P-STAT ®

Basic Statistics

P-STAT: Basic Statistics

Second edition: January 2013
This publication corresponds to **P-STAT Version 3, January 2013**. This publication documents the P-STAT commands which produce basic statistics.

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

CONTENTS

Descriptive Statistics

TTEST: Comparing Two Groups

CORRELATE: Continuous and Discrete Variables

REGRESSION: Linear Relationships

NP.TEST, NP.COR: Nonparametric Statistics

EDA: Exploratory Data Analysis

TURF.SCORES

FIGURES

1
Descriptive Statistics

Descriptive statistics summarize the values of a variable in a sample, as opposed to *sampling* statistics which permit the drawing of inferences about that variable in a population. As well as summarizing a data sample, descriptive statistics permit an immediate check of the surface validity of the input data. Low and high variable scores that diverge markedly from expected values, huge standard deviations, unreasonable mean values, very small numbers of good (non-missing) scores, and other anomalies may suggest further checking of the input data.

1.1 OVERVIEW

A *description* file (a "DES file") of summary statistics may be requested as part of the CONCAT, CORRELATE or MODIFY commands, by using the DES identifier. Statistics in the description file include the mean, standard deviation, low score, high score, and counts of missing and good values. These statistics summarize the variables in a P-STAT system file of data, and are themselves in a P-STAT file. Thus, the description file can serve as the input file for other P-STAT commands. Several description files may be combined using the OVERALL.DES command, which yields a summary description file without the necessity of inputting the original files and recalculating the statistics.

Frequency distributions (displays) are arrangements of data showing the frequency of occurrences of values falling within arbitrarily defined intervals. Frequency displays show the range and shape of data for a variable. The COUNTS command is so fundamental that it is covered in "P-STAT Introductory Manual". It produces frequency counts and summary statistics. The FREQ command is an older command which also produces frequencies and summary statistics. It is included here because the output is in a different format that some users prefer.

The MODIFY command is a vehicle for the P-STAT Programming Language (PPL) as it is used to create new variables and recode existing variables. The usual output file is a file with the surviving variables and cases as they are changed by the PPL. A second output file from MODIY, the description file provides an excellent check on the surface validity of the recoding process.

The PERCENTILES command produces an augmented description file, giving the values of the variables for specified percentiles (quantiles), measures of skewness and kurtosis and, optionally, moments about the mean. The STANDARDIZE command produces standard scores ("z scores") that present a variable in terms of its mean and standard deviation. STANDARDIZE also produces an optional output file in which the mean replaces missing data values.

There are also many P-STAT commands that produce one or more descriptive statistics as part of their normal report or as special options. The COUNTS command produces frequencies of all unique values of character and numeric variables, as well as extensive statistics that summarize the distribution of values. The SURVEY command calculate means, standard deviations, low scores and high scores.

1.2 DES: The Standard Description File

The description file FDes, produced in this command:

```
MODIFY  Tests
  [ GENERATE  Selection.Index  =  2 * Verbal.PSAT  +  Math.PSAT;
    IF  Selection.Index  >=  200,  RETAIN ] ,
 OUT  Finalist,  DES  FDes  $
```

summarizes the file Finalist, which is the result of a modification and selection of the original file Tests. Figure 1.1 shows a description file, also produced by the MODIFY command, which summarizes the entire input file. No actual modifications are done — the MODIFY command is used merely to produce the description file. (Note that the mean, standard deviation, and low and high values are not calculated for character data.)

Figure 1.1 Producing a Description File

----------The Commands------------------------------

```
MODIFY  Tests,  DES  TestsDes  $    p
LIST  TestsDes  $
```

---------The Output File----------------------------

name	type	mean	s.d.	low	high	m1	m2	m3	good	q3
1 Individual	C16	-	-	-	-	0	0	0	15	
2 SAT.Score	DI	599.2857	67.07794	490	710	1	0	0	14	
3 Grade.Average	DI	81.1538	5.66931	70	90	2	0	0	13	
4 Percentile.Rank	DI	81.8571	8.15105	70	95	1	0	0	14	
5 Verbal.Psat	DI	55.3045	5.95430	45	68	1	0	0	14	
6 Math.Psat	DI	62.2891	7.20320	30	77	1	0	0	14	

Producing a description file is “free” with CONCAT, MODIFY or PERCENTILES — that is, it does not require a second pass of the input file as it does with CORRELATE. The statistics in the description file may reflect weighted values in the input file. The WEIGHT identifier is used in the MODIFY command:

```
MODIFY  CopPipes,  WEIGHT  Wt.Factor,   DES   PipesDes   $
```

to indicate the weight variable. Instead of each case “counting as one,” it counts as the value of the weight variable. All statistics in the description file are affected by weighting, except the count of good cases for the weight variable itself. (This “true” count is preserved for use in tests of significance, which are partially a function of the number of cases.)

The description file contains information about each of the variables in the input file. The contents of the description file are the variable name, data type of the variable, mean, standard deviation, low score, high score, counts of the three types of missing data, and number of good (non-missing) cases. Additional statistics may be calculated from these basic descriptive statistics. For example, the coefficient of variation is the standard deviation divided by the mean:

```
LIST  TestDes  [ GEN CV = S.D. / MEAN ]  $
```

Character data type is indicated by the letter C followed by the defined length of the variable. DI indicates double precision integer numeric variables, and SI indicates single precision integer numeric variables. D indicates double precision real numeric variables, and S indicates single precision real numeric variables.

The MODIFY command can be used to obtain descriptive statistics summarizing a subgroup of the input file:

```
MODIFY  InFile [ IF Sex = 1,  RETAIN ],  DES  MaleDes  $
```

This description file summarizes the variables for males.

1.3 Combining Description Files with OVERALL.DES

Description files from existing subsets or related files may be combined, producing a description file for the total group, without joining the data files and recomputing the statistics. The output is a description file summarizing the combined files. This is useful, for example, when a large survey has been built as several separate P-STAT files and a description file is created for each of the files. It is faster to create the description file for the total group using OVERALL.DES than it is to compute it from the concatenated files.

A series of description files, each of which must have the same number of cases (one for each variable) with identical values (variable names) of the variable NAME, may be input to the OVERALL.DES command.

Figure 1.2 OVERALL.DES to Combine Description Files

```
---------The Input Description Files-----------------------

File  MaleDes:

  NAME           TYPE   MEAN   S.D.   LOW  HIGH    M1    M2    M3  GOOD

1 Sex            DI     1.00   0.       1     1     0     0     0   222
2 Occupation     DI     2.32   0.65     1     3     9     0     0   213
3 Work.Status    DI     1.80   0.94     1     3     0     0     0   222
4 Children       DI     1.92   1.96     0     8     0     0     2   220
5 Hrs.Last.Week  DI    42.27  14.18     2    89    79     0     0   143

File  FemDes:

  NAME           TYPE   MEAN   S.D.   LOW  HIGH    M1    M2    M3  GOOD

1 Sex            DI     2.00   0.       2     2     0     0     0   273
2 Occupation     DI     2.23   0.69     1     3    38     0     0   235
3 Work.Status    DI     2.41   0.87     1     3     0     0     0   273
4 Children       DI     2.49   1.95     0     8     0     0     1   272
5 Hrs.Last.Week  DI    35.25  12.07    10    89   182     0     0    91

----------The COmmands and Output File--------------------

    OVERALL.DES  MaleDes  FemDes,  OUT  TotalDes  $
    LIST   TotalDes  $

  NAME           TYPE   MEAN   S.D.   LOW  HIGH    M1    M2    M3  GOOD

1 Sex            DI     1.55   0.50     1     2     0     0     0   495
2 Occupation     DI     2.27   0.67     1     3    47     0     0   448
3 Work.Status    DI     2.14   0.95     1     3     0     0     0   495
4 Children       DI     2.24   1.97     0     8     0     0     3   492
5 Hrs.Last.Week  DI    39.54  13.80     2    89   261     0     0   234
```

The cases must describe variables of the same type — either numeric or character type. Statistics on the character variable Sex may not be combined with statistics on the numeric variable Sex. Thus, if Sex is of character type in one description file, it must be of character type in the other description file. However, if the character length is C6 in one description file and C16 in another, the longer C16 length will be in the output description file. With regard to numeric variables, statistics describing single or double precision values, or integer and non-integer values, may be combined.

Figure 1.2 illustrates using OVERALL.DES to get statistics for the entire group from the description files of two subsets for males and females. The command OVERALL.DES is followed by the names of the input description files. The output file, whose name is supplied after the OUT identifier, summarizes these description files.

If a summary description file of several P-STAT system files that do not have description files is desired, the actual data files may be input to the MODIFY command:

```
MODIFY  F1  F2  F3  F4,  DES  DesF1234  $
```

Figure 1.3 Simple Frequency Distribution

```
------------ The Command ---------------------------------
FREQ  CarFile  ( KEEP  Acceleration ),  DES  CarDes  $


------------- The Output ---------------------------------
        FILE         CarFile

 VARIABLE    1,        Acceleration

                                  ALL

      LOW      HIGH       N  PCT CUM

     8.00      9.50       7    2   2
    10.00     11.00      15    4   5
    11.10     12.50      35    9  14
    12.60     14.00      66   16  30
    14.10     15.50      97   24  54
    15.60     17.30      91   22  77
    17.40     19.20      59   15  91
    19.40     21.00      25    6  97
    21.50     22.20       7    2  99
    23.50     24.80       4    1 100

     MISSING DATA 1                0.
     MISSING DATA 2                0.
     MISSING DATA 3                0.

     GOOD N                      406.

     MEAN                     15.5197
     VARIANCE                  7.8588
     S.D.                      2.8034
```

1.4 FREQ: Frequency Distributions

The FREQ command produces frequency displays for each numeric variable in the input file. The variables may be categorical or continuous. Subgroups may be defined and displays may be produced for each of these groups.Intervals or categories are defined by the frequency command so that the same number of categories are on either side of the mean. Users may specify an arbitrary number of categories, equally sized categories, the printing of category boundaries (rather than the observed low and high scores), and other options.

If, instead of frequency distributions for *grouped* data, frequency distributions for all *individual* values or for *character* data are desired, use the COUNTS command which is documented in "P-STAT Introductory Manual"..

1.5 Output of Frequency Distributions

The output from the FREQ command shows an ordered list of value categories, the counts of the observations in each category ("N"), the percent the counts are of the total number of counts, and the cumulative percent the sum of the counts are of the total number of counts. The mean, standard deviation and variance are also shown.

A frequency distribution provides the central value of a variable and the dispersion or variability of the other scores around that central value. The three most commonly used measures of central value are: 1) the mean or the average score, 2) the median, the score with half of the population below it and half above it, and 3) the mode, the score with the most observations.

The mean is given in the output produced by FREQ. The category containing the median can be found by looking at the cumulative percent. In Figure 1.3, for the category bounded by 14.10 and 15.50, the cumulative percent moves from 30 percent to 54 percent. The median occurs where the cumulative percent equals 50.

The category containing the mode can be found by looking at the column of counts. The category containing the median (14.10 to 15.50) also contains the mode, since N is largest in that category (N is 97).

The mean, rounded to one significant place for classification purposes, is 15.5 and thus also in the same category. In a normal distribution of data, the mean, the median and the mode are the same and the distribution of values is symmetrical around them.

The output from FREQ also gives the variance and standard deviation for each variable. Whereas the mean, the median and the mode are all measures of central tendency, the variance and standard deviation are the most common measures of variability. They are measures of how different each value is from every other value — how spread out or clustered together the values are about the mean.

The range is a simple measure of variability. It is the difference between the largest and the smallest values. Acceleration, in Figure 1.3, has a range of approximately 16.8.

The actual counts or frequencies that are used in the frequency distribution may be weighted. The identifier WEIGHT supplies the name of a variable whose values are used to weight the count. Tests of statistical significance are affected by weighting. Thus, weighting should probably not be used if the significance tests produced by FREQ are of interest. (For further discussion of weighting and calculating weights, see the CORRELATE.)

1.6 Defining Categories

The FREQ command arbitrarily divides the actual range of the data values (obtained from either a pass through the input file or from the low and high values in the description file) into categories so that the display fits on the current output device. In interactive runs when SCREEN is set, the range is divided according to the SCREEN setting so that the entire frequency distribution fits on the screen. This usually permits a maximum of ten categories. If SCREEN is not set, the range is divided into a maximum of 30 categories to fit within the usual LINES setting of 56.

When an alternate print destination is specified, such as:

```
FREQ  Tests,  PR  'PrtFile'  $
```

the number of categories is recalculated and the frequency distribution is reformatted to fit within the number of lines available for the new device.

The general identifier LINES may also be used as part of the FREQ command to set the number of lines for the display. LINES may be used in either interactive or batch runs. If LINES is an identifier in the FREQ command, it takes precedence over the SCREEN setting:

```
SCREEN 22  $
FREQ,  LINES 56  $
```

Here, the categories are calculated to fit within 56 lines even though SCREEN is set to 22. However, if LINES and SCREEN are both used as commands:

```
SCREEN 24  $
LINES  50  $
FREQ   $
```

the SCREEN setting takes precedence over the lines setting in determining how many categories are used.

The number of categories may be reset by the use of the optional identifier NCAT followed by an even number as its argument:

```
FREQ  Tests,  NCAT 12  $
```

There is always an even number of categories. If NCAT is given, the SCREEN and LINES settings have no effect on the number of categories.

In Figure 1.3, the first category has a low score of 8 and a high score of 9.50. The low and high scores that are printed are not the actual category boundaries. The low score for each category is the lowest score observed at or within the category boundaries; the high score is the highest score observed at or within the category boundaries. The optional identifier TRUE may be used to request that the actual category boundaries print. Figure 1.4 shows the effect of the TRUE identifier on the categories.

If there are categories with no values, the FREQ program does not print them. The ALLCAT identifier may be used to request that all categories print even if they are empty. This allows clearer viewing of the shape of data that is not normally distributed. Empty categories (or categories with only few observations) consistently near one end of a distribution may indicate a skewed distribution of values. Empty categories in the middle of a distribution may indicate a multi-modal distribution; that is, a distribution encompassing two sample populations.

The DOWN identifier may be used to request that the categories print in a downwards direction — that is, going from highest to lowest.

1.7 Handling Skewed Distributions

The FREQ program assigns categories evenly on either side of the mean. Half of the categories are above the mean and half are below the mean. When the distribution is skewed, this produces more categories with observations than if the categories are spread evenly over the range. The EQUALCAT identifier may be used to override this provision and force the program to assign equal categories over the range of variables. The settings assumed by FREQ may not suit all variables, but they are usually adequate for the first examination of the data. Information from a simple frequency display often provides the basis for further study of the data (through the use of optional identifiers) and for the selection of appropriate statistical procedures.

1.8 FREQUENCY DISPLAYS WITH SUBGROUPS

The frequency distribution of a variable provides information only about that variable. The FREQ program provides an option for frequency displays of subgroups. Frequency distributions with subgroups give information about the relationship between two variables. The distribution of data values for each subgroup is shown, as well as the distribution for the total group. Figure1.4 shows a frequency distribution with subgroups.

The FREQ program prompts for subgroups when a *semicolon* is used to signal that subcommand information follows:

```
FREQ   CarFile ( KEEP  Acceleration  Origin ),  DES  CarDes ;

Enter definition for group 1,
or H-Q-$ :
American Origin 1 1
```

There is no limit on the number of subgroups in FREQ. However, the order in which the fields occur in the subgroup definitions is important. The group name must be first and may be up to 16 characters long. It must be enclosed in quotes if it does not conform to the rules for legal P-STAT names. The variable follows the group name and is referenced by its name, such as Origin, or by its current position in the file. The variable is followed by values defining the range to be included in the group. Either categorical or continuous variables may be used to define groups.

FREQ also produces frequency distributions for subgroups when the identifier SUB is used. The argument for SUB is the variable whose values define the subgroups:

```
FREQ  Tests,  SUB  Sex  $
```

The values of Sex define two subgroups; frequency distributions are produced for both of these groups.

SUB defines subgroups for *categorical* variables (variables with discrete categories). Using the identifier SUB produces frequency distributions for every subgroup (every category) of a categorical variable. However, the using group definitions at the subcommand level also creates frequency distributions for categorical variables. In addition, it offers the option of regrouping. For example, if a variable consists of 20 discrete categories, it may be more desirable to view only two subgroups, the first consisting of the first ten categories and the second consisting of the second ten categories, than to view all 20 categories as subgroups. The identifier SUB does not provide this option.

The identifier ALLGRP requests that each case be included in every group for which it qualifies. For example, given three values of the variable Origin (country of origin) — American, European and Japanese — four subgroups, such as American, Foreign, Asian and Caucasian, could be formed by various combinations of these cases. If ALLGRP is not used, each case is included only in the first group for which it qualifies.

1.9 Output of FREQ with Subgroups

Frequency displays with subgroups are similar to simple frequency displays. However, when FREQ is used with subgroups an analysis of variance table (“one-way ANOVA”) is calculated. An F value is produced when there are two or more subgroups, and a t value is also produced when there are only two subgroups. (The t value is the square root of the F value when there are two groups). The significance of the t and F values depends on the sample size. (Note that the use of ALLGRP invalidates the t and F statistics).

Of interest is the probability associated with the F or t statistic. In Figure 1.4, the probability equals zero. This means that there is a near zero probability that the difference between the acceleration of American and Foreign cars is attributable to error. An investigator could infer that the country of origin of the automobile is responsible for the difference in average acceleration between the groups. (See the ANOVA for more information on analysis of variance.)

The TTEST and PAIRED.TTEST commands also compute t statistics for testing the difference between two means. If the t test is the portion of the FREQ output that is of interest, these two commands may produce a more accurate statistic because they address issues of equivalent variances for the two groups, significance levels, and correlation of measurements between the groups. (See the TTEST chapter.)

1.10 Chi-Square

The VERBOSITY level associated with FREQ may be set to 4, when subgroups have been defined, to also obtain a frequency distribution of the grouping variable itself and a chi-square statistic:

```
      FREQ   CarFile  ( KEEP  Acceleration  Origin ),
        DES  CarDes,  VERBOSITY 4  $
```

The chi-square tests whether the grouping variable is independent of the variable whose counts or frequencies are being displayed. It measures the discrepancy between the values that would be expected in each category if there were no association between the variables and the actual or observed values. A low probability suggests a relationship between the variables.

Figure 1.4 Frequency Distribution with Subgroups

```
>>   FREQ  CarFile ( KEEP  Acceleration Origin),  DES  CarDes,  TRUE ;
           American Origin 1 1
           Foreign  Origin 2 3
           $

       FILE         CarFile

VARIABLE    2,        Acceleration
```

		ALL			American			Foreign		
LOW	HIGH	N	PCT	CUM	N	PCT	CUM	N	PCT	CUM
8.00	9.50	7	2	2	7	3	3			
9.50	11.01	15	4	5	15	6	9			
11.01	12.51	35	9	14	31	12	21	4	3	3
12.51	14.02	66	16	30	48	19	40	18	12	14
14.02	15.52	97	24	54	50	20	59	47	31	45
15.52	17.38	91	22	77	56	22	81	35	23	68
17.38	19.23	59	15	91	32	13	94	27	18	86
19.23	21.09	25	6	97	12	5	99	13	9	95
21.09	22.94	7	2	99	3	1	100	4	3	97
22.94	24.80	4	1	100				4	3	100
MISSING DATA 1		0.			0.			0.		
MISSING DATA 2		0.			0.			0.		
MISSING DATA 3		0.			0.			0.		
GOOD N		406.			254.			152.		
MEAN		15.5197			14.9406			16.4875		
VARIANCE		7.8588			7.8682			6.3881		
S.D.		2.8034			2.8050			2.5275		

	SUM SQ	VAR	DF	F	PROB=
BETWEEN	227.56	227.56	1.		
WITHIN	2955.26	7.31	404.	31.11	0.000

```
  T   =      5.58
```

1.11 PERCENTILES: the Median, Skewness, Kurtosis

An augmented description file is produced by the PERCENTILES command. The output file contains the same summary information that the regular description file contains, as well as measures of skewness and kurtosis, the median (the 50th percentile), quartiles, deciles, and various other percentiles.

Figure 1.5 illustrates using the PERCENTILES command and shows the output file. PERCENTILES requires an input file and the OUT identifier, followed by a name for the output file. The median, the 50th percentile, is automatically included in the output file, unless specific percentiles are requested. (Note that percentiles are not calculated for character variables.)

Figure 1.5 PERCENTILES: An Augmented Description

```
----------The Commands---------------------------------------

PERCENTILES  Survey,  OUT  SurveyQ,  GET QUARTILES  $

LIST  SurveyQ  [ DROP  Type ]  $


----------The Listing of the Output File----- ---------------
```

Name	Mean	S.D.	Low	High	Miss .1	Miss .2	Miss .3
Sex	1.5515	0.49784	1	2	0	0	0
Age	3.1765	1.46202	1	5	0	0	2
Education	2.1194	0.67700	1	3	0	0	1
Occupation	2.2746	0.67428	1	3	47	0	0
Marital.Status	1.2970	0.45739	1	2	0	0	0
Work.Status	2.1394	0.94738	1	3	0	0	0
Children	2.2378	1.96982	0	8	0	0	3
Siblings	4.4101	3.47220	0	27	0	0	0
Hrs.Last.Week	39.5384	13.80181	2	89	261	0	0

Name	Good	Skewness	Kurtosis	P.25	P.50	P.75
Sex	495	-0.207793	-1.964776	1.00	2	2.00
Age	493	-0.081280	-1.387374	2.00	3	5.00
Education	494	-0.149659	-0.824420	2.00	2	3.00
Occupation	448	-0.393369	-0.810312	2.00	2	3.00
Marital.Status	495	0.891388	-1.210332	1.00	1	2.00
Work.Status	495	-0.281471	-1.831012	1.00	3	3.00
Children	492	0.804045	0.188952	0.25	2	3.00
Siblings	495	1.451907	3.698629	2.00	3	6.00
Hrs.Last.Week	234	0.395600	2.268292	35.00	40	43.25

Any combination of individual values (ranging from .0001 to .9999) and group names may be given. The requested percentiles appear in the output file in low-to-high order, without duplication.

Percentiles are requested with the GET identifier:

```
GET  5  10  90  95  QUARTILES,
```

In this example, the percentiles P.05, P.10, P.25, P.50, P.75 (the quartiles), P.90 and P.95 are arguments of GET. Either 95 or .95 can be entered. Either one produces the 95th percentile (P.95). When GET is used, only the specified percentiles are calculated; that is, the median is not automatically included.

The following predefined group names calculate the specified percentiles:

1.	`GET MEDIANS,`	produces P.50
2.	`GET QUARTILES,`	produces P.25, P.50, P.75
3.	`GET DECILES,`	produces P.10, P.20, P.30, P.40, P.50, P.60, P.70, P.80, P.90
4.	`GET CENTILES,`	produces NINETY-NINE PERCENTILES, P.01 through P.99
5.	`GET EDGES,`	produces P.01, P.05, P.95, P.99

The output file produced by PERCENTILES also includes measures of skewness and kurtosis, which provide information about the shape of the distribution. Skewness refers to the symmetry of the distribution. It has a value of zero when the distribution is symmetrical, is positive when the distribution is positively skewed (low values bunched close to the mean and high values forming a long tail on the right of the curve), and is negative when it is negatively skewed (high values bunched close to the mean and the low values forming a long tail on the left of the curve).

Kurtosis refers to the flatness or peakedness of the distribution. When it is positive, the distribution has longer tails than a normal distribution with the same standard deviation — it is leptokurtic. When it is negative, the distribution is flat-topped with shorter tails — it is platykurtic. The moments about the mean, which figure in the calculation of skewness and kurtosis, appear in the output file when the identifier MOMENTS is included in the PERCENTILES command.

