.14
A Macro With Arguments	5.16

Features and Statistics

P-STAT® is a computing system for data analysis and information management and display. “PSTAT: Introductory Manual” contains a complete introduction to the P-STAT program and is the basic reference manual for all users. . This manual is designed primarily for the new user running P-STAT on PC/windows. This chapter gives a brief introduction to the full capabilities of the P-STAT program. The rest of this book covers the features that are supported with menus .

1.1 P-STAT’S CAPABILITIES

P-STAT capabilities include:

- retrieval of numeric and character information;
- file management;
- data modification;
- data display;
- statistical analyses;
- in-stream and block macros;
- interactive and batch processing.

1.2 Storage and Retrieval of Data

P-STAT accepts information in many different forms. Information may be *numeric*, as is average yearly rainfall or total automobile production, or it may be text or *character*, as is a name or an address. P-STAT accepts information from a variety of sources, including *disk* and the user’s *terminal*. The information may be formatted or unformatted. It may be produced by another software system. Tab delimited files from Excel and SPSS portable files, may be input to P-STAT.

P-STAT holds related information in a rectangular format called a “P-STAT system file”. This file may contain both numeric and character information and is stored in a form that P-STAT can read, write and manipulate quickly. Information may be missing, and representations for three different types of missing data are available. P-STAT system files are *packed* to reduce storage requirements and *automatically saved*. Once a file is changed, a *backup* version exists and is available in case the current version is lost or incorrectly modified.

Other types of files used by P-STAT may be made either within P-STAT or within the host computer operating system. They hold value labels, command sequences, edit files, and data resulting from modifications or analyses.

1.3 File Management

Many commands permit two or more files to be accessed simultaneously. There are commands to join and merge files with either *direct* or *hierarchical* linkages. These commands provide most of the capabilities normally associated with a relational database system. In addition, files can be sorted, divided into subsets, concatenated,

interleaved, updated and transposed. Duplicate cases may be detected and deleted, or summarized into one case. There are also commands to do matrix operations and aggregation.

A utility command, *SHOWBYTES*, is available to display the exact contents of an external file. Another utility command, *REFORMAT*, can be used to change fields in an external file. *EXAMINE* reads an external file and counts how many times each type of ASCII byte occurs.

1.4 Data Modification

The P-STAT Programming Language (PPL) is a powerful tool for *modifying* existing variables, *generating* new variables, and *selecting* cases or variables. PPL can be applied to any P-STAT system file as it is read by any P-STAT command. It can be used on the variables within a single case or across groups of related cases.

The ability to perform data modification across successive cases means that complex operations, which in most systems would require multiple passes through a file, can be done with a single pass using PPL. For instance, given a file of people arranged in household order, it takes only a single step to calculate family size and append that information to the record of each member in that household.

Operators and functions include:

1. wildcards and masks in variable selection;
2. DO loops for repeated calculations;
3. logical testing with IF statements and IF/THEN/ELSE blocks;
4. recoding of data values;
5. system values such as *.DATE.*, the current date;
6. scratch variables and user-defined arrays permit cross-case and cross-command manipulations.;
7. splitting and collecting of cases.

Special functions for *character* variables permit calculation of the length and position of character strings, formation of substrings, conversion of character variables to numeric variables, and location and compression of specific characters. Only a few of the basic PPL features are available in the menus. The more advanced features can be accessed using the command language.

1.5 Data Display

Report writing is straightforward with the *LIST* and *TITLES* commands. Options exist for page layout, data formatting, subgrouping, hierarchical arrangement of variables, headings and summary statistics. Multiple top titles and bottom titles are possible. Simple lists are automatically formatted appropriately for the output device in use. Different output devices with varying attributes may be defined.

The *TEXTWRITER* command produces reports with precision formatting and complex layouts. In addition, information in P-STAT files may be tested and differentially incorporated in the reports. All PPL instructions and functions for data selection and modification may be used in *TEXTWRITER*. *TEXTWRITER* output can be used with PostScript controls to produce camera-ready copy with control over the page layout including choice of font and color.

The *crosstabulation* command, *SURVEY*, may be used to process both numeric and character questionnaire data. It produces stub and banner reports, with titles, multiple banners, nested banners, nets, and other features familiar to survey researchers. Extensive options provide for multiple response tables including nets and subtotals. Other options provide for cells with counts, different percentages, means and sums, flexible formats and layouts, and weighting. *SURVEY* supports PostScript printers and offers a variety of font choices when PostScript is available.

Auxiliary commands for the market research analyst include:

1. **BALANCE**: a command which uses sample balancing techniques to produce weights which have the least impact on cell counts.
2. **SAMPLE**: a command to produce a random sample from a larger file. By variables can be supplied to assure that the sample matches the larger file in critical ways.
3. **TURF**: a command to do Total Unduplicated Reach and Frequency analysis. The **TURF.SCORES** command is a tool that can be used to identify the cases which contributed most to the TURF output.

1.6 Statistical Analyses

P-STAT provides a wide range of statistical procedures from simple descriptive statistics to complex multivariate procedures. *Description files* give means, standard deviations, highs, lows, and counts of good and missing data items. The **COUNTS** command provides an initial overview of data. It produces frequencies and percentages of *all unique values* of character and numeric variables, as well as a multitude of summary statistics. **COUNTS** can also be used to produce frequency distributions for subgroups. **COUNTS** output can optionally be saved in a P-STAT system file. The **PERCENTILES** command provides medians, quartiles, deciles, centiles or arbitrarily specified *quantiles*.

Groups of independent or paired data may be tested using *t tests*. When the measurements are not qualitative data, *nonparametric* tests may be used. *Correlations* for continuous, dichotomous, and ranked variable pairs may be requested, as well as significance values and corrections for grouping errors. Canonical correlation is a multivariate procedure that analyzes the relationship between two sets or groups of variables.

Multiple *regression* analysis can fit linear equations to observed data values, predict dependent values and calculate residuals. The analysis may be a forward stepwise regression, with stepping either program or user controlled, or it may be non-stepwise.

Analysis of variance (ANOVA) determines the effects of multiple treatments in both balanced and unbalanced designs, and detects and reports confounding of effects. Designs may be modified interactively, and error terms and blocks specified. Factorial, block, nested, split-plot and repeated measures designs may be analyzed. Covariates and partitioning of sums of squares may be specified with a cohesive algebraic notation.

Survival analysis is used to analyze lifetime (survival time) data. Both the lifetable (actuarial) and product limit methods are supported. *Cluster analysis* puts cases into a specified number of groups. The cases may be weighted. *Discriminant and factor analyses* permit identification of significant groups and factors. Various factor matrix rotations may be requested. In addition, *matrix* operator commands and a *macro* facility, used in conjunction with the programming language, permit a user to define statistical procedures not included in the P-STAT system.

1.7 USER FEATURES

P-STAT user features include:

- same command language in both interactive and batch modes;
- in-stream and block macros for frequently used activities;
- ease of use with menus on PC/Windows;
- English-like language;
- available on PC/Windows, Linux and SUN Solaris.

1.8 Interactive and Batch Computing

P-STAT can be used in either batch or interactive computing environments. *Batch* computing describes an environment where the entire job is prepared and then submitted to the computer for execution. When the run is complete, the results are printed or written in a disk file. If errors are found, the commands or data must be corrected and the job resubmitted for execution. If an entire job is sent to the computer and the results come back later, it is a batch job.

In interactive computing, each step is entered and executed immediately. Then, depending on the results of that step, the next step is entered. If an error is made, it may be corrected immediately. If a required file name is not supplied, the user is prompted for it. If a command name is misspelled, it may be spelled correctly without reentering the rest of the command. The job proceeds with corrections and additions made when necessary. The results of each step are displayed after each step instead of at the end of the entire job.

P-STAT is designed to be an interactive system, although it may also be used in batch mode. It requires certain “commands” or instructions that P-STAT recognizes in order to process information. These commands tell P-STAT exactly what process the user expects P-STAT to perform. Most new users work interactively so they may correct errors as they occur — they prepare batch jobs when they are more experienced. The *menu system* makes learning P-STAT easier. Selecting menu options causes P-STAT to write the corresponding command, which is then executed. Regardless of whether you type the command directly or P-STAT writes the command based on menu selections, an English-like command results. P-STAT processes the command, following the instructions contained in it, and produces any requested output.

This manual is designed to illustrate the use of menus on PC/Windows. The most frequently used commands and the basic elements of the P-STAT Programming Language are supported in the menu system.

Even if you are running a version of P-STAT with front end menus, you may find that you need to enter a command that is not supported by the windows system. P-STAT expects to find command input in single lines that are no longer than 80 characters. A command can be entered on multiple lines (separated by the carriage-return character) until the end of command is signalled by a single “\$”. Do not enter the carriage return in the middle of a name or token. If you are entering the command from the P-STAT main window, it is easy to see if you are approaching the 80 character limit in time to press the enter key and move to the next line. If you are working from a text editor, *turn off the word wrap feature which automatically moves the cursor to the next line and work from a text window that is no wider than 80 characters.*

1.9 Macros

P-STAT macros contain P-STAT instructions or data. They can be used anywhere to hold any part of a P-STAT command stream. However, they are usually used to hold pieces of P-STAT code or command sequences that are frequently used. All macros can have arguments which may or may not have default values. When a macro is used any values that do not have default definitions must be supplied. Macros can call macros which can then call other macros. There are two basic types of macros: block macros and in-stream macros.

Block macros contain one or more P-STAT commands. When the macro is executed the commands in the macro are executed one after the other. Special features of macros that are not available in a normal job stream are:

1. SUBFILES which provide the ability to execute the commands in the macro for each subgroup specified by the SUBFILES command;
2. Logic which permits you not only to skip commands but also to loop back and re-execute earlier commands.

In-stream macros are not complete commands but can be any part of the command except the command name itself. They can appear within the command text, within programming language statements, and within subcommand or data records. Like block macros, in-stream macros can have key word or positional arguments. The arguments must have values supplied either as defaults in the definition or as argument values when the macro is used.

1.10 LEARNING TO USE P-STAT

It is not difficult to learn the basics of P-STAT despite the fact that it is a large system with extensive capabilities. This manual is intended to provide enough information and illustrations to help a new user get started. There are also some features which even those users who are adept at using a command language will find helpful such as:

1. full screen editing of command input and error correction;

2. access to the Print Manager for printing.

A basic understanding of the way the P-STAT language works and the use of files is helpful even for users with simple needs that can be handled completely by the front end menus. P-STAT is a language and there are some basic rules and some required punctuation when using the language. One way to become familiar with the rules is to look at the commands that are created by menu requests.

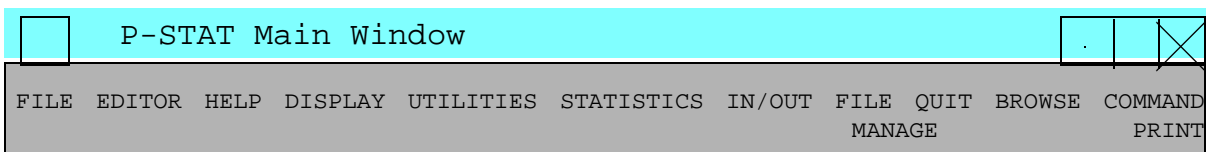
P-STAT commands require that the data they are given be in the form of a P-STAT system file. These files contain information about the variables and pack the data into a compressed, binary form that can only be read by P-STAT. This means that you must learn how to:

1. Convert your data into a P-STAT system file. If you have raw data, you may use the MAKE command to provide your own variable names or possibly TEXTFILE.IN if generated names are acceptable. If you have data in another system such as Excel, it can be exported from that system and then imported into P-STAT. (See IN/OUT in the menu headers.)
2. How to process that data. The menu system supports a number of commands to perform statistics and display the data.
3. How to get the data back out, perhaps to a printer. Since a P-STAT system file is in a binary form that cannot be read by other programs, a P-STAT command such as LIST or a command to create the appropriate form for another package must be used. If the input to the P-STAT PRINT command is a P-STAT system file, a LIST command is automatically generated and that output sent to the printer.

1.11 P- STAT Under PC/Windows

P-STAT in a windows environment is basically identical to P-STAT in an environment where windows are not available in the sense that the underlying command structure and the total capabilities are not changed by the windowing environments.

Figure 1.1 The P-STAT Main Window



.This area is at the bottom of the screen. It is often used for informative messages.

1.12 Basic Information: Data values and Names

In P-STAT data values are either numeric or character. Numeric values are always entered in double precision and numeric operations are always double precision. This permits great accuracy when calculations are performed.

A single character datum can have up to 50,000 characters. When a character value is created in P-STAT it is assumed to contain 40 characters unless a different value is specified. Specialized variables such as date and time values can be stored as character values and manipulated with one of more than 50 date functions and commands. Character values that have one or more internal blanks must be enclosed in quotes.

Names for external files can be arbitrarily long and depend on the operating system. Under PC/Windows the names can be entered and referenced using upper or lower case characters. Under Linux or Sun Solaris, file names are case sensitive. Within P-STAT, file and variable names are NOT case sensitive. The program remembers the original use of the name and uses that for displays and printout even though the name can be referenced in whatever is the most convenient form.

P-STAT system files store file names, variable names, and a notation of the command which created them, as well as the data values. File names can be up to 16 characters long and can contain only letters, numbers and decimal points. The first character MUST be a letter. Case is unimportant. P-STAT remembers the file name as it is first entered and uses that representation in all reports. Whatever, the name of a P-STAT system file, it is ALWAYS stored on the disk in upper case characters with a 3 character extension. The default extension is "PS" followed by either a 1 or a 2.

Each variable has a unique name which may be up to 64 characters and contain letters, numbers, decimal points, and underscores. The use of a double colon (::) is allowed to divide the variable name into 2 parts: the first up to 16 characters is known as the "tag", followed by the double colon (::) and finally additional text for a maximum of 64 characters. Not only must the variable have a unique name, tags, if they are used, must also be unique. If you do not provide a name, P-STAT generates a name in the form of "VARn" where n is the variable number.

It can be awkward when you wish to use the P-STAT Programming Language (PPL) to reference a variable if all 64 characters are needed.

```
IF q33::this.is.my.very.long.variable.name = 1, DELETE;
```

If you choose to use long variable names, TAGS and wildcards are an essential feature when using PPL.

```
IF q33 = 1, DELETE
```

or, for a variable such as Age.oldest.child

```
IF Age.o? GT 21, DELETE
```

1.13 P-STAT DOCUMENTATION

The following manuals describing the P-STAT system are currently available and can be downloaded from our web page (www.pstat.com) in Acrobat format that has been zipped to save space.

1. P-STAT: Introductory Manual (psintro.zip)
This manual covers the basic features of the system with the emphasis on the command language. The command language is identical on all supported computers. It also covers basic commands such as MAKE, TEXTFILE.IN, LIST, COUNTS and TURF analysis.
2. P-STAT: An Introductory Guide for PC/Windows (pcintro.zip)
This manual covers the Menu system and provides an example using TURF to illustrate a run from data entry through examination of the results. In addition full screen usage of the P-STAT editor is described.
3. P-STAT: File Management (psmang.zip)
This manual covers the commands used to bring your data into P-STAT and how to write it out so other commands can use it. The MAKE and MAKE.FIXED commands are covered in detail.

4. P-STAT: Utility Commands (psutil.zip). This includes the features of the P-STAT editor.
5. P-STAT: A Guide to the P-STAT Programming Language (PPL) (psppl.zip). The final sections discuss the TEXTWRITER command and P-STAT Macros.
6. P-STAT: The Survey Command, BALANCE and SAMPLE (pssurv.zip)
7. P-STAT: Basic Statistics (psstat1.zip)
8. P-STAT: Advanced Statistics (psstat2.zip)
9. P-STAT: Plots, Graphs and Postscript.support (psplot.pdf)
10. P-STAT: Master Index (pstatIX.zip)

1.14 INSTALLATION

The actual process of installation depends on the computer system. Each CD is sent with instructions that are appropriate for that system. In general:

- 1A. Web page download: Place the downloaded files into a temporary directory such as c:\temp. For the standard release, click on W6FILES.EXE. You need a password to begin the installation. This password can be requested by filling out the form on the P-STAT web site and providing us with a name and an email address. This is all that is needed for an installation that permits you to use your own small (20 X 200) data files.
- 1B. CD install: move to the pstwin folder on the cd. It will have a name such as pstwin.3.01
- 2. Click on PSINSTALL.EXE. You will be given a chance to select the directory and supply the folder for the P-STAT module. The default installation folder is "C:\Program Files\PSTAT".

There are several environment variables that can be set which make it easier to locate data sets. Most of the information that is contained in these variables can be supplied or changed later by P-STAT commands. The only environment variables that have no equivalent command form are PSKEY and PSTART.

1. PSKEY The PSKEY is not required for the basic demo version. The demo version permits all P-STAT commands to be used with a reduced file size. Your PSKEY is supplied when you order a full version of P-STAT.
2. PSHELP supplies the name and location of the P-STAT help file. This is necessary if the on-line help file is to be used. Since this location is determined at installation time, it makes sense to set it once with an environment variable and forget about it. On the PC, this information is automatically stored in PSTAT.INI at install time.
3. PSFILES supplies the name of the home or current working directory. This is where P-STAT expects to find or create all files except those explicitly located elsewhere. If this information is not available, P-STAT considers the directory from which P-STAT is invoked to be the current working directory. If P-STAT is invoked from the program manager, the assumed working directory is the directory where P-STAT is installed.

There are three types of files to be located: 1) P-STAT system files (PSAUTO) created by P-STAT commands, 2) temporary work files (PSTEMP) used by P-STAT commands such as SORT, and 3) external data files (PSDATA) including your data input, labels files, command files and files exported to and from other systems.

PSFILES sets values for PSAUTO, PSDATA and PSTEMP. PSAUTO, PSDATA and PSTEMP, if used, take precedence over the PSFILES setting. If PSFILES is used after the start of the run, PSTEMP will not be changed. PSFILES can be accessed in the front end menus under "Utilities".

4. PSAUTO provides the full path to a directory where P-STAT system files are located. If PSAUTO is not specified, system files will be written to and read from the current directory. What determines the current directory depends on the operating system. PSAUTO can be changed any time during a P-STAT session.
5. PSTEMP provides the full path to a directory where temporary files are to be placed. PSTEMP may be executed only at the beginning of the P-STAT session.
6. PSDATA provides the full path to the directory where other external files are located. This includes raw data files, command files, value label files and files to be exported or imported into spreadsheets or other packages. PSDATA, like PSAUTO can be changed any time during a P-STAT session.
7. PSTART provides the full path and the name of a file which contains P-STAT commands to be executed as the run begins. The PSTART file may contain any P-STAT command including PSHELP. A command in the PSTART file takes precedence over the equivalent environment variable.

The PSTART environment variable is not automatically installed by the Windows INSTALL program. If the PSTART environment variable is set and points to an existing file, the commands in that file will always be executed at the beginning of a P-STAT session.

If there is no PSTART environment variable, P-STAT looks for a file named PSTART in the PSDATA directory. If the PSDATA directory has not been specified by either a PSFILES or PSDATA environment variable, P-STAT looks for PSTART in the current working directory. PSTART is fully described in the manual “P-STAT: Utility Commands”.

1.15 Examples of the Three Different Types of Files

The three different file types are:

1. P-STAT system files created by P-STAT commands;
2. External raw data files including label files and files containing P-STAT commands;
3. Temporary files which are used by individual commands to hold intermediate results.

Suppose these environment variables are set:

1. PSAUTO to 'C:\olddata'
2. PSDATA to 'D:\input'
3. PSTEMP to 'C:\tempdir'

The command:

```
TABFILE.IN X, READ 'ABC.TXT' $
```

uses the following files:

1. **X** is the name for the P-STAT system file (an autosave file) which is created by the TABFILE.IN command. It is written to the PSAUTO directory as:

```
C:\olddata\X.PS1
```

Note: P-STAT uses extensions of PS1 and PS2 on system file names. The pathname and the extensions are NEVER used in P-STAT commands. File X.PS1 is always referred to as “X”. The P-STAT system files on disk are ALWAYS stored with their names in uppercase characters.

2. **ABC.TXT** is an external file being read. When it does not have its own path it defaults to a PSDATA path and is read as:

```
D:\input\ABC.TXT
```

3. **TABFILE.IN** also creates and then erases several temporary files which have, in all likelihood, unique names. Given the definition of PSTEMP in the example above, a name for such a file on PC/Windows would be something like

```
C:\tempdir\P_0Z7CR29J2$$$$.TMP
```

1.16 MIGRATE

P-STAT Version 3 on the PC cannot read binary (unformatted) files created by P-STAT Version 2. This does not matter with P-STAT system files which are automatically converted to Version 3. However, P-STAT edit files are NOT automatically converted. The new MIGRATE command is available on the PC only and does the conversion for you.

```
MIGRATE "c:\myfile\test.edt", OUT "c:\newstuff\test.edt" $
```

If you do not know the origin of your P-STAT files, use the FILETYPE command.

```
FILETYPE $
```

This examines a file and simply reports if the file is LF90 unformatted, LF95 unformatted, CR-LF formatted, or unknown.

SUMMARY

Environment variables may be created which tell P-STAT where it should find existing files or write new files. On the PC under windows operating systems such as WINDOWS 7 the INSTALL program will create the environment variables for you. They are stored in a file named PSTAT.INI which is stored in the directory associated with the system environment variable WINDIR. This is usually C:\WINDOWS. This file can be edited in an ordinary text editor or updated with PSTATINI.EXE which is provided as part of the P-STAT installation.

PSKEY **key value**

PSKEY is required for all except demo installations.

PSHELP **full path and file name**

PSHELP provides the complete path and name of the PSTAT help file. The following assumes that the P-STAT module is on the C drive in the PSTAT directory.

```
SET PSHELP=C:\PSTAT\PSTAT.HLP
```

PSTART **full path and file name**

PSTART provides the complete path and name of a *start-up* file of P-STAT initialization commands.

```
SET PSTART=d:\joe\psinit.trn
```

If the PSTART environment variable exists, the contents of the file it cites will be executed at the beginning of any P-STAT session.

PSFILES **full path**

PSFILES provides the complete path to a directory for PSAUTO, PSTEMP and PSDATA if they are not individually specified. If PSFILES is used after the start of the current P-STAT session, only PSAUTO and PSDATA are changed.

```
SET PSFILES=C:\susie\study32
```

PSAUTO **full path**

PSAUTO provides the complete path to a directory where P-STAT system files are located and new system files are to be written. It takes precedence over the PSFILES directory.

```
SET PSAUTO=C:\susie\study32\myspf
```

PSDATA **full path**

PSDATA provides the complete path to a directory where external data files are located.

```
SET PSDATA=E:\proj32\data
```

PSTEMP **full path**

PSTEMP provides the complete path to a directory where temporary work files are stored. These files are automatically erased when P-STAT runs are terminated with an END\$ command.

```
SET PSTEMP=C:\TEMP
```


2 Getting Started

This chapter is an introduction for users who are not familiar with P-STAT or its implementation on PC/Windows. The first part of this chapter provides an overview of the menus. The second part discusses the way the information is organized in P-STAT. The rest of the chapter provides illustrations to help you get your data into P-STAT and to display that input. Chapter 3 illustrates a single project from beginning to end.