A WEIGHT variable with either integer or fractional values may be specified in the PERCENTILES command. By default, PERCENTILES interpolates when the requested quantiles fall between actual data values, unless fractional weights are used. There is no interpolation when the requested quantiles fall a certain fraction into a group of equal values.

The OPTION identifier may be used to specify whether or not interpolation is desired. OPTION 1 uses interpolation to determine the various quantiles; OPTION 2 does not use interpolation. Interpolation (OPTION 1) is not an option when there are fractional weights.

Figure 1.6 shows a data file containing two weight variables, one with integer values and one with fractional values. The chart following the file shows the median (50th percentile) calculated by PERCENTILES with various options.

The PERCENTILES command requires a sizeable amount of computer memory to calculate many percentiles for large files. Therefore, only the percentiles actually desired should be requested. When many percentiles are needed, multiple passes are made through the input file. The number of passes depends upon:

1. the size of P-STAT in use, which affects how many of the requested variables can be done in one step, and

2. how rapidly the search converges for each requested percentile.

Figure 1.6 Interpolation Depends on OPTION and if Weight is a Fraction or Integer

----------FILE Test:--------------------------------

X	Integer Wt	Fraction Wt
10	1	1.5
11	1	0.5
11	1	1.5
12	2	1.5
14	2	3.0
15	3	2.0

----------Median Computed given OPTION and Weight-----

Median of X	Weighted	Weight Variable	Option	Default	Interpolated
11.5	no	-	1	yes	yes
11.0	no	-	2	no	no
13.0	yes	Integer.Wt	1	yes	yes
12.0	yes	Integer.Wt	2	no	no
12.0	yes	Fraction.Wt	1	no	no
12.0	yes	Fraction.Wt	2	yes	no

1.12 Subgroups With PERCENTILES

If the input file is *sorted or grouped* by one or more variables, PERCENTILES can produce summary information on those subgroups. The identifier BY is followed by up to 15 variables that define the subgroups. The output file contains one case for each subgroup of each variable in the file. (That is, the number of variables in the input file multiplied by the number of values of each of the BY variables is the number of cases in the output file.) The BY variables may be any combination of numeric or character variables. Summary statistics are not computed for the BY variables.

Figure 1.7 illustrates PERCENTILES with subgroups. Since the file is not grouped by the variable Sex, the SORT command must be used first. Two subgroups multiplied by the nine variables in the input file results in an output file with 18 cases, one for each variable subgroup combination. The LIST command is used with variable selection to present the information in a concise and attractive manner.

Figure 1.7 PERCENTILES with Subgroups

```
----------The Commands-----------------------------

SORT  Survey,           BY  Sex,  OUT  SurvSort  $
PERCENTILES  SurvSort,  BY  Sex,  OUT  SurvStat  $


LIST  SurvStat [ KEEP  Name Sex Good Mean S.D. P.50 ;
                 RENAME P.50  TO  Median ],
      STUB  Name,  SKIP.VAR  Name,  PLACES 2,
      LABELS  'SurLabs',  LEFT  $


----------The Listing-----------------------------
```

Name	Sex	Good	Mean	S.D.	Median
Age	male	221	3.26	1.53	3
	female	272	3.11	1.40	3
Education	male	222	2.11	0.75	2
	female	272	2.12	0.61	2
Occupation	male	213	2.32	0.65	2
	female	235	2.23	0.69	2
Marital.Status	male	222	1.31	0.46	1
	female	273	1.29	0.45	1
Work.Status	male	222	1.80	0.94	1
	female	273	2.41	0.87	3
Children	male	220	1.92	1.96	2
	female	272	2.49	1.95	2
Siblings	male	222	4.43	3.68	3
	female	273	4.40	3.30	3
Hrs.Last.Week	male	143	42.27	14.18	40
	female	91	35.25	12.07	40

1.13 Descriptive Statistics for Subgroups

Means and totals for subgroups can be obtained using the LIST command. The commands AGGREGATE and DUPLICATES produce a variety of subgroup statistics in P-STAT system files. If none of these commands provide exactly what is needed, the P-STAT Programming Language permits calculation of user-defined statistics. Scratch variables, the permanent vector, the wildcard character “?”, the COLLECT and SPLIT instructions, and the FIRST, LAST, FIRST.GOOD, LAST.GOOD, SUM and SUM.GOOD functions facilitate all types of calculations.

Figure 1.8 Standardizing Data

----------The Commands----------

```
STANDARDIZE  Cars,  SDATA  CarsStan,
      CARRY  Year  Origin  Cylinders  $

LIST  CarsStan [ IF  Year = 80,  RETAIN ],  GAP 1,  PLACES 2  $
```

----------Listing of the Standardized Output----------

Model	Mpg	Cylinders	Displacement	Horsepower	Weight	Acceleration	Year	Origin
VW RABBIT	2.30	4	-0.92	-0.75	-0.99	-0.29	80	2
TOYOTO COROLLA T	1.87	4	-1.01	-1.16	-1.19	1.17	80	3
CHEVROLET CHEVET	1.10	4	-0.92	-0.90	-1.01	-0.01	80	1
DATSUN 310	1.75	4	-1.04	-1.03	-1.13	0.31	80	3
CHEVROLET CITATI	0.57	4	-0.42	-0.39	-0.36	0.35	80	1
FORD FAIRMONT	0.37	4	-0.52	-0.44	-0.13	0.92	80	1
AMC CONCORD	0.10	4	-0.42	-0.39	0.03	1.63	80	1
DODGE ASPEN	-0.56	6	0.29	-0.39	0.47	1.13	80	1
AUDI 4000	1.38	4	-0.93	-0.70	-0.93	0.10	80	2
TOYOTO CORONA LI	0.80	4	-0.58	-0.39	-0.32	-0.01	80	3
MAZDA 626	1.00	4	-0.71	-0.78	-0.52	0.71	80	3
DATSUN 510 HATCH	1.73	4	-0.72	-0.34	-0.64	-0.19	80	3
TOYOTO COROLLA	1.11	4	-0.83	-0.78	-0.84	-0.11	80	3
MAZDA GLC	2.95	4	-1.04	-1.03	-1.03	0.85	80	3
DODGE COLT	0.56	4	-0.37	-0.00	-0.21	-0.40	80	1
.....								
DATSUN 210	2.21	4	-1.05	-1.03	-1.03	1.31	80	3
VW RABBIT C (DIE	2.66	4	-1.00	-1.47	-1.06	2.20	80	2
VW DASHER (DIESE	2.54	4	-1.00	-1.47	-0.76	2.92	80	2
AUDI 5000S (DIES	1.65	5	-0.70	-0.98	-0.03	1.56	80	2
MERCEDES-BENZ 24	0.83	4	-0.47	-0.98	0.32	2.24	80	2
HONDA CIVIC 1500	2.70	4	-0.99	-0.98	-1.33	-0.61	80	3
RENAULT LECAR DE	2.22	4	-1.05	0.	-1.35	0.64	80	2
SUBARU DL	1.32	4	-0.93	-0.98	-0.99	0.88	80	3
VOLKSWAGEN RABBI	0.80	4	-1.01	-1.11	-1.34	-0.08	80	2
DATSUN 280-ZX	1.18	6	-0.26	0.69	-0.08	-1.47	80	3
MAZDA RX-7 GS	0.02	3	-1.19	-0.13	-0.66	-1.08	80	3
TRIUMPH TR7 COUP	1.47	4	-0.69	-0.44	-0.57	-0.15	80	2
FORD MUSTANG COB	0.01	4	-0.52	0.	-0.09	-0.44	80	1
HONDA ACCORD	1.14	4	-0.84	-0.85	-0.81	0.53	80	3

1.14 STANDARDIZE

The STANDARDIZE command produces an output file with standardized data values. Options permit replacing missing data with their mean, weighting the data prior to standardization, and carrying unstandardized variables into the output file. Standardized data, called *standard* or *z scores*, have a mean of zero and a standard deviation of one.

Figure 1.8 illustrates the use of the STANDARDIZE command. The input file and a name for the output file of standard scores are the only required items. CARRY is an optional identifier that specifies variables that should be carried directly from the input file to the output file without being standardized:

```
STANDARDIZE  Cars,  SDATA  CarsStan,
      CARRY  Year  Origin  Cylinders  $
```

Typically, categorical variables that serve primarily to classify or group cases are carried and not standardized. Any character variables are automatically carried into the output file.

The output file that is produced when the SDATA identifier is used substitutes the mean of the standardized data, which is zero, for values that are missing in the input file. This is the assumed option in the STANDARDIZE command, unless the STAY.MISSING identifier is used to request that missing values remain missing.

The STANDARDIZE command may also be used to calculate standard scores based on the mean and standard deviation of another data file. A description file, produced in a previous command and summarizing one file, is input to STANDARDIZE using the DES identifier:

```
STANDARDIZE  TestFile,  DES  TestDes,  SDATA  StanFile   $
```

The mean and standard deviation from this DES file are used to standardize the data in the input file. If the DES file is a description file of some file other than the input file, the means and standard deviations of the output data may be some values other than zero and one. When a description file is not input, the STANDARDIZE program computes the means and standard deviations from the input file.

The data values in the input file may be weighted before they are standardized. The WEIGHT identifier supplies the name of the weight variable:

```
STANDARDIZE  Spots,  WEIGHT  Freq,  SDATA  SpotsST  $
```

Instead of “counting as one,” each case is counted as the value of the weight variable.

An optional output file containing the *original data* (not standard scores), but with the mean of each variable substituted for missing values of that variable, may be requested using the MDATA identifier. This output file may be input to commands that do not permit missing data. For example, DISCRIM, the command that does discriminant analysis, automatically omits any case with any missing data. If the mean is substituted for missing data values, these cases are included in the analysis. The CORRELATE command computes correlation coefficients using pairwise deletion — any pair of variables with a missing value is deleted from the calculation. If the mean is substituted, these variables are included in the analysis.

SUMMARY

FREQ

```
FREQ  Scores  ( KEEP  Math )  $

FREQ  Scores  ( KEEP  Math Sex ),  DES  ScoreDes ;
  Male    Sex  1
  Female  Sex  2
  $
```

The FREQ command produces frequency distributions for all *numeric* variables in the input file. Specific variables may be selected using the PPL instructions KEEP and DROP. Frequency distributions may be requested for subgroups by: 1) ending the command with a semicolon and entering subgroup definitions at the subcommand level, or 2) using the identifier SUB at the command level.

Required:

FREQ fn

supplies the name of the required P-STAT system file. If a name is not supplied, the most recently referenced file will be used.

Optional Identifiers:

ALLCAT

requests that all the categories print even if they are empty. The FREQ program normally prints only those categories that have data.

ALLGRP

requests that each case be included in every group for which it qualifies. If ALLGRP is not used, each case is included only in the first group for which it qualifies. Using ALLGRP invalidates the t or F statistics.

DES fn

supplies the name of a description file. When this identifier is not used, a pass is made through the data file to obtain the ranges.

DOWN

specifies that categories print from high to low. UP is assumed unless DOWN is specified.

EQUALCAT

divides the categories equally over the range of the variables. When this is not used, half the categories are above the mean and half are below the mean.

LINES nn

specifies the number of lines to be used for the output. The SCREEN command may also be used in interactive runs (prior to FREQ) to set the screen line setting. The SCREEN command takes precedence over LINES unless LINES is explicitly used as an identifier in the FREQ command.

NCAT nn

specifies the number of categories. This must be an even number. If NCAT is not used, either the LINES setting or the SCREEN setting determine the maximum number of categories.

SUB vn

specifies the variable defining subgroups. The SUB variable should have values of 1, 2, 3, and so on. Each value defines a separate subgroup.

SUMMARY

causes a one line summary, containing t or F values, to print for each variable after the full frequency printout for all the variables is complete.

T.TESTS

requests a t test for each pair of subgroups. Use with caution when the number of subgroups is more than three. *All* pairwise comparisons are printed.

TRUE

causes the actual category boundaries to print. If this is not used, the lowest and highest observed values in each category are printed.

UP

specifies that categories print from low to high. UP is assumed.

WEIGHT vnp

supplies the name of the variable to be used for weighting.

VERBOSITY nn

sets the verbosity level. When it is less than 4, the variable that defines the groups is not included in the frequencies. Chi-square is produced only when the verbosity level is 4.

MODIFY

```
MODIFY  Survey,  DES  SurvDes  $
```

The MODIFY command may be used to request an output description file that summarizes the variables in the input file. PPL modification phrases enclosed in parentheses may follow the input file. Description files may also be requested using the CONCAT and CORRELATE commands.

Required:

MODIFY fn

provides the name of the required P-STAT system file.

Optional Identifiers:

DES fn

provides a name for and requests an output description file.

OUT fn

provides a name for an output file containing all modifications done to the input file.

WEIGHT vn

provides the name of a variable whose values are used for integer or fractional weighting of each case.

OVERALL.DES

```
OVERALL.DES  Des1  Des2  Des3,  OUT  Des123  $
```

OVERALL.DES reads a number of *description files* and produces a single description file that summarizes the input files. Any pair of variables whose statistics are to be combined must either be both numeric or both character type. However, numeric variable pairs may be single and double precision, or integer and non-integer.

Required:

OVERALL.DES fn fn

specifies the names of the required input description files. s

Required Identifiers:

OUT fn

provides a name for the output description file.

PERCENTILES

```
PERCENTILES  Survey,  OUT  SurvStat,  GET  QUARTILES  $
```

The PERCENTILES command produces an augmented description file, including medians, various percentiles, and measures of skewness and kurtosis. Moments about the mean are an option. Descriptive statistics for subgroups may be req3quested using the BY identifier, which is followed by the variables whose values define the subgroups. The input file must already be *grouped or sorted* by those variables.

If desired, a WEIGHT variable may be specified. Interpolation is assumed by PERCENTILES, unless there are fractional weights. The OPTION identifier specifies whether or not interpolation is desired.

Required:

PERCENTILES fn

specifies the input P-STAT system file.

Required Identifiers:

OUT fn

provides a name for the output P-STAT system file which will contain the various statistics and percentiles.

Optional Identifiers:

BY vn vn

specifies up to 15 variables whose values define subgroups and requests descriptive statistics for each group. The BY variables may be numeric or character variables, or a mixture of both types. The input file must already be sorted or grouped on the BY variables.

GET nn nn

requests that the specified percentiles be computed. The percentiles in the list may either be in the form of numbers (10 or P.10), or in the form of standard intervals. The following group names or intervals are supported: MEDIANS, QUARTILES, DECILES, EDGES and CENTILES.

MOMENTS

requests that the second, third and fourth moments about the mean be included in the output file.

OPTION nn

specifies whether or not interpolation is desired. OPTION 1 uses interpolation to determine the various quantiles; OPTION 2 does not use interpolation. Normally, PERCENTILES uses interpolation (OPTION 1) unless there are fractional weights.

WEIGHT vn

specifies the name of a variable whose values are used to weight each case in the input file. When their are fractional weights, PERCENTILES does not use interpolation in computing quantiles.

STANDARDIZE

```
STANDARDIZE  Survey,  SDATA  ZScores  $
```

STANDARDIZE produces an output file with standardized data (the mean is set to zero and the standard deviation to one). Missing values are replaced by the mean (zero) of the standardized data. Options permit an output file of the original data with the mean of each variable substituted for missing data, an output file of standardized data without means replacing missing values, weighting the data prior to standardization, and carrying unstandardized variables into the output file.

Required:

STANDARDIZE fn

specifies the name of the required input system file.

Optional Identifiers:

CARRY vn vn

specifies variables that are to be carried directly from the input file to the output file, without being standardized. This is often a suitable option for categorical variables. Character variables are automatically carried into the output file.

DES fn

provides the name of a *description file* produced in a previous command. The means and standard deviations from this description file are used to standardize the input file. If DES is not provided, the program computes the means and standard deviations from the input file.

MDATA fn

provides a name for an output file that has the original data in the input file, with the mean of each variable substituted for any missing values of that variable.

SDATA fn

provides a name for an output file of standardized scores. Missing values in the input file are given the mean value of the standardized data (zero) in the SDATA file, unless STAY.MISSING is used. (Thus, the standard deviation of the resultant data may become less than one if missing scores are replaced by means.)

STAY.MISSING

indicates that missing scores are to remain missing. The mean value of the standardized data (zero) is not to be substituted for the missing values.

WEIGHT vn

specifies a variable whose values are used to weight each case. Data is weighted before it is standardized.

2
TTEST:
Comparing Two Groups

TTEST compares the means of two sample groups and tests the significance of the difference between the means. The groups may either be independent (different cases with common variables) or correlated (paired different variables). The TTEST command is used when the groups are independent, and the PAIRED.TTEST command is used when they are correlated.

Cases with missing values on specific variables are excluded on a variable-by-variable basis. If desired, whole cases with any missing variables may be deleted as the file is input to either TTEST or PAIRED.TTEST by using the P-STAT programming language.

Similar nonparametric tests for comparing two samples are available in the NP.TEST command. They are typically used when the data do not meet the assumptions of parametric tests such as the t test. (See the discussion of assumptions and hypothesis testing in the beginning of the chapter on nonparametric statistics later in this manual.)

2.1 TTEST: Independent Variables

TTEST tests the hypothesis that the difference between the means of two sample groups is zero. The groups may be defined at either the subcommand level or with two description files. The output includes the value of the t statistic and the associated two-tail probability, the mean, standard deviation, standard error, and the appropriate pooled or separate variance estimate. If a one-tail t test is desired, merely halve the probability value in the output to obtain the probability associated with a one-tail test.

Sample groups containing independent (unrelated) cases with variables in common are defined by one or two background variables at the subcommand level. The groups may also be defined by two input description files containing the mean and standard deviation for each group, rather than by the individual cases. The significance of the difference between the variances of each group is tested with an F statistic. If the probability associated with the F statistic is less than or equal to the assumed threshold value of .05, separate variance estimates are calculated. If the F probability is greater than .05, a pooled variance estimate is calculated. The t statistic is calculated using the appropriate variance estimate. Other values for the threshold for the F value may be provided. In addition, the calculation of both variance estimates and both the corresponding t statistics may be requested.

If desired, ranked data for two groups may be input to TTEST. (See the RANK command in the CORRELATE chapter in this manual.) The resultant t value will approximate Wilcoxon's W (rank sum test) when the errors are not serially correlated; that is, when the errors are random. The initial distributions need not be normal.

The NP.TEST command performs other two sample tests similar to t test: for independent samples — the Median test, the Mann-Whitney U test, the Kolmogorov-Smirnov test, the Wald-Wolfowitz runs test, and the squared ranks test for equal variances; for paired samples — the Sign test and the Wilcoxon matched-pairs signed-ranks test.

Figure 2.1 Defining Groups with Subcommands

-----The Commands----------

```
TTEST  Solar,   OUT  SolarT ;
Short  Treatment  1
Long   Treatment  2  $

 LIST  SolarT,  STUB  Variable,  PLACES  2  $
```

-----The TTEST Output File----------

variable	group	good	mean	std dev	std error	f.var
Cell.No	Short	18	10.17	5.74	1.35	1.04
	Long	19	10.00	5.63	1.29	1.04
Efficiency	Short	18	4.06	1.25	0.30	10.67
	Long	19	5.60	0.38	0.09	10.67
Fill.Factor	Short	18	0.50	0.09	0.02	10.90
	Long	19	0.62	0.03	0.01	10.90
Voltage.OC	Short	18	0.77	0.02	0.01	1.27
	Long	19	0.80	0.02	0.00	1.27
Current.Density	Short	18	10.21	1.48	0.35	14.59
	Long	19	11.26	0.39	0.09	14.59
Process	Short	18	2.00	0.84	0.20	1.02
	Long	19	1.95	0.85	0.19	1.02

.....

variable	f.prob	variance est	t. value	df	t.prob
Cell.No	0.93	Pooled	0.09	35.0	0.93
	0.93	Pooled	0.09	35.0	0.93
Efficiency	0.	Separate	-4.99	20.0	0.
	0.	Separate	-4.99	20.0	0.
Fill.Factor	0.	Separate	-5.50	19.9	0.
	0.	Separate	-5.50	19.9	0.
Voltage.OC	0.62	Pooled	-4.25	35.0	0.
	0.62	Pooled	-4.25	35.0	0.
Current.Density	0.	Separate	-2.91	19.2	0.01
	0.	Separate	-2.91	19.2	0.01
Process	0.97	Pooled	0.19	35.0	0.85
	0.97	Pooled	0.19	35.0	0.85

2.2 Defining Groups with Subcommands

The TTEST command defines groups either at the subcommand level with group definitions (Figure 2.1), or from information in two input description files (Figure 2.2). All input variables are analyzed. Any variable selection is done using the programming language as the file is input to TTEST.

The TTEST command requires the name of the input file containing the data and a name for the output (OUT) file of statistics:

```
TTEST  Solar,  OUT  SolarT ;
```

The semicolon at the end of the command indicates that subcommand information follows. Group definitions are provided at the subcommand level. The definitions take the following form:

```
Group.Name.1  Variable.Name  Low  High
Group.Name.2  Variable.Name  Low  High
```

The group name may be any valid name; that is, a name of no more than 16 characters that includes only letters, numbers and periods, and that begins with a letter. The variable name is the name of the variable whose values define group membership. Only two groups may be defined. The low and high values are the bounds for inclusion in a group. Only one of these values need be provided when the low and the high bounds are equal.

Two defining variables, each with a low and high value, may be used:

```
Senior.Males     Age  65  100  Sex  1
Senior.Females   Age  65  100  Sex  2
```

The asterisk may be used as a shortcut in defining the second group when all cases not included thus far comprise the second group:

```
Males    Sex  1
Females  *
```

In this example all cases with values other than 1 for the variable Sex comprise the second group, Females.

TTEST is illustrated in Figure 2.1. The groups are defined by subcommands:

```
Short  Treatment  1
Long   Treatment  2  $
```

Cases with a value of 1 for the variable Treatment are members of the “Short” group, and cases with a value of 2 are members of the “Long” group. (The solar cells had either a short or a long baking period.) Since there can only be two groups, the command ending $ may be omitted.

A threshold value of .05 is assumed for the probabilities associated with the F ratios testing whether or not the group variances for each of the test variables are equal. The probabilities associated with the F ratios for Efficiency, Fill.Factor and Current.Density are less than or equal to .05, so separate variance estimates are computed for these variables. Pooled variance estimates are computed for the remaining variables, because the probabilities associated with their F ratios exceed .05.