2.1 GETTING STARTED

The first step in using P-STAT is getting the P-STAT computer package running. You can either click on the P-STAT icon that is created at installation time or you can access the P-STAT module using the system "RUN" facility.

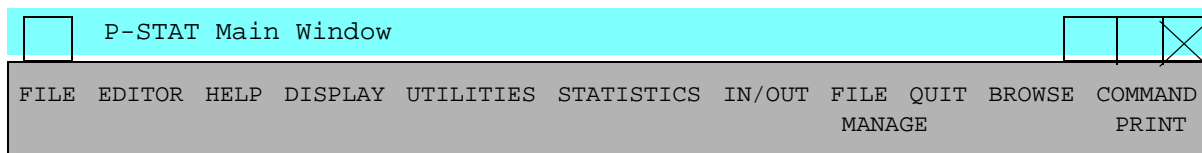
If the initial greeting message:

```
P-STAT, version ....
```

does not appear, see the local person who installed P-STAT on your system or refer to the installation instructions that came with the software. The usual reason for such a problem is that P-STAT has been installed in an unusual location or that the P-STAT icon was not created.

2.2 OVERVIEW OF THE MAIN MENU

The two lines at the top of the screen are used for the highest level menu. The top line contains the eleven major menu entries. Seven of these items have sub-menus which lead directly to commands. To select an item move the mouse to that item and click the left mouse button. To see the contents of the sub-menus hold the left mouse button down and move the mouse slowly across the area without making a selection. As you move down the submenus, the status bar at the bottom of the screen contains a brief description of each menu item.



The four items which do not have sub-menus are:

1. Editor which places you directly in the full screen P-STAT editor. This editor has its own top menu.
2. Quit which cancels the action in progress and returns control back to the P-STAT main window.
3. Browse which brings up a scrolling window that contains up to 1024 lines of input and output.
4. Command Print which causes the output from the most recent command to go to the print manager. You then have your choice of printer and print setup.

The File menu contains eleven entries.

1. System Files displays the names of the P-STAT system files stored on your current system file directory.

- | | |
|----------------------|--|
| 2. Active Files | displays the names of all the P-STAT system files that you have referenced in the current P-STAT job. |
| 3. Status | displays settings such as the output width and the active print destinations. |
| 4. Show Contents | displays the variables in an existing P-STAT system file. |
| 5. Examine a file | examines the bytes in an external file. An optional output P-STAT system file contains counts for each character that is found. |
| 6. Show the bytes | displays the bytes in a file. |
| 7. Reformat | allows you to create a new external file with selected fields altered. This can be used, for example, to change a PC file to MAC or Unix format. |
| 8. Define Print File | sets the attributes of a print destination. This includes the output width, lines per page, and the page change character. |
| 9. Select Print File | specifies which print destination is to become the current printer. All printed output will be directed to this destination until another print destination is selected. |
| 10. Print | invokes the print manager to print either a P-STAT system file or a disk file of print output (see “Define Print File” and “Select Print File” above.) If you select “P-STAT system file”, a LIST command is first executed to create an ASCII text file for the printer. In either case you have your choice of printer and can tune the printer’s setup. If the file is an active print file in P-STAT, it is closed before printing so that you do not lose the final print buffer. |
| 11. Exit P-STAT | ends the run by submitting an END\$ command. |

The Help menu contains two entries:

- | | |
|--------------------|--|
| 1. Menu Help Items | opens your default browser with entries for all the commands that are included in the menu system. |
| 2. Help File | opens a browser with entries for all the topics and commands that are covered in the full P-STAT help file. The Display menu contains four items |

The Display menu contains four entries:

- | | |
|-------------------------------|--|
| 1. Save Labels | saves value labels in an external file so they can be used by commands such as LIST and SURVEY. Value labels provide text to replace values in reports and tables. |
| 2. Define Titles | provide titles for use in reports and tables. |
| 3. List a file | lists a P-STAT system file with a variety of format controls. |
| 4. Produce Tables With Survey | produces tables. It has extensive controls for labels, titles, cell contents and the arrangement of the rows and columns. |

The Utilities menu contains fourteen items:

- | | |
|----------------------|--|
| 1. Check file | checks the status of a P-STAT system file. In addition to verifying that it is readable, it reports the number of variables and the number of cases. |
| 2. Compare two files | compares two P-STAT system files and saves the differences in a file. |

- | | |
|---------------------------------|---|
| 3. Locate a File | provide a link to a P-STAT system file that is not located in the current PSAUTO directory |
| 4. Erase current | erases the current version of a P-STAT system file (only if there is a previous version). |
| 5. Erase previous | erases the previous version of a P-STAT system file (only if there is a current version). |
| 6. Erase | erases both current and previous versions of a P-STAT system file. |
| 7. Erase external | erases an external file such as a labels file or a raw data file. |
| 8. Change all Folders (PSFILES) | issues a PSFILES command. This makes the designated folder the default for all files at the start of the run. After the run has begun PSFILES changes only the PSAUTO and PSDATA directories. |
| 9. Change Save Folder (PSAUTO) | specifies the directory where P-STAT system files are to be found and saved if they are not specifically located elsewhere. |
| 10. Change Data Folder (PSDATA) | specifies a directory for external data such as labels files, transfer files and data exported from other programs. |
| 11. Show Pathnames | prints the names of all the defined/assumed pathnames. |
| 12. Transfer file | specifies the name of a file containing P-STAT commands to be executed. |
| 13. Run a Macro | causes a popup menu of all the currently defined macros and generates an appropriate macro RUN command. |
| 14. Open and run a macro | uses the file manager to list all files with extensions such as .txt and .mac which might be macros, executes the macro to make it active and generates the macro RUN command. |

The Statistics menu contains seven items:

- | | |
|---------------------|--|
| 1. Frequencies | use the COUNTS command to produce counts, percentages and univariate statistics for the variables in a P-STAT system file. |
| 2. Description file | produces summary statistics for a P-STAT system file in P-STAT system file format. This uses the MODIFY command. |
| 3. Correlate | produces correlation coefficients for the numeric variables in a P-STAT system file. |
| 4. Regression | does stepwise regressions. |
| 5. TTest | produces t-tests and stores the results in a P-STAT system file. |
| 6. Turf | produces Total Unduplicated Reach and Frequency output. |
| 7. Combinations | produces the number of combinations of N items taken N at a time. A range of sizes can be provided. |

The In/Out menu contains seven items:

- | | |
|-----------------------------|---|
| 1. Textfile input | creates a P-STAT system file from text with a choice of delimiter. |
| 2. Tabfile input | creates a P-STAT system file from text with tabs as delimiters. |
| 3. SPSS portable file input | creates a P-STAT system file from an SPSS export file. |
| 4. Textfile output | causes a P-STAT system file to be written out as a textfile with a choice of delimiter. |

- | | |
|-----------------------|---|
| 5. Tabfile output | causes a P-STAT file to be written as a text file with tabs as delimiters |
| 6. SPSS format output | causes a P-STAT file to be written in SPSS export format. |
| 7. Unmake file | creates an ASCII file that has all the information needed to move a P-STAT system file to another operating system. |

The Manage menu contains six items:

- | | |
|--------------------------|--|
| 1. Modify a file | modify a P-STAT system file, selecting variables and cases to create a new system file and an optional description file. |
| 2. Concatenate files | joins as many as 20 files sequentially. The variables in the output file are a super set of all the variables found in the input files. The number of cases in the output file is the total of the cases found in the input files. |
| 3. Lookup table files | joins the cases in one file to the cases in one or more tables files. The output file contains all the variables in the input file plus any new variables in the table files. The default is to have a row in the output file for every row in the input file even if there is no matching row in the table files. |
| 4. Sort | creates an output file that is arranged in sort order by one or more numeric or character variables. The output file can be in either ascending (assumed) or descending order on the BY variables. |
| 5. Aggregate (summarize) | produces means, sums, standard deviations, minimum scores, maximum scores and counts for an entire file or for subgroups as defined by “BY” variables. The file must be sorted or ordered on the BY variables. |
| 6. Duplicate detection | checks a file for extra (duplicate) entries. The contents of the output file depend on the selection criteria. For example, the first case or the last case in a group might be selected. Groups are defined by their values on “BY” variables. The file must be sorted or ordered on the BY variables. |

Additional features cause an output file to have all the cases in the input file with additional variables for the size of each group and the sequence number of each case within the group.

2.3 ORGANIZATION OF INFORMATION

Computer programs use *information* stored in *files*. A file drawer in an office may contain many different files, each with information about a different subject, and each organized in a particular fashion. Similarly, a collection of files in a computer may differ both in their content and in their organization.

External files are files which contain the labels or the data that you are bringing into P-STAT. When you request an external file, the program manager is displayed and you select your file from files that are listed.

2.4 Basic Elements of a P-STAT System File

The basic unit handled by the P-STAT system is a single piece of information. This may be a *number* or a *character string*. If it is a number, it may be an integer or a real number of any size. If it is a character string, it may be anywhere from 1 to 50,000 characters in length. The information or *data* shown in Figure 2.1 has both numbers and character strings.

A P-STAT system file is a rectangular arrangement of data. In addition to the data, the name of the file, the names of the variables and the data type of each variable are contained in the P-STAT system file. Figure 2.2 illustrates the different elements of a P-STAT system file.

The *name* of this file is “Tennis”. Each P-STAT system file has a name. There are five categories of information in the Tennis file.

They are Last Name, First Name, Sex, Age and USTA (United States Tennis Association) Rating. These are *variables* because the information in each of these categories varies or is different for each person. Each line in the file is a *case*. There are three cases in this file.

Figure 2.1 The Information (Data) for the File

Allen	Robert	m	-	4.5
Jones	Karen	f	13	3.0
Prince	Susan	f	14	3.5

The information that a secretary places in a file comes from some source that is external to the file itself. Similarly, the material that P-STAT puts into a file comes from some *external source*: a terminal, a disk or a tape. Just as the secretary must be supplied with the information to place in a file, P-STAT must similarly be told where the input data can be found. Sometimes the data may exist only on a piece of paper or in your head. Since most computers can’t mind read, the data must be made readable — it must be typed on the terminal or in a file. Figure 2.1 shows *raw data* for a file as it might appear on a piece of paper or in a file on disk.

P-STAT system files and external data files are the two types of files that are accessed through these menus. P-STAT system files are files created by P-STAT commands. P-STAT system files are discussed more fully later in the next section of this chapter. When the menu item expects an existing P-STAT system file, you are given a list of the existing system files to choose from. When the menu item expects a new P-STAT system file you are prompted for a name and a check is made to ensure that this name is a legal name for a P-STAT file. A similar procedure is used when a variable name is expected.

Figure 2.2 Elements of a P-STAT System File

File Name:	Tennis				
Variable Names:	Last Name	First Name	Sex	Age	USTA Rating
Data:	Allen	Robert	m	-	4.5
	Jones	Karen	f	13	3.0
	Prince	Susan	f	14	3.5

P-STAT system files are arranged in this rectangular *case-by-variable* format. The variables are the *columns* and the cases are the *rows*. These files are said to be in a rectangular format because each case contains the same number of variables, in the same order. The cases generally correspond to persons, subjects, measurements, or other similar items. The variables are information about varying aspects of the cases. There may be many variables for each case and there may be many cases. A P-STAT system file can have an *unlimited* number of cases and 6,000 to 250,000 variables, depending upon the size of the P-STAT system being used.

P-STAT system files are written in a form that cannot be read by any program except P-STAT. The data are packed to conserve space and the amount of I/O (input/output) that is needed to read them. P-STAT system files

are referenced by name. The name can be up to 16 characters long. It must start with a letter and it can contain only letters, numbers and decimal points. The case of the letters does not matter. Enter it the way that you want it to print. Then you may use either upper or lower case when you refer to it.

2.5 Missing Data

Sometimes information is *missing*. This can happen in a survey when a person refuses to answer a question or in medical research when a patient misses a visit. Missing data are permitted for both numeric and character variables.

Three different missing values may be defined for different types of missing data. This makes it possible to distinguish, for example, among the person who was not asked the question, the person who refused to answer, and the person who replied, “I don’t know”. The three types of missing data are represented by *dashes*: in the printed output

```
MISSING1  -          MISSING2  --          MISSING3  ---
```

Robert’s age is not known (see Figure 2.2), so a dash is used to indicate missing data. Similarly, if Karen’s sex were missing, one, two or three dashes would be used to represent the data value for that variable.

2.6 Legal Names

Each file and each variable in the file must have a name. P-STAT only recognizes certain names, and these are referred to as “legal names”. Legal *file* names must:

1. start with a letter,
2. have no more than 16 characters, and
3. contain only letters, numbers, and decimal points.

Legal *variable* names must:

1. start with a letter,
2. have no more than 64 characters, and
3. contain only letters, numbers, decimal points, and underscores.
4. may have an optional 16 character tag delimited by a double colon (::)

```
Test1::English_composition
Test2::Introduction_to_algebra
```

Variables can be referenced by either their tags, their full names or by using the wildcard (?) to construct a shortcut. In printout you can select whether the variable should be represented by its TAG, its TEXT or the full name. You can also select the way variable names are to be presented in the printout. Not all commands behave the same way with variable names. Commands that print numeric matrices may print only short names while the SURVEY command has subcommand controls that permit selective labelling. For example, banner points can to be labelled with TAGS while stub variables are labelled with the full TEXT. The menus will often display TAGS when the full name cannot fit the display area.

Most names are one or more “words” that describe the variable — however, the words must be connected by a decimal point or underscore to form a single name *without any embedded spaces*. “Last.Name” or “Last_Name” is correct. When the file is listed, P-STAT omits the decimal point or underscore.

There are 2 other special variables: *system variables* and *scratch variables*. System variables have similar rules except they are limited to 16 characters and always begin and end with a decimal point. Scratch variables have similar rules except that they are limited to 16 characters and always begin with one or two pound (#) signs. For example, .DATE. is the system variable for the current date and ##counter is a permanent scratch variable created by the user to hold a single number. Since system variables and scratch variables are limited to 16 char-

acters, they cannot have tags. Scratch variables which begin with a single # sign exist only for the duration of the current command. Scratch variables which begin with double ## signs exist for the duration of the current P-STAT job..

The keyword TO is a *reserved* word in P-STAT and may not be used for a variable or a file name. The use of reserved words simplifies the P-STAT language. It would be wise not to use BY, OUT, or any common prepositions as variable or file names because it is possible they may become reserved words in the future.

It is also a good idea not to use single letter names for variables, especially the letters H, Q, E and V. In some situations, confusion may exist because these letters have special meanings in P-STAT as abbreviations for HELP, QUIT, EDITOR and VARIABLE.

The names in P-STAT may be any combination of upper case and lower case characters. P-STAT remembers the way the names are entered and uses those representations when printing them. However, references to the name may be made in upper case, lower case or mixed case, *regardless* of how the name was entered originally. Thus, a P-STAT system file with the name Myfile can be referred to as “myfile”, “MYFILE”, or even as “myFILE”.

The next step in using P-STAT is to create one or more P-STAT *system files* from your data. The easiest way for new users to get started is to use the menus. These menus create P-STAT commands in a command language format.

2.7 CREATING A P-STAT SYSTEM FILE

If your data are stored in Excel use “Save As” and select the type as “Text (tab delimited)(* .txt)”. Provide a name for the file and be sure that Excel stores this data in the folder that you will use for your P-STAT run. If your data are stored in a text file created by some other program or editor, note where the file is stored and the character that is used to delimit each variable (tab, comma, blank).

One important feature of the data which has not been discussed is the difference between variables that contain just numbers and variables that contain other characters. The TEXTFILE.IN command looks at the data and determines the mode (numeric or character) for each of the variables. The MAKE command, which can handle more complex data files, must be told the data type for each variable. Similarly, if you use PPL (the P-STAT Programming Language) to create a new variable, character data must be explicitly declared when the new variable is created.

2.8 The First Step

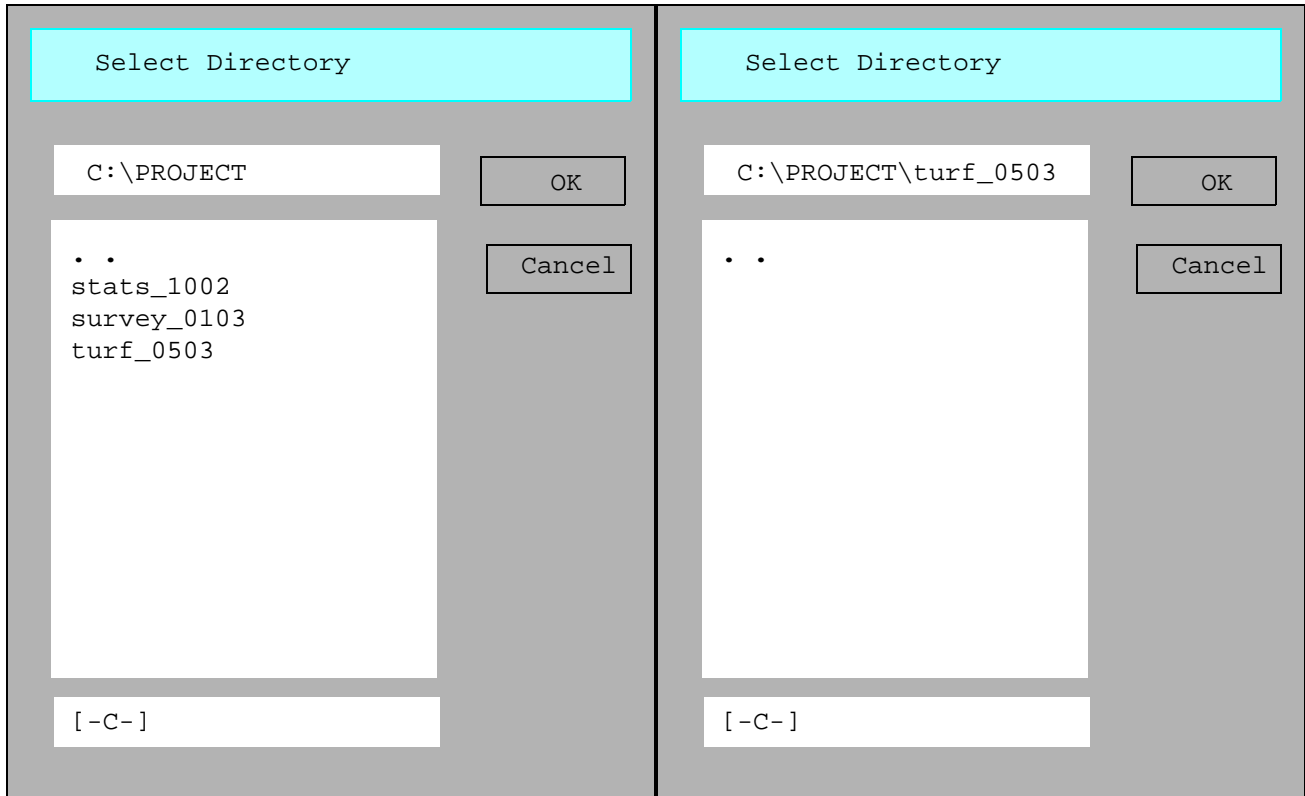
1. Click on the P-STAT icon or use RUN to point to the P-STAT module.
2. In the top menu click on Utilities.
3. Click on “Change All Folders (PSFILES)”. This causes the “Select Directory” dialog box to appear. Use the mouse to move to the disk and the folder that is to be the working directory for your current project. This should be the folder (directory) where your input data file is stored.

In the illustration on the next page,, the initial folder is C:\PROJECT. It contains 3 folders. If you click on OK at this moment, the folder that is selected for the PSFILES command is “C:\PROJECT”. You can move down the tree of folders by selecting one of the sub-folders and using a double click of the left mouse button. If that folder itself contains folders, you may select one of them. If you wish to move up the tree, select (double click) the line with the double dots (..).

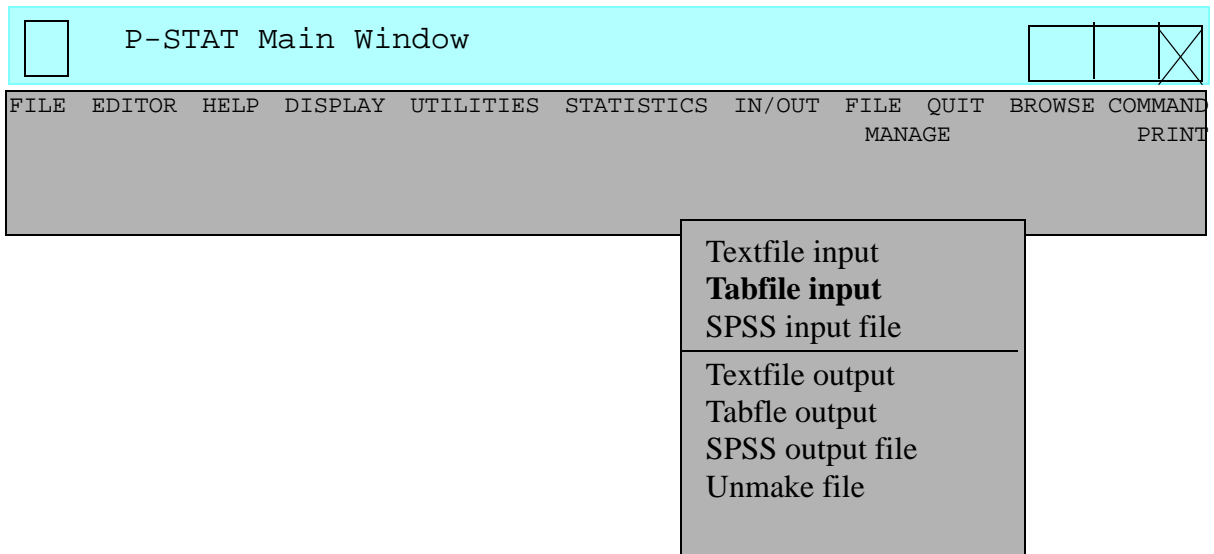
Do NOT select “OK” until the complete path to the folder you want is echoed in the line to the left of the “OK” button.

You will now see echoed in the P-STAT window a line such as:

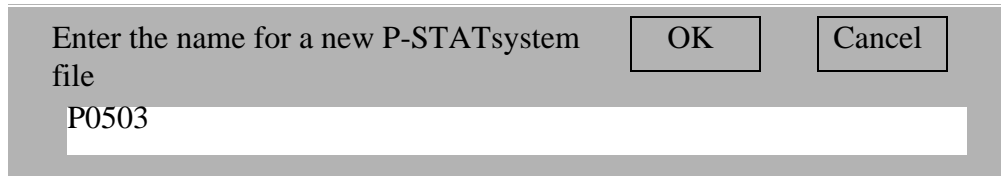
```
PSFILES 'C:\PROJECT\TURF_0503' $
```



In the top P-STAT menu click on “IN/OUT” and then on “Tabfile Input”. The process of creating a TABFILE.IN command begins when you select (click on) the command.