Figure 2.2 Defining Groups with Description files

----------The Commands----------

```
MODIFY  Solar  [ IF  Treatment  =  1,  RETAIN ],  DES  ShortDes  $
MODIFY  Solar  [ IF  Treatment  =  2,  RETAIN ],  DES  LongDes   $
TTEST,  DES.1  ShortDes,  DES.2  LongDes,  OUT   TestDes $
LIST  TestDes,  STUB  Variable,  PLACES  2  $
```

----------The Output File----------

variable	group	good	mean	std dev	std error	f.var
Cell.No	ShortDes	18	10.17	5.74	1.35	1.04
	LongDes	19	10.00	5.63	1.29	1.04
Fill.Factor	ShortDes	18	0.50	0.09	0.02	10.90
	LongDes	19	0.62	0.03	0.01	10.90
Voltage.OC	ShortDes	18	0.77	0.02	0.01	1.27
	LongDes	19	0.80	0.02	0.00	1.27
Current.Density	ShortDes	18	10.21	1.48	0.35	14.59
	LongDes	19	11.26	0.39	0.09	14.59
Process	ShortDes	18	2.00	0.84	0.20	1.02
	LongDes	19	1.95	0.85	0.19	1.02
Treatment	ShortDes	18	1.00	0.	0.	9999.00
	LongDes	19	2.00	0.	0.	9999.00

.....

variable	f.prob	variance est	t.value	df	t.prob
Cell.No	0.93	Pooled	0.09	35.0	0.93
	0.93	Pooled	0.09	35.0	0.93
Fill.Factor	0.	Separate	-5.50	19.9	0.
	0.	Separate	-5.50	19.9	0.
Voltage.OC	0.62	Pooled	-4.25	35.0	0.
	0.62	Pooled	-4.25	35.0	0.
Current.Density	0.	Separate	-2.91	19.2	0.01
	0.	Separate	-2.91	19.2	0.01
Process	0.97	Pooled	0.19	35.0	0.85
	0.97	Pooled	0.19	35.0	0.85
Treatment	0.	Separate	9999.00	-	-
	0.	Separate	9999.00	-	-

There are 12 variables in the TTEST output file. They are the Variable, the Group, the Good (non-missing) count, the Mean, the Std.Dev (standard deviation), the Std.Error (standard error), the F.Var (F testing equality of variances), the F.Prob (probability associated with the F value), the Variance.Est (type of estimate chosen), the T.Value (t value testing equality of means), the DF (degrees of freedom associated with the t test), and the T.Prob (probability associated with the t value).

The count of cases in the input file may be weighted using the WEIGHT identifier followed by the variable whose values are the weights. (See the discussion on weighting in the CORRELATE chapter.) WEIGHT may only be used when groups are defined at the subcommand level; that is, when description files are not input to TTEST.

2.3 Defining Groups with Description Files

Groups may be defined at the command level by using the identifiers DES.1 and DES.2 to input two description files. Each of the description files should contain summary information about one of the groups. Some variables must be common to both of the files, although the order of the variables in the description files does not matter. If there are extra, unmatching variables in either description file, they are ignored. An input file containing values for both groups should not be provided when DES.1 and DES.2 are used.

Figure 2.2 illustrates TTEST with two description files in the command. The output file contains one case for every variable which appears in both of the input description files. In this case, the results are identical to those in Figure 2.1, because the description files were created from the same input file and all of the variables are present and in the same order.

If the names of the corresponding variables in the two input description files are not the same, the NO.MATCH identifier may be used to indicate this. The first variable in the DES.1 file is compared with the first variable in the DES.2 file, the second in DES.1 with the second DES.2, and so on. This continues until one of the files has no more cases (variables).

2.4 Resetting the THRESHOLD Value

The identifier THRESHOLD may be used to specify an F probability to compare with the probabilities associated with the F values testing equality of group variances. When the F probability for a variable is less than or equal to the threshold value, the two groups have statistically significant different variances and separate variance estimates are used in calculating the t values. When the F probability is greater than the threshold value, the two groups have non-differing variances and a pooled variance estimate is used. When THRESHOLD is not used, an F probability of .05 is assumed.

The calculation of either a separate or pooled variance estimate may be forced . POOLED forces the calculation of a t value based on the pooled variance estimate, and SEPARATE forces the calculation of a t value based on separate variance estimates.

The identifier BOTH may be used to specify that the t values are to be computed both ways — appropriately for both the same and differing variances. The output file will have twice as many cases because it will include both separate and pooled variance estimates and the corresponding t values. This file is best listed using the BY identifier in the LIST command:

```
LIST  AutoT,  BY  Variable  Variance.Est  $
```

The user may select the appropriate t value based on differing criteria at different times.

Figure 2.3 Using PAIRED.TTEST

```
.----------The Data File: Students------------------------

 Pre  Post

  31    60
  75    85
  69    68
  83   100
  72    74
  79    75
  83    88
  90    94
  80    82


----------The Commands-----------------------------------


PAIRED.TTEST  Students,  OUT  PairedT  $

------PAIRED.TTEST completed--------
9 cases were read from file students

LIST  PairedT,  PLACES  2  $


----------The PAIRED.TTEST Output File------------------

vari                 std     std          mean    s.d..   s.err
able      mean       dev     err     h     dif     dif     dif      cor

Pre      73.56   17.16    5.72     9    7.11   10.25    3.42    0.80
Post     80.67   12.70    4.23     9    7.11   10.25    3.42    0.80

...........:

   cor
  prob       t     df  t.prob

  0.01    2.08      8    0.07
  0.01    2.08      8    0.07
```

2.5 PAIRED.TTEST: Correlated Variables

If the data are composed of pairs of correlated measurements, the TTEST procedure is not appropriate. PAIRED.TTEST uses the difference method to test the null hypothesis that the mean difference between the two

sample groups is zero. In effect the difference between the variables is treated as a variable and the mean of this difference is compared to zero.

Sample groups containing correlated variable pairs such as “before” and “after” measurements are defined by the order of the variables in the input file. The output includes the value of the statistic and the associated two-tailed probability. The mean, standard deviation and standard error of the mean, as well as the mean difference, the standard deviation and standard error of the difference are included. In addition, the correlation coefficient and its associated two-tailed probability are given.

PAIRED.TTEST is illustrated in Figure 2.3. File Students contains scores for some students taken before and after three weeks of intensive study. Because the data are paired and thus most likely correlated, the PAIRED.TTEST is the appropriate command to use.

PAIRED.TTEST assumes an even number of input variables, the first half of which are “pre” values and the second half of which are “post” values. With ten input variables, the first variable will be assumed to be paired with the sixth, the second with the seventh, and so on for a total of five t tests.

If the paired data values are adjacent variables (the first paired with the second, and so on), they may be reordered into the expected order using the KEEP instruction with a MASK as the file is input to PAIRED.TTEST:

```
PAIRED.TTEST  Hist101
  [ KEEP  V(1) .ON. ( MASK 01 )  V(1) .ON. ( MASK 10 ) ],
     OUT  HistTest  $
```

Although designed as a two-sample test, the PAIRED.TTEST command may to used to test if the difference between the mean of a single sample and a population mean is zero — that is, if the sample mean equals a known mean. As the input file is read by PAIRED.TTEST, a variable is generated equal to the population mean:

```
PAIRED.TTEST  Region12
  [ KEEP  Income;   GEN  Pop.Income  =  15,500 ],
    OUT  IncomeTT  $
```

The output file produced by the PAIRED.TTEST command takes approximately 132 print positions. The identifier LONG specifies 132 print positions and is assumed in the PAIRED.TTEST command. When there are many variable pairs, SKIP 2 in the LIST command improves the readability of the output. A shorter (narrower) output file may be requested by including the SHORT identifier in the PAIRED.TTEST command. An output file with approximately 80 print positions will be produced. This will list more satisfactorily on a terminal screen.

SUMMARY

TTEST

```
TTEST  Autos,  OUT  AutosT ;
  American  Origin  1  1
  Foreign   Origin  2  3
  $

TTEST,   THRESHOLD .01,
  DES.1  AmerDes,  DES.2  ForDes,
  OUT    AutosT  $
```

The TTEST command computes two-tailed t statistics for testing hypotheses that the means of variables for two comparison groups are equal. The groups are defined either by: 1) definitions at the subcommand level, or 2) two input description files, each with variables in common that summarize that group. Group definitions, given at the subcommand level, take the form:

```
Group.Name.1  Variable.Name  Low  High
Group.Name.2  Variable.Name  Low  High
```

The group name is any valid name. The variable name is that of the variable whose values define group membership. Low and high values are the bounds for inclusion in a group.

A threshold value specifying an F probability to use in determining whether or not the group variances are equal may be provided. A value of .05 is assumed for the threshold value when none is specified.

Required:

TTEST fn

supplies the required input file. If DES.1 and DES.2 are used, the original data file need not be input.

OUT fn

supplies a name for the required output file of statistics.

Optional Identifiers:

BOTH

requests that both t values be computed — the t appropriate when separate variance estimates are correct and the t appropriate when a pooled variance estimate is correct. The output file contains F values and their associated probabilities for each variance estimate. S ince this output contains twice as many cases, it is best listed using BY:

```
LIST  Height.t,  BY  Variable  Variance.EST  $
```

BOTH and THRESHOLD should not be used together. If they are, the THRESHOLD value will be ignored.

DES.1 fn

specifies the name of a description file whose variables give summary values for the first group. DES.2 must also be used and must have variables in common with DES.1. Its variables should give summary

values for the second group. An input file containing values for both groups should not be provided when DES.1 and DES.2 are used.

DES.2 fn

specifies the name of a description file whose variables give summary values for the second group. DES.1 must also be used and must have variables in common with DES.2. Its variables should give summary values for the first group. An input file containing values for both groups should not be provided when DES.1 and DES.2 are used.

NO.MATCH

may be used when DES.1 and DES.2 are used, if the names of the variables in the description files do not need to match. The first variable in the DES.1 file will be compared with the first variable in the DES.2 file. This continues until one of the files has no more cases.

POOLED

specifies that a pooled variance estimate be used to calculate the t values.

SEPARATE

requests that separate variance estimates be used to calculate the t values.

THRESHOLD nn

specifies an F probability value. If the probability associated with the F statistic testing equality of the group variances is less than or equal to the specified threshold value, separate variance estimates will be used to calculate the t values. If the F probability is greater than the threshold value, a pooled variance estimate will be used in calculating the t values. When THRESHOLD is not used, an F probability of .05 is assumed.

WEIGHT vn

specifies the name of a variable whose values will be used to weight the count of cases in the input file. WEIGHT may not be used with DES.1 and DES.2.

PAIRED.TTEST

```
PAIRED.TTEST  DrugTest,  OUT  DrugT  $

PAIRED.TTEST  Bearings
  [ KEEP  V(1) .ON. ( MASK 01 )  V(1) .ON. ( MASK 10 ) ],
  OUT  BearingT  $
```

The PAIRED.TTEST command tests the significance of the difference between the means of two groups, where the values in the groups are paired or correlated. The values may be construed as “pre” and “post” scores on a measurement.

PAIRED.TTEST assumes an even number of input variables, the first half of which are “pre” values and the second half of which are “post” values. With ten input variables, the first variable will be assumed to be paired with the sixth, the second with the seventh, and so on for a total of five t tests.

If the paired data values are adjacent variables (the first paired with the second, and so on), they may be reordered into the expected order using KEEP with a MASK. (This is shown in the second example given under the heading PAIRED.TTEST.)

Required:

PAIRED.TTEST fn

specifies the name of the required input file. Its variables should be ordered as explained above.

Required Identifiers:

OUT fn

provides a name for the output file. This file requires close to 132 print positions. Therefore, it is best listed on a printer. The use of SKIP 2 when the output file is listed will improve readability.

Optional Identifiers:

LONG

requests a complete output file of about 132 print positions. This file can be listed on most printers. LONG is assumed and need not be explicitly specified.

SHORT

requests a shorter (narrower) output file of about 80 print positions. The file can be listed satisfactorily on a terminal screen. When SHORT is not specified, LONG is assumed.

3
CORRELATE: Continuous and Discrete Variables

The CORRELATE command produces Pearson product-moment correlation coefficients for bivariate data (pairs of measurements) in which the variables are continuous interval or ratio measurements. For other types of variables (discrete, dichotomous, nominal and/or ordinal), SPEARMAN, BISERIAL, POINT BISERIAL, PHI or TETRACHORIC correlations can be produced. These correlations are either variants of the product-moment correlation, or estimates of it. Thus, the introductory discussion of the product-moment correlation also provides background information for these other types of correlations.

Appropriate use of these correlations is as follows:

1. Pearson r — Continuous quantitative variable pairs
2. Spearman — Ranked variable pairs
3. Biserial — Variable pairs comprised of one continuous and one dichotomous variable (with the variable underlying the dichotomy assumed to be continuous and normal)
4. Point Biserial — Variable pairs comprised of one continuous and one dichotomous variable (with the variable underlying the dichotomy inherently discrete)
5. Phi — Variable pairs comprised of two dichotomous variables (with the variables underlying the dichotomies assumed to be discrete)
6. Tetrachoric — Variable pairs comprised of two dichotomous variables (with the variables underlying the dichotomies assumed to be normally distributed)

The variance/covariance and the cross-product matrices are optional output files from the CORRELATE command.

When some variables in a file are continuous and others are dichotomous, the variables for each type of correlation may be selected (using KEEP or DROP phrases) as the file is input to the appropriate command. The resultant separate output matrices (files) of correlation coefficients may be joined together using the SJOIN command to form one matrix of correlations for all variables in the input file.

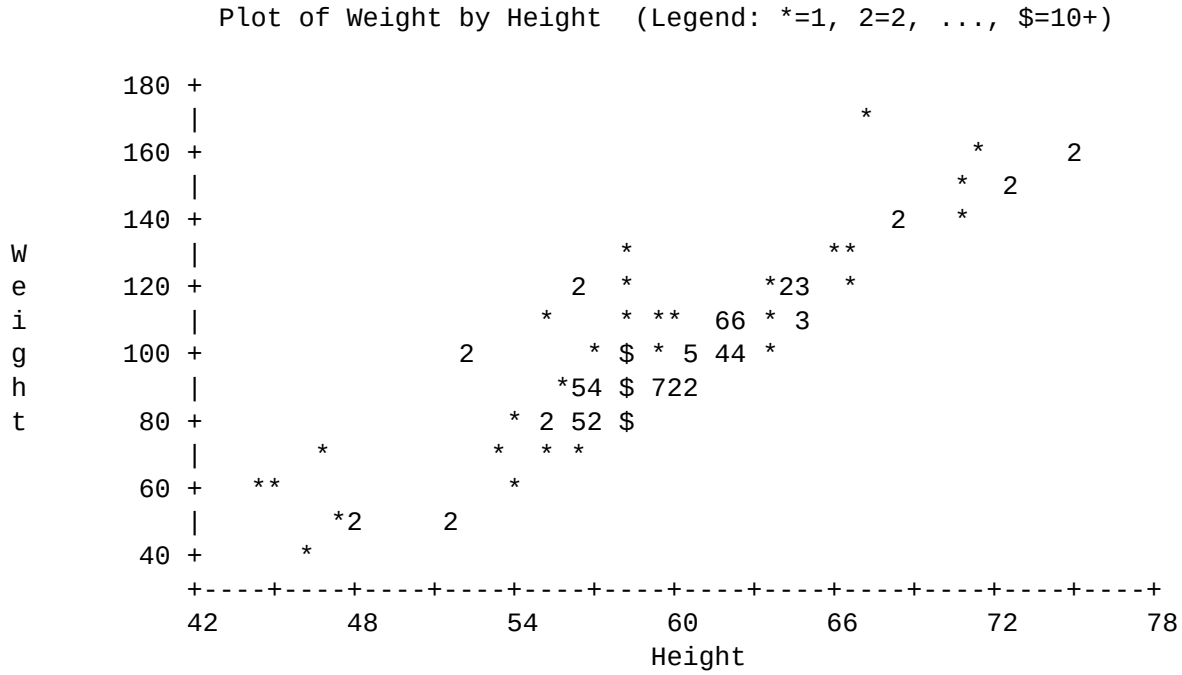
Some matrix operations and their products that may be useful in conjunction with correlation and multivariate regression analysis are as follows: 1) inversion of matrices, 2) production of matrices of partial and multiple correlations, 3) production of matrices with squared multiple correlations on the diagonal, 4) calculation of the determinant, and 5) normalization of the rows or columns of matrices.

3.1 BACKGROUND

Correlation measures the degree of linear relationship between two variables which are paired. For example, the data might be height and weight measurements of school children, or absenteeism and productivity measurements of employees. The correlation coefficient provides a description of the relationship between children's height and weight, or of the relationship between employee absenteeism and productivity. When one variable increases, the second may also increase — as with height and weight, or when one variable decreases, the second may increase — as with absenteeism and productivity. Both of these relationships may have high correlations. Height and weight will almost always have a positive relationship, while absenteeism and productivity will have a nega-

tive or inverse relationship. Employee height and productivity would probably have no correlation. Note that while correlation is indicative of a relationship, it is not necessarily a causal one.

Figure 3.1 Scatter Plot Showing Positive Correlation



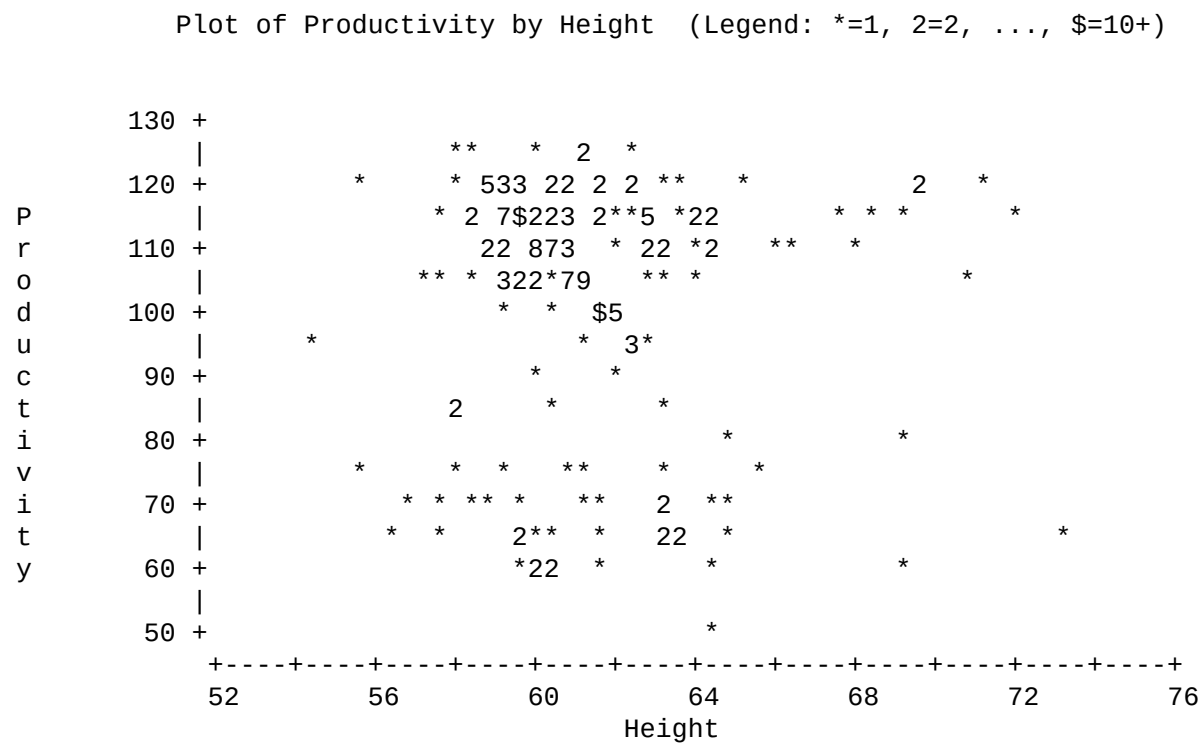
The value of the correlation coefficient, also called the product-moment correlation, “Pearson’s r” or just “r”, ranges from -1.0, a perfect negative relation (the first variable decreases linearly as the second one increases), to +1.0, a perfect positive relation (the first variable increases linearly as the second one increases). An r value of 0 (zero) describes either the absence of a relation — the first variable is independent of the second one or has only a random relation to it, or the possibility of a nonlinear relationship.

3.2 Scatter Plots

Scatter plots tell much about the relationship between variables. For example, Figure 3.1 shows a scatter plot of hypothetical data for height and weight measurements for school children. There is a relatively strong positive correlation between height and weight, as indicated by the illustrated linear relationship in the positive direction. Figure 3.2 shows a scatter plot for hypothetical data for employee height and productivity. As would be expected, there does not appear to be any systematic relationship among the data. Scatter plots also show negative relationships — a linear relationship in the negative direction, and various curvilinear relationships.

Scatter plots may be obtained by using the PLOT command. It is often useful to plot data and obtain a visual idea of any relationship before calculating a correlation coefficient or attempting to fit a line to the data.

Figure 3.2 Scatter Plot Showing Lack of Relationship



3.3 Prediction and Variance

One way of understanding correlation is in terms of prediction — correlation makes prediction possible. Often, in addition to a description of a relationship, an experimenter may want to predict one variable from knowledge of the other variable. A linear regression equation (a straight line) describes a relationship more completely and permits prediction. Linear regression usually uses the method of "least-squares" to fit the best possible line to pairs of measurements. (This is discussed further in the chapter on REGRESSION.*)* The correlation coefficient tells how well the regression line fits the data; it is a measure of the closeness of fit of this line. The sign of r, either + or - , is the slope direction of the line. The slope is positive when r is positive, and negative when r is negative. An r value close to zero means that a straight line is a poor fit to the data, though a curvilinear relationship may exist.

Correlation may also be understood in terms of variance: the correlation coefficient describes the extent to which the variation in one variable is linked to the variation in the other. This is referred to as "concomitant variation."

The variable being predicted has a variance, which we can speak of as the observed variance. If all pairs of observations fall exactly on the regression line, prediction would be perfect and the correlation coefficient would be either +1 or -1. Thus, the observed variation of the predicted variable would equal the variation of the predictor variable (the variable being used to predict the other). We would say that all of the observed variation in the predicted variable is "explained" by the variation in the predictor variable. Conversely, if the correlation coefficient were 0, prediction would be no better than guessing. None of the observed variation in the predicted variable would be explained by the variation in the predictor variable — the concomitant variation or the explained variation would be 0.

Almost all correlation coefficients are between these extremes of 0 and -1 or +1. Thus, only a portion of the observed variation in the predicted variable is explained by the variation in the predictor variable, and the remaining portion is “unexplained”. The square of the correlation coefficient (R squared) is the ratio of the explained variance to the observed variance. (Observed variance is sometimes called original or total variance.)

For example, suppose the correlation coefficient is .7. Then R squared is .49, and one can say that 49 percent of the observed variation in one variable is explained by the other variable. The remainder of the variation, 51 percent, is unexplained. Even with a correlation of .9, the unexplained variance is still 19 percent (1 - .81) of the observed variation. Unexplained variance is associated with error.

3.4 Underlying Assumptions

The main assumptions underlying the interpretation of the correlation coefficient are:

1. a linear relationship, and not a curvilinear one, between the two variables, and
2. similarly shaped (although not necessarily normal) distributions for the two variables.

Figure 3.3 CORRELATE Command

```
CORRELATE solar $

-------Correlate completed-------
Input file solar has 64 cases
and 4 numeric variables.
There was no missing data.

..............
```

variable	statistic	process	treatment	efficiency	fill factor
process	cor	1.	-0.03983	-0.05601	-0.01308
	n	64.	64.	64.	64.
	sig(2)	0.	0.75467	0.66022	0.9183
treatment	cor	-0.03983	1.	0.66048	0.69663
	n	64.	64.	64.	64.
	sig(2)	0.75467	0.	0.	0.
efficiency	cor	-0.05601	0.66048	1.	0.9927
	n	64.	64.	64.	64.
	sig(2)	0.66022	0.	0.	0.
fill.factor	cor	-0.01308	0.69663	0.9927	1.
	n	64.	64.	64.	64.
	sig(2)	0.9183	0.	0.	0.