You are first asked to provide a name for the new P-STAT system file that TABFILE.IN will create. This name must follow the rules for legal names described earlier in this chapter.

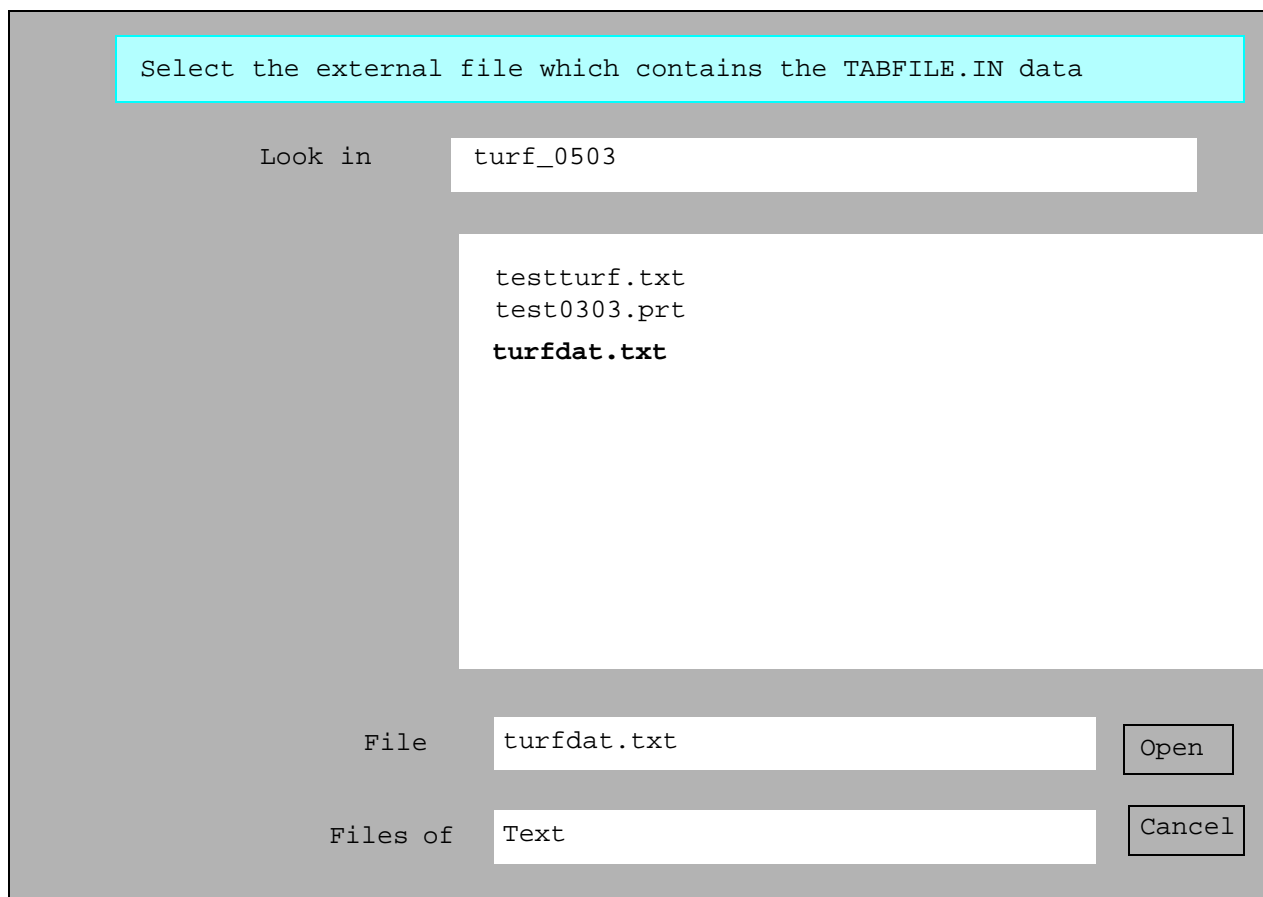


Enter the name for a new P-STAT system file

P0503

OK Cancel

The TABFILE.IN command also needs to know the name of the data file that you exported from EXCEL. If the folder you selected is the one where you stored your tab delimited data, you will see it listed in the file manager dialog box that is now displayed. Select the desired file and click on “Open”.



Select the external file which contains the TABFILE.IN data

Look in turf_0503

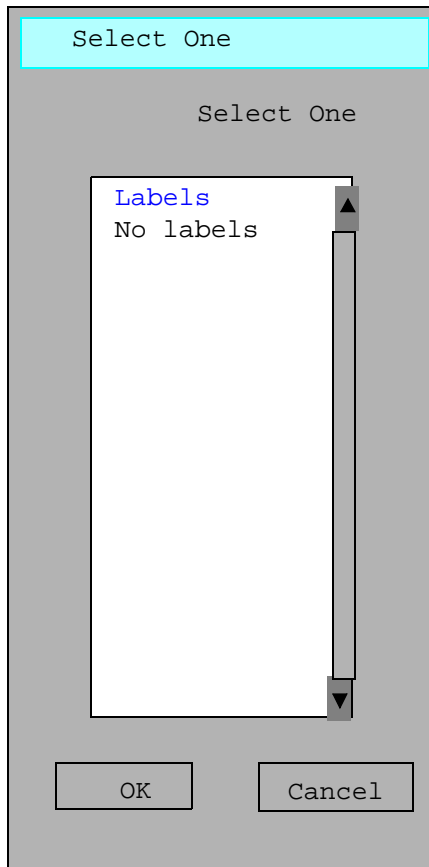
testturf.txt
test0303.prt
turfdat.txt

File turfdat.txt Open

Files of Text Cancel

Finally P-STAT needs to know whether the file that you exported from Excel had a first line with variable names (labels) or whether P-STAT must create them. This is a required identifier. Since “Labels” is highlighted you need only press the “OK” button if the first row of your data has variable names. If these names do not conform to the P-STAT rules for variable names, they will be truncated and characters converted as needed to make legal labels.

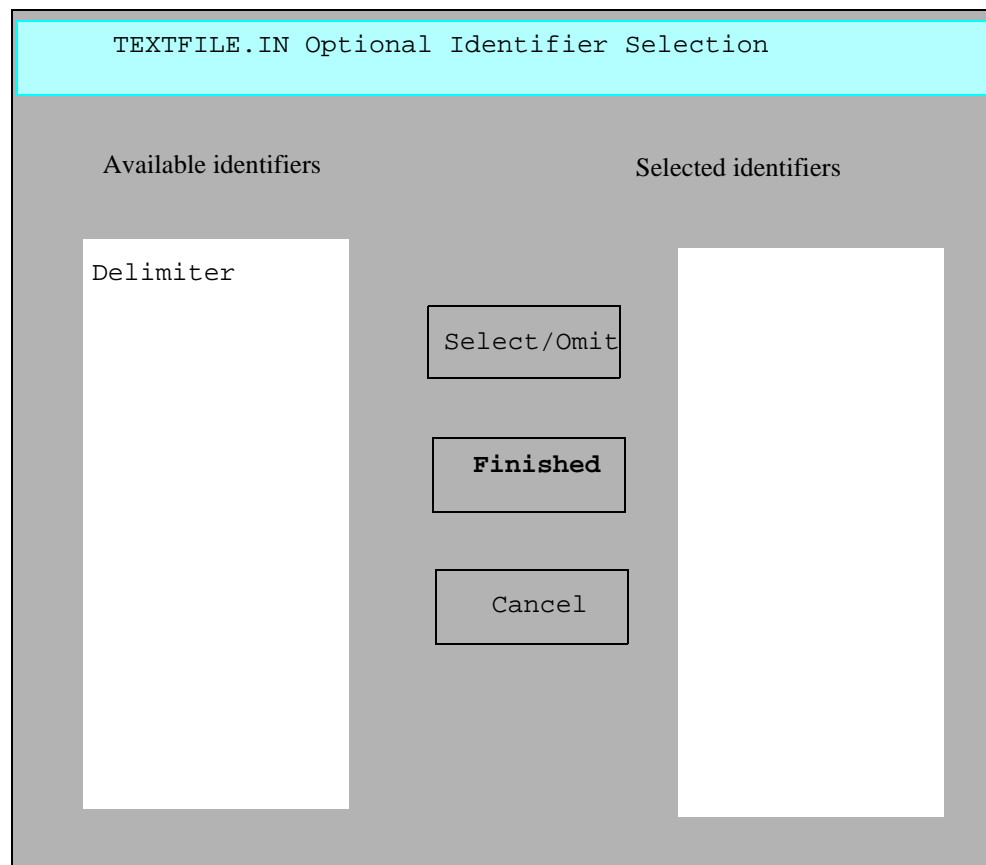
If your data file has no labels, select “No labels and press the “OK” button. The P-STAT generated labels will be in the form of “var1”, “var2”, etc..



The graphic on the left illustrates the Labels/No labels menu.

Most P-STAT commands have one or more optional identifiers. These identifiers are presented in the double menu format illustrated below. This is the same format that is used when selecting variable names.

TEXTFILE.IN has a single optional identifier, DELIMITER, which can be used to specify that the delimiter is a TAB, a COMMA or a BLANK. This means that TEXTFILE.IN can be used to import both comma delimited (.CSV) and BLANK or space (.PRN) delimited files as well as TAB delimited files. TAB delimited is the default and does not require the DELIMITER identifier.



When you have finished making your selections, you will see on your screen the completed command text which will look something like:

```
TABFILE.IN p0503, READ 'C:\PROJECT\TURF_0503\turfdat.txt', LABELS $
```

“p0503” is the name you supplied for your P-STAT system file. The argument for the identifier READ is the folder you selected with the PSFILES command and the file that you selected from the file manager. In addition you will see either “Labels” or “No labels”. The TABFILE.IN command is automatically submitted for execution and results are written on your terminal. The file “p0503” is now available for use in a P-STAT command such as TURF, SURVEY or LIST.

2.9 The LIST Command

LIST is the command that displays P-STAT system files. It lists the *most recently referenced* file, unless it is given a specific file name as its argument. The steps to list the file you have just created are:

1. Select Display from the top menu
2. Select List from the drop down menu
3. Select the desired file from the list of files.

Click on OK

See the next chapter for a real project with an example of modification and information on printing the results

4. Click on the “Finished” button in the LIST Optional Identifier selection dialog box.
The optional LIST identifiers are not needed for this exercise.

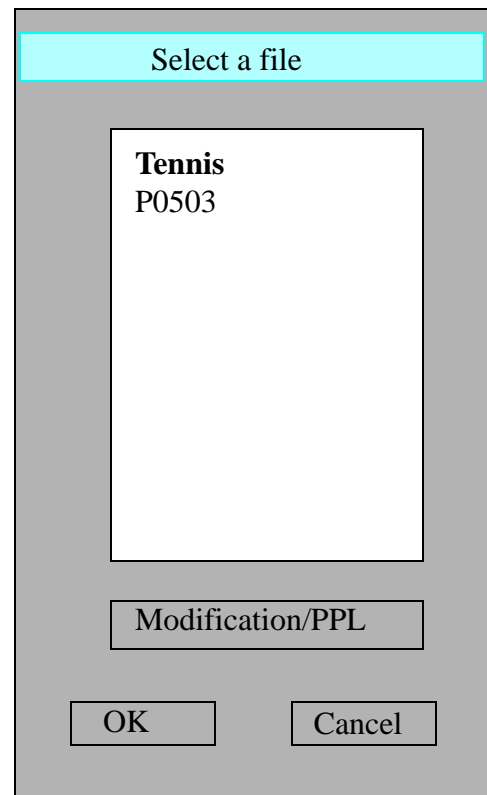
The following listing appears on your terminal

Last Name	First Name	Sex	Age	USTA Rating
Allen	Robert	m	--	4.5
Jones	Karen	f	13	3.0
Prince	Susan	f	14	3.5

LIST automatically sets the proper spacing between the variables so that the listing is easy to read and yet concise.

LIST has many optional identifiers that allow the user to control spacing, layout and data format. To direct the listing of a file to a diskfile, select the optional PR identifier and provide a name for the file on disk before selecting Finish.

It is a good idea to be selective in the extensions that you use for external file names. “.txt” is a good extension for an ASCII input file. “.lab” is a good extension for a labels file. Printfile extensions of “.prt” or “.txt” are both good extensions for print files stored on disk. Consistency in selecting these extensions will simplify housekeep-



ing of your directories. P-STAT uses “.TMP” as the extension for temporary files. P-STAT uses “.PS1” and “.PS2” as the extensions for P-STAT system files. Do NOT rename the P-STAT system files or P-STAT will not be able to access the data stored in those files.

2.10 Ending the Run

To end the P-STAT session:

1. Move the mouse to the top menu
2. Select button in the top right-hand corner of the “P-STAT Main Window” line

Any temporary files are now erased.

The P-STAT window may be minimized (become an icon though still active) and be maximized (use the entire screen). The exit button is only enabled in the main window. It is not in effect in sub-windows such as the P-STAT editor.

2.11 Autosave and External Files

P-STAT is an *autosave* system — *system files* are *automatically saved*. After you make a system file, it is on your disk and available for use at any time. P-STAT adds the suffix of “.PS1” or “.PS2” to the file’s name. Refer to your file by the name that you give it and do NOT include the extension. P-STAT selects the one that is more recent. The FILES command lists all P-STAT system files and external files *referenced in the P-STAT session*. It shows both the name you gave the file and the full *pathname* on disk. Figure 2.3 shows a report from the FILES command.

To execute the FILES command:

1. Select File from the top menu
2. Select Active Files from the drop down menu

In the FILES report, autosave files are displayed first; external files are displayed next. The file named “Tennis” is stored on disk as “TENNIS.PS1”. If this file is modified, the original file becomes a *backup* file and the modified file becomes “TENNIS.PS2” — that is, both *current* and *previous* versions of the file then exist. In the report in Figure 2.3, there is only a current version of Tennis. You cannot tell by the extensions which file is the current file as the extensions (PS1 and PS2) alternate each time that the file is modified. P-STAT uses an internal counter to keep track of which is the current and the backup version of each file — you need only refer to the file by its name “Tennis” to get the most recent version.

Figure 2.3 **Displaying Autosave and External File names**

```

-----autosave files-----
name          current          previous
Tennis        TENNIS.PS1

-----external files-----
name          usage          pathname
              External use  tennis.dat
              Printer    Tennis.txt
-----

```

This use of file extensions can be a convenience providing one level of backup for your data. It is also a 2-edged sword. In a learning situation it is useful because you can execute and re-execute commands without

change. However, if you use this feature with your working data and make an error, you may accidentally delete your backup.

External files are displayed after the autosave files. External files include raw data files such as “tennis.dat” and printer files such as “Tennis.txt”. Other external files that might appear in the FILES report are labels files, edit files, and files containing P-STAT commands which are referred to as “transfer” files. These files are discussed later in this manual.

When you use P-STAT in a subsequent session, select the file you want from the list of P-STAT system files. These are the files that are located in your autosave directory. Your autosave directory depends on whether you have set PSFILES or PSAUTO environment variables and the way that you invoke the P-STAT command.

1. If you invoke P-STAT using the system RUN command or by clicking on the P-STAT icon and have no environment variables defined, the assumed autosave directory is either the directory where the P-STAT module is stored or your user folder under the “Documents and Settings” folder.
2. If you open a command window and invoke P-STAT from that window, the assumed autosave directory is the folder you were in when you invoked the P-STAT module.
3. If you have a PSFILES or PSAUTO environment variable set, that takes precedence no matter how you invoke the P-STAT module.
4. If you have a PSTART file in the folder where the run begins and that PSTART file has either a PSFILES or PSAUTO command, that takes precedence over any environment variables.

A good way to begin the run with the certainty that P-STAT can find your files and that P-STAT will write any new files where you want them is to:

1. If you think that you have PSFILES correctly set, you can check the settings by selecting the “Utilities” menu and clicking on “Show Pathnames” \$
2. If the report is not what you expect, Select “Utilities” and then select “Change All Folders (PSFILES)” and provide the correct path.

You can change the assumed locations for your P-STAT system files (PSAUTO) and for your external data files (PSDATA) any time during the P-STAT session. You may want to store your P-STAT system files in a folder other than the folder where you have your input data. Any time that you are unsure where P-STAT is writing its output or cannot figure why P-STAT cannot find your data, the PATHNAMES command is useful.

2.12 MORE FEATURES

The explanations in this introductory chapter are not complete, but they are sufficient to allow a new user to begin using P-STAT. There are other ways to create files, and there are many optional identifiers to control various ways in which commands operate. If your data have already been entered into a PC program, you need not reenter it. Most PC programs can export data in tab delimited text formats. If your data are in SPSS export format you can import them into P-STAT using the same steps used for TABFILE.IN:

1. Select IN/OUT from the top menu
2. Select the appropriate format from the menu
3. Supply a name for the P-STAT file and the name of the file where your program has stored the data.

Perhaps the best way to learn how to use the P-STAT system is to create small files like the one shown in this chapter and experiment. Cycle through the top menus and their sub-menus and try each item in turn. For advanced features consult the manuals and the on-line help file.

The on-line help file contains information on both commands and general topics. The help file can be accessed by using the Help button on the top menu. There are two choices. The first entry brings up the table of contents for a helpfile which has just the commands supported by the top menus. The second gives you the keyword names for the full P-STAT HELP file.

To get help on the current release:

1. Select Help from the top menu
2. Select “Full Help File”
3. Select “Utility Topics”
4. Select “NEWS”

The NEWS feature is important because P-STAT is an evolving system — enhancements are made regularly. They are included in the latest version, and new versions are available on-line for those P-STAT sites that are under maintenance. The NEWS contains information about new features and instructions on how to use them.

SUMMARY

P-STAT MENUS

1. P-STAT menus can be accessed from the bar at the top of the P-STAT window. The items in this top menu are:

File	Editor	Help	Display
Utilities	Statistic	In/Out	File Manage
Quit	Browse	Command Print	

7 of these items have their own menus which create P-STAT commands. Editor transfers control to P-STAT's internal full screen editor. Quit is used to interrupt runaway printout on the screen. Browse is used to search through and edit the most recent 1000 lines that have scrolled off the screen. Command Print sends the output from the most recently executed command to the system printer.

2. A click of the mouse outside the menus removes any pop-up menus and returns the system to command mode.
3. Menu items can be selected by using a mouse, the arrow keys and a carriage return, or the first letter of the menu selection.

P-STAT NAMES

File Names in P-STAT

File names must begin with a letter and can be up to 16 characters long. They may contain letters, numbers and decimal points.

Variable names in P-STAT

Each variable has a unique name which may be up to 64 characters and contain letters, numbers, decimal points, and underscores. The use of a double colon (::) is allowed to divide the variable name into 2 parts: the first up to 16 characters is known as the "tag", followed by the double colon (::) and finally additional text for a maximum of 64 characters. Not only must the variable have a unique name, tags, if they are used, must also be unique. If you do not provide a name, P-STAT generates a name in the form of var1 to var999.

LIST

After a file is built you can display it by using the LIST command. LIST is accessed by selecting Display from the top menu and the pressing the List button.

PATHNAMES

PATHNAMES is a command which tells you where P-STAT looks for P-STAT system files, input data files and temporary work files.

PSAUTO

PSAUTO is a command that may be used any time during a P-STAT session to specify where P-STAT should look for and store P-STAT system files. This overrides any definition supplied in a PSFILES command.

PSDATA

PSDATA is a command that may be used any time during a P-STAT session to specify where P-STAT should look for and store external data files including input data, labels files, transfer files containing P-STAT commands and print output files. This overrides any definition supplied in a PSFILES command.

PSFILES

PSFILES requires as an argument the full drive and directory (folder) where all the files for the current run are to be found and all new files written. If PSFILES is issued after the run has begun it changes PSAUTO and PSDATA but does not change the folder for PSTEMP. PSFILES is located in the “Utilities” menu with the label “Change all folders (PSFILES)”.

LIST

After a file is build you can display it by using the LIST command. LIST is accessed by selecting Display from the top menu and then pressing “List a file”..

TABFILE.IN / TEXTFILE.IN

These commands are used to take a text file of data which has tabs as the delimiters between values and create a P-STAT system file. This tab delimited format provides an easy way to move from Excel to P-STAT. This can also be used to import a file with a delimiter other than a tab. If the first record does not have variable names, P-STAT will generate them for you. TABFILE.IN and TEXTFILE.IN are identical except that you do not need to supply a delimiter for TABFILE.IN if your data are indeed tab delimited.

3

A Single Project

This chapter contains step by step illustrations for a single project. Some of this will be a repetition of steps from the previous chapter.

The project is to do a TURF (Total Unduplicated Reach and Frequency) analysis. The instructions assume that there is an existing folder for the project and that the data exist in an Excel spreadsheet. The full path to this folder is “C:\PROJECTS\TURF_0503”. The following steps are illustrated:

1. Export the data from Excel as a tab delimited file.
2. Enter P-STAT and tell P-STAT where the files for the project are located.
3. Import the tab delimited file into P-STAT creating a P-STAT system file.
4. Run the TURF command selecting basic options.
5. Print the results.

3.1 EXPORTING THE DATA FROM EXCEL

The data file has 30 variables; a case id and 29 items. These items are coded either 0, which represents a NO, or 1 which represents a YES to a question about the television programs that the respondent watches. The purpose is to choose the 5 programs that reach the most people. These will be the programs that make the best use of our advertising budget.

The first step is to export this data from Excel as a tab delimited file. The process is:

1. Click on “File”
2. Click on “Save As”
3. Select the “Save as type” as “Text (Tab delimited)(*.txt)” and provide a name. For this project the name is “turfdat.txt” and it is to be saved in C:\project\turf_0503

3.2 PREPARING P-STAT

The first step is to tell P-STAT where all of the files for this project are located. This is done using the PS-FILES command. PSFILES provides a default folder for all the files, input and output, that might be used during the run. The previous chapter covers this in some detail. The following is a summary of the steps that are needed.

1. Click on “Utilities” in the top menu
2. Select “Change all Save Folders (PSFILES)”
3. Work your way through the “Select Directory” dialog box double clicking to make your selections.

In this example, the starting folder at the beginning of the P-STAT run is “C:\psource”. This is the starting directory for this example. There are three sections to the “Select Directory” dialog box illustrated in the previous chapter::

1. The selected directory/folder line
2. All the subdirectories under the current folder, starting with “..” which moves control to the directory above the current directory.