3.5 PRODUCT-MOMENT CORRELATION

The command name is CORRELATE. When it is used without an output file, the results is a rectangular printout which shows the correlation coefficient, the number of non-missing cases and the significance values for each pair of variables. Figure 3.3 shows the command and the resulting printed output. To get this same output as a system file use:

```
CORRELATE Solar, STATS Solstats $
```

The new system file Solstats will be listed immediately and will look much like the output in Figure 3.3. To create an output file without the listing add the identifier "CC". A LIST command that produces the same appearance as the file in Figure 3.3 is:

```
LIST  Solstats,  STUB Variable,  TRIM.ZEROS $
```

Figure 3.4 Producing and Printing a Correlation Matrix

```
----------The Commands--------------------------------


CORRELATE  Survey,  OUT  SurvCor  $

Correlate completed.
495 cases were read.
233 is the smallest good n for any pair of variables.
The variables are Children and Hrs.Last.Week .

BPRINT  SurvCor,  THRESHOLD .2  $


----------The BPRINT Output-------------------------
```

POSITION		1	2	3	4	5	6	7	8	9
	LABEL					Marital Status	Work Status	Children	Siblings	Hrs Last Week
1	Sex	100					32			-25
2	Age		100	-38				29	24	
3	Education		-38	100	-47		-20		-38	
4	Occupation			-47	100				22	
5	Marital.Status					100		-22		
6	Work.Status	32		-20			100			-59
7	Children		29			-22		100		
8	Siblings		24	-38	22				100	
9	Hrs.Last.Week	-25					-59			100

The OUT identifier provides a name for an output matrix (file) which contains just the correlation coefficients:

```
CORRELATE  Survey,  OUT SurvCor  $
```

When OUT is used, the resulting P-STAT file contains only the correlation coefficients. This has the advantage that the new P-STAT system file can be used as input to the REGRESSION command or to a factor analysis. The disadvantage is that the information about the significance valises is not part of the CORRELATE command and requires an extra step. Figure 3.4 illustrates the use of CORRELATE with the OUT identifier to provide a name for the output matrix of correlation coefficients. (The file name must be a new name — one that does not yet exist in this run.)

There are several other optional output files that may be created by the CORRELATE command. NMAT is used to request a matrix that shows the *number of cases* used in computing the correlation coefficient for each pair of variables. This is necessary if the COR.SIG command will be used to compute significance values for the correlation coefficients, and if there are missing data. COV is used to request a the *variance/covariance matrix* — this file contains the variances of each of the variables on the diagonals and the covariances of each of the pairs of variables on the off-diagonals. (Covariances are the products of the standard deviations of each variable and r. They measure how the values of each pair of variables vary jointly from their respective means.) CROSS is used to request a file of *cross products*. (Cross products are the summations of the products of each pair of variables.) DES is used to request a description file of summary statistics. Any or all of these output files may be produced in a single call to the CORRELATE command. However, producing the description file requires a separate pass through the input data.

The report in Figure 3.4 states that 495 cases were read and that 233 was the smallest “good N” (number of non-missing pairs of values) for any pair of variables. The CORRELATE command assumes that there may be missing data and calculates each correlation coefficient on the maximum number of data pairs available for any pair of variables. In Figure 3.4, the output file SurvCor is printed using BPRINT (mnemonic for Blank Print), a print program specifically designed to print correlation matrices attractively. Each value is multiplied by 100 and printed without decimal places.

The identifier THRESHOLD defines a number indicating the lower limit of coefficients to be printed. Any absolute values below this threshold are replaced by blanks. Because the argument for THRESHOLD in Figure 3.4 is .2, the printing of any coefficient with an absolute value less than .2 is suppressed. Thus, the larger coefficients stand out in the printout. The identifier DOTS may be used in the BPRINT command to request that dots replace the blanks when values are below the specified threshold . Unless the general identifier LINES is used, 18 columns and 50 rows are printed per page.

3.6 Missing Value Options

If the results of the correlation are to be used in a regression, it is sometimes better statistically to delete any cases with missing data on any of the variables. This is called case-wise deletion. (The procedure normally assumed by the CORRELATE command is pairwise deletion — the program uses the maximum number of data pairs available to calculate the correlation coefficient for each pair of variables.) Figure 3.5 illustrates how to get case-wise deletion by using the identifier COMPLETE. This specifies that either 1) the input file has no missing data, or 2) only complete cases are to be used.

If it is known that there are no missing data, or that only complete cases are to be used, the identifier COMPLETE should be used to save computer time and space. If CORRELATE knows that the data file is complete (either because it was that way initially or because incomplete cases have been dropped), it does not have to compute and keep track of the good N’s (number of non-missing pairs of values) and other information for each coefficient. This means that a complete data correlation will be both faster and less expensive to run. A further advantage is a better use of space. If there are missing data and the N’s must be computed, they must also be stored. This leaves less room for the correlation coefficients and means that fewer variables can be handled in a single pass of the data.

3.7 The Weighting Procedure and Its Effects

Weights may be used to modify a sample which theoretically should be an accurate estimate of the population, but which is in fact somewhat biased. Weights change the effective number of scores for certain cases in a sample. This may be desirable when those sample cases are not representative of their actual proportion in the population.

It is usually best to select weights such that the total number in the sample remains the same, even though the number of scores in each category changes. This is because significance tests are affected by the total number of observations, even though the correlation coefficient itself is not.

Figure 3.5 Case-wise Deletion of Variables

```
----------CORRELATE Command and Report----------------


CORRELATE  Survey,  COMPLETE,  OUT  SurvCor  $

Correlate completed.
495 cases were read.

262 cases had some missing data and were ignored.
That left 233 cases.


 ----------BPRINT Command and its Output--------------


 BPRINT  SurvCor,  THRESHOLD .15  $


  PAGE=     1,  THRESHOLD= 0.15
CORRELATIONS OF  Survey                              FILE=SurvCor

POSITION                1     2     3     4     5     6     7     8     9
          LABEL                                Marit
                                                  al  Work              Hrs
                                     Educ Occup   Sta   Sta  Chil  Sibl  Last
                      Sex   Age ation ation   tus   tus  dren  ings  Week

   1   Sex            100                                              -25
   2   Age                 100   -27                      36    16
   3   Education           -27   100   -44                      -30
   4   Occupation                -44   100                       17
   5   Marital.Status                        100         -32
   6   Work.Status                                 100               -59
   7   Children             36               -32         100
   8   Siblings             16   -30    17                      100
   9   Hrs.Last.Week   -25                           -59               100
```

For example, a sample of school children might include 40 girls and 60 boys. The actual population of school children may be 50 percent girls and 50 percent boys. Rather than delete 20 of the boys' scores, one could weight the girls' scores more heavily than the boys' scores. The programming language (PPL) is used to create a weighting variable as the file is input to CORRELATE (1 = male, 2 = female in the code for Sex in the example that follows.) The identifier WEIGHT tells the CORRELATE command to weight the scores by using the weight values in the variable Weight.Factor:

```
CORRELATE  School
  [ IF  Sex = 2 ,  T.GENERATE    Weight.Factor = 1.25,
```

```
F.SET   Weight.Factor =  .83  ],
WEIGHT  Weight.Factor,  OUT  SchCor  $
```

These weights were obtained by figuring out a number that, when multiplied by the actual N, gives the desired N. The desired N is the number that reflects the actual number that should be found in a sample of this size. 1.25 x 40 = 50, which is the number of girls that should be present in a sample of size 100 if girls are 50 percent of the actual population. Similarly, .83 x 60 = 50 (approximately). Any case with a negative, zero or missing value on the weight variable is ignored.

Weighting affects all other output files that are requested in the CORRELATE command when WEIGHT is used. The mean and the standard deviation in the description file, and the "good N" (the number of non-missing scores) will usually change, if the weights were not chosen to keep N the same. The direction of the changes depends on whether extreme values were given more or less relative weight. Changes may also occur in the correlation, covariance and cross product files.

The weighting of cases in the calculation of correlation coefficients does not affect the regression procedure. The weight variable is present in the description (DES) file, but is itself unweighted. Since the weighting procedure drops any cases with missing data (or non-positive values) of the weight variable, the weight variable's good N in the DES file is the unweighted count of cases that were actually used. REGRESSION will not be affected by the weighting procedure, since it uses the good N for the weight variable from the DES file.

Tests of the statistical significance of the correlation coefficient are affected by weighting. When weighting is used, the calculation algorithms that normally give each value a weight of 1 instead give each value the designated weight. N then becomes the sum of the weights. The validity of the weighting procedure is somewhat questionable. Increasing the total N of the sample size inflates the level of subsequent significance tests. Even when the sample size is held constant, the change in the correlation coefficient may be appreciable if the values of the weighted scores are not representative of those in the actual population. In summary, weighting should probably not be used, or used only with care, if significance tests are desired.

3.8 Size Constraints

The CORRELATE command will produce all the correlations that are requested even if more than one pass through the data file is necessary. The number of variables that can be handled in a single pass depends on: 1) the size of P-STAT that is being used and 2) the presence of complete or missing data,

3.9 Asymmetric Files

The normal output file from CORRELATE is a square matrix with each variable correlated with every other variable. Sometimes the desired result is an asymmetric matrix with selected variables correlated with the rest of the variables in the file. This can be done by: 1) using variable selection to rearrange the variables in the file so that the selected variables are first, and 2) including the ROWS identifier in CORRELATE to indicate the number of initial variables that will be correlated with the remaining variables in the file.

The following command produces an output file of correlation coefficients with 10 rows and 40 columns:

```
CORRELATE  X [ KEEP  V(46) TO V(50)  .OTHERS. ],
  ROWS 10,     OUT   XCor  $
```

The ten variables that were in positions 46 TO 50 and 1 TO 5 are correlated with the rest of the variables in the file (those that used to be in positions 6 TO 45).

3.10 SPEARMAN RANK CORRELATION

For data where no meaningful significance may be attached to the intervals between values, as in the case of children scored on friendliness, Spearman's coefficient of rank correlation may be used. Spearman's coefficient is a particular case of Pearson's product-moment correlation coefficient.

First, the actual data are replaced by their ranks, and then the data are input to CORRELATE:

```
RANK       Test,      OUT  RankTest  $
CORRELATE  RankTest,  OUT  RankCor   $
```

The values in each case of the input file are converted into ranks by the RANK command, and then the file of ranks is used as input to the CORRELATE command. (See also Kendall's coefficient of rank correlation, tau, in the nonparametric statistics chapter.)

3.11 CORRELATIONS FOR DICHOTOMOUS VARIABLES

Biserial and point biserial correlations are appropriate for correlations between dichotomous and continuous variables. However, the assumptions regarding the variables underlying the dichotomies differ.

3.12 Biserial Correlation

BISERIAL is a correlation command that may be used when one variable is continuous and the other variable is dichotomous (that is, has only two values). The variable underlying the dichotomy is assumed to be continuous and normally distributed. For example, scores of 0 and 1 might represent failing and passing a test, where the original scores of 64 and below were equated with failing, and those of 65 and above with passing. (If the variable underlying the dichotomy is inherently discrete, point biserial correlation is appropriate.)

Variable selection should be used to arrange the input file so that the continuous variables are on the left, followed by the dichotomous variables. This would produce an output file of biserial correlations named B:

```
BISERIAL  A [ KEEP  V(1) TO V(4)  .OTHERS.  V(5) TO V(7) ],
  OUT  B,  NCV 12  $
```

Required identifiers are: 1) OUT with an output file name, and 2) NCV with the number of continuous variables (which should be on the left.) The rows in the output file correspond to the continuous variables (here, 12 in number — NCV 12), and the columns represent the remaining variables in the file. The number of output rows and columns allowed depends on the size of P-STAT being used. The maximum number of rows times columns is 19,000 in the Jumbo size which is the size that is usually supplied.

If the dichotomous variables are not coded 0 and 1, then the identifier ZERO nn, where nn is the score which represents the absence of the quality that has been dichotomized, should be used. For example, if the dichotomous variable is Musical.Talent, coded 1 for untalented and 2 for talented, ZERO 1 would cause scores of 1 to be changed to 0. All other scores, 2 or otherwise, would be changed to 1. If the data are coded 0 and 1, ZERO 1 can be used to reverse the sign of the correlation.

A high biserial correlation occurs when high continuous scores are associated with the non-zero value. That is, if the dichotomous scores are 0 and 1, and ZERO 1 is not used, a high biserial correlation coefficient would occur when high scores of the continuous variable were associated with scores of 1 of the dichotomous variable.

3.13 Point Biserial Correlation

Point biserial correlation is appropriate when one variable is continuous and the other is inherently dichotomous, that is, definitely discrete. This is the case when the variable is sex — it has but two values, male and female, or when the variable is video disk ownership — it has two values, ownership and non-ownership. Point biserial is defined in terms of the product-moment correlation, and thus CORRELATE is the appropriate command to use.

It is sufficient to assign values of 0 and 1 (or 1 and 2, and so on) to the dichotomous variable, and input the data to CORRELATE. The result is a point biserial correlation coefficient, which is independent of the particular values assigned to the dichotomous variable scores. The identifier ZERO is not necessary here, and thus is not an option. However, for a positive coefficient, the high scores on the continuous variable should be associated with the higher score on the dichotomous variable. If, for example, ownership of a VCR had been coded 1 for ownership and 2 for non-ownership and ownership was correlated with income, this would reverse that coding for a positive correlation coefficient:

```
CORRELATE  File1
  [ KEEP   Income  V.C.R.Ownership )
  ( IF  V.C.R.Ownership = 1,  SET  V.C.R.Ownership = 3 ],
  OUT   PtB  $
```

Completely accurate prediction from biserial and point biserial correlations is not possible. Loss of information occurs when a continuous variable is reduced to a dichotomous variable, as in biserial correlation. Loss of information also occurs when a continuous variable is predicted from a dichotomous variable, as in both biserial and point biserial correlations.

3.14 The Phi Coefficient

The phi coefficient is applicable when both the variables in the variable pairs are discrete and dichotomous. It is assumed that the variables underlying the dichotomies are discontinuous and that the two categories of each variable can be appropriately represented by two point values. However, the phi coefficient is sometimes calculated even when the underlying variables may be continuous. It is commonly used by experimenters to find the correlation between dichotomously scored test items on psychology tests.

The phi coefficient is also a particular case of the product-moment correlation. Thus, it is sufficient to assign two values, such as 0 and 1 (any two values may be used), to represent the two categories and to input the data to CORRELATE:

```
CORRELATE  Test8  [ KEEP  Item.1  TO  Item.4 ;
  DO #J = 1, 4;
  GENERATE V(#J) ( "Coded" d ) = 0;
  ENDDO;
  DO #J = 1, 4;
  IF V(#J) = 'pass', SET V#J+4) = 1;
  ENDDO;  DROP v(1) to v(4) ],
OUT    PhiCorr  $
```

In the preceding example, four test items have been selected. They are character variables coded “PASS” or “FAIL”. Four new variables are generated equal to zero — they are named Coded.Item.1, and so on, and they are set to 1 if the original item was a “PASS”.

Phi has the same minimum and maximum values as r, -1.0 and +1.0. However, these can be attained only when the values of the variables are evenly divided into the two categories; that is, the number of “passes” equals the number of “fails,” and so on, for all variables. For uneven divisions, phi has a more restricted range; its minimum is greater than -1.0 and its maximum is less than +1.0. The range of values depends on the particular number of scores in each of the dichotomous categories.

3.15 Tetrachoric Correlation

Tetrachoric correlations are used when all the variables are dichotomous. However, the assumption is that the variables underlying the dichotomies are normally distributed. The dichotomous variables often result when the original multi-categoried or graduated variable scores have been reduced to two categories. A common usage is with 2 x 2 tables, which may be reductions of larger tables. The tetrachoric correlation is an estimate of the product-moment correlation that would have been obtained if the two variables were continuous and normally distributed. Some loss of information has occurred in the reduction of multi-categoried variables to two-categoried ones, and in the initial scoring or coding of continuous variables into discrete values.

The command name is TET. The identifier OUT provides a name for the output file of tetrachoric correlations. This command would be used to produce a file named B containing tetrachoric correlation coefficients:

```
TET  A,  OUT  B  $
```

The input data are usually composed entirely of the scores 0 and 1. Missing data are not allowed with the TET command. There is a limit on the number of variables in the input file, which depends on the P-STAT size being used. The maximum number of variables is 300

If the input data are numbers other than 0 or 1, the identifier ZERO can be used to provide a substitute value for zero. For example, if the data for a variable were entirely composed of the values 3, 4, and 5, ZERO 4 would set the value 4 to 0 (the absence of the quality or quantity involved) and 3 and 5 to 1 (its presence).

The identifier CROSS requests an output cross-count matrix and provides a name for it. CTET is an optional identifier which requests an output file of cleaned up tetrachoric coefficients. Any variables with off-diagonal correlations of +1 or -1 are dropped from the CTET file. Any variable with too large a percentage of zeros (see the SPLIT identifier in the next paragraph) will also be dropped from the CTET file. The correlations in the CTET file will be the same as those in the TET file except that some may have been dropped. An output file cleaned in this way is generally better able to support a factor analysis than the uncleaned version.

The magnitude of error in a tetrachoric correlation increases with the extremeness of the dichotomies. Thus, if a file has variables which are almost entirely zero, the correlations involving those variables will not be accurate. The identifier SPLIT specifies the split-test value. Any variables whose percentage of zeros is above this value will be dropped from the input file. TET has an assumed SPLIT test value of .95 . This means that any variable with a percentage of zeros greater than 95 will be deleted from the CTET output file.

3.16 SIGNIFICANCE OF THE CORRELATION COEFFICIENT

It is often important to find out whether the correlation coefficients are significant and at what level. The command COR.SIG can be used to obtain the two-tailed significance levels of a correlation matrix. The primary input is a correlation file, which may be a symmetrical or an asymmetrical matrix.

A typical COR.SIG command, with its identifiers, is:

```
COR.SIG  Corr,  NMAT  NInput,  OUTCOR   Corr2,
  LEVEL .01,  OUTSIG  Sig,     OUTSIG1  OneMinus  $
```

NMAT is required if the original data were incomplete, and its argument is the name of the matrix of "good N's" — the number of non-missing pairs of values used in calculating the correlation coefficient for that pair of variables.

Figure 3.6 illustrates the use of COR.SIG. The first step is to do the correlation and produce both a file of correlations (using OUT) and a file of the numbers of good cases (using NMAT). The required input to COR.SIG is the correlation file and either the NMAT matrix or the identifier N and the number of good cases. If the data had been complete, the identifier NMAT and its argument SurvGood could have been omitted and replaced by N nn, where nn was the number of cases in the input file. This number is used in the calculation of the significance values. The default level of significance is .05. This can be changed by using LEVEL with some other value, such as LEVEL .01.

There are three optional output files. OUTCOR produces a file the same shape as the input correlation file. Any correlations that are significant at the designated level remain unchanged. Any correlations that are below the significance threshold are converted to zeros. In Figure 3.7, BPRINT is used with a threshold of .0001 so that any values in C that are below .0001 are replaced by blanks.

OUTSIG produces a file of the significance levels for each input correlation. OUTSIG1 produces a file in which each value is 1 minus the OUTSIG values. BPRINT can then be used for printing either file of significance levels in an attractive format.

Figure 3.6 Significance Levels of a Correlation

```
----------The Commands-------------------------------

CORRELATE  Survey,  OUT  SurvCor,  NMAT  SurvGood  $

Correlate completed.
495 cases were read.
233 is the smallest good n for any pair of variables.
The variables are Children and Hrs.Last.Week .

COR.SIG   SurvCor,  NMAT  SurvGood,  OUTCOR  C,  OUTSIG  S,
  LEVEL .01,   OUTSIG1 S1  $

----------BPRINT and its Printout-----------------

BPRINT  C,  THRESHOLD .0001  $

  PAGE=     1,  THRESHOLD= 0.0001
SIGNIFICANT CORRELATIONS FROM FILE  SurvCor           FILE=C
```

POSITION		1	2	3	4	5	6	7	8	9
	LABEL					Marital Status	Work Status	Children	Siblings	Hrs Last Week
		Sex	Age	Education	Occupation					
1	Sex	100					32	14		-25
2	Age		100	-38			17	29	24	
3	Education		-38	100	-47		-20	-16	-38	
4	Occupation			-47	100		13	13	22	
5	Marital.Status					100		-22		
6	Work.Status	32	17	-20	13		100	12	14	-59
7	Children	14	29	-16	13	-22	12	100	17	
8	Siblings		24	-38	22		14	17	100	

SUMMARY

CORRELATE

```
CORRELATE  File1 $
CORRELATE  File1,  OUT  File1Cor  $
```

The CORRELATE command calculates Pearson product-moment correlation coefficients for each pair of variables in the input file. The variables should be continuous. Ranks of data values may be input to CORRELATE for Spearman coefficients. Data values, where one variable is continuous and the second is inherently dichotomous, may be input to CORRELATE for point biserial correlation coefficients. The dichotomous variable should have only two integer values (such as 0 and 1).

Required:

CORRELATE fn

specifies the name of the required input file, for which correlation coefficients are desired. If no output file is requested the correlation coefficients are printed with the good N's and significance levels.

Optional Identifiers:

COMPLETE

specifies that there are no missing data values, or that only those cases which do not contain missing data be used.

COV fn

requests an output file of variances and covariances, and provides a name for it. This variance/covariance matrix contains the variances of each of the variables on the diagonals and the covariances of each of the pairs of variables on the off-diagonals.

CROSS fn

requests an output cross product matrix, and provides a name for it.

DES fn

requests an output description file and provides a name for it. This option requires a separate pass through the input data file.

MISSING

indicates a file with missing data. This is assumed when neither MISSING nor COMPLETE is used.

NMAT fn

requests an output matrix (file) of the "good N" for each correlation coefficient and provides a name for it. This matrix gives the number of non-missing pairs of data values used in calculating the correlation coefficient for that pair of variables. This is required when OUTSIG is used to calculate the significance values of the correlation coefficients, and there are missing data.

NO LIST

prevents the automatic listing of the P-STAT system file produced with the STATS identifier.

OUT nn

provides a name for the output matrix (file) of correlation coefficients. COR is a synonym for OUT.

ROWS nn

gives the number of output rows if the correlation matrix is not square. Each cross product is based only on the N available to it.

STATS fn

produces a P-STAT system file which contains the correlation coefficient, the count of cases involved and the significance for each of the cells in the matrix. This is usually listed automatically unless the identifier NO LIST is used.

WEIGHT vn

gives the name or position of a variable to be used for weighting the actual count of data values.

BISERIAL

```
BISERIAL    File1 [ KEEP  V(6)  TO  V(15)  .OTHERS. ]
  OUT  File1B,  NCV 10  $
```

The BISERIAL command calculates correlation coefficients appropriate for continuous data correlated with dichotomous data in which the values underlying the dichotomy are assumed to be continuous. The continuous variables should precede the dichotomous variables. A variable selection phrase may be used to reposition the continuous variables so that they are on the left side of the input file.

Required:

BISERIAL fn

gives the name of the required input file.

Required Identifiers:

NCV nn

gives the number of continuous variables. These must be arranged so that they are on the left side of the input file (possibly by using a variable selection phrase).

OUT fn

provides a name for the output file of biserial correlations.

Optional Identifiers:

ZERO nn

specifies a substitute value for zero. This is not necessary if the data are in 0, 1 form, unless you wish to reverse the meanings of 0 and 1.

BPRINT

```
BPRINT  File1Cor,  THRESHOLD  .2  $
```

The BPRINT ("blank print") command prints a correlation matrix of coefficients or significance values in a more readable format. The coefficients are multiplied by 100 and printed as integers. Values below the specified threshold are blanked out. Unless the general identifier LINES is used, 18 columns and 50 rows are printed per page. Missing data are printed as -, -- and --- (Missing1, Missing2 and Missing3).

Required:

BPRINT fn

provides the name of the required input file.

Optional Identifiers:

DOTS

requests that dots replace the blanks that are present when values are below the requested threshold. This is sometimes useful when the matrix is rather sparse and is also printed on unlined paper.

LOWER

requests that only the lower triangle of the correlation matrix print.

THRESHOLD nn

gives a number which is the lower limit of coefficients which will be printed. If a threshold value such as .2 is given, any absolute value below that threshold will be replaced by a blank. This permits significant values to stand out, even in a large correlation matrix.

UPPER

requests that only the upper triangle of the correlation matrix print.