3. A list of the available drives.

You can change drives by selecting a different drive in the bottom section. You can work your way up and down through the folders by making selections from the middle section or you can type your desired drive and folder directly in the top section. To move up the directory tree *double click* the “..”. To move down the tree select a folder from the list of available folders with a *double mouse click*. When you have located the directory/folder that is to be your current working folder, click on the OK button. Be sure that you have selected the folder you want and that the *selected directory line* at the top of the dialog box contains the full path before clicking on OK.

3.3 IMPORTING THE TAB DELIMITED FILE INTO P-STAT

The previous sequence creates and executes the PSFILES command which is echoed in the main P-STAT window. The next step is to read the tab delimited file into P-STAT. Move the mouse to “IN/OUT” in the main menu. The “IN/OUT” list contains 7 items. The second one is used to read a tab delimited file and convert it into a P-STAT system file. This sequence is illustrated in the previous chapter. The following is a summary of the steps that are needed.

When you click on “Tabfile input” P-STAT asks for a name for the P-STAT system file you are about to create. Enter a name that is appropriate for the task and that follows the rules for legal P-STAT names and click on “OK”. If you click on “Cancel”, the program returns to the main P-STAT command window. A single click selects items from the menu. Unlike the directory selection a double click from the menus is the equivalent of “OK” and causes an exit from the menu or dialog box.

After you have entered the name for the new P-STAT system file, the program manager appears so that you can select and open the text file exported from Excel. The program manager will list all the files which have extensions such as.txt located in the folder designated by the PSFILES command.

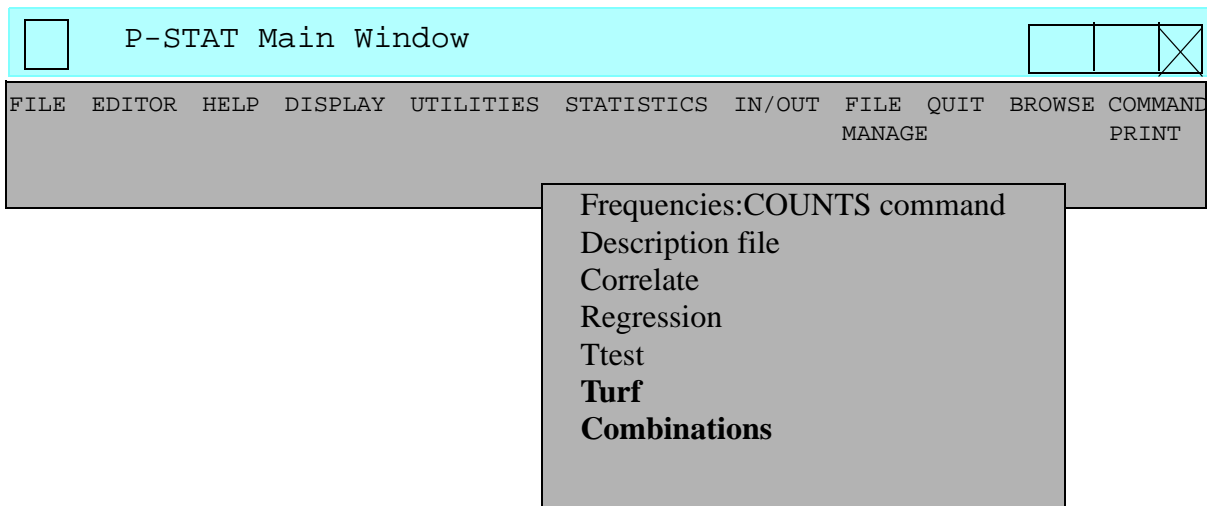
Most commands require some information in addition to file names. The TABFILE.IN command needs to know whether the first row of the tab delimited file contains variable names (labels). In this case, since the first row in the Excel file does contain variable names, “Labels”, the first identifier on the list is the one to select.

Unless your input data were created with a delimiter other than a tab you do not need to make a selection from the optional identifiers. Press the “Finished button” to complete the command.

After P-STAT executes the TABFILE.IN command the third step is complete. You have a P-STAT system which contains the data for your TURF analysis. The name of this file is the name that you provided in the previous step

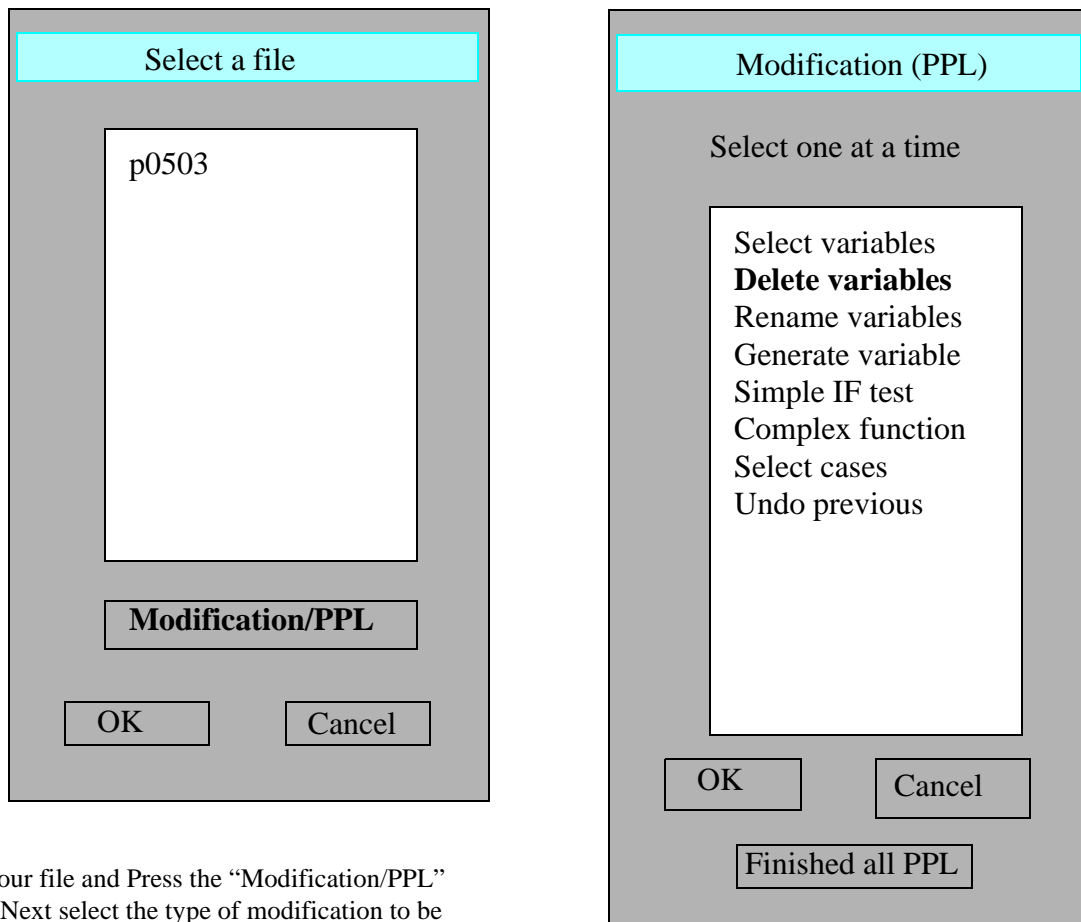
3.4 EXECUTING THE TURF COMMAND

The next step is to click on the TURF command which is found under the heading “Statistics” in the main menu.



After you select the TURF command, you will see a menu with all the P-STAT system files that you have created during the current P-STAT session. Here it is the single file “P0503”. TURF expects to use all the variables in the input file as either analysis or weight variables. The file in this example contains the case ID as well as the analysis variables. Since TURF analyzes all the variables in the file except a specified weight variable, the id variable must be removed before the analysis can be done.

A P-STAT system file can be modified ANYTIME that it is input to any P-STAT command. Variable selection is one example of a modification. The modification process can be done from the file selection menu.



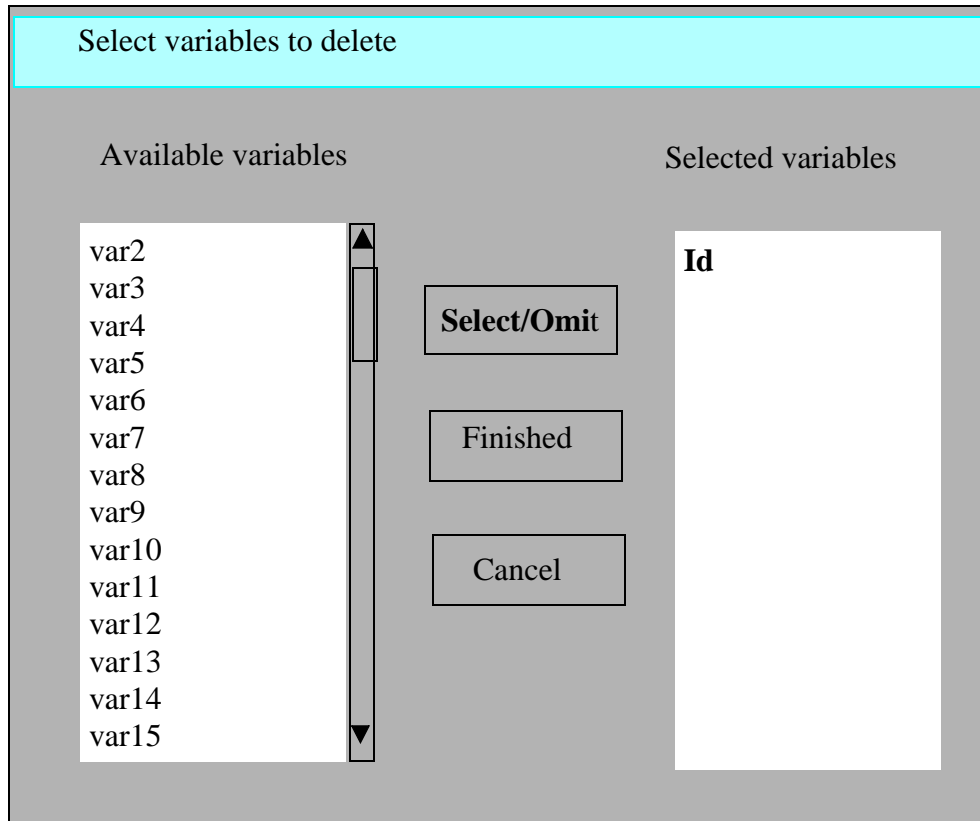
Select your file and Press the “Modification/PPL” button. Next select the type of modification to be done.

There is virtually no limit to the number of modifications which can be done. The command that you are about to execute will receive the data in modified format while the input file remains unchanged. If you wish to make permanent modifications you may use the MODIFY command (see “File Manage” in the top menu). MODIFY is a vehicle for PPL. It can be used to create a new P-STAT system file which reflects all selections and modifications.

It does not matter whether you select or delete variables. Use whichever is easier for the particular situation. The “Select Variables” and “Delete Variables” dialog boxes each have two menus: one contains a list of the available variables; the second contains the selected variables. The procedure is similar to the procedure used in selecting identifiers.

A variable is selected by moving your mouse, highlighting one or more variables and then clicking on the “Select/Omit” button. This moves your choices to the “Selected variables” menu. It is not enough to highlight your choice. It *must be moved* to the column of selected choices. When you are satisfied that the list of selected variables is correct, click on the “Finished” button.

When you are selecting variables, the variables that will be available for the analysis are those in the selected list. When you are deleting variables, the variables that remain are those still in the list of “Available variables”. When the variable selection/deletion is complete press the ‘Finished’ button and control returns to the PPL menu.

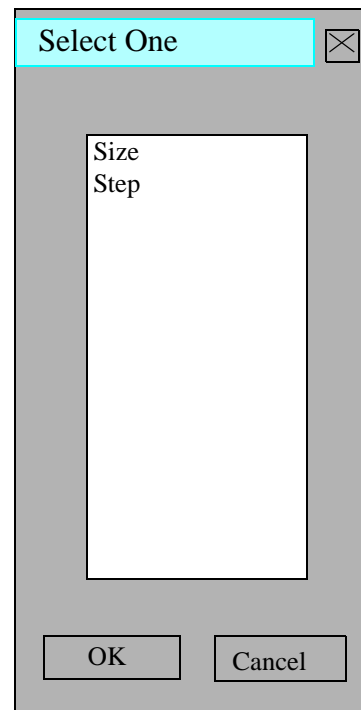


Since there are no more modifications needed for the current project, click the “Finished all PPL” button. The filename and all of the modifications requested for that file satisfy the first requirement for a TURF analysis. You will see the beginning of the TURF command appearing in the main P-STAT window. It looks something like:

```
Turf P0503
[ DROP Id ;
]
```

There are 2 different ways to run TURF. The basic way is to do all possible combinations of one or more selected sizes. An alternate way is to do a stepwise run. In a stepwise run the variables that are determined to be the best combination in a given step are forced into the next step. This results in fewer passes through the data file but the final result may not be the best combination overall. STEP is used when the number of possible combinations is so large that the run may take hours or days to complete. It is also used to produce cascading step where the best single variable is computed and becomes the basis for the next step. The variables are added one at a time with each step computing the best combination given the preceding combinations.

After selecting the file and the variables, you must decide whether to do a SIZE run or a STEP run. For this project we select Size.



The other requirement for a TURF analysis is at least one size for the combinations. Size 3, for example, tells the command to look at the items (variables) in groups of 3. The first group to be analyzed will include var2, var3 and var4. The second group will include var2, var3 and var5, etc. through var28, var29 and var30. Size combinations can be 1 to 60.

It is possible to do more than one size in a single TURF run. The graphic below is the graphic that appears to enable SIZE or STEP selection. Select the desired size or step, click on the “Select/Omit” button to move that size to the selected column and then click the “Finished” button when you have finished selecting sizes. The menu with all of the other available options for the TURF command appears in the next dialog box.

The rules are different for SIZE and STEP. SIZES may be entered in any order and they will appear in the order in which they are selected. STEP sizes must ascend.

Sizes Available	Sizes Selected
1	5
2	
3	
4	
6	
7	
8	
9	
10	
11	
12	
!3	
14	
15	
16	

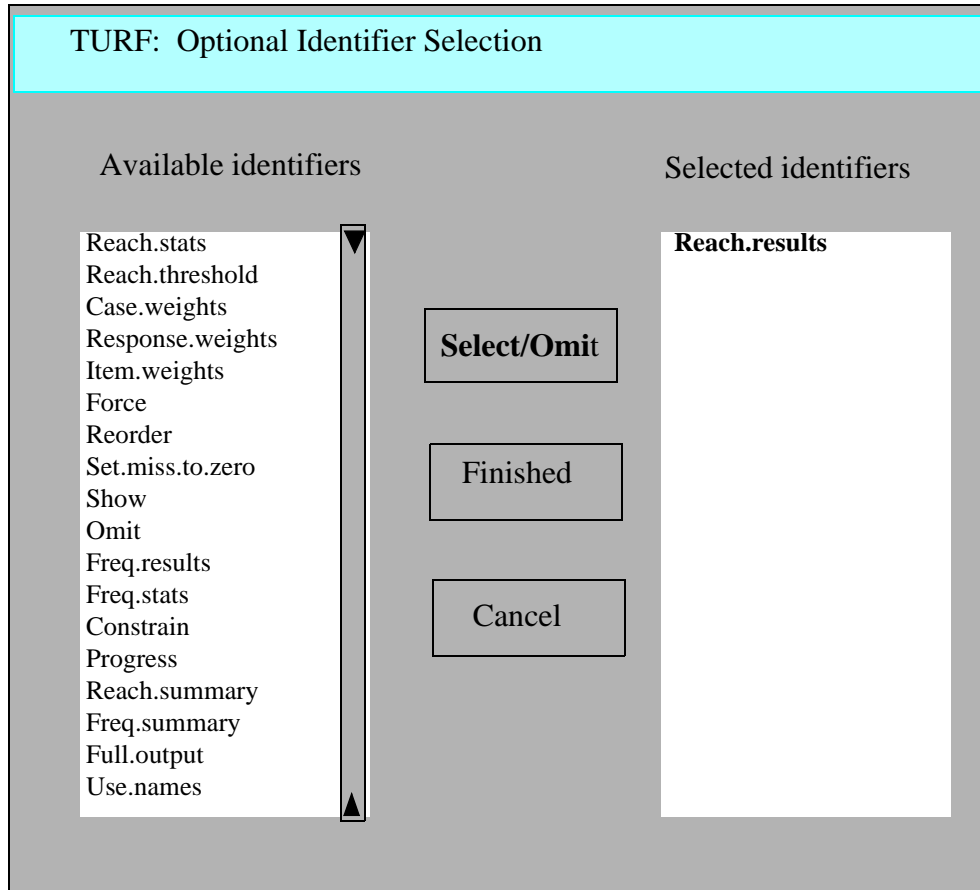
Buttons: Select/Omit, Finished, Cancel

The file name and the number of combinations (SIZE or STEP) are always required. In a SIZE run this is sufficient, but TURF has many optional features and you will probably want to select at least one of the output files. The most useful selection is the “Reach.results” file. This is a file in which the rows contain the variable names for each combination. These rows are sorted so that the first row contains the variable names for the combination that “reached” the most people. If there are ties, the frequency is used to determine the final sort order.

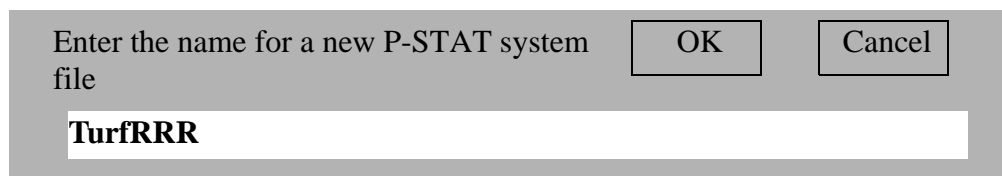
In a STEP run you must select either the REACH.RESULTS or the FREQ.RESULTS identifiers for the run to execute.

The procedure in the “Identifier Selection” dialog box is to select the identifier you want by a single mouse click or by moving the down arrow key. Then press the “Select/Omit” button to move that identifier to the “Selected identifiers” list. (NOTE: a double click is interpreted as “Finished”.)

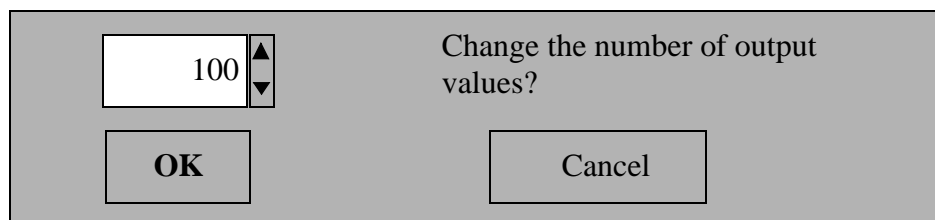
You may select as many steps/sizes as you need. To change the selections click on a selected item in the “Sizes Selected” menu and then on “Select/Omit”. This moves the item from selected to available. The position is changed. A de-selected item becomes the last item in the available list.



The REACH.RESULTS identifier has two arguments. The first is the name to be given to the “reach.results” file. The “reach.results” file is a P-STAT system file and the name must follow the rules for a P-STAT system file name. For example:



The second is the number of cases to be written to the “reach.results” file. If you do not change it, the number is set to 100 cases. NOTE: if you select multiple sizes the default for output values is 100 for each size.



After you click “OK” the next screen is a return to the “Identifier Selection” window. The other identifiers have similar arguments. However, for many runs “REACH.RESULTS” will be sufficient. Press the “Finished” button when you have completed your selections. The TURF command is now automatically executed and the report displayed on your terminal

Figure 3.1 Turf Report

```
-----TURF analysis for file P0503 completed-----
OPTIONS: none

    29 items were read.
    550 cases were read and used.
    213 cases had at lease one politive response,
        making that the maximum possible reach.

SIZE   5 evaluated 118,755 combinations:
    213 was the best REACH, found in 12 combinations.
    746 was the best FREQ in those 12 combinations.
    867 was the best FREQ in any size 5 combination.

The FREQ score for a combination is the count
of the non-zero responses for that combination,
summed over the reached cases.

REACH.RESULTS file TurfRRR has the 100
combinations with the highest reach scores.
The items are ordered by their REACH contribution.
Cumulative reach is shown.

Time: less than 0.1 second.
```

If you wish to print this report, click on the “Command Print” button at the top left of the menu in the main P-STAT window.

The TURF information of the most interest is in the new P-STAT system file, TurfRRR. Since this is a P-STAT system file, it is binary and can only be read by a P-STAT command. The LIST command reads a P-STAT system file and reformats it as text. If LIST is not told otherwise, it prints the text on the screen. The LIST command is selected by clicking first on “Display” in the top menu and then on “List a file”.

LIST needs to know which of our files to LIST and then which LIST identifiers to use. First select TurfRRR and since there are no modifications click on the “OK” button. LIST has many optional identifiers to control the appearance of the printout. To execute LIST using the defaults, click the “Finished” button.

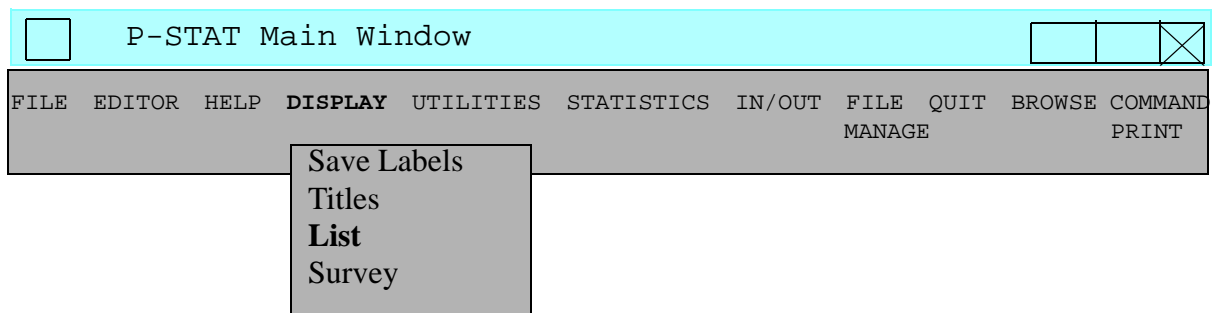


Figure 3.2 Turf: Reach.results Output File

```

FILE TurfRRR

TURF results ordered by REACH, from input file P0503 using 29,5    PAGE 1

size
and
rank    reach    pct    freq    stats    item    item    item    item    item
          reached
5_1      213    38.727    746          VAR3    VAR4    VAR5    VAR23    VAR13
          cum.r    186    208    210    212    213
5_2      213    38.727    744          VAR3    VAR4    VAR5    VAR23    VAR15
          cum.r    186    208    210    212    213
5_3      213    38.727    720          VAR3    VAR4    VAR5    VAR28    VAR13
          cum.r    186    208    210    212    213
5_4      213    38.727    720          VAR3    VAR4    VAR5    VAR23    VAR19
          cum.r    186    208    210    212    213
5_5      213    38.727    718          VAR3    VAR4    VAR5    VAR28    VAR15
          cum.r    186    208    210    212    213

holding .