COR.SIG

```
COR.SIG  File1Cor,  N  355,  OUTSIG  File1Sig  $
```

The COR.SIG command calculates two-tailed significance levels associated with the coefficients in a correlation matrix. Either NMAT or N is required to provide the number of data pairs used in calculating the correlation coefficients.

Required:

COR.SIG fn

provides the name of the input matrix of correlation coefficients for which significance values are desired.

Required Identifiers:

LEVEL nn

specifies a desired level of significance. The value .05 is assumed when none is specified.

N nn

gives the number of cases in the file (the number of pairs of values used in calculating the correlation coefficients) if there were no missing data. The calculated significance values will be based on this number. Either the identifier NMAT or N is required.

NMAT fn

provides the name of the matrix which contains the "good N" — the number of non-missing pairs of values used in calculating the correlation coefficient for that pair of variables. This is required if the original file contained missing data, so that the calculated significance values take into account the number of data values from which the correlation coefficients were computed. Either this input file (requested when the correlation coefficients are calculated) or the identifier N is required.

Optional Identifiers:

OUTCOR fn

provides a name for the output matrix of significant correlations. Values that are not significant are converted to zero.

OUTSIG fn

provides a name for the output file of significance values.

OUTSIG1 fn

provides a name for an output file that has values of 1 minus the significance values. This provides for the subsequent use of BPRINT for an attractive listing of the significance levels.

TET

```
TET  File1,  OUT  File1Tet  $
```

The TET command produces tetrachoric correlation coefficients. It is appropriate when both the variables are dichotomous — a 2-by-2 table type of classification, for example. The data values should be either 0 or 1. Missing data are not allowed with the TET command.

Required:

TET fn

provides the name of the required input file. It must be composed of non-missing dichotomous data.

Optional Identifiers:

CTET fn

requests an output file of cleaned up tetrachoric correlations and provides a name for it. Variables with off-diagonal correlations of +1 or -1 are dropped from the input file, as are variables whose percentage of zeros are above the split-test value (See SPLIT). These corrected values are generally better able to support a factor analysis.

CROSS fn

requests an output cross-count matrix, and provides a name for it.

OUT fn

provides a name for the output file of tetrachoric correlations.

SPLIT nn

specifies the split-test value. The value .95 is the assumed split-test value when none is specified. Any variables whose percentage of zeros is above this value will be dropped from the input file.

ZERO nn

specifies a substitute value for zero. This is not needed if the input data are in 0, 1 form.

4
REGRESSION: Linear Relationships

Regression analysis fits a linear equation to observed values of multiple independent variables and a dependent variable. Equations may also be fit to time series data using *polynomial distributed lag* models and to all possible *subsets* of independent variables. The independent variables may be transformed to achieve linearity prior to the regression analysis using the TRANSFORM identifier or the P-STAT programming language. (Thus, an equation may be fit to a seemingly nonlinear relationship between X and Y, when a linear relationship exists between the log, inverse or square of X and Y.) The regression is a *forward stepping* procedure, unless options force the inclusion of independent variables.

The input to the REGRESSION command may be the data values for the independent and dependent variables, or a correlation matrix and a description file summarizing those values. The dependent and independent variables are specified at the *subcommand* level. The printed output is a step-by-step or summary report. An output file containing the residuals, the predicted values and the original input values may be requested. Other optional output files contain summary statistics, the regression coefficients and the standardized regression coefficients.

The RESIDUALS command computes residuals and predicted values from new data, using the coefficients calculated by the REGRESSION command on similar data. The PRE.POST command does a regression analysis and calculates residuals when the variables are before and after measures. The DURWAT command produces Durbin-Watson values to test for autocorrelation of the residuals from regression analysis. The POLY.FIT command does a polynomial regression of the requested degree, using powers of the independent variable to predict the dependent variable.

4.1 BACKGROUND

Regression analysis considers the relationship of a dependent variable (Y) to one (X) or several (X_1, X_2, X_3, ...) independent variables. The regression procedure fits a linear equation to observed values of Y and X. The general form of the equation in which there is one independent or predictor variable (X) and one dependent or response variable (Y) is the equation of a line:

$$Y = \beta X + Constant$$

β represents the slope of the line. It is also called the coefficient or the constant of proportionality. It describes the proportional relationship between X and Y. The constant in the equation is the place where the line crosses the Y-axis; it is the Y-intercept. t is a sample estimate of the actual population value. The constant makes adjustments for differences between the means of X and Y.

1. Regression has many uses. It permits you to:
2. determine if there is a relationship between X and Y,
3. predict the most likely value of Y for given values of X, and
4. estimate the change in Y that will accompany a change in X.

For example, as children grow taller they generally also gain weight. Weight is the dependent variable that changes when changes occur in the independent (predictor) variable, height. Regression analysis would confirm

that this relationship does exist, permit prediction of the most likely weights for given heights (forecasting), and estimate the change in weight that accompanies a change in height.

When the X and Y variables are before and after (pre and post) measurements, the PRE.POST command should be used. (See PRE.POST in the summary section at the end of this chapter.)

4.2 Multiple Regression

There are few experimental or real-life situations where the dependent variable is a direct function of only one independent variable. With regard to the example of height and weight, there are other variables such as age, bone structure, diet and metabolic rate that also contribute to weight. The statistical technique used to analyze the relationships and measure the contributions of a number of independent variables to a dependent variable is multiple regression analysis.

In a multiple linear regression, all the independent variables are entered into the equation at once. In a multiple *stepwise* regression, the variables are either added one at a time (forward stepping), or they are all included at the start and then selectively removed (backward stepping). The goal of a stepwise regression is to include all the independent variables that contribute significantly to the dependent variable and to exclude those variables that have little additional effect on the dependent variable.

A stepwise regression is both more selective and more robust than a non-stepwise regression. If two of the independent variables are very highly correlated, that is, they measure the same thing, only one is likely to be included in the stepwise procedure. When highly correlated variables are forced into the equation in the non-stepping procedure, one of them will contribute very little to the equation and may even cause the program to stop with an error message.

Some procedures that may be useful in conjunction with more advanced correlation and multivariate regression analysis are as follows:

1. inversion of matrices,
2. production of matrices of partial and multiple correlations,
3. production of matrices with squared multiple correlations on the diagonal,
1. calculation of the determinant, and
2. normalization of the rows or columns of matrices.

4.3 The REGRESSION Command

The P-STAT REGRESSION command uses a forward stepping algorithm. Required input files are either the P-STAT system file of data values for the independent and dependent variables, or both a correlation matrix and a description file based on those values.

The identifier NOSTEP can be used to prevent stepwise regression and to force all the variables into the equation as long as computational accuracy can be maintained. Other options that control the variables included in the regression equation are also available. The REGRESSION program is in double precision for all supported computers. The maximum number of variables that can be handled in a single regression is 150.

Figure 4.1 illustrates regression using a P-STAT system file containing data values for the independent and dependent variables as input.[1] The argument for REGRESSION is the name of the file containing the input data. The semi-colon (;) indicates that subcommand information follows. The DEPENDENT subcommand gives the name of the dependent variable. The other variables in the file are assumed to be independent variables. The output shown is the fourth and final step of the regression analysis. Figure 4.2 shows the final summary and the regression equation.

---

[1] The data for this run is a subset of the Werner blood chemistry data as adapted in Dixon and Brown, *BMDP Biomedical Computer Programs P-Series*, University of California Press, Berkeley, 1977, p. 73.

Figure 4.1 REGRESSION Command and the Fourth Step

```
      REGRESSION  BCD ;

      DEPENDENT  Cholesterol  $


                  -----PARTIAL OUTPUT-----
                  -----THE FOURTH STEP-----

 Step 4 on dependent variable Cholesterol:  Entering variable Birth.Pill.

 Multiple R                    .499535
 Multiple R squared            .249535
 Adjusted R squared            .217937
 Std. Error of Est.          40.125426
 Constant                   -37.421691


 Analysis of                     Sum of            Mean          F     Prob.
    Variance      DF           Squares          Square       Ratio   Level
 Regression        4      50858.624752     12714.656188      7.8971   .0000
 Residual         95     152954.735248      1610.049845

 Adj. Total       99     203813.360000


                    Variables in Equation

         Variable    coefficient     Stand Coef      Std Error    F-delete

              Age       1.460117       .3272909       .4072015     12.8575
        Uric.Acid       8.116034       .1965715       3.941875      4.2392
          Calcium       16.94369       .1628916       9.783185      2.9995
        Birth.Pill       11.25154       .1243891       8.042218      1.9574


                    Variables NOT in Equation

         Variable   Partial   Tolerance     F-enter

          Albumin     .0532    .8571028       .2669
           Weight    -.0062    .8488726       .0036
           Height    -.0708    .9277863       .4733

 None of the remaining variables can be entered.
```

Figure 4.2 Regression Final Summary

```
************************************************************************
    File is BCD

Final summary of Regression on dependent Variable Cholesterol:

Multiple R                    .499535
Multiple R squared            .249535
Adjusted R squared            .217937
Std. Error of Est.          40.125426
Constant                   -37.421691


Analysis of                     Sum of            Mean          F     Prob.
   Variance      DF           Squares          Square       Ratio   Level
Regression        4      50858.624752     12714.656188      7.8971   .0000
Residual         95     152954.735248      1610.049845

Adj. Total       99     203813.360000


  S  Num   Mult   Mult  Change         Variable            B,        Stand
  T vars      R      R  in Rsq          entered           raw        error
  E  now           Sq.                (* shows         coef-           of
  P   in                              deleted)       ficient            B

  1    1  .3893  .1515   .1515             Age      1.460117     .4072015
  2    2  .4588  .2105   .0589        Uric.Acid      8.116034     3.941875
  3    3  .4838  .2341   .0236          Calcium      16.94369     9.783185
  4    4  .4995  .2495   .0155       Birth.Pill      11.25154     8.042218


  S         Variable         BETA,     Final    F when    Simple   Partial
  T          entered        Stand.         F   entered      cor.   cor. in
  E         (* shows         coef-        to        or      with     final
  P         deleted)       ficient    delete   deleted      dep.      step

  1              Age      .3272909   12.8575   17.5037     .3893     .3453
  2        Uric.Acid      .1965715    4.2392    7.2376     .3211     .2067
  3          Calcium      .1628916    2.9995    2.9603     .2698     .1750
  4       Birth.Pill      .1243891    1.9574    1.9574     .1211     .1421


Regression Equation:

Cholesterol (pred)  =  1.46012 Age  +  8.11603 Uric.Acid
                    +  16.9437 Calcium   +  11.2515 Birth.Pill
                    +  -37.4217
```

If many different regression analyses are to be computed from the same input data, it may be more efficient to correlate the data and produce a matrix of correlation coefficients and a description file. These files may then be used as input for a series of regressions. In addition, the correlation and description files are available for other procedures such as factor analysis.

4.4 Interpreting the REGRESSION Output

The regression output includes a report of each step in the step-wise regression (unless NOSTEP has been specified) and a final summary. The summary, shown in Figure 4.2, provides the following statistics:

1. **Multiple R**
   is the multiple Pearson correlation coefficient that describes the goodness of the fit of the regression line to the data. It is a measure of association between the dependent variable (Y) and an optimal combination of independent variables (X_1, X_2, and so on).

2. **Multiple R squared**
   is the squared multiple correlation coefficient. It measures the strength of the linear relationship between the independent variables and the dependent variable by giving the proportion of variance in the dependent variable that is explained by the independent variables.

3. **Adjusted R squared**
   is the squared multiple correlation coefficient adjusted to reflect the sample size and the number of independent variables in the regression equation. It is sometimes referred to as the "shrunken" R squared, and the magnitude of shrinkage is larger for small values of R squared and for large numbers of independent variables relative to the sample size. The adjusted R squared is a better estimate of the population value of R squared.

4. **Std. Error of Est**
   is the standard error of estimating the mean value of Y for a particular value of X using the regression equation.

5. **Constant**
   is the Y-intercept of the regression line. (It is zero when a regression line through the origin is requested.)

6. **Analysis of Variance**
   (ANOVA) tests the hypothesis that all of the coefficients in the regression equation are equal to zero, that is, that the independent variables are not related to the dependent variable.

7. **Sum of Squares**
   is the variation among the dependent values partitioned into two parts — the variation among the regression estimates of the dependent variable, and the residual variation not accounted for by the regression. (The sum of squares is a sequential or Type I SS. In the step summaries, the SS gives the total SS thus far. The difference between the SS in sequential steps is the reduction in the residual SS that the variable entered into the regression equation in that step is responsible for.)

8. **DF**
   are the degrees of freedom, which are similarly partitioned into the contributions of the regression estimates and of the residual variation.

9. **Mean Square**
   is the sum of squares divided by its degrees of freedom. It is a measure of average variability — it is a variance estimate.

10. **F Ratio**
    is the regression mean square divided by the residual mean square. It is a test of significance of the relation of the means of the dependent variable to the independent variables.

11. **Prob. Level**
    is the probability of achieving the given F value by chance.

12. **STEP**
    lists the steps in the regression (unless a non-stepwise regression was requested). The independent variables are shown in the order in which they were entered in the regression equation. The multiple R, the multiple R squared, and the change in R squared are shown for the variables in the equation at that particular step. Deleted variables are indicated with an asterisk.

13. **B, raw coefficient**
    is the partial regression coefficient for the independent variable in question. It is the average amount of change expected in this independent variable, given any values for the other independent variables.

14. **Stand error of B**
    is the error associated with estimating B for the sample. It is often used in calculating confidence intervals about B.

15. **BETA, stand. coefficient**
    is the standardized partial regression coefficient. It is B converted to standard units to make comparison of the strengths of the relationships between the different independent variables and the dependent variable easier by removing the differences in variation of the independent variables.

16. **Final F to delete**
    is the F value required to delete the variable once it is in the regression equation. It is the F value associated with a minimum probability level. (This is equivalent to a t test testing whether the regression coefficient differs significantly from zero.)

17. **F when entered or deleted**
    is the F value of the independent variable when it was either entered into or deleted from the regression equation.

18. **Simple cor. with dep**
    is the correlation of the independent variable with the dependent variable.

19. **Partial cor. in final step**
    is the correlation of the independent variable with the dependent variable, adjusted for its correlation with other independent variables.

20. **Regression Equation**
    is the sample regression that is used to predict the mean value of the dependent variable for given values of the independent variables. The coefficients are slopes and the constant (if the regression was not through the origin) is the Y-intercept.

4.5 OPTIONS

The REGRESSION command requires an input P-STAT system file of data, or both an input correlation matrix and a description file. It also requires the DEPENDENT subcommand indicating the dependent variable. The INDEPENDENT subcommand is optional. Numerous other options, specified either as identifiers in the REGRESSION command or subcommands at the subcommand level, permit:

1. regression analyses for subgroups,
2. output files of predicted and residual values, and coefficients,
3. analyses of complete cases or cases with missing data,
4. regression through the origin,
5. weighting,

6. analyses of time-related measurements, and
7. fitting equations to all-possible subsets of independent variables.

4.6 Specifying Independent and Dependent Variables

The REGRESSION *subcommand* language specifies which variables are the dependent ones and which are the independent ones. The REGRESSION command ends with a semicolon indicating that subcommands follow. The DEPENDENT and INDEPENDENT subcommands specify exactly which variables are the dependent and independent variables:

```
REGRESSION  Bcd ;
  DEPENDENT    Cholesterol,
  INDEPENDENT  Age  Height  Weight ;
```

In this example, a regression is done with the variable Cholesterol used as the dependent variable, The independent variables are Age, Height and Weight.

The independent variables need not be specified. When that is the case, *all* numeric variables not used as the dependent variable, BY variables, or the WEIGHT variable are used as independent variables. Subcommands requesting additional regression analyses or analyses with other options may follow. A $ ends the REGRESSION command.

The subcommand language permits abbreviations as long as there is no ambiguity. In addition, variables may be specified with a range:

```
DEP  Chol,  IND  Age  TO  Uric  Hemo ;
```

In this example, the variables from the one named Age through the one named Uric.Acid, and the variable named Hemoglobin, are used as the independent variables.

Regressions may be repeated with changes either in the pool of variables to be used or in the options. This is especially useful in interactive usage, where an initial regression analysis may suggest changes for subsequent analyses. AGAIN repeats an analysis, possibly with modifications:

```
DEPENDENT  Cholesterol ;
AGAIN,  NOSTEP ;
```

Here, the DEPENDENT subcommand requests a regression analysis with Cholesterol as the dependent variable. The AGAIN subcommand requests that the same analysis be repeated, but without stepping.

AGAIN may also be used to request the previous analysis with a change in the pool of independent variables:

```
REG  InFile ;
  DEP  Hrs,
  IND  Age  To  Children ;
  AGAIN,  IND  Age  Education  Work.Status  $
```

A new correlation matrix is computed if the new list of independent variables contains variables that are not in the original list.

A regression may be repeated with a change in the F-to-enter, the print destination, or a request for an output file:

```
REG  InFile ;
  DEP  Hrs,  SUMMARY  ;
  AGAIN,  F.ENTER  1. ;
  AGAIN,  REPORT,  PR  'PrtFile' ;
  AGAIN,  OUT  OutFile,  NO REPORT  $
```

In the first regression, only the final summary is printed. The second regression is to be done with a change in the F-to-enter (see the final section on changing the thresholds). Because SUMMARY is still in effect, the step information is not printed. In the third regression, the REPORT subcommand requests that all of the step information, as well as the summary, be printed. The PR subcommand requests that the output be directed to the external file named "PrtFile". The final regression produces no printed output, but it creates the P-STAT system file OutFile. OutFile contains all the variables in file InFile, plus the residuals and predicted values of the variable Hrs.

When new regressions are specified and all of the variables specified are included in the correlations computed for the previous regression, a new pass through the data file is not made unless a change in the options requires a new correlation matrix. For example, if the previous regression is done with missing data permitted, a new regression with the COMPLETE option requires that the correlations be recomputed (see the subsequent section on missing data). Options that require recomputation can only be specified if the input is from a system file of data values, and not from correlation and description files.

RESET restores all settings to their original values:

```
REG  InFile ;
  DEP  Hrs, IND   Age  TO  Children,  MISSING ;
  RESET,  DEP  Income  $
```

The second regression is done with COMPLETE data only and has all the numeric variables in the file (except for the dependent variable Income) as independent variables.

4.7 BY for Subgroup Analysis

If the input file of data is sorted or ordered on up to 15 numeric or character variables, a series of regressions for each subgroup can be specified in a single step. This command, using the BY *identifier*,

```
REGRESSION   Bcd,  BY  Hospital ;
  DEPENDENT  Cholesterol  $
```

produces as many analyses as there are values of Hospital. The BY variables can either be character or numeric. BY cannot be used when the input to REGRESSION is from a correlation matrix and a description file, and BY must be an identifier in the REGRESSION *command* and not a subcommand.

4.8 Optional Output Files

There are several optional output files that may be requested. An output file containing the residual and predicted values for the dependent variables and the original input data values is produced for each set of regressions when the subcommand OUT is used. The subcommand OUT requires that the original input data be available even if a correlation matrix and description file are provided.

The variables for residuals and predicted scores in the output file have the same names as the variables in the input file, except that they are prefixed with an appropriate prefix: "Res." for the residuals and "Pre." for the predicted scores. If the variable names are too long when the prefix is added, they are truncated on the right.

The COEF subcommand requests a file of the regression coefficients. The COEF file contains a column for every dependent variable and a row for every independent variable, plus a row for the constant. The STAND.COEF subcommand requests a file of standardized regression coefficients. Files of regression coefficients can only be produced for the entire file. Therefore, COEF and STAND.COEF cannot be used with BY. However, OUT may be used with BY.

Figure 4.3 illustrates a regression that produces an output file The new file is then used as input to PLOT. A number of plots are produced, although only one is shown in the figure. Typically, plots (and histograms) of the residuals and the analysis variables are made as diagnostic checks of the regression model.

The output files produced by REGRESSION may be used as input to several other commands. The file of coefficients may be input to the RESIDUALS command to compute predicted values and residuals from different related data. The residuals may be input to the DURWAT command to test for autocorrelation.

Figure 4.3 Output Files from REGRESSION

```
REGRESSION  BCD,  OUT  BCD2 ;
DEPENDENT   Cholesterol,  SUMMARY  $

PLOT  BCD2 ;
  P   Uric.acid  *  Res.Cholestrol ;
```

```
Plot of Uric.Acid by Res.Cholesterol (Legend: *=1, 2=2, ..., $=10+)

        9 +
          |             *       *             *
        8 +                             *
U         |
r       7 +                           *
i         |      *           *
c       6 +  *    *   2       *             *** **
.         |          *    *   3 *               *           *
A       5 +            * 2 *2* 22***  2 * *  *
c         |         * **    22*22*2  *   * 2*  *
i       4 +       *    **  *2  2  * 4*2*2       *
d         |             ***   23***   *         *
        3 +            *
          |                                      *
        2 +
           +----------+----------+----------+----------+----------+
          -100       -50         0          50        100        150
                                Res.Cholesterol
```

4.9 Missing Data

The REGRESSION command permits missing data values. However, COMPLETE is assumed unless MISSING is explicitly specified. When the input is from a system file of data values, COMPLETE means that only complete cases without any missing data are used in the regression analysis. Any case with missing values of any of the variables involved in that regression is omitted from the analysis.

The subcommand MISSING requests that cases with missing data (incomplete cases) be included in the analysis. The non-missing pairs of values are used in the calculations. In other words, MISSING specifies *pair-wise* deletion of missing data, whereas COMPLETE specifies *case-wise* deletion.

The treatment of missing, when the input to REGRESSION is from correlation and description files, depends on whether the input to the correlation had missing data and what options were selected at that time. The CORRELATE command assumes pair-wise deletion of missing values (MISSING), but the COMPLETE subcommand may be used to specify that only complete cases be included in the analysis.

The USE.MEAN subcommand requests that the mean score be used in place of missing values in the original input file in calculating predicted and residual values. This permits a regression analysis with missing data for independent variables, but with prediction of all dependent values.

4.10 Regression Through the Origin

A regression equation passing through the point (0,0) may be requested. This is appropriate in some experimental situations, when the physical situation implies that the regression equation should clearly yield a line or linear surface that passes through the origin. For example, in measuring the dispersion of a gas at given times, the dispersion (measured in particles per unit volume) at the start of the experiment is zero. At time zero, the gas has not been released, so dispersion is also zero.

The REGRESSION program will most always yield a regression equation with a constant or Y-intercept not exactly equal to zero. The constant in this case is associated with errors of measurement. The subcommand ORIGIN, used in the REGRESSION command, specifies the calculation of a regression equation passing through the origin.

4.11 Weighting

Weighting in REGRESSION is requested by using the WEIGHT identifier:

```
REGRESSION  Auto,  WEIGHT  W.Factor ;
```

If the input is from a correlation matrix and description file,

```
REGRESSION  Auto,  COR  AutoCor,  DES  AutoDes,  WEIGHT  W.Factor ;
```

WEIGHT must also be used when the correlations are calculated — that is, WEIGHT is used in the CORRELATE command to produce weighted correlation and description files. In the DES file, the weight variable is itself unweighted, thus preserving the actual number (N) of cases. (Since the weighting procedure drops any cases with missing data (or non-positive values) of the weight variable, the weight variable's good N in the DES file is the unweighted count of cases that were actually used.)

WEIGHT specifies the name of the weight variable, and REGRESSION uses the good N taken from that variable in the DES file. Thus, the degrees of freedom are kept consistent with the actual number of cases. (See the complete discussion of weighting in the CORRELATE chapter.)

4.12 MORE COMPLICATED REGRESSION MODELS

Several other procedures are available in the REGRESSION command for analyses of more complex regression models, such as nonlinear but intrinsically linear relationships, time-related measurements, time series data and subsets of independent variables. Also, see the FORECAST command for additional procedures for time series data with and without trend.

4.13 Transforming Variables

Variables are typically transformed either to stabilize their variance or to "linearize" the relationship between them when it is nonlinear, but intrinsically linear — linear in the way the coefficients (βs) enter into the regression equation. The dependent and independent variables may be transformed using either the TRANSFORM identifier or the PPL language.

TRANSFORM requests log transformations of *all* the variables in the input file, except any BY and WEIGHT variables. TRANSFORM is followed by LOG or LOG10 to request natural logs to the base *e* or logs to the base 10, respectively:

```
REGRESSION  File5,  TRANSFORM  LOG ;
```

This log transformation of all the variables linearizes this simple regression equation:

$$Y = \alpha X^{\beta}$$

(as well as the multiple regression form of it that contains X_1, X_2, ... , X_N). The equation fit by the REGRESSION command is:

```
LOG Y  =  LOG α  +  β LOG X
```

The PPL language may be used to produce any arbitrary transformations of any variables in the input file. For example, this transformation uses PPL to set Y to its reciprocal:

```
REGRESSION  Study  [ SET  Y =  1 / Y ] ;
```

This often stabilizes the variance of a variable with numerous values near zero but with some large values. This could be the case if Y is the response time to complete a puzzle. Taking the reciprocal of Y changes time per puzzle to number of puzzles per unit time. See the many textbooks on regression for further information on linearizable functions and variance stabilizers.

4.14 Autoregressive and Time Trends

If there are one or more independent variables that are measurements over time, several new independent variables may be created from each one by using LAG. Thus, regression analysis of a series of time-related measurements — measurements with an *autoregressive* trend — is possible. For example, if the quantity of pollutants is measured at a given station (place) along a stream over a period of time, the values of the variable Pollutants are most likely related serially. The pollutants in the water passing the station now reflect those just passed and those soon to pass.

LAG is used in the INDEPENDENT subcommand, enclosed in parentheses:

```
REGRESSION  Stream ;
DEP  Pollutants,  IND  Station ( LAG 0 2 4 6 ) Season ;
```

This regression analysis is done with five independent variables: Station, Station.2, Station.4, Station.6 and Season. Station.2, Station.4 and Station.6 are created from the values of Station taken from previous cases. The values of Station.2 are those of Station two cases back, those of Station.4 are those of Station four cases back, and so on. LAG 0 means that the value of Station in the current case is also used as an independent variable.

Using LAG creates missing data for the cases before the lag length. Station.2 (in the preceding example) has missing data for the first two cases passed to the REGRESSION command. Since the REGRESSION command assumes COMPLETE data, all cases before the largest LAG length are omitted from the regression unless MISSING is used as a subcommand:

```
DEP  Pollutants,  MISSING,
  IND  Stream      ( LAG 2 TO 4 )
       Temperature ( LAG 1 3 7  )  Month ;
```

LAG may be used with several variables. However, the number of non-LAG variables combined with the number of LAG variables cannot be larger than the maximum number of variables permitted in REGRESSION in the size of P-STAT in use. (Constraining the number of lag terms to a relatively small number increases the preciseness of the coefficient estimates by reducing the amount of multicollinearity.)

Models with autoregressive terms generally describe short term fluctuations well. For long term trends with a constant absolute growth over time, terms incorporating powers of a time variable may be included in the regression equation. When the data are ordered and evenly spaced, without missing values, a time variable may be generated and used in the regression analysis:

```
REGRESSION  Stat56  (GEN  Time = .N.) (GEN  Time.Sq = Time ** 2) ;
  DEP  Measuremts,   IND  Time  Time.Sq  Measuremts  ( LAG 1 2 ) ;
```

In this example, the independent variables are Time, Time.Sq (Time squared), and two lags of the dependent variable (Measuremts.1 and Measuremts.2). (See the FORECAST command for additional procedures that incorporate smoothing and autoregressive and time trends.)

4.15 Polynomial Distributed Lag Models

Time series data — ordered and equally spaced measurements — may be fit using *polynomial distributed lag* (PDL) models. These models (sometimes called "Almon Lag models") are appropriate when the effects of an independent variable are distributed over time, as in econometric relationships.

The equation for a PDL model includes a constant and terms for each lag. This regression equation is for a model with lag two:

$$Y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + CZ_t + \varepsilon$$

The term (or terms) with Z is a optional covariable in the regression equation. The pattern of the lag in this model will be approximated by a second-degree polynomial. (Constraining the form of the distributed lag to a polynomial decreases the multicollinearity.) Thus, the equations for the three coefficients in this model are:

$$\beta_i = b_0 + b_1 i + b_2 i^2$$

The variable i (time) goes from 0 to 2 in this example. Substituting the equations for the β terms in the regression equation and factoring yields:

$$Y_t = b_0(X_t + X_{t-1} + X_{t-2}) + b_1(X_{t-1} + 2X_{t-2}) + b_2(X_{t-1} + 4X_{t-2}) + \varepsilon_t$$

The parameters b_0, b_1 and b_2 are estimated from this last equation using least-squares regression analysis. The βs in the initial regression equation are estimated by substitution.

PDL models are fit by using the REGRESSION command with the POLY and LAG options after the independent variable:

```
REGRESSION  K188 ;
  DEP  Consumption,  IND  Income  (POLY 2,  LAG 1 TO 2) ;
```

POLY requests a PDL model and specifies the *degree* of the model; it must follow the independent variable specification and come directly after the parenthesis. LAG specifies the *length of the lag*, which must be equal to or greater than the degree of the polynomial. The LAG specification must be a sequential range of lag periods beginning with 1. Additional independent variables may follow in the IND list, but there may be only one POLY request and it should be part of the specification of the *first* independent variable.

Figure 4.4 illustrates specifying a PDL model. The data are consumption and disposable income in billions of current dollars.[2] The REGRESSION report gives the equation with the parameter estimates (b_0, b_1 and b_2) and the equation with the coefficient estimates (β_0, β_1 and β_2). The report also includes various statistics about the analysis — Income.dl0, Income.dl1 and Income.dl2 refer to the three terms composed of lags of the independent variable to which the REGRESSION command fits the equation. The OUT, STATS, COEFS and STAND.COEF subcommands may be used to request the various output files that are normal REGRESSION options.

Figure 4.4 Polynomial Distributed Lag Model

```
REGRESSION  K188 ;

 DEP  Consumption,  IND  Income  ( POLY 2,  LAG 1 TO 2 )  $

 2 cases were dropped because of missing data.

All variables are now in.
```

---

[2] This data is from *Econometric Theory and Applications* by David A. Katz. The filename K188 refers to the data set in this book on page 188.

****

File is k188.

Final summary of Regression on dependent variable Consumption:

Multiple R	.999729
Multiple R squared	.999458
Adjusted R squared	.999341
Std. Error of Est.	3.064102
Constant	1.554711

Analysis of Variance	DF	Sum of Squares	Mean Square	F Ratio	Prob. Level
Regression	3	242192.462917	80730.820972	8598.7035	.0000
Residual	14	131.442083	9.388720		
Adj. Total	17	242323.905000			

STEP	Num vars now in	Mult R	Mult R Sq.	Change in Rsq	Variable entered (* shows deleted)	B, raw coef-ficient	Stand error of B
1	1	.9996	.9992	.9992	Income.dl0	.7150460	.1446340
2	2	.9997	.9994	.0002	Income.dl1	-1.096812	.6446477
3	3	.9997	.9995	.0001	Income.dl2	.4156654	.3192884

STEP	Variable entered (* shows deleted)	BETA, Stand. coef-ficient	Final F to delete	F when entered or deleted	Simple cor. with dep.	Partial cor. in final step
1	Income.dl0	2.172519	24.4414	18880.791	.9996	.7974
2	Income.dl1	-3.115690	2.8948	5.8861	.9991	-.4139
3	Income.dl2	1.942709	1.6948	.0000	.9989	.3286

Regression Equation with Parameter Estimates:

Consumption (pred) = .715046 Income.dl0 + -1.09681 Income.dl1
+ .415665 Income.dl2 + 1.55471

Regression Equation with Estimated Coefficients:

Consumption (pred) = .715046 Income + .0338992 Income.1
+ .184083 Income.2 + 1.55471

There should be *no* missing values of the lagged independent variable. Any missing values cause REGRESSION to stop with a fatal error. The initial *n* cases, where *n* is equal to the length of the lag, are excluded from the analysis. Using the MISSING subcommand requests that cases with missing values of any covariables be included in the analysis. The REGRESSION command uses a *non*-stepwise procedure to enter the terms composed of lags of the independent variable and any covariables into the equation.

4.16 Regression Analyses of Subsets

Subsets of independent variables that best predict the dependent variable may be identified using the ALL.POSSIBLE subcommand. ALL.POSSIBLE fits equations to *all possible subsets* of independent variables and either writes the best subsets, selected by one of three criteria, or all subsets in a P-STAT system file. The ATLEAST, ATMOST and START subcommands (discussed also in the subsequent sections) may be used to control the size and contents of the subsets of independent variables.

The ALL.POSSIBLE subcommand is typically followed by a name for the output P-STAT system file that is to contain the subsets:

```
REGRESSION  Hpd,  WEIGHT  Wt ;
  DEP  HH,  ALL.POSSIBLE  HpdAP  $
```

In this example, weighted regressions of the dependent variable HH on all possible subsets of the independent variables are requested. The single best subset or regression model of each size is written in the output system file HpdAP. When specific independent variables are not specified, as is the case in this example, the number of variables in the input system file, excluding the dependent and weight variables, is the size of largest subset of independent variables. The smallest subset is one independent variable. Figure 4.5 shows the output file from this all-possible subsets regression.

ATLEAST and ATMOST may be followed by numbers giving the sizes of the smallest and largest subsets, respectively. The START subcommand may be followed by the number of initial variables in the input file to be included in every subset. ORIGIN may be used to request regressions through the point (0,0).

The criterion for selecting the "best" subset or regression model is R^2, unless another criterion is specified. Other possibilities for selecting subsets are Mallows' C_p statistic and the adjusted R^2 (R^2 adjusted for the degrees of freedom or the number of independent variables in the subset). The TEST subcommand specifies the criterion to use in sorting and selecting the subsets:

```
DEP  Response,  ALL.POSSIBLE  S17AllP,  TEST  MALLOW  $
```

The possible arguments for TEST are RSQ, MALLOW and ADJ.RSQ. The highest values of RSQ and ADJ.RSQ and the lowest values of MALLOW are selected. (Note that the "best" subsets are best only for the particular sample in question and that, because the regression model is not specified beforehand, the regular regression statistics are considered biased.)

One subset of each size is included in the output file, unless the MAX subcommand specifies another maximum. MAX 3, for example, includes the three best subsets of each size in the output file. MAX ALL may be specified to include *all* subsets in the output file. When no output file name follows the ALL.POSSIBLE subcommand, the regression subsets are written in a *temporary* P-STAT system file. The single best subsets of each size in the temporary file are automatically listed when MAX is one, the default MAX. (The temporary file contains all subsets.)

The output system file produced when ALL.POSSIBLE is used contains the following variables: Num.Ind.Vars, R.Square, Mallows.Cp, Adjusted.Rsq, Dependent.Var and Ind.0001, Ind.0002, Ind.0003, and so on. The cases are unique subsets and, when there is more than one subset of each size, they are sorted by the default or specified criterion within each size subset. The REGRESSION command automatically generates the appropriate SORT command, exits after producing the output file, and causes the execution of the generated SORT command. (Thus, ALL.POSSIBLE is always the final REGRESSION subcommand.) The output file may be displayed using the LIST command. The step information and summary regression statistics are obtained by running REGRESSION with the variables in the best subset specified as the independent variables.

Figure 4.5 All-Possible Subsets Regression

```
REGRESSION  Hpd,  WEIGHT  Wt ;

DEP HH,  ALL.POSSIBLE  HpdAP  $

511 regressions were performed.

File HpdAP contains the best single subsets of each size from the
all-possible regression.  The previous version of file HpdAP contains
all subsets.


LIST  $


FILE HpdAP
```

Num Ind Vars	R.Square	Mallows Cp	Adjusted Rsq	Dependent Var	Ind. 0001	Ind. 0002	Ind. 0003	Ind. 0004	Ind. 0005
1	0.642214	22.14412	0.623383	HH	EL	-	-	-	-
2	0.766578	10.53785	0.740642	HH	BC	CC	-	-	-
3	0.844285	4.03624	0.816806	HH	BC	AH	CC	-	-
4	0.863113	3.97628	0.828892	HH	BC	AB	AH	CC	-
5	0.881012	4.01807	0.841349	HH	BC	AB	AH	F2	EL
6	0.888150	5.23706	0.840215	HH	BC	AB	AH	F1	F2
7	0.891644	6.85483	0.833299	HH	BC	AB	AH	SH	F1
8	0.898883	8.06289	0.831471	HH	BC	AB	AH	SH	F1
9	0.899457	10.00000	0.817195	HH	BC	AB	AH	CC	SH

Ind. 0006	Ind. 0007	Ind. 0008	Ind. 0009
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	-	-
EL	-	-	-
F2	EL	-	-
F2	EL	ST	-
F1	F2	EL	ST

4.17 THE STEPWISE PROCEDURE

The first independent variable to be entered into the equation is the one with the highest absolute correlation with the dependent variable. In Figure 4.1, it is the variable Age. When a variable has been entered in the equation, its contribution is computed (the F-to-delete). The potential contribution of each of the variables that have not been entered is also computed. If any of the variables that have been entered have weakened because of overlap with other entered variables — that is, the F-to-delete is below a threshold (usually 1.0) — that variable is deleted from the equation unless:

1. there is only one variable entered,
2. START is in use and has not been satisfied,
3. NODELETE is in use,
4. NOSTEP is in use, or
5. ATLEAST is in use and would be violated.

In these cases, the analysis of that dependent variable is finished. These subcommands are discussed in the subsequent section on controlling the variables to be entered.

If there are no variables to delete, the next step begins. The variable which has the highest partial correlation with the dependent variable is entered next if: 1) it has an F-to-enter above a threshold which is usually set to 1.5, or, if a probability-to-enter has been specified, it has a probability associated with the F value that is less than that, and 2) it has a tolerance greater than or equal to the tolerance threshold which is usually set to 0.005. The tolerance is a check that a variable is not too highly correlated with independent variables already in the equation. It can be changed by using the subcommand TOL. The F-to-enter may be changed using F.ENTER. The probability-to-enter, which overrides the F-to-enter, may be changed using the P.ENTER subcommand.

A variable which has been deleted in one step may be reentered later on. This process continues until: 1) too many steps have occurred due to enter-and-delete oscillation, 2) no entered variable can be deleted, and no remaining variable can be entered, 3) all the variables are entered, or 4) ATLEAST or ATMOST forces the end.

When the stepping has concluded, a report is printed. The report includes step information, as well as a final summary. (The summary is the entire output when REGRESSION is run with the NOSTEP option.) REPORT is assumed; SUMMARY may be used to request that step information be suppressed, and just the final summary printed. NO REPORT turns off all printed output.

4.18 Oscillation and Missing Data

P-STAT calculates correlation coefficients based on either: 1) complete data, or 2) data containing some number of missing values. The correlation of Age and Cholesterol in a complete data matrix reflects the entire sample, as does every other coefficient in that matrix. Complete data correlations seldom cause problems in a regression run.

However, missing data correlations can cause problems. If the data file has missing data, a missing data correlation of Age and Height represents only those cases which had non-missing scores on *both* Age and Height. The number of values used in computing that coefficient (the N associated with those variables) could be far less than the sample size. Each correlation coefficient could be based on a different portion of the sample.

It is reasonably safe to use a missing data correlation matrix in a regression only when: 1) the smallest N used in the correlation is quite a bit more than the number of variables, and 2) the occurrence of missing data is random. Oscillation of variables as they are entered and deleted is almost always caused by inconsistencies in the missing data correlation matrix in which different coefficients represent different samples.

4.19 Controlling the Variables To Be Entered

There are a number of options to control the number of variables which are included in the equation. NOSTEP enters the variables in *order* (the order they have in the input file), one at a time, starting with the first, until all are entered or until entering the next one would cause computational problems. ATLEAST specifies a minimum num-

ber of variables to be included in the equation. ATMOST specifies a maximum number of variables to be included.

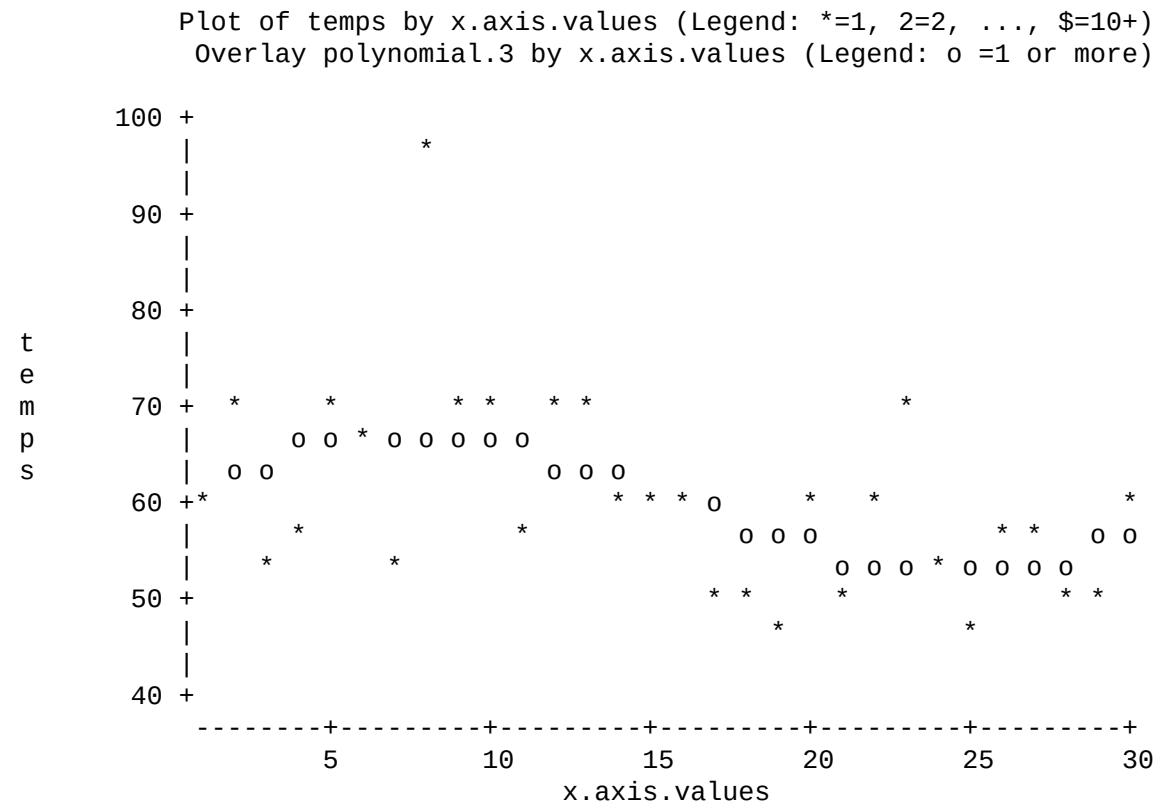
ORDER specifies that the variables are to be entered in the order in which they appear in the input file. Variable selection can be used with ORDER to ensure that variables are entered in a particular sequence. Stepping stops when the next variable to be entered has an F-to-enter that is below the threshold (or a probability associated with that F that is greater than the user-specified P.ENTER value), or when ATMOST, if it is in use, is satisfied. START specifies the number of variables from the left of the file to be entered at once, before stepping begins.

4.20 Changing the Thresholds

The various thresholds may be changed. F.DELETE changes the F-to-delete value; it is normally set to 1.0. F.ENTER changes the F-to-enter value; it is normally set to 1.5. A threshold can be specified by providing a probability rather than an F value to determine whether a variable should be included. If P.ENTER or P.DELETE are specified, the program computes the F value and the degrees of freedom, and then checks if the probability associated with the F is greater or less than the P.ENTER or P.DELETE that was specified. The variable is entered into the equation only if the probability is less than the value of P.ENTER. A variable that has already been entered into the equation is deleted only if the probability is greater than the value of P.DELETE. P.ENTER and P.DELETE override F.ENTER and F.DELETE.

TOL changes the tolerance threshold; it is normally set to .005. The tolerance of a variable is the proportion of its variance *not* accounted for by other independent variables in the equation. No variable can be entered if its tolerance is below the tolerance threshold.

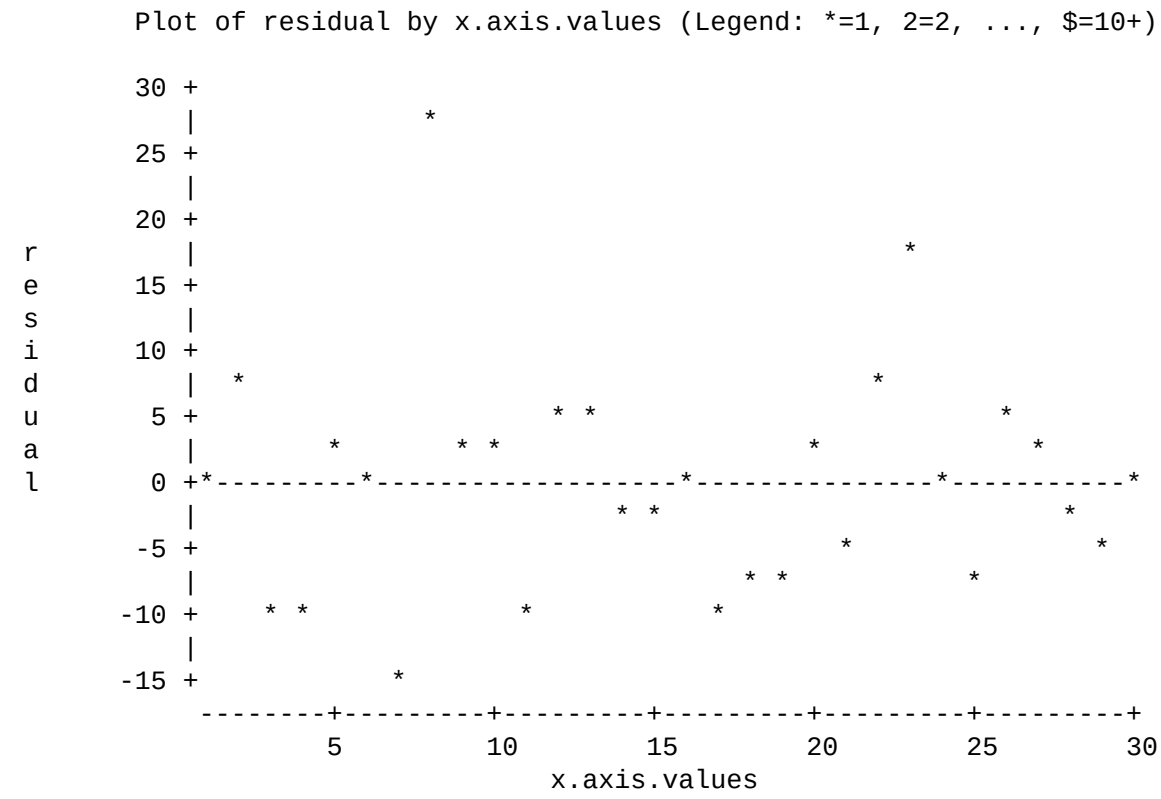
Figure 4.6 Fitting a Polynomial: Degree 3 Plot Output



4.21 POLY.FIT COMMAND

This command does a polynomial regression of the requested degree, using powers of the independent variable to predict the dependent variable. Figure 4.6 has the dependent variable by the independent variable with the polynomially fitted curve overlaying it. Figure 4.7 has residuals of the dependent variable by the independent variable.