```

Figure 3.2 shows the first screen of the printout. The “holding .” at the bottom of the screen is there so you can look at the output before it scrolls off the screen. Press the return key to see the next screen. If you do not wish to look at the rest of the printout, enter a “q” or click on “Quit” in the top menu.

The 5 variables which reached the most people and had the highest frequency of hits were VAR3, VAR4, VAR5, VAR23 and VAR13. The reach statistics show that VAR3 reached 186 people. Adding VAR4 improved the reach by 22 for a total reach of 208. Adding VAR5 only improved the reach by 2 people, etc.

There are 11 other combinations each with the same reach but a lower frequency. A reach of 213 is the number of respondents who had positive values on all 5 of the indicated variables. The frequency is the total frequency of hits for the 5 selected variables.

In Figure 3.2, there are 213 people who selected at least one of the 5 variables in the first combination. This means that the minimum frequency for those 5 variables is 213. The maximum frequency for the same 5 variables occurs only if all 213 people select all 5 of the variables. Thus 5 times 213 gives the maximum possible frequency of 1,065. The second combination in Figure 3.2 also has a reach of 213 but the frequency of 744 is less than the frequency of 746 for the first combination.

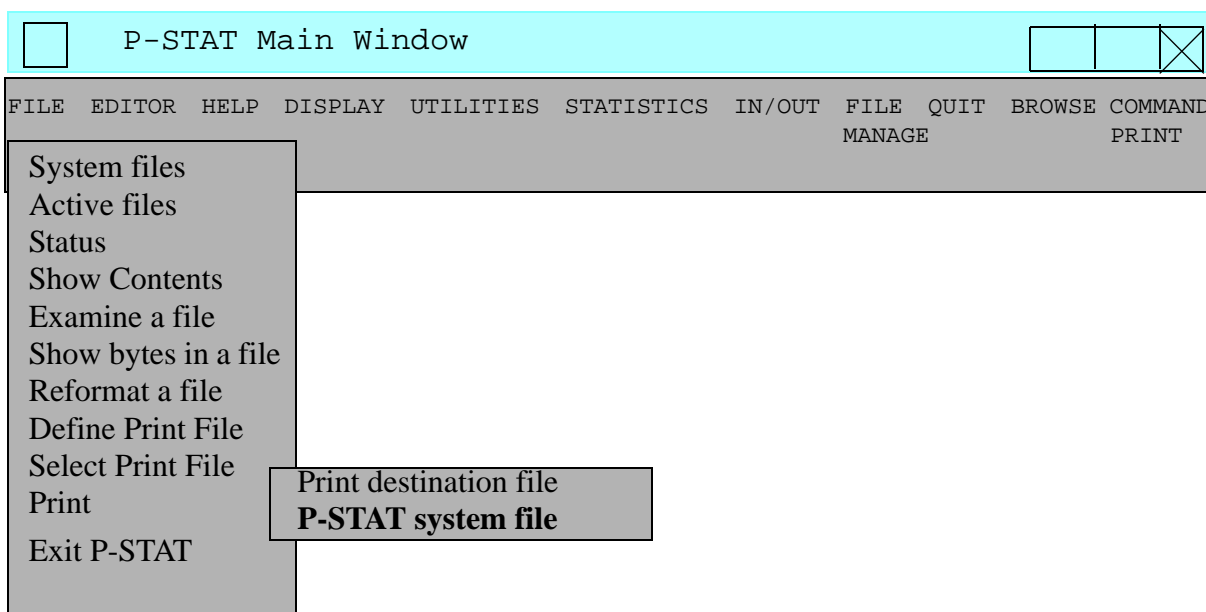
3.5 PRINTING THE RESULTS

When the listing is complete you can get a printed copy by pressing “Command Print” in the top menu. However, if you press “Command Print” before the command actually finishes printing or if you quit the command, the print window will have an error message stating “Error opening the file”. Print files are buffered. Buffering

means that the lines of output are not actually written until a print buffer is full and print buffers are often very large. If you quit the printout before the file has finished printing, “Command Print” shows only the lines that have actually been written to the screen.

The “Command Print” button is very useful for command output such as the TURF final report and for listing small output files. However, when a file has 500 or more rows, what you may really want is to look at the first screen or two; quit; and then have the full file printed.

P-STAT system files such as P0503 and TurfRRR are binary files. That means that they are written to be very efficient on the computer. This advantage has one disadvantage; because P-STAT binary files can only be read by a P-STAT command such as the LIST command, they cannot be sent directly to the printer. However there is an easy way out. The “Print P-STAT system file” item in the “Files” menu will automatically generate the necessary LIST command and send that output to the Print Manager.



When you select “Print” you can choose your printer and also specify your page setup. The LIST command that is generated is programmed to print in “portrait” rather than “landscape” format. Be sure to check that the page setup has the correct orientation.

To finish the run you can either click on the X in the top right-hand corner of the screen or click on “Exit P-STAT” at the bottom of the “File” list of items.

You can use the PRINT command to print any appropriate file anywhere in your file system. However, you cannot directly print binary files such as P-STAT system files or executable programs. To print P-STAT system files you have 3 choices:

1. List the file at the terminal and when the listing is complete click on “Command Print” in the top menu.
2. Select “File”, “Print” and “P-STAT system file”. This will generate the LIST and the PRINT commands for the selected file. This is easy but you have no control over the format of the output.
3. LIST the file and select the “PR” identifier to supply your own print destination name. This allows you to use LIST’s formatting options. (Selecting “Blank.missing” enhances the listing of the REACH and FREQ results files.) Then press “File”, “Print” and “Print destination file” and select the filename you provided for the PR identifier.

Figure 3.3 REACH.STATS : All possible statistics

The generated TURF command

```
TURF P0503 [ DROP Id ], SIZE 5, REACH.RESULTS TurfRRR,
  REACH.STATS ALL,
```

FILE TurfRRR: partial listing

TURF results ordered by REACH, from input file p050 using 29,5 PAGE 1

size	and	rank	reach	pct reached	freq	stats	item .1	item .2	item .3	item .4	item .5
5_1		213	38.727	746			VAR3	VAR4	VAR5	VAR23	VAR13
						cum.r	186	208	210	212	213
						add.r	186	22	2	2	1
						%cum.r	33.82%	37.82%	38.18%	38.55%	38.73%
						%add.r	33.82%	4.00%	0.36%	0.36%	0.18%
						cum.f	186	364	542	628	746
						add.f	186	178	178	86	118
						unique	16	8	2	2	1

Figure 3.4 REACH.STATS Controls the Rows. SHOW Controls Columns

The generated TURF command

```
TURF P0503 [ DROP Id ], SIZE 5, REACH.RESULTS TurfRRR,
  REACH.STATS cumulative.reach additional.reach
  cumulative.freq unique.reach,
  SHOW size rank reach freq $
```

FILE TurfRRR: partial listing

TURF results ordered by REACH, from input file p0503 using 29,5 PAGE 1

size	rank	reach	freq	item .1	item .2	item .3	item .4	item .5
5	1	213	746	VAR3	VAR4	VAR5	VAR23	VAR13
				186	208	210	212	213
				186	22	2	2	1
				186	364	542	628	746
				16	8	2	2	1

3.6 OTHER TURF IDENTIFIERS

The manual “P-STAT: Introductory Manual” contains complete information about the TURF command and all the identifiers. A brief description of TURF’s optional features follows here;

3.7 The Output Files

The identifiers that control the REACH output are:

1. REACH.RESULTS supplies a new P-STAT system file name for the REACH output
2. REACH.STATS There are 9 possible arguments for REACH.STATS
 ALL
 NONE
 CUMULATIVE REACH
 ADDITIONAL REACH
 UNIQUE.REACH
 CUM.REACH.PCT
 ADD.REACH.PCT
 CUMULATE.FREQ
 ADDITIONAL.FREQ
3. REACH.THRESHOLD permits the user to control what constitutes a successful reach. The default is 1. If the case has a single positive value on any of the variables in a given combination, the case is added to the reach total for that combination.
4. SHOW and OMIT control the columns in the REACH.RESULTS and FREQ.RESULTS output files. The items are always in the output files but any or all of the first 5 columns can be selected. Also available are RANK, SIZE and PCT.OF.MAX.REACH. SHOW NONE and OMIT ALL have the same result: only the items are included in the output files. See Figure 3.4 which causes the printout to have the same columns as earlier releases of the TURF command.
5. REACH.SUMMARY a P-STAT system file that summarizes each of the reach values that were found. .

The identifiers that control FREQ Output are:

1. FREQ.RESULTS supplies a new P-STAT system file name for the FREQ output
2. FREQ.STATS default value is CUMULATIVE.FREQ. Other options are ADDITIONAL.FREQ, ALL and NONE
3. FREQ.SUMMARY a P-STAT system file that summarizes the freq values .

Identifiers which can be used to provide weights.

1. CASE.WEIGHTS provides a variable in the file which is used to weight each case
2. RESPONSE WEIGHTS the data values themselves are the weights. The default is to use a value of 1 for any item that is greater than zero.
3. ITEM.WEIGHTS provides a file with information on the weighting to be given to individual variables

3.8 General Identifiers

There are a number of other identifiers which provide additional control over the way that items are included and the way that the data are represented.

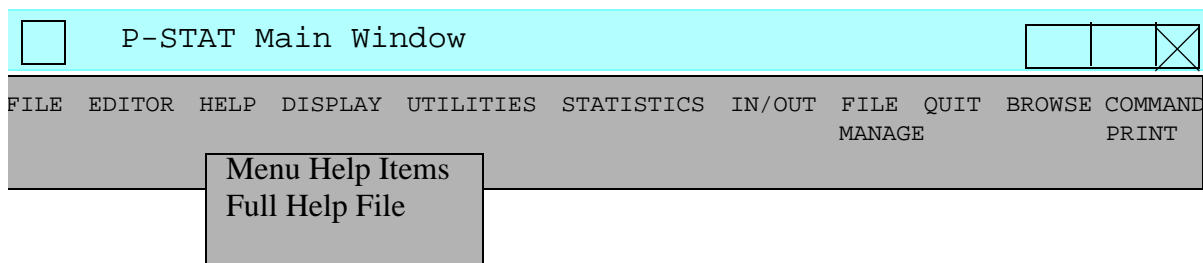
- | | |
|----------------|---|
| 1. FORCE | needs a list of variables that are to be forced into the output regardless of their contribution. |
| 2. STEP | automates the forcing. This provides a way to get an approximation when the number of combinations is too large to be practical. There are two other identifiers that apply to STEP runs: SHOW.ALL.COMBOS and FULL.REPORT. |
| 3. CONSTRAIN | provides a limitation on the number of combinations to be tried by defining clusters of variables to be treated together. |
| 4. FULL.OUTPUT | provides a name for a P-STAT system file with all the combinations. |
| 5. USE.NAMES | used with FULL.OUTPUT. The names of the variables are used instead of the variable position. |
| 6. PROGRESS | Reports the percent of combinations performed. The reports are done every 1,000,000 combinations unless the PROGRESS identifier is used to change the value. Pressing the F2 key can be used to interrupt the run without losing results for any sizes that have been completely processed. |

3.9 HELP OPTIONS

For more information on the other options in the TURF command you can see the manual “P-STAT: Introductory Manual”, or you can use the P-STAT help file.

Click on “Help/Menu Help Items” for information about any of the commands that are supported in the menu system. For TURF, click on “Statistics” and then on “Turf”. You will also get help information if you select “--HELP--” from the menu of Turf’s optional identifiers.

An alternate source for this information is the P-STAT help file which contains information about all P-STAT commands not just those supported by the menus. The final 3 items listed under “HELP” invoke the basic P-STAT help file.



“Help Topics” and “Help Commands” contain information about the topics and commands that are supported in the help file. “Help Keyword” requests a keyword from the list of topics or commands. To get the TURF information select “Help Keyword” and enter “Turf” as the keyword.

A further source if help is the single line at the bottom of the main P-STAT window. As you select a command or an identifier, this line is used to detail just what the command or the identifier does and how to use it.

3.10 OTHER DATA INPUT

If your data are not in an Excel spreadsheet, the first 2 steps illustrated here will be somewhat different. If your data are on paper, you can use P-STAT's MAKE command to enter the data. This is described in the manual "P-STAT: Introductory Manual".

If your data were entered into some other PC program, the first step is to export that data in one of the following formats:

1. Tab delimited, i.e., like the Excel export format but without the variable labels. Use TABFILE.IN to import the data into P-STAT.
2. ASCII but delimited by some other character such as a comma. You can use TABFILE.IN with the DELIMITER identifier which tells P-STAT what character that was used as the delimiter. If you do not have variable names as the first record in the data file and do not want to have P-STAT generate the names, click on "FILE" and then on "Create File Free Format Data".
3. SPSS portable format. To import use Export/Import: SPSS input.
4. See the MAKE and MAKE.FIXED commands for more complex data formats

4 The Editor

The P-STAT editor contains a log of P-STAT commands as they are entered. Each command has a substructure containing any subcommands and data records. There are editor instructions to move from one command to another. There are instructions to move from a command to its subcommands and data records. Editor files can be saved and reused and, most important, the contents can be changed and errors corrected.

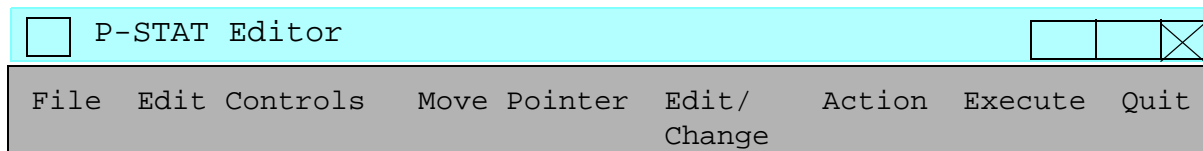
There are two ways to work with a P-STAT edit file:

1. Instructions can be entered manually. This permits the P-STAT editor to be used on computers other than PC/Windows. It also permits users who are accustomed to entering P-STAT commands to continue operating in a familiar way.
2. Commands can be selected from the Editor's top menu bar. Command text is generated depending on the options you select.

In PC/Windows all changes to a command or its data records are done in a full screen editor using standard options such as cut and paste.

The P-STAT editor control window is entered whenever an error is encountered or the "Editor" button is selected from the main menu. The edit control window manages the P-STAT edit file which contains a copy of every command that is entered during a P-STAT session. Edit files can be saved for reuse in later runs. The commands in the edit file can be located either by their position in the file or by matching text strings in the command. The subcommands or data records that belong to a given command are only available when the edit file pointer is located at that command.

Figure 4.1 **An Empty Editor File**



```
P-STAT, version 3.01, rev 3 (Dec 1, 2012)
WHOPPER 2 (6,000 variable) size with storage options 222.
Copyright (c) 1972 to 2012, P-STAT Inc.
Use HELP NEWS$ for general news about this version.
```

```
P-STAT starting... 15:46:02 Aug 1, 2012
```

```
Enter a command:
```

```
editor $
```

```
The editor file is empty.
```

```
At the TOP
EDITOR:
```

The first section in this chapter covers the menu items available in the editor control window. The second section illustrates how to save the commands in the edit file for use in later runs. The third section illustrates using a saved edit file, changing a command and executing the changes. The final section illustrates command input in the editor.

4.1 THE EDITOR'S MAIN MENU

To make changes to a given command or its associated subcommand/data records, you must first locate the command. If you enter the editor because of an error situation, the edit file is positioned at the command which contains the error. If you enter the editor independently, the edit file is pointed to the final command and you may need to move the pointer to the desired location.

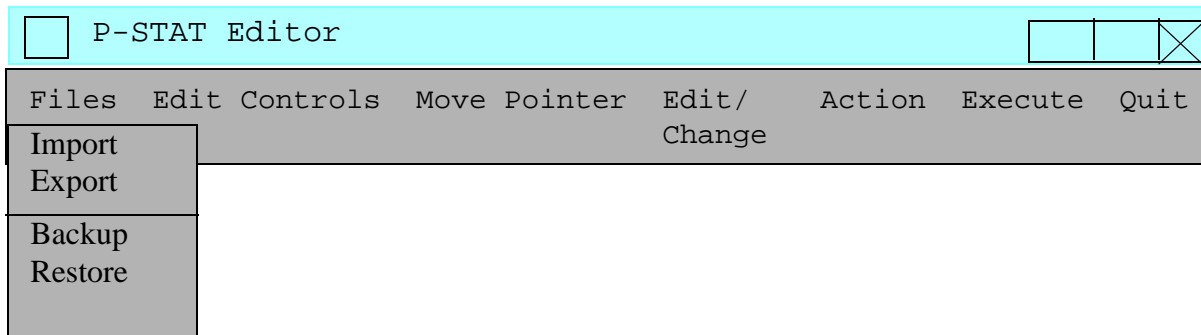
Figure 4.1 shows the P-STAT window after the "Editor" button is selected at the beginning of the run. No P-STAT commands have been entered and the editor file is empty. The editor control window has its own set of menu buttons. The "Edit/Change", "Execute", and "Quit" buttons are the only ones that do not have a pull-down submenu. If you press the "Quit" button or enter the letter "Q", control returns to the main P-STAT window.

The rest of this chapter contains:

1. a brief summary of each of the editor file control buttons;
2. saving an edit file containing the commands that were entered in the previous chapter;
3. locating a command and making changes;
4. executing the changed commands.

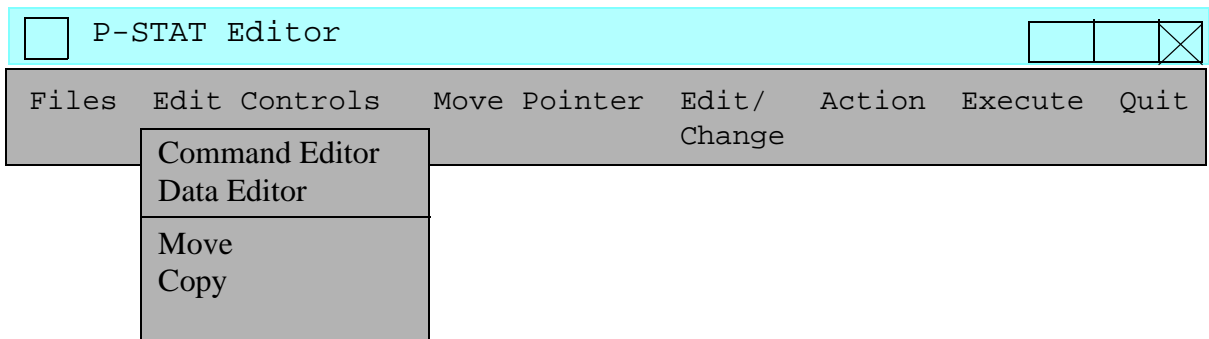
4.2 The Editor "File" button

The "Files" button is used to save the current edit file or restore an existing edit file. The P-STAT edit file can be saved and restored in either a text format or a binary format. Import and Export are used for text format. Backup and Restore for binary format. For most purposes text format is satisfactory and has the advantage that it can be easily read by any text editor (Notepad, Wordpad, MicroSoft Word, etc.).



4.3 The "Edit Controls" button

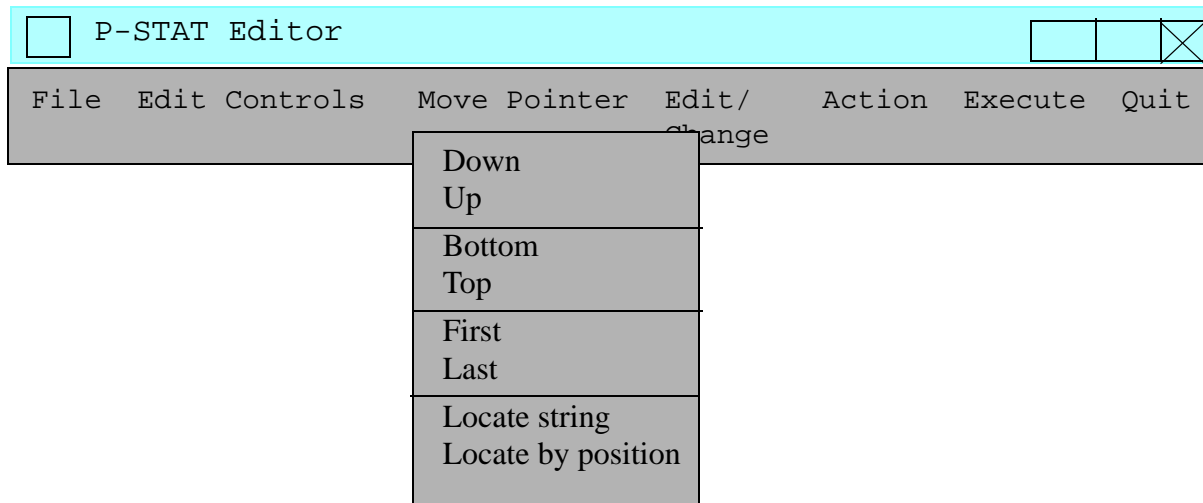
The "Edit Controls" button is used to move between the command editor and the data editor. It is also used to move or copy one or more P-STAT commands to another location in the edit file.



4.4 Moving the Pointer in the Editor

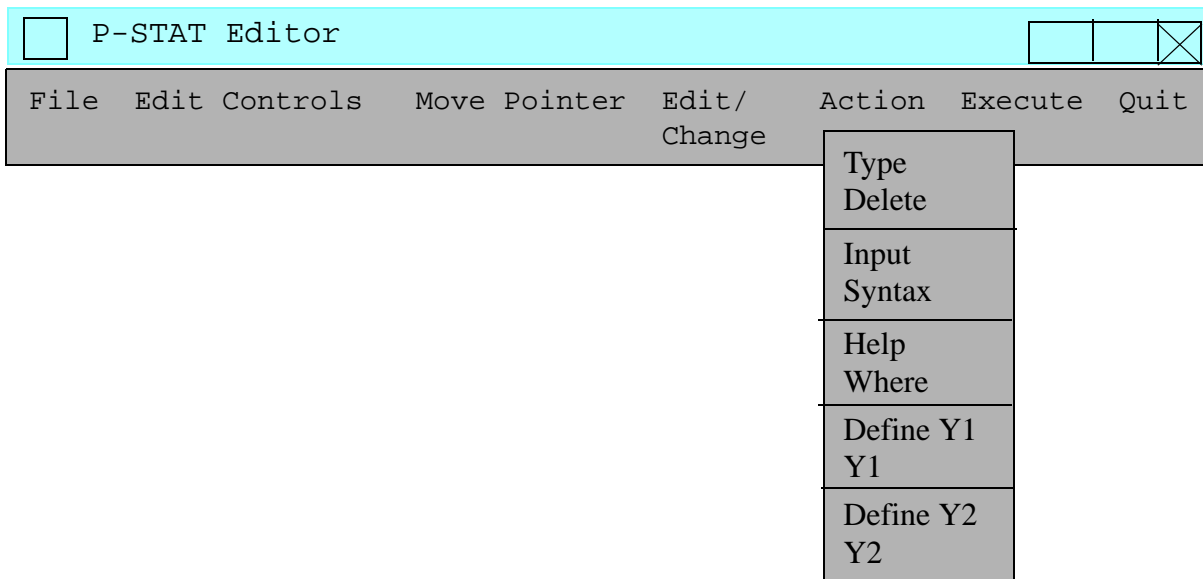
The “Move Pointer” button is used to find your way through the edit file. Only the currently active command can be edited or changed. The pointer must be located at the command to be changed. The ways to move the pointer include:

1. Move up or down one or more command positions.
2. Move to the top (before the first command) or bottom (after the last command).
3. Move to the first command or to the last command.
4. Move the pointer to a specified position or to the command containing a specified string.



When you select “Down” or “Up” you are given the choice of moving 1 position or more than one position. If you select “More than one”, you may either accept the default value of “2” or enter the appropriate number.

4.5 Editor Actions



The “Action” button is the final Editor button with choices. These are the things that you can do aside from changing the contents of the command. The first four are the most useful:

1. Type one or more commands and (optionally) their data records.
2. Delete one or more commands.
3. Input a command.
4. Check the syntax of a command.

The final three buttons on the Editor control window are:

1. Edit/Change which causes the current command to be placed in a full screen edit window.
2. Execute which causes one or more commands to be submitted for execution.
3. Quit which exits from the edit file and returns to the main P-STAT window.

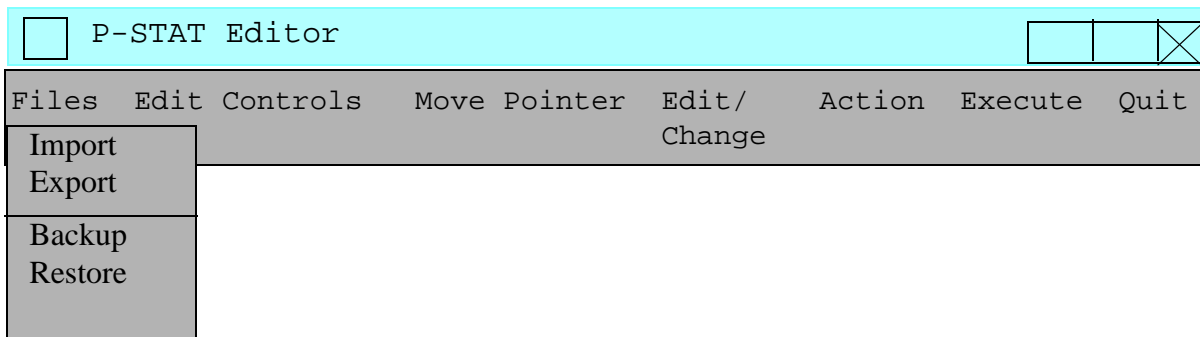
4.6 AN EXAMPLE WITH A COMMAND FILE

In the previous chapter 5 commands were executed to bring data into P-STAT, run a TURF analysis and print the results. When PSFILES, is executed before any files are opened it does not appear in the edit file. These are the four remaining commands stored in the edit file:

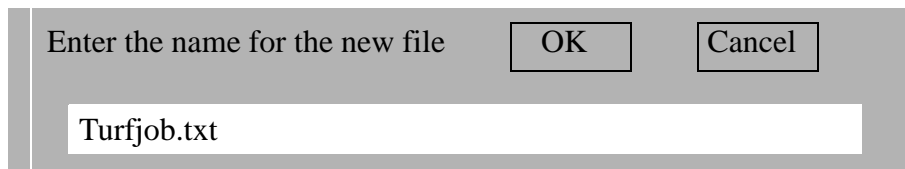
1. TABFILE.IN p0503, READ 'turfdat.txt' \$
2. TURF p0503, SIZE 5, REACH TurfRRR \$
3. LIST turfrrr \$
4. PRINT turfrrr \$

To save this file as an ASCII text file:

1. click on Editor in the main P-STAT window;
2. click on “Files” in the P-STAT Editor menu;
3. click on “Export.



Export needs a name for the file to be stored on disk in the PSFILES folder. A name such as Turfjob.txt is appropriate.



The file is stored in c:\project\turf_0503 because the PSFILES command provides a default folder for all the files created in the current run. Since this is an ASCII file you can edit it in Notepad or even in MicroSoft Word (as long as you save it as text). The edits can include changes or additional commands. The example in the next section illustrates using this saved file of P-STAT commands in a subsequent run.

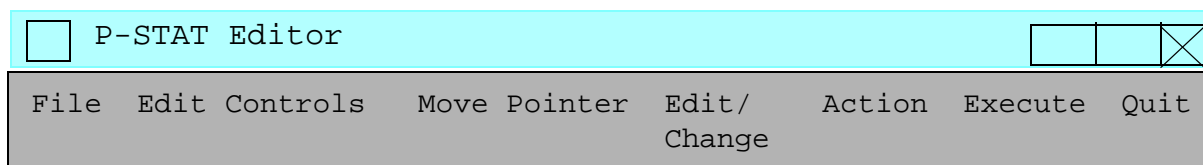
4.7 Using the Saved Edit File

This time after selecting “Utilities” and “Change All Save Files (PSFILES)” go directly into the editor. It is empty because the PSFILES command was executed before the editor began. After selecting “Editor” click on “Import”. “Import” needs to know the name of the file of commands.

The windows file manager is used to locate existing files. The file manager begins its search in the currently defined “PSFILES” directory. You may, if necessary, move to any other folder in your file system. When you find the file that contains the commands click on OPEN.

Figure 4.2 illustrates the appearance of the screen when the open is complete. Control is still within the editor and the pointer is located below the final command. Since this is the beginning of the run, the editor, which was empty before the file was imported, now contains the 4 commands that comprise this job.

Figure 4.2 Import an Existing Editor File



P-STAT, version 3,01, rev 1 (Jan 12, 2012)

WHOPPER 2 (6,000 variable) size with storage options 222.

Copyright (c) 1972 to 2012, P-STAT Inc.

Use HELP NEWS\$ for general news about this version.

P-STAT starting... 15:46:02 Jan 20, 2012

Enter a command:

PSFILES "C:\PROJECT\TURF_0503" \$

Enter a command:

Editor \$

The editor file is empty

At the TOP

EDITOR:

4 commands successfully imported from file C:\project\turf_0503\Turfjob.txt

At the BOTTOM

EDITOR:

The goal of this exercise is to locate the TURF command, change the SIZE identifier and then execute the last 3 commands so that you can see the results with a changed SIZE parameter.

The first command in the editor is the TABFILE.IN command. Since P-STAT system files are automatically saved you do not need to do this command again. You know that the TURF command is the second command and the current pointer is beyond the last command. The instructions which move the pointer are located under the “Move Pointer” item.

The steps to locate the 2nd command are:.

1. Click on “Move Pointer”
2. Click on “Locate by position”
3. Enter the number 2

The screenshot shows a dialog box titled "Enter LOCATE position". On the left, there is a small rectangular input field containing the number "2". To the right of this field is a vertical scroll bar with up and down arrows. Below the input field are two buttons: "OK" on the left and "Cancel" on the right.

Locate by position works as you would expect no matter where the pointer is located initially. In “Locate by string” the movement of the pointer is usually from the current position to the bottom of the edit file. However, if you are already at the bottom it cycles back and starts at the top. This means that an alternate way to locate the TURF command is to use “Locate by string” and provide a set of characters that is unique to the TURF command such as “REACH”.

In these examples, case is irrelevant. In P-STAT the only time that case matters is when you create a variable or a file. The initial entry is remembered for listing or printing reports but comparisons are case independent. In these examples upper case is used for keywords and mixed case for the names that you provide. For Windows case is also irrelevant as PC/WINDOWS treats all file names as if they were upper case.

There are times when you do not know where the command is located and are unsure of a unique string. When this occurs, the first step should be a listing of the edit file. To see the entire contents of the edit file:

1. Click on “Action”
2. Click on “Type”
3. Click on “All”
4. Click on “OK”

The screenshot shows a dialog box titled "TYPE". The title bar is highlighted in cyan. Below the title bar are three radio button options: "One", "More than one", and "All". The "More than one" option is selected. To the right of the "More than one" option is a small input field containing the number "2" with a vertical scroll bar. Below these options is an unchecked checkbox labeled "Include all data records". At the bottom of the dialog box are two buttons: "OK" and "Cancel".

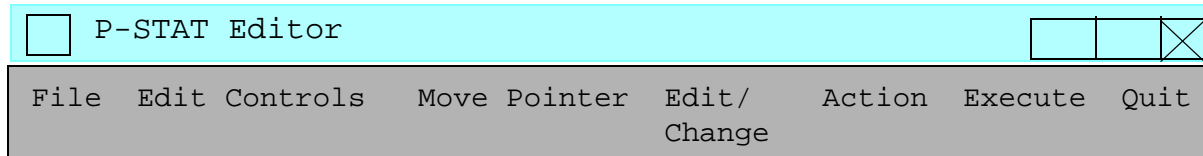
When you click on “Action” and then on “Type” you will see that it has several options. You can type the current command, some specified number of commands or all commands from the current command to the end of

the edit file. In addition you can request just the command text or you can request both the commands and any subcommands or data records.

The TYPE controls are a combination of radio boxes (only one can be selected), a check box which is either checked or not checked, and an integer counter. After making the selections, press “OK” and the screen displays the requested commands.

Figure 4.3 shows the screen as it looks when the TURF command is located.

Figure 4.3 An Editor File



Enter a command:

Editor \$

The editor file is empty

At the TOP

EDITOR:

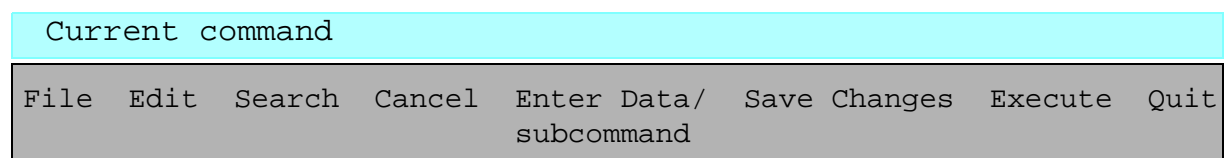
2 commands successfully imported from file C:\project\turf_0503\Turfjob.txt

At the BOTTOM

EDITOR:

```
2...TURF p0503[ DROP Id;
      ],
      SIZE 3,
      REACH TurfRRR $
```

To edit the command, press the “Edit/Change button. The selected command is transferred to an edit window which has standard edit capabilities with “File”, “Edit” and “Search”. In addition there are buttons that are very specific to P-STAT. The “Quit” button leaves the original command unchanged, exits from the editor and returns to the main P-STAT window. The “Cancel” button also leaves the text unchanged but returns to the P-STAT editor control window. “Save Changes” saves the current contents and returns back to the editor control window. “Execute” saves the current contents and then submits the changed command (and optionally the following commands) for execution.



The “Enter Data/Subcommands” button transfers control to the data/subcommand editor. Since none of the commands in the TURF project uses data or subcommands, the data editor is not useful here. However, for a commands such as SURVEY the data editor is usually needed. If the data editor does not exist, it is created. The

controls in the data editor behave exactly the same as the controls in the command editor. You can move back and forth between the editors by clicking the mouse inside the editor's boundaries.

This is the selected command as it looks in the Edit/Change edit window after you change the SIZE argument from 3 to 6 and have selected the "Execute" item. If you select "all commands", the updated TURF command and all commands following it in the edit file will be executed.

Current command							
File	Edit	Search	Cancel	Enter Data/ subcommand	Save Changes	Execute	Quit

TURF p0503[DROP Id;
l,
SIZE 6,
REACH TurfRRR
\$

one command
more than one command
all commands

A selection of "more than one command" produces another dialog box where you can enter the number of commands to be executed.

Since the command editor uses the system's clipboard to store information, you can cut and copy information from one command, exit the command editor with Cancel or Save Changes, select another command to edit and paste in the material that is stored on the clipboard.

To exit from the command editor there are 4 choices:

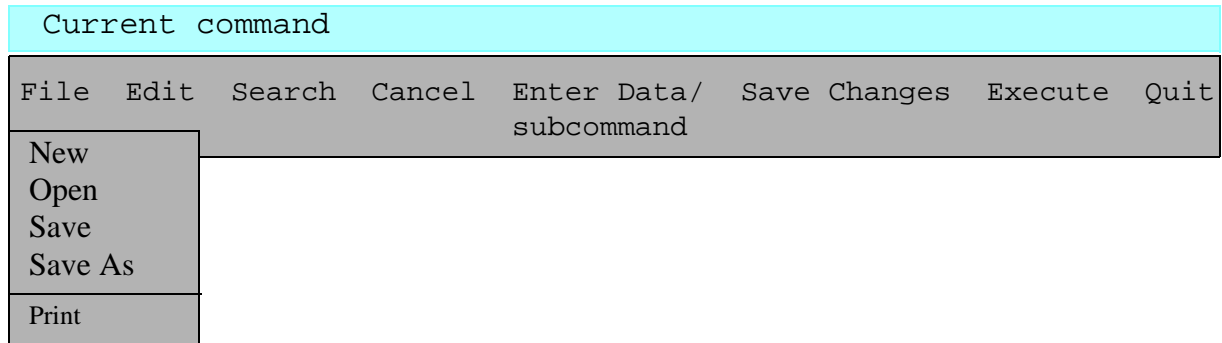
1. "Cancel" exits the command editor and returns to the P-STAT editor control window. The command is unchanged.
2. "Quit" exits both the command editor and the P-STAT editor and returns to the main P-STAT window. The original command is unchanged.
3. "Save Changes" exits the command editor and returns to the P-STAT editor control window ready to do more editor activity.
4. "Execute" submits the changed command (and optionally any following commands) for execution. If there is an execution problem, control returns again the editor control window.

4.8 MORE COMMAND EDITOR FEATURES

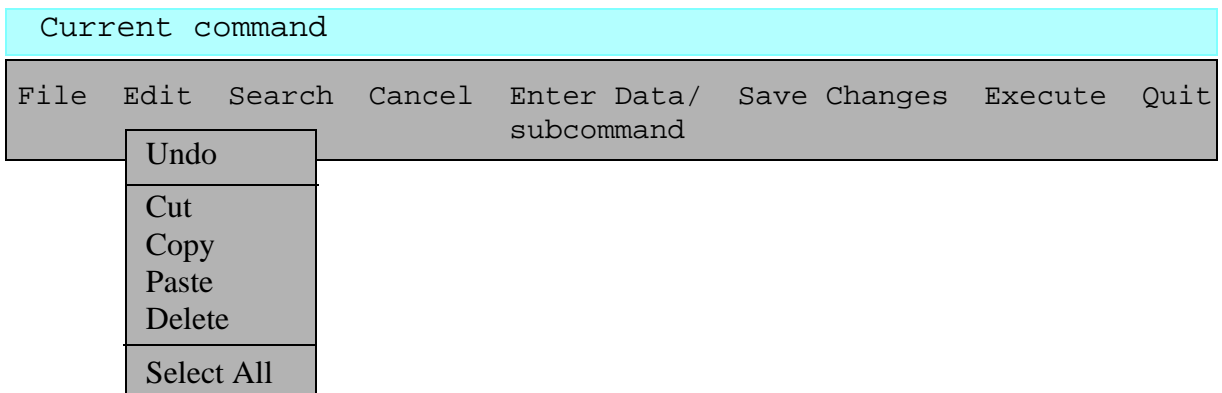
While the basic purpose of the command editor is to handle individual P-STAT commands with any associated data or subcommand records, this editor can be used in a more general way. The "Enter Data/subcommands", "Save Changes" and "Execute" features only make sense when used on single commands. When you enter the "Edit/Change" editor from the P-STAT main editor only the current command is available. However, if you select "Input" from the "Action" menu of the P-STAT editor control window, you begin with an empty edit file.

If you enter a single P-STAT command in the command editor and exit with either “Save Changes” or “Execute”, the new command is placed in the editor following the command which had the pointer when the command editor was entered. You can also use an empty command editor to edit any text file. Unless the contents of the file are a single P-STAT command, you cannot execute the changes but you can save them.

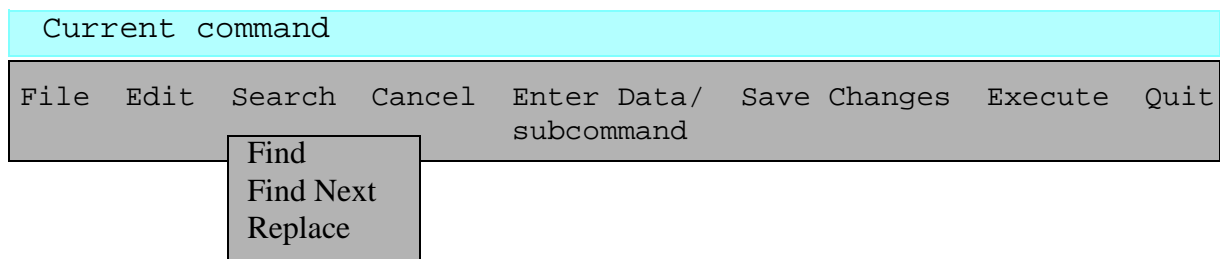
This mechanism can be used to solve the problem of multiple changes. For example using the 4 command file, Turfjob.txt, you can use the editor to change all references to turfrrr into turfxxx in a single step, remove the first command, save the changes and then execute a TRANSFER command to execute the entire new job.



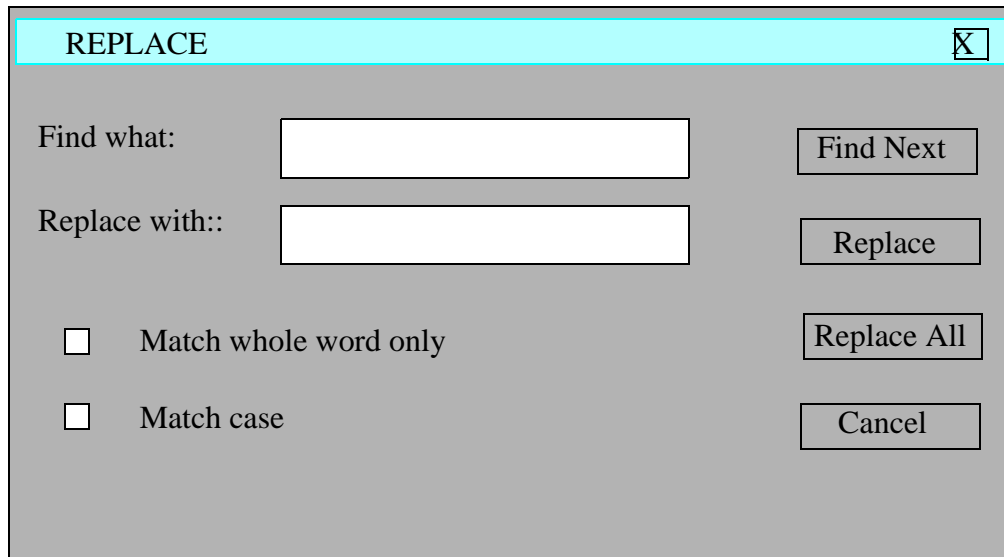
1. Select: File, Open and then select Turfjob.txt. The entire contents of the file of commands is brought into the editor.
2. Highlight the TABFILE.IN command, click on “Edit” and then on “Cut”. (Pressing either the delete key or the backspace key also works.)



3. Select Search and Replace.



4. Enter the find field as “turfrrr:”.
5. Enter the replace field as “turfxxx”.



7. Click on File and Save (or Save As).
8. Quit the "Change" editor and return to the P-STAT main editor

Do NOT use either the "Save Changes" or "Execute" key when you have more than one command in the edit file. The file containing your P-STAT changes is now ready to execute. This can be done either by:

1. Import the file of changes into the editor, move to the first command in the file and execute all the commands.
2. In the main menu press "Utilities" and "Transfer to Command File". Select the file you have just saved.

5 A Larger Project

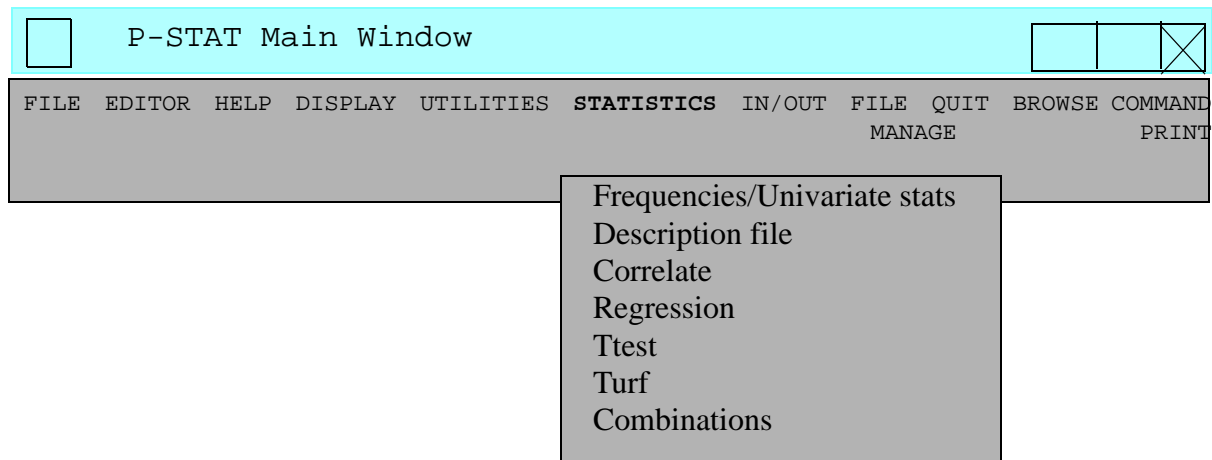
Chapter 3 described a simple project: import data from an Excel spreadsheet and do a TURF analysis. The examples showed how to delete a variable and print the results. This chapter uses the same project with additional features. The input data includes a gender variable: males are coded as 1 and Females as 2. Since females are more likely to fill out a survey form, they are over represented in this study. The first example shows:

1. how to use the gender variable to create a weight variable and
2. how to use the weight variable in the TURF analysis.

If you do not know the best parameters for your TURF analysis or if you have many similar projects, it may become tiresome to do the analysis one step at a time, repeating most of the steps while changing only one or two arguments. The second section discusses ways to process the data more efficiently.

5.1 A WEIGHTED ANALYSIS

When you are weighting on the basis of demographic information such as gender, you must know both the real population percentages and the counts in your sample. The real numbers can be obtained from census data. The counts in your data can be determined by counting. If the sample is large, an easy way is to let P-STAT do the counting for you.