Figure 4.7 Polynomial: Plot of the Residuals



```
POLY.FIT file, DEGREE 4, INDEPENDENT days, DEPENDENT pollen $
```

The INDEPENDENT (or IND) identifier is optional. If it is not used, values of 1, 2, 3, etc. are generated. The DEPENDENT (or DEP) identifier is required when the file has more than one variable. The DEGREE identifier is needed to specify what degree of polynomial should be used.

The OUT identifier can be used if a results file of the 2 input variables and the fit values should be kept after PLOT finishes. If OUT is not used, a WORK file is created. COEF file causes the command to write a file containing the polynomial coefficients and the intercept.

The default overlay plotting character is 'o'. CHAR 'x' or such causes that character to be used instead. NO PLOT prevents the two plots from occurring. NO FILL prevents the code from trying to expand the overlay plot with interpolated independent points. The filling is done when possible, to make the fitted curve easier to see.

The plots are created by a call to the PLOT command. In a PostScript block, the plots will automatically be printed with PostScript controls. Additional PLOT identifiers can be supplied by the USE identifier. USE is followed by a quoted string which contains additional PLOT identifiers.

```
POLY.FIT Cows, DEP Temps, DEGREE 7 $
```

PostScript output can be created directly from POLY.FIT by:

```
POLY.FIT Cows, DEP Temps, DEGREE 7,
         POSTSCRIPT,  PR cows.ps ' $
```

The printed output from this POLY.FIT command is:

```
 Degree 7 POLY.FIT on variable Temps completed.
 30 observations were processed.
 Since no X variable was given, values of 1, 2, 3, etc were used.
 The mean squared residual is 57.7418519
```

The POSTSCRIPT command provides more control over the plot size, fonts, etc. than just requesting POST-SCRTIPT through the USE identifier in POLY.FIT. The commands that would make a PostScript plot from the data in figure 4.7 are:

```
POSTSCRIPT,
   MARGINS .2 6.8 .5 1.5,
   PR cows7.ps,
   PORTRAIT $
POLY.FIT cows,  DEPENDENT, Temps, DEGREE 7 $
POSTSCRIPT.CLOSE $
```

The MARGINS were chosen so that the plots would be approximately half a page in this document. The various PostScript commands are documented in the manual "P-STAT: Plots, Graphs and PostScript Support".

REFERENCES

1. Almon, Shirley. (1965). "The Distributed Lag Between Capital Appropriations and Expenditures", *Econometrica*, 33, 178-196.

2. Chatterjee, Samprit and Price, Bertram. (1977). *Regression Analysis by Exampl*e, John Wiley & Sons, New York.

3. Draper, Norman and Smith, Harry. (1966). *Applied Regression Analysis*, John Wiley & Sons, New York.

4. Katz, David A. (1982). *Econometric Theory and Applications*, Prentice-Hall, Englewood Cliffs, New Jersey.

5. Montgomery, Douglas C. and Peck, Elizabeth A. (1982). *Introduction to Linear Regression Analysis*, John Wiley & Sons, New York.

6. Pedhazur, Elazar. (1982). *Multiple Regression in Behavioral Research*, Holt, Rinehart and Winston, New York.

7. Pindyck, Robert S. and Rubinfeld, Daniel L. (1981). *Econometric Models and Economic Forecasts*, McGraw-Hill Book Company, New York.

8. Weisberg, Sanford. (1980). *Applied Linear Regression*, John Wiley & Sons, New York.

SUMMARY

REGRESSION

```
REGRESSION  InFile ;
  DEPENDENT   Income  $

REGRESSION  IO,  COR  IOCor,  DES  IODes ;
  DEPENDENT  Cpu,  INDEPENDENT  Day Hour,  OUT  IOAllVar  $

REGRESSION  Speed,  WEIGHT  Wt.Var ;
  DEP  SP,  ALL.POSSIBLE  SpeedAP,  TEST  MALLOW  $
```

The REGRESSION command fits a linear equation to observed values of multiple independent and dependent variables. Equations may also be fit to autoregressive and time trend terms, polynomial distributed lags, and all possible subsets of independent variables. The independent variables may be transformed to achieve linearity or to stabilize the variance prior to the regression analysis using the programming language or the TRANSFORM identifier. The regression is a forward stepping procedure, unless options are specified to force the inclusion of independent variables.

Either a P-STAT system file of data values or input correlation and description files are required. The dependent (and independent) variables are specified at the *subcommand* level. An output file that includes the original data values, the predicted scores, and the residuals may be requested using the subcommand OUT. Other output files are also available. The command name REGRESSION may be abbreviated to REG.

Required:

REGRESSION fn

specifies the P-STAT system file of data values. If there are existing correlation (COR) and description (DES) files, they may be used instead. *Either a file of input data or both COR and DES files are required.*

Required Identifiers if no data file input:

COR fn

specifies the name of the input correlation matrix (produced by the CORRELATE command). COR and DES are used only when a system file of data values is not input to REGRESSION.

DES fn

specifies the name of the input description file. (Description files are optional output files from the BUCONCAT, CORRELATE and MODIFY commands.) If COR is used, DES must also be used.

Optional Identifiers:

BY vn vn ....

specifies from 1 to 15 numeric or character variables by which the file is either sorted or ordered. A separate regression is performed for each subgroup defined by the BY variables. (BY may not be used when the ALL.POSSIBLE subcommand specifies regression analyses of all possible subsets.)

TRANSFORM LOG / LOG10

requests log (to the base e) transformations of the dependent and independent values. The BY and WEIGHT variables (if any) are not transformed. TRANSFORM LOG10 requests log transformations to base 10. *All* of the variables in the file are transformed. PPL may be used to transform just some of the variables or to specify any arbitrary transformations.

WEIGHT vn

provides the name of the weight variable. Its values may be integer or fractional weights.

Required Subcommands:

DEPENDENT vn

specifies the dependent variable. DEPENDENT may be abbreviated to DEP. (See the INDEPENDENT subcommand also.)

Optional Subcommands:

AGAIN

requests that the previous regression be repeated, usually with some change in the options.

ALL.POSSIBLE fn

specifies that linear equations be fit to *all* possible subsets of the independent variables. The subsets range in size from one to the total number of independent variables (n) and the total number of subsets is $2^n - 1$. The best subsets (in terms of R^2) of each size are written in the specified output system file; when no output file name is supplied, all subsets are written in a temporary system file and the best subsets of each size are listed. (See the TEST and MAX subcommands for relevant options.)

ATLEAST nn

specifies that the program is to accept at least that many variables, if possible.

ATMOST nn

provides an upper limit on the number of variables that are to be accepted.

COEF fn

provides a name for an output file of regression coefficients. This can be used by the RESIDUALS command.

COMPLETE

indicates that only cases with no missing data on the dependent and independent variables are to be used in the regression. COMPLETE is assumed unless MISSING is specified. It is only used when the input is a file of data values. When a correlation file is used as input, the treatment of missing data depends on the identifiers used in the CORRELATE command.

F.DELETE nn

changes the F-to-delete threshold, which is normally set to 1.0. A variable already entered into the regression equation is deleted when its F value becomes less than the F.DELETE value.

F.ENTER nn

changes the F-to-enter threshold, which is normally set to 1.5. A variable is entered into the regression equation if its F value is greater than the F.ENTER value.

INDEPENDENT vn vn ....

specifies a list of independent variables. It may be abbreviated to IND. When IND is not used, any numeric variables not otherwise in use are included as independent variables.

LAG may be used with the IND subcommand to create additional independent variables from an initial variable whose values are serially related:

```
REGRESSION  Stream ;
  DEP  Salt.Concentrate,  IND  Location  ( LAG 0 2 4 6 ) ;
```

The independent variables in this analysis are Location, Location.2, Location.4 and Location.6.

POLY may be used after the *first* independent variable specification to request that REGRESSION fit a polynomial distributed lag model to time series data:

```
REGRESSION  Yrs80.89 ;
  DEP  Y,  IND  X  (POLY 2,  LAG 1 TO 4) ;
```

LAG should follow and give the length of the lag, which should be a sequential range starting with one. Missing values of the lagged variable (X in this example) cause a fatal error.

MAX nn

specifies the number of best subsets of each size to include in the output file of all-possible regressions. MAX is used only when ALL.POSSIBLE is also used. When the maximum number of subsets is not specified, MAX 1 is assumed — the output file includes the single best subset of each size. "Best" is determined by the associated R^2, unless the TEST subcommand is used to specify another criterion. MAX ALL may be specified to include *all* subsets in the output file.

MISSING

indicates that cases with missing data are to be included in the regression.

NO DELETE

specifies that a variable may not be deleted once it has been entered into the regression, even if its F value falls below the F-to-enter value.

NO STEP

forces *all* independent variables into the regression without any stepping, in the order that they appear in the input file, unless computational impossibilities prevent this. If the first variable is rejected because of its F value, the REGRESSION procedure stops. Only the summary prints when NO STEP is used.

ORDER

specifies that the variables be used in the order in which they appear in the input file until the ATLEAST setting is reached. If ATLEAST is not used, the variables are used in order until they are all in the regression analysis.

ORIGIN

specifies that the regression equation line or surface should pass through the origin (0, 0). (The equation will not have a constant term.)

OUT fn

provides a name for the output file, which includes all the variables in the input file as well as the residuals and predicted values of the dependent variables. The variables of predicted values have the same names as the original variables, except that they are prefixed with "Pre.". The variables of the residuals have the same names as the original variables except that they are prefixed with "Res.".

PR 'fn'

requests that the printout be directed to the specified external disk file or on-line printer.

P.DELETE nn

provides a probability that is to be used instead of the F-to-delete. A variable already entered into the regression equation is deleted when the probability associated with its F value becomes greater than the P.DELETE value.

P.ENTER nn

provides a probability that is to be used instead of the F-to-enter. A variable is entered into the regression equation if the probability associated with its F value is less than the P.ENTER value.

REPORT

specifies that the printed output include both steps and summary. NO REPORT may be used to turn off the printed output — it is typically used when a regression is repeated so that an output file may be requested:

```
AGAIN,  OUT  RegOut,  NO  REPORT ;
```

RESET

resets all the subcommand options to their original settings. This includes the lists of dependent and independent variables.

STAND.COEF fn

provides a name for an output system file of standardized regression coefficients.

START nn

gives the number of variables (starting from the left in the file) to be entered at the start of the regression procedure, before normal stepping begins. These variables must still pass the tolerance threshold, however.

STATS fn

requests and provides a name for an output system file containing summary statistics from the regression analysis.

STEP

requests a stepwise regression. This is assumed unless NO STEP is used to turn off stepping.

SUMMARY

requests that only the final summary be printed; step information is thus suppressed.

TEST RSQ / MALLOW / ADJ.RSQ

specifies a criterion to use in selecting the best subsets of independent variables produced when regression equations are fit to all-possible subsets (see ALL.POSSIBLE). One of the following criteria may be specified as the argument after TEST:

`RSQ` (for R^2) `MALLOW` (for Mallows' C_p) `ADJ.RSQ` (for Adjusted R^2)

When TEST is not used, TEST RSQ is assumed, and the best subsets of each size are those with the highest value of R^2. (See the MAX subcommand also.)

TOL nn

changes the tolerance threshold, normally set to 0.005, to the specified value. A variable is not entered into the regression equation unless its tolerance value is greater than or equal to the TOL value. TOL equals:

$1 - R^2$.

As more variables are entered into the regression equation, the value of TOL for additional variables is lower. Thus, it becomes increasingly more difficult to enter additional variables.

USE.MEAN

specifies that the mean score be used in place of missing data in the original input file for predicting the dependent variable and for calculating residuals. This permits missing data for independent variables and yet prediction of the dependent variable.

RESIDUALS

```
RESIDUALS  File1,  COEF  F1Coef,  RES  F1Res,
  PRED  F1Pred  $
```

The RESIDUALS command computes predicted values and/or residuals from new data, using the coefficients calculated from a regression on related data.

Required:

RESIDUALS fn

provides the name of a P-STAT system file containing independent and dependent variables. The dependent variables may be missing if the purpose of the run is only to compute predicted scores and calculate residuals.

Required Identifiers:

COEF fn

provides the name of a coefficient file from a previous regression which used either these input data or other input data with the same variables. The coefficient file has one row for each independent variable, plus one row for the constant, and one column for each dependent variable. (A coefficient file with 21 rows and 3 columns represents a regression with 20 independent variables and 3 dependent variables.) Each of the independent and dependent variables must also be present in the input file.

Optional Identifiers:

DES fn

provides the name of the description file which must be supplied if USE.MEAN is specified.

PRED fn

provides a name for an output file of predicted values. It will contain one row for each row in the input file and one column for each column from the COEF file.

The variables in the file will have the same names as those in the original file, except that they will be prefixed with “Pre.”. Thus, the PRED file may be joined back onto the input data or with the RES file without the presence of duplicate variable names. If there are more than 12 characters in the original names, characters from the right end will be dropped to make room for the prefixes.

RES fn

provides a name for an output file of residuals. It will contain one row for each row in the input file and one column for each column from the COEF file. OUT is a synonym for RES.

The variables in the file will have the same names as those in the original file, except that they will be prefixed with “Res.”. Thus, the RES file may be joined back onto the input data or the PRED file without the presence of duplicate variable names. If there are more than 12 characters in the original names, characters from the right end will be dropped to make room for the prefixes.

USE.MEAN

requests that the mean be used in place of any missing data in the independent variables when calculating predicted and/or residual scores. If the mean is not used, the predicted or residual score will be set to missing for any case with a missing value on any independent variable needed in that prediction. USE.MEAN is used in conjunction with a DES file.

PRE.POST

```
PRE.POST  File1,  OUT  Fpp  $
```

The PRE.POST command performs regression analysis and calculates residuals and predicted values when the variables in the file are measures from before and after a treatment.

Required:

PRE.POST fn

specifies the name of the required input file of data. The file should contain an even number of variables. The first half of the variables should be pre-test scores and the second half, post-test scores. If this is not the case, PRE and POST should be used.

Optional Identifiers:

COR.N fn

specifies a name for an output file of correlations and good N’s. It will have two columns and a row for each pair of pre-test and post-test variables.

OUT fn

provides a name for the output file of residuals.

POST vn vn

specifies the name or position of the first of the post-test variables and the name or position of the last of the post-test variables. If PRE and POST are not used, the program assumes that the first half of the variables in the file are the PRE variables and the second half are the POST variables. The file must have an even number of variables.

PRE vn vn

specifies the name or position of the first of the pre-test variables and the name or position of the last of the pre-test variables. If PRE and POST are not used, the program assumes that the first half of the variables in the file are the PRE variables and the second half are the POST variables. The file must have an even number of variables.

SI.IN fn

provides the name for an input slope and intercept file from a previous PRE.POST run. This file should have two columns and a row for each pair of pre-test and post-test variables. These slope and intercept values are used to compute the residuals of the new data.

SI.OUT fn

requests and provides a name for a slope and intercept output file. If there are NV pre-test variables (and therefore NV post-test variables), this file will have NV rows and two columns.

DURWAT

```
DURWAT  CarRes  [ KEEP  Res? ],  OUT  CarDw  $
```

DURWAT produces a vector of Durbin-Watson values. The output file has one row for every column of the input file.

The Durbin-Watson statistic d is used to test for autocorrelation of the residuals from regression analysis. The range of d is from 0 to 4. The closer the value of d to 2, the firmer the evidence that there is no autocorrelation present in the error. Values of d for different percentage points, tabulated by Durbin and Watson, are available in statistics texts.

Required:

DURWAT fn

specifies the name of the input file of residuals. Typically, this file is an output file from the REGRESSION or RESIDUALS command. Missing data values are not permitted.

Required Identifiers:

OUT fn

provides a name for the output file of Durbin-Watson values. This file will have one variable and as many cases as there are variables in the input file.

POLY.FIT

```
POLY.FIT Cows, DEPENDENT Temps,  DEGREE 4 $
```

POLY.FIT does a polynomial regression to the requested degree.

Required:

DEGREE nn

specifies the degree of the polynomial

DEPENDENT vn

required when the file has more than one variable. DEPENDENT can be abbreviated to DEP.

Optional:

INDEPENDENT vn

If this is not used, values of 1, 2, 3, etc. are generated. IND can be used as an abbreviation.

OUT fn

produces a results file which with the input variables and the fit value. This is the file that is used for the plots. If it is not supplied a work file is generated.

COEF fn

requests a file containing the polynomial coefficients and the intercept.

CHAR 'cs'

supplies a character to be used for the polynomial curve which overlays the data values in the first plot. This is used in character plots only. Symbols are used in XWindow plots and a period with a drawn line is used in the PostScript plots.

NO FILL

requests that extra points not be generated. These make the curve smoother and easier to read.

NO PLOT

requests that the plots be omitted.

USE 'cs'

supplies a string which contains additional plot identifiers.

5
NP.TEST, NP.COR: Nonparametric Statistics

Nonparametric statistics are often called “distribution-free tests” or “rank tests.” These terms highlight the advantages of nonparametric tests — they do not assume that the data are from a normally distributed population, nor do they assume that the data are quantitative measurements. In addition, nonparametric tests are often more appropriate for small sample sizes. The two P-STAT commands that produce nonparametric tests are the NP.TEST and NP.COR commands.

NP.TEST calculates the following nonparametric statistics:

- ONE-SAMPLE TESTS

Binomial	BINOMIAL
Chi Square	CHI
Kolmogorov-Smirnov One-Sample Test	KS1

- TWO-INDEPENDENT-SAMPLE TESTS

Median Test	MEDIAN
Mann-Whitney U Test	U
Kolmogorov-Smirnov Two-Sample Test	KS2
Wald-Wolfowitz Runs Test	RUNS
Squared Ranks Test for Equal Variances	SQUARED.RANKS

- TWO-PAIRED-SAMPLE TESTS

Sign Test	SIGN
McNemar Test for Significance of Changes	MCNEMAR
Wilcoxon Matched-Pairs Signed-Ranks Test	WILCOXON

- K-INDEPENDENT-SAMPLE TESTS

Median Test	MEDIAN
Kruskal-Wallis 1-Way ANOVA by Ranks	KW
Squared Ranks Test for Equal Variances	SQUARED.RANKS

- K-PAIRED-SAMPLE TESTS

Cochran Q Test	COCHRAN
Friedman 2-Way ANOVA by Ranks	FRIEDMAN
Kendall Coefficient of Concordance	CONCORDANCE

NP.COR calculates the following nonparametric statistics:

- TWO-PAIRED-SAMPLE TESTS

Spearman Rank Correlation Coefficient	
Kendall Rank Correlation Coefficient	KENDALL

The identifier that should be used in each command is shown to the right of each test. Spearman is assumed in the NP.COR command.

These nonparametric tests are described in *Nonparametric Statistics for the Behavioral Sciences,* by Sidney Siegel, McGraw-Hill Book Company, Inc., New York, 1956, (examples used by permission of McGraw-Hill Book Company, copyright 1956) and in *Practical Nonparametric Statistics,* by W. J. Conover, John Wiley & Sons, New York, 1980 (examples used by permission of John Wiley & Sons, copyright 1980). The file names that

begin with “S” in the examples illustrating each test refer to problems on pages in Siegel’s text, and those that begin with “C” refer to pages in Conover’s text. For example, “S40” refers to the problem on page 40 in Siegel.

5.1 BACKGROUND

Nonparametric tests are used with data that do not meet all of the assumptions made by parametric tests. For example, the assumptions that must be met for parametric tests, such as the t test of the differences between means, are:

1. independent (random) observations,
2. observations from normally distributed populations,
3. equal variances in the populations, and
4. quantitative (interval or ratio) measurements. (Siegel)

For nonparametric tests, only the first assumption of independent observations must be met. The observations need not come from a normally distributed population, the populations do not need to have equal variances, and the data generally need be only qualitative (ordinal or ranking) measurements.

Parametric tests are generally more powerful than nonparametric ones. There is less possibility of rejecting the null hypothesis that no difference exists between groups when it should not be rejected. However, the power of nonparametric tests can be increased by increasing the sample size. Thus, parametric tests should be used if the assumptions can be met. If not, nonparametric tests must be used, and the sample size should be increased if possible. Even if the sample is fixed and small in size, nonparametric tests are often better because they generally calculate exact probabilities when the sample is small and use approximations only for large samples.

5.2 Hypothesis Testing

An experimenter, investigating a problem, collects data from a sample. On the basis of this data, the experimenter would like to infer something about the population from which the sample came. This process of statistical inference is called “hypothesis testing.” (Hypothesis testing is the basis of parametric statistical inference, as well as nonparametric inference.)

The first step for the experimenter is to formulate a “null hypothesis. ” This is typically (though not always) a hypothesis stating that no difference exists — for example, between the measurement observed in the sample and that expected in the population, between the measurements observed in two samples and the population, or between before and after measurements observed in a sample and the population. An alternative hypothesis, opposite to the null hypothesis, is also formulated. This is the prediction or hypothesis of interest to the experimenter.

A statistic is selected to test the null hypothesis. The model associated with this statistic has certain assumptions and measurement requirements that should be met by the data. For example, the sample must be random, the data may need to be from a continuous or a symmetric distribution, and the measurements may need to be ordinal values (ordered values) or interval values (ordered values with equal intervals between them).

A rule is stated, in terms of values of the test statistic, for rejecting the null hypothesis. When the null hypothesis is rejected, the alternative hypothesis is accepted. The rule is typically a statement that if the probability of observing a measurement (such as a median or a variance) is smaller than a specified fraction (such as .05 or .01), the null hypothesis will be rejected and the alternative accepted.

Thus, hypothesis testing involves the following steps:

1. Formulate the null hypothesis and its alternative in terms of the population.
2. Choose the test statistic and check that any assumptions and requirements are met by the sample data.
3. State a decision rule for rejecting the null hypothesis and accepting the alternative.
4. Evaluate the test statistic using data from a random sample from the population and make a decision regarding the null hypothesis. (Conover)

Hypotheses may be nondirectional or directional. If the null hypothesis is that the difference between the medians of two samples is zero:

$$H_0: \ Median_1 \ - \ Median_2 \ = 0,$$

then the alternative hypothesis is also nondirectional:

$$H_1: \ Median_1 \ - \ Median_2 \ \hat{} = 0.$$

The alternative hypothesis says merely that the difference between the medians is not zero. It says nothing about the direction of that difference — whether the first median is larger than the second or whether the first median is smaller. If the null hypothesis is that the difference between the first median and the second median is greater than or equal to zero:

$$H_0: \ Median_1 \ - \ Median_2 \ >= 0,$$

then the alternative hypothesis is also directional:

$$H_1: \ Median_1 \ - \ Median_2 \ < \ 0.$$

The alternative hypothesis says that the difference between the first and second medians is less than zero; that is, the first median is smaller than the second. (The directions in the null and alternative hypotheses could be reversed.)

A nondirectional test is also called a “two-tailed” test because the two tails of the test statistic distribution are used in estimating probability. A directional test is called a “one-tailed” test because only one tail of the distribution is used. A two-tailed test is appropriate when one is concerned with the absolute magnitude of the difference between two medians, for example, and a one-tailed test is appropriate when one is concerned with the direction of that difference.

For a two-tailed test with a probability or significance level of .05, using the normal distribution for example, a test statistic of 1.96 is required. The area of the distribution above 1.96 and below -1.96 sum to .05 (each is .025). For a one-tailed test with the same .05 probability, a test statistic of either 1.64 or -1.64 is required. (The expected direction should be incorporated into the alternative hypothesis, and not chosen after-the-fact.) The area above 1.64 and the area below -1.64 are each .05.