The COUNTS command is accessed by selecting “STATISTICS” and then “Frequencies/Univariate stats”. COUNTS provides a wide variety of univariate statistics for every variable in the input file. Like most P-STAT commands, COUNTS works with a P-STAT system file which you select from the list of the available P-STAT system files.

After selecting your file, press the “Modification/PPL (P-STAT Programming Language) button and select the variables of interest, in this case “Gender”. After selecting Gender press “Finished all PPL”. By keeping the single variable Gender, that will be the only variable that the COUNTS command receives. The next dialog box shows the various identifiers that are available in the COUNTS command.

The default COUNTS output is the values, the totals, counts of missing and 16 univariate statistics. This is more than you need. Selecting VALUES from the list of COUNT identifiers produces enough information to determine if the distribution for variable Gender is appropriate. After you select “Values”, Press “Select/Omit” to move it to the selected column and then press “Finished”.

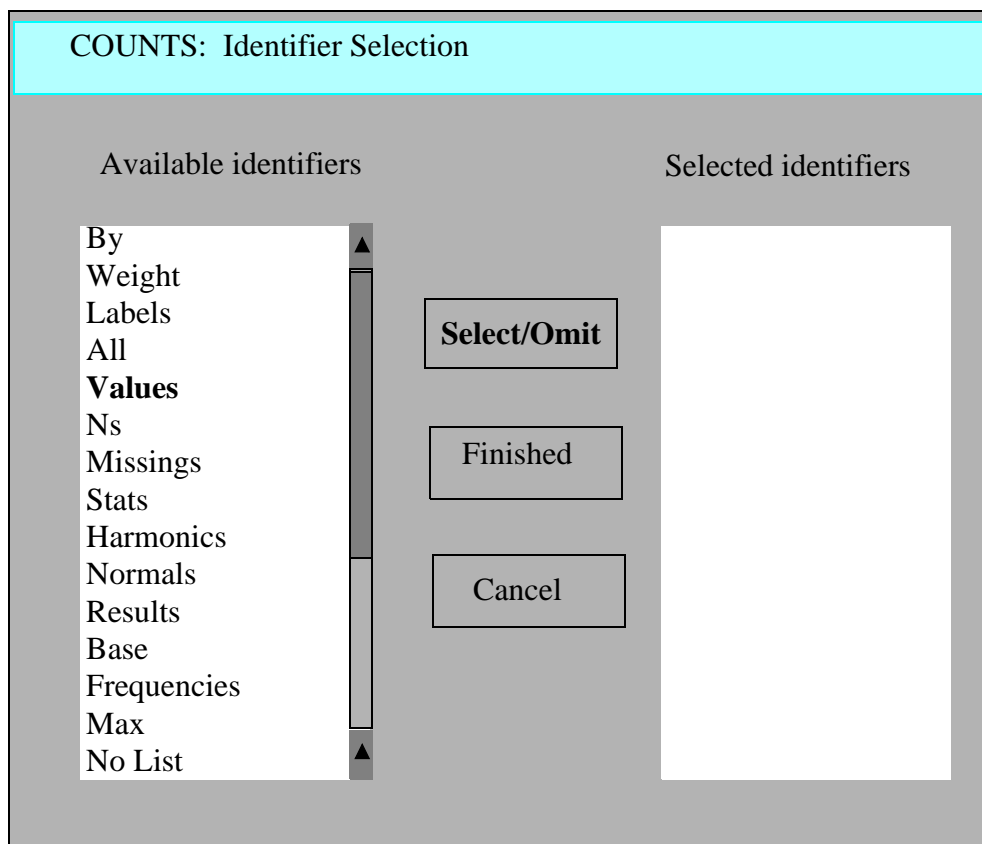
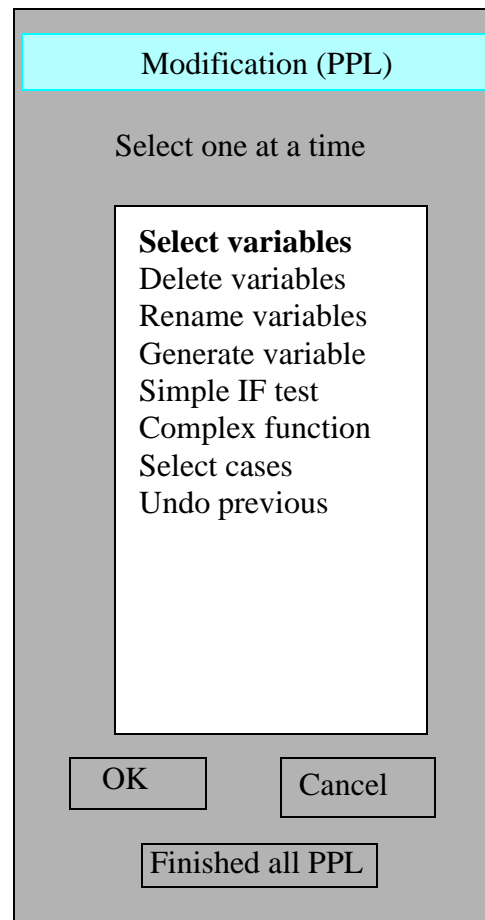
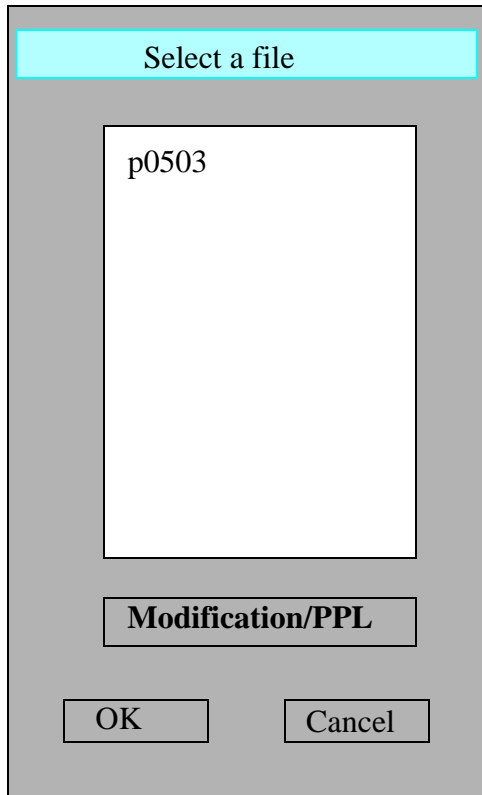


Figure 5.1 **The COUNTS Command**

The Command

```
COUNTS [ KEEP Gender; ],
RESET, VALUES $
```

The Printout

```
-----Counts of File P0503 completed-----
550 cases and 1 variable were processed
```

variable	value	count	or stat	cum count	pct	cum pct
Gender	1	241		241	43.82	43.82
	2	309		550	56.18	100.00

From Figure 5.1 it is obvious that the percent of males and females does not represent the actual distribution in most populations. The next step is to create a weight variable that will give the under-sampled males more representation in the TURF analysis. Giving each female a weight of .905 and each male a weight of 1.12 produces weighted values for men of 49.12 and for females of 50.88. These weights can be computed by using more features available with “Modification/PPL”.

5.2 Modifications: Temporary and Permanent

There are two forms of modifications in P-STAT:

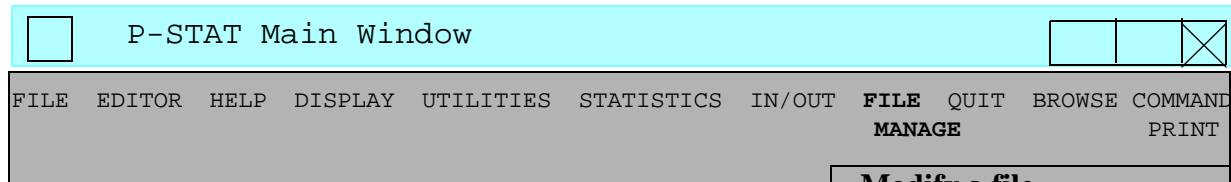
1. Temporary modifications exist for the duration of a single command and do not change the contents of the original input file. Modifications that are made when there is no output file are always temporary modifications. Temporary modifications may be done anytime that a P-STAT system file is input to any command.
2. Permanent modifications can only occur when a command produces a P-STAT output file. The MODIFY command takes a P-STAT system file as input and produces a new P-STAT system file as output. The output file contains any modifications, new variables or recoded variables. The input file is unchanged. Since it is likely that the weight variable will be useful in more than just a single TURF command, it is appropriate to generate the weight variable as a permanent variable in a new output file.

5.3 More About P-STAT System Files

P-STAT system files are sometimes referred to as “autosave” files. This is because any output file given a legal name that does not begin with “WORK” and has no more than 16 characters is automatically saved and is given the extension of “PS1”. If you modify this file you have the option of using the original file name for the output file or providing a completely new file name. If you reuse the original file name, it is given an extension of “PS2”. You now have 2 versions of the file available. If you make yet another change and continue to use the same file name, the newer version replaces the older of the two previous uses. Thus the extension of a file name

alternates between PS1 and PS2. P-STAT knows which is the more recent version because the file contains an internal counter. The input to any command, unless specified otherwise, is always the more recent version.

5.4 Generating a Weight Variable



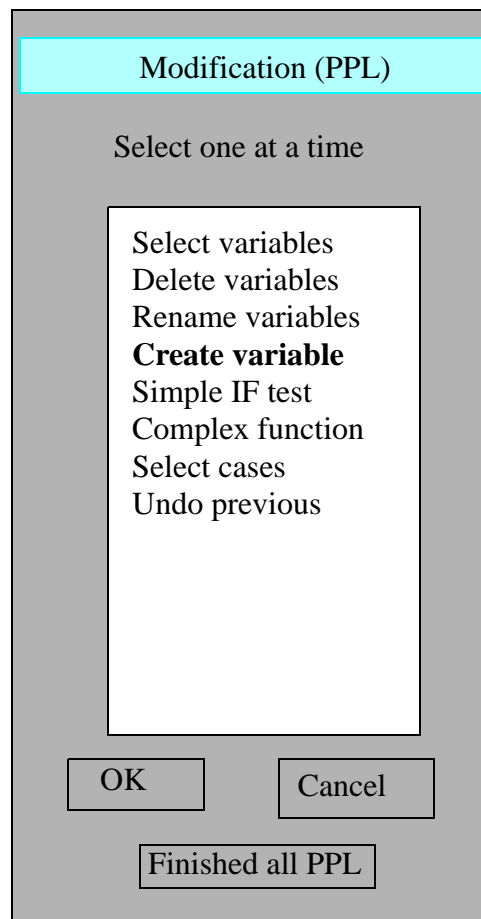
The MODIFY command is located under the main menu “File Manage” item. The first dialog window that appears is the “Select a file” window. Select your file and then click on “Modification/PPL”.

When you create a new variable you must give it a name that conforms to the rules for legal names in P-STAT. These are:

1. 1-64 characters
2. Start with a letter
3. Contain only letters, numbers, decimal points, underscore characters
4. An optional tag followed by 2 colons at the beginning.

If every case in the file has a good value for variable Gender, you can assign an initial value to the new weight variable and then with a single IF statement assign the weight for the other Gender category.

Modify a file
 Concatenate files
 Lookup - table files
 Sort
 Aggregate (summarize)
 Duplicate detections



Enter the new variable name

Weight

Select the data type

Numeric

Character

Weight is to be a numeric variable initialized to the value .905.

Set the variable to ?

Increase the value

Decrease the value

Value

Missing 1

Missing 2

Missing 3

Another variable

A function

Enter the value to be used

After the new variable is created and given an initial value, control returns to the “Modification (PPL)” dialog box. Select “Simple IF test”. Creating an IF test requires several steps:

1. Select the variable to be tested from the menu of variable names.
2. Select the 3 radio buttons in the “Simple PPL IF statement” dialog box. The initial settings are “Equal”, “Value” and “Delete the case”. For this test change “Delete the case” to “Recode or set”.

Simple PPL IF statement

If variable Gender

Select 1 of these 6 operations	Compare the variable to	If compare is true do the following
<input checked="" type="radio"/> Equal	<input checked="" type="radio"/> A value	<input type="radio"/> Keep the case
<input type="radio"/> Not Equal	<input type="radio"/> A variable	<input type="radio"/> Delete the case
<input type="radio"/> Less than		<input checked="" type="radio"/> Recode or set
<input type="radio"/> Less or equal		
<input type="radio"/> Greater		
<input type="radio"/> Greater or equal		

3. Selecting “A value” produces a numeric dialog box. Since the initial value of .905 is the weight for the females which are coded as a 2 for variable Gender, you need to test Gender for the value of 1.

Enter the comparison value

▲▼

4. When the result of the compare is either to keep the case or to delete the case, pressing the OK button completes the PPL statement. However, when the result is to recode a variable or to set a variable to a new value, there are more steps. The first additional step is to select the variable which is to have the new value. In this case you are testing variable Gender and recoding variable Weight when the IF statement is true.

Select One

Variable to Set

Id
Gender
var2
var3
var4
var5
var6
var7
var8
var9
var10
var11
var12
var13
var 14
var15

OK Cancel

Select One

Variable to Set

var16
var17
var 18
var19
var20
var21
var22
var23
var24
var25
var26
var27
var28
var29
var30
Weight

OK Cancel

When a variable is generated, it is placed at the end of the file. Since there are too many variables in the file to fit in a single window, use the scroll bar on the right to move to the end. Select weight and click on “OK”. The dialog window “Set the variable to ?” appears again. Enter the desired weight and press OK.

1.12

Enter the value to be used

OK Cancel

When the value is set, control returns again to the “Modification (PPL)” dialog box. Click “Finished PPL”. The next prompt is for an output file name for the modified file. You can use the same name for the output file as you used for the input. The final step is to click on “Finished” in the “Identifier Selection menu”.

Enter P-STAT system file name

OK Cancel

P0503

Figure 5.2 **Generate Weight: the Command**

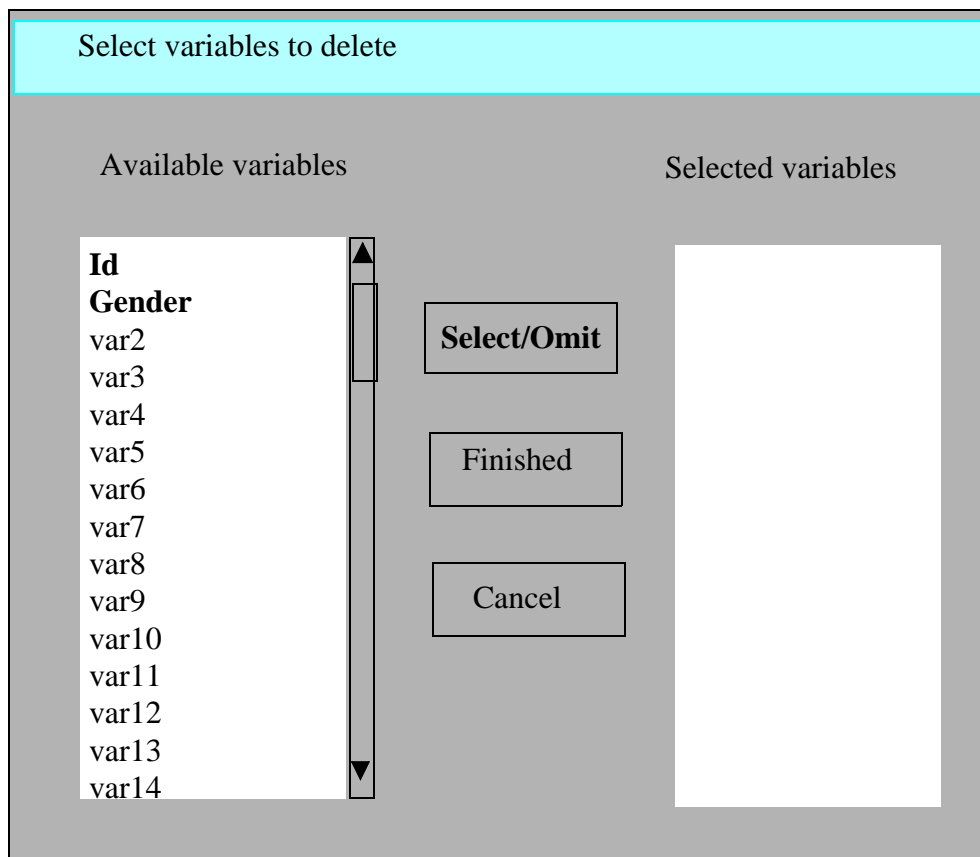
```

MODIFY P0503 [ GEN Weight = .905;
IF Gender EQ 1 SET Weight = 1.12;
]
Out P0503
$

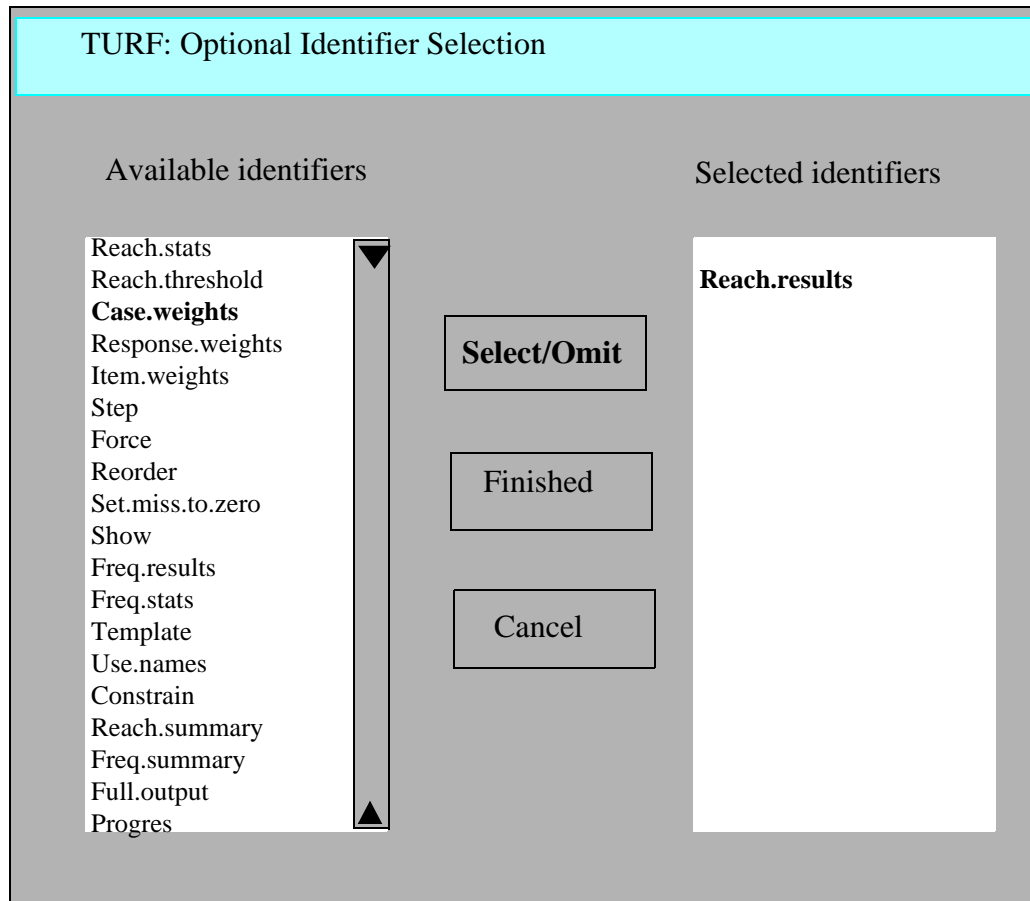
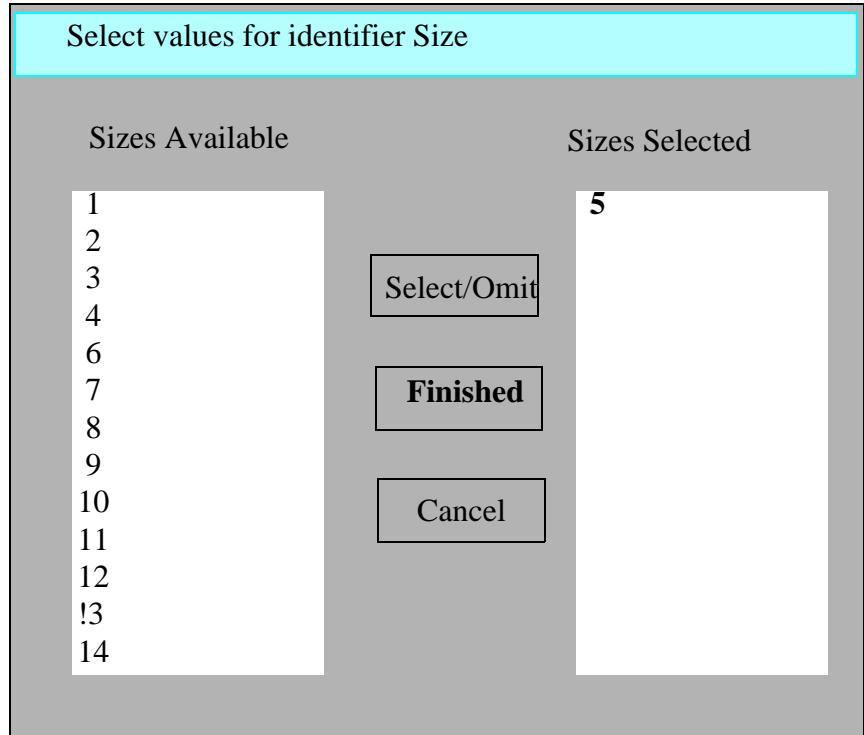
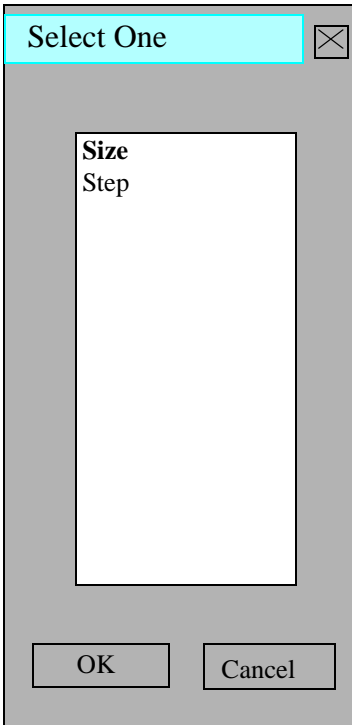
```

Figure 5.2 contains the command that is created by this series of selections and mouse clicks. The next step is to use P0503 with its weight value to produce a weighted TURF analysis. The beginning steps are exactly the same as the procedure in the previous chapter.

1. Click on “Statistics” and “TURF”
2. Select p0503 as the input file and click on “Modification/PPL”
3. Delete variables Id and Gender.



4. Click on Select/Omit to move the variables to the selected column and click on “Finished”.
5. Next select SIZE as the type of TURF run. SIZE looks at all combinations of the variables at the selected size. STEP produces a stepwise analysis where results from a given step are forced into the analysis for the following step. Select one or more sizes using the same procedure used to select the variables.



6. In the “Identifier Selection” menu click on “Reach.results” and supply a name for the file which is to contain the best 100 reach combinations for each size
7. Click on Case.Weights
8. Select variable Weight from the list of variables.
9. Select any other options that you wish to try and finally click on “Finished”.

Figure 5.3 shows the TURF report when the CASE.WEIGHT option is used.

Figure 5.3 Weighted TURF Report

```
-----TURF analysis for file P0503 completed-----
OPTIONS: case.weights (in variable weight)

      29 items were used in the analysis.
      550 cases were read and used; the sum of
          the weights for these cases was 549.565.

SIZE   5 evaluated 118,755 combinations:
      215.5500 was the best REACH, found in 12 combinations.
      754.035 was the best FREQ in those 12 combination.
      877.515 was the best FREQ in any size 5 combination.

The FREQ score for a combination is the count of
non-zero responses in the case for that combination,
times the caseweight, summed over the reach cases.

REACH.RESULTS file TurfRRR has 100
combinations with the highest reach scores.
The items are ordered by their REACH contribution.
Cumulative reach is shown

Time: .2 seconds.
```

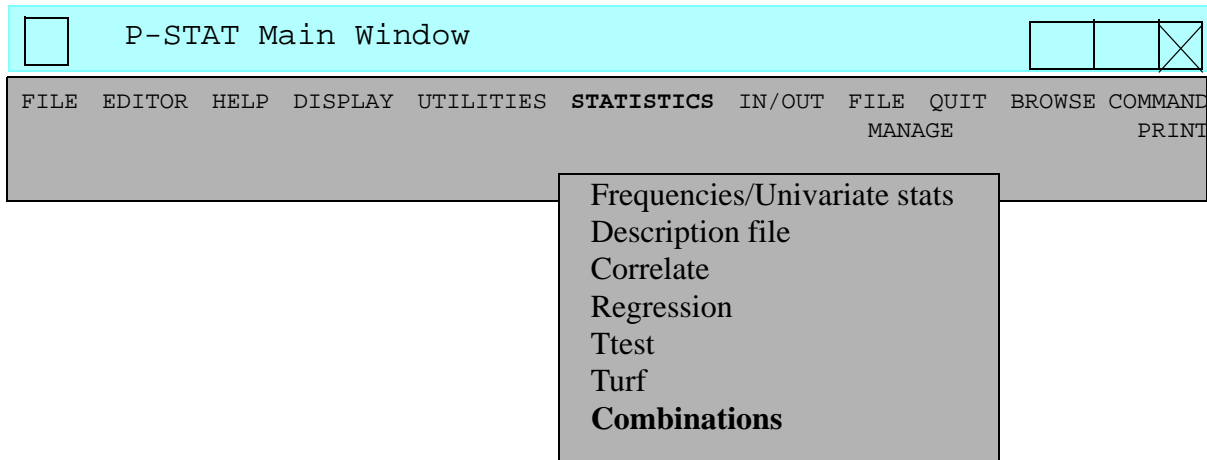
5.5 COMBINATIONS

When you have a file with a large number of items, it is helpful to have an idea of how many combinations are required for various sizes.

Consider 40 items. Depending on number of cases and on options:

1. Size 4 takes 91,390 iterations. Seconds.
2. Size 6 takes 3.8 million iterations. Minutes.
3. Size 10 takes 847 million iterations. An hour.
4. Size 15 takes 40 billion iterations. A day.
5. Size 20 takes 137 billion iterations. A week.

PUT with the COMBINATION function can be used to display the number of iterations that are required for a given number of items and sizes. This can be run by selecting "Combinations" just below TURF in the statistics pull down menu.



Selecting Combinations causes 3 requests for integers. The first is the total number of items to be used. The second is the Size to be considered. Thus the numbers 20 and 4 would provide the number of combinations as 20 things taken 4 at a time. If the final number is the same as the first, a single report is made. If the final number is larger, the number of combinations is computed for all sizes from the first to the last..

Enter the total number of items to be included

40

OK Cancel

Enter the starting combination SIZE

4

OK Cancel

Enter the final SIZE to be considered

8

OK Cancel

Size	Combinations
4	91,390
5	658,008
6	3,838,380
7	18,643,560
8	76,904,685

5.6 The STEP Identifier

The STEP identifier can be used when you have a situation such as 40 items taken 15 at a time. It produces an approximation of the desired result by doing the calculations in steps. For example:

```
STEP 8 12 15
```

The first 8 are processed in the normal fashion. In the second step, the selected 8 variables are forced into the calculations and sizes 9-12 are then calculated. A third step forces the initial 8 variables plus the 4 variables calculated in the second step and finishes with steps 13-15. This will take far less time and will produce a result which may not be the absolute best but provides a reasonable solution for a problem that might otherwise be beyond the capabilities of your computer. STEP and cascading STEPS are illustrated in the manual "P-STAT: Introductory Manual"

5.7 TURF.SCORES

The TURF.SCORES command is used after a TURF run to identify the cases which contributed to the reach results. There are two ways to use the TURF.SCORES command. One way provides the item information that was used by TURF plus background variables. The easy way creates a TEMPLATE file in the TURF command and then uses the original input file including any demographic variables as input to TURF.SCORES. The TURF.SCORES command is documented in the manual "P-STAT: Basic Statistics". TURF.SCORES can only evaluate a REACH.RESULTS file for a single step size.

5.8 EFFICIENT PROCESSING

The step by step click and select procedures described in the chapter 3 and in the first section of this chapter make it easy for the new user or the occasional user to process his data. However, if you want to try TURF with the SIZE parameter set to values 3, 4, 5 and 6 both with and without weighting, the step by step procedure becomes tedious. TURF has the ability to provide multiple values for the SIZE identifier in a single run. You may use any of the following:

```
TURF . . . . ., SIZE 3 5 TO 7,
TURF . . . . ., SIZE 8 7 4 TO 6
TURF . . . . ., SIZE 1 4 3 2
```

There will be a single REACH.RESULTS or FREQ.RESULTS output file with results from all of the sizes. The number of variables will depend upon the number needed for the largest SIZE selection. The default output rows for each size is 100. Thus if there are 4 sizes requested, the reach output file will have 400 rows unless an override is provided.

If the largest size is 8 and the smallest is 4, the output rows for size 4 will have missing supplied for the 4 columns that are not relevant. Some of the output options such as FULL.OUTPUT are not available for multiple size runs.

Using PPL, the P-STAT Programming Language, in the menus requires setting values in several dialog boxes. If you are going to do a series of TURF commands changing only the SIZE parameter or the output files, use MODIFY to create a file which has exactly the variables that you want. This way you need to do the PPL steps only once. Rerunning TURF from the menus will require fewer mouse clicks and selections if you only need to select or change TURF identifiers.

An easy way to make simple changes to the command rather than re-entering it from the beginning is to use the P-STAT editor. The P-STAT editor was described in the previous chapter. It provides one way to simplify the procedure of repeated analysis with minor changes. When you finish looking at the results of an analysis, select the "Editor" item from the P-STAT main menu. This places you in the editor positioned at the most recent command. (Note: the number at the left is the sequence number of the most recent command and is NOT part of the command text.) Select "Edit/Change". Make your changes in the edit window and then select "Execute".

If you handle many large data sets, you may find it easier to use the saved edit file as a template and use an editor such as Notepad to construct a job stream with multiple steps. Figure 5.4 contains such a job stream. The data file has a variable designating the region of the country. The task is to do a separate analysis for each region.

There are 4 ways to use a file of commands.

1. Start a P-STAT run and enter the editor. Import the file of commands and selectively execute 1 or more of those commands. See the chapter on the Editor for examples
2. Start a P-STAT run, click on “Utilities” and on “Change All Folders (PSFILES)”, click on “Utilities” and on “Transfer to Command File”. Select the name you provided for the file of commands. Each command will be executed in turn. If there are errors, execution stops and the P-STAT editor is entered.
3. Run P-STAT in batch mode. Batch mode is automatic when the P-STAT is given both an input file and an output file in the command line.

```
PSTAT C:\xxx\turfcommands.txt c:\xxx\turfcommands.out
```

When running in batch mode a small window echoes the command name as each command is processed. The “xxx” in the previous illustration is the folder which contains the command file. If there are errors in the run, the error text will be found in the output file provided as the second argument in the command line. Because the first command is PSFILES, P-STAT knows

- 4.
5. where to find the input and where to write the output.
6. Design a P-STAT macro to contain the commands. Use the P-STAT RUN command to execute the macro.

Figure 5.4 **A File of Commands**

```
PSFILES 'C:\project\turf-0304' $
PR P0304.txt $

TURF P0304 [ IF Region = 1, RETAIN;
            DROP Id Region ],
            SIZE 3 to 6, REACH.RESULTS P0304R1 30
$
LIST P0304R1 $

TURF P0304 [ IF Region = 2, RETAIN;
            DROP Id Region ],
            SIZE 3 to 6, REACH.RESULTS P0304R2 30
$
LIST P0304R2 $

TURF P0304 [ IF Region = 3, RETAIN;
            DROP Id Region ],
            SIZE 3 to 6, REACH.RESULTS P0304R3 30
$
LIST P0304R3 $

PRINT P0304.txt $
```

A macro must be brought into P-STAT and executed before it can be used. This can be done by using the TRANSFER command which is found under “Utilities” in the main menu. Type in the RUN command with the argument in parentheses, remember to enter the “\$” and press the return key.

If you are going to run the macro several times you can package the macro with several run commands in a single file and use this file in any of the 3 ways that command files can be used. One way is to take a command file such as the one in Figure 5.4 and sandwich it between the 2 commands

```
MACRO macro.name $      and
ENDMACRO $
```

The TRANSFER command and the RUN command can be selected from the “Utilities” submenus or entered directly from the terminal

```
TRANSFER "mymacro.txt" $
```

which brings the macro, an external file with the name “mymacro.txt”, into P-STAT and activates it. This is then followed by:

```
RUN macro.name $
```

The macro is the equivalent of a super P-STAT command and the name must follow the rules for legal names in P-STAT. Thus “macro.name”, while meaningless, is a legal macro name.

Using an editor to create and edit copies of P-STAT commands can be both tedious and a source of error if you have many subgroups to process. Figure 5.5 illustrates the same problem using the SUBFILES command, a special feature available only within a macro, to loop through an indefinite number of subgroups.

Figure 5.5 Macro with Subfiles

THE MACRO in file turfrun.mac

```
MACRO turfrun $

PSFILES 'C:\project\turf-0304' $
PR 'Turf.by.region.txt' $

SUBFILES P0304, BY Region $
TITLES 'TURF Analysis for Region #REGION' $
TURF SUBFILE [ DROP Id Region ],
    SIZE 3 to 5,
    REACH.RESULTS Work1 $
LIST Work1, TITLES $
END SUBFILES $

PRINT 'Turf.by.region.txt' $

ENDMACRO $
```

USING THE MACRO

```
TRANSFER "turfrun.mac"

RUN turfrun $
```

To use this macro copy it in a text editor. Start each command on a new record and if your editor supports word wrap, turn it off. Be sure to copy the macro changing only the file, variable names, etc. to the names you are using in your project. Be especially careful with the punctuation, and save it as text file with a name such as "turfrun.mac". To activate the macro, enter P-STAT and click on "Utilities" and then on "Transfer to command file". Supply the name of the file containing the macro. If there are errors, you will be placed into the P-STAT editor where you can select "Edit/Change" to make corrections and then execute the corrected macro.

If there are no errors, click on "Utilities" and then on "Run a macro". The final entry in the "Utilities" menu can be used to both activate and run the macro. When you are sure that your macro is working correctly, this does the job in a single step.

5.9 The Macro: Step By Step

The Macro in Figure 5.5 has 10 P-STAT commands. The rest of this section explains each of these commands in detail.

```
MACRO turfrun $
```

The MACRO command must have a name for the macro which conforms to the rules for legal file names in P-STAT: 1-16 characters starting with a letter and containing only letters numbers and decimal points. This name may but need not be reflected in the name of the external file where it is stored. However, if the external file contains only one macro, it is easier to remember if the name of the external file corresponds to the macro name.

```
PSFILES 'C:\project\turf-0304' $
```

The PSFILES command provides the full path to the folder which contains the data.

```
PR 'Turf.by.region.txt' $
```

The PR command provides a name for an output file which is where all subsequent printout is to be written. When the run is complete this output file will contain all the information generated by all the subsequent commands.;

```
SUBFILES P0304, BY Region $
```

The SUBFILES command is a special feature of macros. SUBFILES provides a BY capability for all the commands within its provenance. SUBFILES is similar to MACROS in that its domain begins and ends with a P-STAT command. For SUBFILES, the ending command is ENDSUBFILES \$

The input to the SUBFILES command is a P-STAT system file. This will usually be the only time that file is referenced in the SUBFILES loop. The file name "SUBFILE" is used to refer to the current subgroup that is being processed regardless of the original input file name.

The SUBFILES command also requires the name of one or more variables in that file which control the looping. The file need not be sorted on the BY variable. The default is to loop through the file using the values of the BY variable in ascending sort order.

```
TITLES 'TURF Analysis for Region #REGION' $
```

Each time through the subfiles loop a scratch variable is created for each of the BY variables which contains the current value. In Figure 5.5, the current value if variable Region is stored in the scratch variable #Region. If this is used in a TITLES command it is available to identify the subgroup when it is listed.

```
TURF SUBFILE [ DROP Id Region ],
  SIZE 3 to 5,
  REACH.RESULTS Work1 $
```

The input to the TURF command is not our system file, P0304, it is the current SUBFILE. The rest of the TURF command looks exactly like all the examples we have seen so far except that the input is named "SUBFILE". Because of the PR command the TURF report does not appear on the screen but is written to the designated disk file.

```
LIST Work1, TITLES $
```

The REACH.RESULTS file is listed. Because this output has information about 3 sizes and the number of cases to be saved was not specified, the default of 100 per size is used to produce a 300 case output. The addition of the identifier TITLES to the LIST command produces a printout where the region that is analyzed is fully identified.

```
END SUBFILES $
PRINT 'Turf.by.region.txt' $
ENDMACRO $
```

“END SUBFILES” causes the looping to be evaluated. If all of the values of variable Region have not been analyzed the loop variable is increased and the next region analyzed. Only when all the values for variable Region have been used does the control move to the PRINT command. Finally the ENDMACRO command signals the end of the macro.

This macro can be used repeatedly with a RUN command. This can be entered from the keyboard or from the menus. If you want to run it on another file, for example P0404, you need only edit the macro and change the file name. You can also edit the macro and change the number of sizes analyzed for the print file name.

This is not the end of macro usefulness as macros can have arguments which would make it possible to supply information at RUN time:

```
RUN turfrun ( p0604, 4 to 7 ) $
```

Figure 5.6 illustrates how this is done.

Figure 5.6 A Macro With Arguments

THE MACRO in file turfrun.mac

```
MACRO turfrun ( file, size )$

PSFILES 'C:\project\turf-0304' $
PR 'Turf.by.region.txt' $

SUBFILES &file, BY Region $
TITLES 'TURF Analysis for Region #REGION' $
TURF SUBFILE [ DROP Id Region ],
    SIZE &size,
    REACH.RESULTS Work1 $
LIST Work1, TITLES $
END SUBFILES $

PRINT 'Turf.by.region.txt' $

ENDMACRO $
```

USING THE MACRO

```
TRANSFER "turfrun.mac"

RUN turfrun ( p0604, 3 To 5 7 )$
```


- SystemVariables
- .DATE. 2.6
- A
- Active files 2.2
- AGGREGATE 2.4
- ASCII 3.13
- B
- BALANCE 1.3
- Batch computing
 - defined 1.3
 - job processing 5.13
- Blank
 - (space) delimited files 2.10
- Bytes
 - examine the bytes in an external file 2.2
 - show the bytes in an external file. 2.2
- C
- CASE.WEIGHTS
 - in TURF command 5.10
- Cases
 - defined 2.5
- COMBINATIONS
 - PC menus 2.3
- Combinations
 - calculating 5.10
- Comma delimited
 - files 2.10
- Command
 - print 3.7, 3.8
 - summary for new users 2.15
- Command files 5.13
 - executing 2.3
- Command.Print
 - file 2.2
- CONCAT 2.4
- CONSTRAIN
 - in TURF command 3.12
- CORRELATE 2.3
- COUNTS 1.3, 2.3, 5.1
- Create
 - a new variable 5.4
- Crosstabulation 1.2, 2.2
- D
- Description files 2.3
- Directory
 - for all files
- PSFILES 2.3
 - for external data 2.3
 - for P-STAT system files 2.3
 - selection 3.1
- Display
 - PC Windows
 - menu 2.2
- DUPLICATES 2.4
- E
- END 2.12
- ENDSUBFILES **5.15**
- Environment variables
 - PSAUTO 1.8
 - PSDATA 1.8
 - PSFILES 1.7
 - PSHELP 1.7
 - PSKEY 1.7
 - PSTART 1.8
 - PSTEMP 1.8
- ERASE 2.3
- ERASE.CURRENT 2.3
- ERASE.EXTERNAL 2.3
- ERASE.PREVIOUS 2.3
- EXAMINE 1.2
 - the bytes in a file 2.2
- EXCEL 1.5, 3.1
- Exit from
 - P-STAT 2.2, 2.12
- Export data
 - ASCII text
 - UNMAKE 2.4
 - SPSS export format 2.4
- F
- File names 2.15
- Files
 - active 2.12
 - external
 - erasing 2.3
 - how does P-STAT find them 2.12
- FILETYPE 1.9
- FREQ.RESULTS
 - in TURF command 5.12
- FREQUENCIES
 - in COUNTS command 2.3
- Frequency
 - in TURF command 3.8

H

HELP 3.12

Help

PC Windows
menu 2.2

Help file

using 2.13

I

Installation 1.7

Introduction to using P-STAT 2.1

J

Join files

add cases with CONCAT 2.4
add variables with LOOKUP 2.4

L

Legal names 5.4

defined 2.6
file 2.15
rules for 2.15
variables 2.15

LIST 2.2, 2.11, 2.15, 2.16, 3.7

LOOKUP 2.4

Lower case

use of 2.7

M

MACRO

for TURF analysis 5.15
open and run 2.3

Macros 1.4

MAKE 1.5, 3.13

MAKE.FIXED 3.13

Manuals

list of 1.6

MENUS

PC Windows
File submenu 2.1
using 2.15

MIGRATE

from P-STAT version 2 1.9

Missing data

defined 2.6

Mixed case

use of 2.7

MODIFY 2.3

changing a P-STAT system file 2.4

N

Names, legal

rules for 2.15

NEWS 2.14

P

PATHNAMES 2.3, 2.13, 2.15

PERCENTILES 1.3

Permanent modifications 5.3

PPL

see P-STAT Programming Language

variable selection

PC Windows 3.3

PPL Functions

numeric

COMBINATIONS 5.10

PRINT

to a disk file 2.11

Print

changing destination 2.2
set attributes of print destination 2.2

Print Manager 1.5

Printing

from PC/Windows 3.9

PSAUTO 1.8, 1.10, 2.3, 2.13, 2.16

PSDATA 1.8, 1.10, 2.3, 2.16

PSFILES 1.7, 1.10, 2.3, 2.13, 2.16, 3.1, 5.15

PSHELP 1.7, 1.10

PSKEY 1.7, 1.10

PSTART 1.8, 1.10

summary 1.10

P-STAT commands 1.5

P-STAT editor 4.1

control window 4.2
delete 4.4
edit or change 4.4
execute 4.4
full screen example 4.8
input 4.4
moving the pointer 4.3
quit 4.4
saving 4.2
syntax 4.4
type 4.4

P-STAT system file 1.5, 5.3

check the status 2.2

comparing 2.2

- erasing 2.3
- format 2.4, 2.5
- legal names 2.6
- limits 2.5
- listing 2.11
- locating 2.3
- missing data 2.6
- names of available files 2.1
- show contents 2.2
- PSTEMP 1.8, 1.10
- R
- REACH.RESULTS
 - in TURF command 3.6, 5.10
- REFORMAT 1.2, 2.2
- REGRESSION 2.3
- Report writing 1.2
- RUN
 - a macro with arguments 5.16
 - Executing a macro 2.3
- S
- SAMPLE 1.3
- SAVE.LABELS 2.2
- Saving a P-STAT system file 2.12
- Scratch variables 2.6
- SHOWBYTES 1.2, 2.2
- Simple PPL IF statement 5.6
- SIZE
 - in TURF command 5.12
- SORT 2.4
- SPSS 3.13
 - files, reading and writing 2.4
- SPSS.IN 2.3
- SPSS.OUT 2.4
- Statistical analysis 1.3
- STATUS
 - of system attributes 2.2
- SUBFILES 1.4
 - used in TURF macro 5.15
- Summary statistics
 - using AGGREGATE 2.4
- SURVEY 1.2, 2.2
- System files
 - defined 2.4
- System variables
 - .DATE. 2.6
- T
- Tab delimited file 3.1
 - as input 2.3
 - as output 2.4
 - importing 3.2
- TABFILE.IN 2.3, 2.16
 - import tab delimited file 3.2
- TABFILE.OUT 2.4
- Tables
 - creating 2.2
- TAGS 1.6
- Temporary modifications 5.3
- TEXTFILE.IN 1.5, 2.3, 2.16
- TEXTFILE.OUT 2.3
- Titles 2.2
- TRANSFER 2.3
 - command files 5.13
- TTEST 2.3
- TURF 1.3, 3.1, 3.3, 5.1
 - identifiers
 - CASE.WEIGHTS 3.11, 5.10
 - CONSTRAIN 3.12
 - FORCE 3.12
 - FREQ.RESULTS 3.11, 5.12
 - FREQ.STATS 3.11
 - FREQ.SUMMARY 3.11
 - FULL.OUTPUT 3.12
 - ITEM.WEIGHTS 3.11
 - OMIT 3.11
 - PROGRESS 3.12
 - REACH.RESULTS 3.6, 3.11, 5.10
 - REACH.STATS 3.11
 - REACH.SUMMARY 3.11
 - REACH.THRESHOLD 3.11
 - RESPONSE WEIGHTS 3.11
 - SHOW 3.11
 - SIZE 5.12
 - STEP 3.12, 5.12
 - USE.NAMES 3.12
 - Total Unduplicated Reach and Frequency
 - 2.3
- TURF.SCORES 1.3, 5.12
- U
- Univariate statistics 5.1
- Upper case
 - use of 2.7

V

Value labels

- saving 2.2

Variable

- names 1.6, 2.15

Variable names 2.15

Variables

- defined 2.5

- naming 2.6

- system 2.6

- temporary 2.6

W

Weighted analysis 5.1

Wildcard 1.6

- ? 2.6