With regard to reading the output from a computer program, where probability is given not just at select points (such as .05) but continuously, a two-tailed probability should be halved to obtain a one-tailed probability. Similarly, a one-tailed probability should be doubled to obtain a two-tailed probability.

5.3 Selecting Statistical Tests

An experimenter must choose an appropriate statistic to test the null hypothesis using the specified decision rule. If the sample data meet the assumptions for parametric tests (given earlier in the beginning of the section “BACKGROUND”), those tests should be used because they are more powerful. However, the assumptions for parametric tests are more stringent than those for nonparametric tests, and often it is not possible to meet them. When that is the case, the most appropriate nonparametric test should be selected.

Nonparametric tests typically have requirements about the scale of measurement used and, occasionally, about the distribution underlying those measurements. Measurements may be ordered by the strength of their level or scale. The four levels of measurement, from “weakest” to “strongest”, are nominal, ordinal, interval and ratio.

Nominal and ordinal measurements are *qualitative.* Qualitative data are usually sufficient for nonparametric tests. The *nominal* level assigns numbers to observations as names for categories. For example, assigning females the value “0” and males the value “1” is a nominal scale of measurement. Different measurements may be equal or unequal. The *ordinal* level assigns numbers to observations to name and order categories. For example, rating beverages with the values “1” through “5” to indicate “dislike” to “like very much” is an ordinal scale of measurement. Different measurements may be equal or unequal, and they may also be less than or greater than one another.

Interval and ratio measurements are *quantitative* measurements. Quantitative data are typically required for parametric tests. The *interval* level assigns numbers to observations to both order measurements and to give the difference between measurements. An arbitrary start point or "zero" and unit steps between measurements exist. For example, temperature measurements are an ordinal scale of measurement. Different measurements may be equal or unequal, they may be less than or greater than one another, and the distance between any two measurements may be equal, less than, or greater than another distance. The *ratio* level assigns numbers to observations to order measurements, to give the difference between measurements, and to give the ratio between measurements. An absolute or natural zero point and unit steps between measurements exist. For example, weight and income are ratio scale measurements. The ratio between different measurements is independent of the unit of measurement.

The following sections discuss the various nonparametric tests and give the required levels of measurement, any assumptions regarding the distribution of those measurements, and the null hypothesis that the statistic tests. Also, some idea of the comparative power of the tests is given. The tests are grouped according to the number of samples with which they deal and the independence or relatedness of those samples.

5.4 ONE-SAMPLE TESTS

One-sample tests are "goodness-of-fit" tests. They test if a random sample comes from ("fits") a population with a specified distribution. The Binomial, Chi-Square and Kolmogorov-Smirnov tests are nonparametric one-sample tests. A comparable parametric test is a t test of the difference between the observed or sample mean and the expected or population mean. (A one-sample t test may be done using the PAIRED.TTEST command, where one variable has data from one sample and the second variable is generated equal to the population mean.)

The P-STAT command for nonparametric tests, other than rank correlation tests, is NP.TEST. It is followed by the name of the input file. An identifier, such as BINOMIAL, CHI.SQUARE or KS1, specifies the desired test. Other identifiers specify any options. The output is printed on the terminal, unless it is directed elsewhere (using the command or general identifier PR).

Any of the nonparametric tests may be weighted by including the identifier WEIGHT in the NP.TEST command. Its argument is the name of the variable whose values are counts observed in the corresponding category. The weight for a case should be a positive integer, or that case will be skipped. When WEIGHT is not used, each case in the input file counts as one observation.

5.5 Binomial Test

The binomial test calculates the probability that, in a two-category population, k observations out of N total observations are in one category. The proportion of observations expected in one category is P, and the proportion expected in the other category is 1 - P or Q. The binomial distribution is the sampling distribution of observed proportions in random samples from a two-category population.

The null hypothesis is that the proportion expected in the population equals P. In other words, the null hypothesis is that the sample with this observed proportion in one category comes from a population with proportion P in that category. The alternative hypothesis is that the population proportion does not equal P. The data need be only nominal (categorical) values.

Figure 5.1 Data for Binomial Test (from Siegel, Page 40)

	METHOD CHOSEN		TOTAL
	First-Learned	Second-Learned	
Frequency	16	2	18

Figures 5.1 and 5.2 illustrate using the binomial test. The data deals with college students who were taught to tie the same knot two different ways. Half the students learned one method for tying the knot first and half learned the other first. The prediction was that under stress, the students would use the method they had learned first. If stress has no effect on the method using, the proportion using each method should be the same (.5). Thus, the null hypothesis is that the proportion using the first-learned method would be less than or equal to the proportion using the second-learned method. (Siegel) The alternative hypothesis is that a higher proportion would use the first-learned method than would use the second-learned one.[1]

Figure 5.2 Binomial Goodness-of-Fit Test

```
      NP.TEST,  BINOMIAL,  COUNTS  16  2  $


-------- BINOMIAL test on 16 and 2 --------
-------- the test P is .5

          CASES
             16  first  count (9.00 were expected)
              2  second count
           ----
             18  TOTAL

                 (binomial prob; 1-tailed unless otherwise noted)
          .9999  prob of 16 or fewer occurrences
          .0007  prob of 16 or more  occurrences (2-tailed P= .0013)
          .0006  prob of exactly 16  occurrences
```

The data is shown in Figure 5.1 and the way to specify the binomial test and the output is shown in Figure 5.2. Since the frequencies observed in each category are already tallied, it is not necessary to have an input file. The frequencies or counts are given using the COUNTS identifier with the BINOMIAL identifier in the NP.TEST command:

```
       NP.TEST,  BINOMIAL,  COUNTS  16  2  $
```

[1] Siegel specifies null hypotheses exclusively as hypotheses of no difference — as nondirectional tests. However, his alternative hypotheses and decision rules are often directional, and when a null hypothesis is rejected, he accepts the alternative directional hypothesis. In this manual, we shall specify directional "null" hypotheses when appropriate. Rejecting the null hypothesis and thereby accepting its opposite is intuitively more logical.

It is assumed that the proportion expected in each category is .5, unless another proportion follows BINOMIAL as its argument.

Figure 5.3 Data for Binomial Probability (0 is Heads, 1 is Tails)

FILE Tosses:

Var1
1
0
0
1
0
0
0
0
0
1

The output gives three probabilities. The one typically of interest is the probability of 16 or more occurrences in one category. (This is the same as the probability of two or less in one category.) Since this probability is very small, .0007 for a directional test, the null hypothesis is rejected and the alternative accepted. It is highly likely that the sample comes from a population where the proportion using the first-learned method is greater than .5.

When the counts of cases in each category have not been tallied, the file of data is input to the NP.TEST command. The binomial probability is calculated for each variable in the input file, excluding a possible weight variable. Figures 5.3 and 5.4 illustrate this. The input file in Figure 5.3 contains the results of 10 tosses of a coin (0 represents a “head” and 1 represents a “tail”). The probability of obtaining seven or more heads (.1719) is shown in Figure 5.4. This probability is not sufficiently small so that we could conclude that this is a biased coin — so that we could reject the null hypothesis that the proportion expected in each category is .5.

The NP.TEST command, used with the BINOMIAL identifier, needs to know the number of cases in each of the two categories. The COUNTS identifier may be followed by two numerical arguments giving this information (as in Figure 5.2), or a boundary-defining identifier such as EQ may be used (as in Figure 5.4). The identifier EQ in this command:

```
NP.TEST  Tosses,  BINOMIAL,  EQ  0  $
```

specifies that all values equal to zero are in one category and all other non-missing values are in the other category. There are six boundary-defining identifiers:

```
EQ  equal       GE  greater than or equal   LE  less than or equal
NE  not equal   GT  greater than            LT  less than
```

The identifiers are shown in uppercase letters and their meanings are in lowercase. The identifier LT, for example, followed by the number 5, specifies that all values less than 5 are in one category and all other non-missing values are in the other category.

Figure 5.4 Probability of Getting 7 Heads in 10 Tosses of a Coin

```
      NP.TEST  Tosses,  BINOMIAL,  EQ  0 $


-------- BINOMIAL test: file Tosses -----------
-------- variable Var1,  the test P is .5


          CASES
              7  eq 0 (5.0 were expected)
              3  other
           ----
             10  TOTAL


                 (binomial prob; 1-tailed unless otherwise noted)
          .9453  prob of 7 or fewer occurrences
          .1719  prob of 7 or more  occurrences (2-tailed P= .3438)
          .1172  prob of exactly 7  occurrences
```

Another way to provide the number of cases in each category is to input a file with two cases containing the counts in each of the two categories. The identifier COUNTS, without any arguments, indicates this:

```
      NP.TEST  S40,  BINOMIAL,  COUNTS  $
```

In this example, the input file contains two cases — the first case is the count in one category and the second case is the count in the other category. If file S40 contained 16 and 2 in each of the two cases (this corresponds to the data in Figure 5.1), the output would be the same as that shown in Figure 5.2.

Thus, the required information giving the counts in each category may be input in any of three different ways:

1. Use COUNTS, followed by two numbers giving the number in each category, without any input file.
2. Use a boundary-defining identifier (EQ, NE, GE, GT, LE or LT) and the input file of data.
3. Use COUNTS without any arguments and an input file with two cases, each containing the count in one of the categories.

When P, the probability expected in each category in the population, does not equal .5, the expected probability should be supplied after the identifier BINOMIAL. Here, the expected probability is .17:

```
      NP.TEST  DieRolls,  BINOMIAL .17,  EQ  1  $
```

This is approximately one-sixth, the expected probability of occurrence of any one number on a die.

Figure 5.5 shows an input file created to calculate the probability of rolling two sixes in five rolls of a single die. The five cases represent the total of five rolls, and the “1’s” and “2’s” represent “hits” and “misses” (or “wins” and “losses”). (A hit is a six and a miss is anything else.) Any arbitrary two numbers could be used to represent the hits and misses, or a file of real data (containing the numbers actually rolled) could be input. Figure 5.6 shows the command and the output. The probability of rolling exactly two sixes is .1652.

Figure 5.5 Data for Binomial Probability (1 is Win, 2 is Lose)
FILE DieRolls:

```
Var1

   1
   1
   2
   2
   2
```

An exact binomial probability is calculated. The sampling distribution of the binomial is the sum of all terms in the binomial (P + Q) to the nth power, where P is the probability that an event will occur and Q is the probability that it will not, and n is the number of observations.

Figure 5.6 Probability of Getting 2 "sixes" in 5 Rolls of Die

```
     NP.TEST  DieRolls,  BINOMIAL .17,  EQ  1  $


-------- BINOMIAL test: file DieRolls -----------
-------- variable Var1, the test P is .17


          CASES
              2  eq 1 (.8 were expected)
              3  other
           ----
              5  TOTAL


                 (binomial prob; 1-tailed unless otherwise noted)
          .9625  prob of 2 or fewer occurrences
          .2027  prob of 2 or more  occurrences
          .1652  prob of exactly 2  occurrences
```

5.6 Chi-Square Test

The chi-square test calculates the probability that, in a multi-category population (as opposed to a two-category one), the observed frequencies in each category of the sample are those expected in the population. The number of categories may be two or more. The chi-square distribution is the sampling distribution of discrepancies between observed and expected frequencies in random samples from a multi-category population. The distribution differs for different numbers of categories or degrees of freedom (the number of categories minus one).

The null hypothesis is that the sample comes from a population with certain expected frequencies in each category. In other words, the null hypothesis is that no difference exists between the observed and expected

frequencies. The alternative hypothesis is that the sample does not come from this population; that is, a difference exists between the observed and expected frequencies. The data need be only nominal (categorical) values.

Figure 5.7 Data for Chi-Square Test

FILE S45:

Post Position	Wins
1	29
2	19
3	18
4	25
5	17
6	10
7	15
8	11

Figure 5.8 Chi-Square Goodness-of-Fit Test

```
     NP.TEST  S45,  CHI,  WEIGHT  Wins  $


------- CHI-SQUARE test: file S45 -------
------- variable Post.Position


             observed    expected         cell    (weighted by
 category       cases       cases   chi-square     var Wins)
        1          29       18.00         6.72
        2          19       18.00          .06
        3          18       18.00         0.00
        4          25       18.00         2.72
        5          17       18.00          .06
        6          10       18.00         3.56
        7          15       18.00          .50
        8          11       18.00         2.72
                  ----
                   144


                chi-square       DF    significance
                   16.3333        7           .0222
```

Figures 5.7 and 5.8 illustrate using the chi-square test. The data in Figure 5.7 show the number of wins for different post positions for a month at one horse racing track. The null hypothesis is that there is no difference in

the number of wins expected at each post position (1 is closest to the inside rail and 8 is furthest from the rail). The alternative hypothesis is that there is a difference in the number of wins at each post position. (Siegel)

Figure 5.8 shows the command and the output. The identifier CHI requests the chi-square test:

```
NP.TEST  S45,  CHI,  WEIGHT  Wins  $
```

CHI is short for CHI.SQUARE or CHISQUARE, either of which could be used instead, if desired. The identifier WEIGHT is used in this command because each case is not a single observation, but a count of observations in the corresponding category or post position. The chi-square statistic is calculated for each variable in the input file, excluding the weight variable.

The output in Figure 5.8 shows both the observed and the expected frequencies in each category. The expected frequencies are the total number of observations divided by the number of categories, since the proportion expected in each category is the same. If the frequencies expected in each category are not equal, the EXPECTED identifier should be used to provide different expected values.

Figure 5.9 Chi-Square Goodness-of-Fit Test with Expected Values

```
     NP.TEST  S45,  CHI,  WEIGHT  Wins,
        EXPECTED  22  20  18  18  18  18  16  14  $


------- CHI-SQUARE test: file S45 -------
------- variable Post.Position


             observed    expected         cell    (weighted by
 category       cases       cases   chi-square     var Wins)
        1          29       22.00         2.23
        2          19       20.00          .05
        3          18       18.00         0.00
        4          25       18.00         2.72
        5          17       18.00          .06
        6          10       18.00         3.56
        7          15       16.00          .06
        8          11       14.00          .64
                  ----
                   144


                chi-square       DF    significance
                    9.3160        7           .2308
```

Figure 5.9 shows different expected values provided for the same position-versus-wins problem. Eight expected frequencies must be given when there are eight categories:

```
NP.TEST  S45,  CHI,  WEIGHT  Wins,
   EXPECTED  22  20  18  18  18  18  16  14  $
```

The order of the frequencies should correspond to the category numbers. Here, more wins are expected for the first two post positions and less for the last two. If the supplied expected frequencies do not sum to the total number of observations, each number divided by the sum will be the proportion of the total observations expected in that category. For example, if these values were given for the position-versus-wins problem:

```
EXPECTED  15  15  10  10  10  5  5  5,
```

15 divided by 75 (the first expected value divided by the sum of all the expected values), or 20 percent of the total number of observations, would be the frequency expected in the first (and second) category.

The outputs in both Figures 5.8 and 5.9 give the chi-square for each cell or category, as well as the overall chi-square and its corresponding probability (significance level). This is the probability of obtaining that large a chi-square (measure of discrepancy) in a population with these expected frequencies. It is a "two-tailed" (nondirectional) probability, even though only the right tail of the chi-square distribution is used. This is due to the fact that chi-square is always positive because the discrepancies between the observed and expected frequencies are squared. In goodness-of-fit tests, nondirectional or two-tailed hypotheses are generally most appropriate. However, if a one-tailed probability is desired, the two-tailed value should be halved.

The significance of the chi-square statistic in Figure 5.8 is .0222. This is sufficiently small to reject the null hypothesis and accept the alternative one. There is a difference in the number of wins observed for each post position. The significance of the chi-square in Figure 5.9 is .2308. This is not sufficiently small to reject the null hypothesis; therefore, it must be accepted. Given the expectations of more wins for the first two post positions and less wins for the last two, the observed frequencies do not differ from the expected ones.

Only categories with data are included in calculating chi-square; empty categories are not included. If all categories should be included in the calculations, a range of categories should be specified after the CHI identifier:

```
NP.TEST  S45,  CHI  1  8,  WEIGHT  Wins  $
```

Because a range going from 1 through 8 is specified here, all of these categories are included in calculating chi-square, even if some of them do not have any data. If a category value is a not an integer, just the integer portion is used to place the case in a category. For example, if the value of Post.Position is 2.6, that case is placed in the second category.

The chi-square statistic is the sum of the squared differences between the observed and expected frequencies divided by the expected frequency. Large values of chi-square are more likely to indicate significant discrepancies of the observed frequencies from the expected ones. The chi-square sampling distribution under the null hypothesis is used to obtain the probability of a chi-square statistic when the degrees of freedom are less than 500. The normal approximation is used when the degrees of freedom are 500 or above.

There are a few restrictions on the size of the expected frequencies for categories. When there are only two categories, each expected frequency should be 5 or greater. When there are more than two categories, each expected frequency should be 1 or more and no more than 20 percent of the expected frequencies should be less than 5. A warning appears in the output when these restrictions are not met. To use a chi-square test when there are small frequencies that do not meet these requirements, combine several adjacent categories if that can be done meaningfully.

5.7 Kolmogorov-Smirnov One-Sample Test

The Kolmogorov-Smirnov goodness-of-fit test calculates the probability that the observed distribution of a sample does not differ from an hypothesized distribution. It compares the observed cumulative frequency distribution with either a uniform, normal or Poisson cumulative frequency distribution having the observed parameters (range, mean and/or standard deviation) or specified parameters.

The Kolmogorov-Smirnov test statistic is the largest vertical distance between the two distribution functions. The approximate two-tailed probability of obtaining a difference this large is calculated using the Smirnov

formula (1948). The null hypothesis is that the test sample has the same distribution as the hypothesized distribution; the alternative hypothesis is that it does not. The data should be at least ordinal (ordered) values.

Figure 5.10 shows the data set used in illustrating the Kolmogorov-Smirnov one-sample (KS1) test. Is this data normally distributed? The null hypothesis is that it is; the alternative one is that it is not. This command:

```
NP.TEST  Test,  KS1,  NORMAL  $
```

inputs the data set to NP.TEST, specifies the Kolmogorov-Smirnov one-sample test, and specifies the normal distribution as the hypothesized distribution.

Figure 5.10 Data for Kolmogorov-Smirnov One-Sample Test

FILE Test:

VAR1
0.023
1.570
0.125
0.211
0.894
0.127
0.204
0.132
0.255
0.971
0.024
0.111
0.272
0.173
0.005

Figure 5.11 shows the output. The observed mean and standard deviation, the largest absolute deviation of the observed cumulative frequency distribution from the normal one (as well as the largest positive and negative differences), the test statistic and the associated two-tailed probability are given. A value of .041 is sufficiently small enough to reject the null hypothesis and accept the alternative one. This data set is not normally distributed.

The identifiers NORMAL, UNIFORM or POISSON may be used with KS1 to specify one of three possible distributions. The WEIGHT identifier may also be used if the data are weighted. With NORMAL, the hypothesized distribution has the observed mean and standard deviation unless alternate ones are specified:

Figure 5.11 Kolmogorov-Smirnov Goodness-of-Fit Test

```
     NP.TEST  Test,  KS1,  NORMAL  $


 --- Kolmogorov-Smirnov goodness of fit test: file Test ---
 --- variable VAR1

     NORMAL fit using observed mean,sd of .33980 and .4466319

            .....LARGEST DIFFERENCES......
  cases    absolute   positive   negative      K-S Z    2-tailed P
     15      0.3603     0.3603    -0.2267      1.396         0.041
```

```
       NP.TEST  Zscores,  KS1,  NORMAL  0 1  $
```

This command compares a data set of standardized scores with a normal distribution having a mean of zero and a standard deviation of one. With UNIFORM, the low and high values of the distribution may be specified after the UNIFORM identifier. A continuous uniform distribution with that range is assumed. If the distribution is discrete (distinct values), the identifier DISCRETE should also be used. With POISSON, the mean may be specified, and the data must be positive integer values.

Figure 5.12 shows the Kolmogorov-Smirnov test used with the post position and wins data set (in Figure 5.7). The null hypothesis is that the same number of wins is expected for each post position at the track — that is, the observed frequencies are a sample from a uniform population distribution The alternative hypothesis is that different numbers of wins are expected for each position — the population is not a uniform distribution. The DISCRETE identifier is used along with KS1 and UNIFORM in the NP.TEST command because the post positions are eight distinct values.

Figure 5.12 Kolmogorov-Smirnov Goodness-of-Fit Test

```
      NP.TEST  S45,  KS1,  UNIFORM,  DISCRETE,  WEIGHT  Wins  $


 --- Kolmogorov-Smirnov goodness of fit test: file S45 ---
 --- variable Post.Position

     UNIFORM fit using observed range of 1 to 8

            .....LARGEST DIFFERENCES......
  cases    absolute   positive   negative      K-S Z    2-tailed P
    144      0.1319     0.0       -0.1319      1.604         0.013
```

The very small probability .013 of obtaining that large a difference between the cumulative frequency distributions of this data and a uniform distribution with a range going from one to eight forces rejection of the null hypothesis. The observed frequencies are not a sample from a uniform distribution. This is the same conclusion reached after the chi-square test.

The Kolmogorov-Smirnov one-sample test is generally considered to be more powerful than the chi-square test, both because it is suitable for small data sets and because information is not lost in categorization when either the data set is initially continuous or existing categories must be combined due to small expected frequencies.

5.8 TWO-INDEPENDENT-SAMPLE TESTS

Two-sample nonparametric tests may compare independent measurements or paired measurements. Independent measurements are typically the result of two samples from possibly different populations. Paired measurements result when subjects selected from two samples are matched with regard to one or more background variables, so that any observed differences are due to the treatment or variable of interest. With either independent or paired samples, the investigator typically wants to compare the samples with regard to central location of their distributions; that is, he (she) wants to determine if the measurements in one sample are larger than they are in the other. The comparable parametric procedure is a t test comparing the means of two independent or related samples. (See the TTEST and PAIRED.TTEST commands.)

The tests for two *independent* samples that may be requested in NP.TEST are the Median test, the Mann-Whitney U test, the Kolmogorov-Smirnov test, the Wald-Wolfowitz Runs test, and the Squared Ranks test for equal variances. The NP.TEST command requires one identifier to indicate which test is desired and the GROUP identifier. The identifiers for the statistical tests are: MEDIAN, MANN.WHITNEY (or U), KS2, RUNS and SQUARED.RANKS. If desired, the WEIGHT identifier may be used. (It is described in the beginning of the earlier section "ONE-SAMPLE TESTS.")

The GROUP identifier specifies the variable whose values define membership in the two groups:

```
NP.TEST  S114,  MEDIAN,  GROUP  Absent.Present,
                WEIGHT   Count  $
```

Here, the variable Absent.Present, shown in Figure 5.13, has values of 1 and 2 indicating to which group each case belongs. If the grouping variable does not have values of 1 and 2, the values that define group membership should be specified after the variable name:

```
NP.TEST  S130,  KS2,  GROUP  Class  7  11  $
```

Here, all cases in seventh grade classes are in one group and all cases in eleventh grade classes are in the other group. Cases with other values of Class are not included in either group. If only one number follows the variable name, all cases with that value of the grouping variable are in one group and all other cases with any moonshining values of that variable are in the other group.

5.9 Median Test

The Median test calculates the probability that the medians of two independent samples are not statistically different. (The Median test may also be used when there are more than two groups. See the subsequent section "K-INDEPENDENT-SAMPLE TESTS.") The sampling distribution is the hypergeometric distribution, although this is rarely used because of computational difficulties. The Fisher exact probability is computed when the sample size is 50 or less, and the chi-square approximation is used to obtain the probability when the sample is over 50.

The null hypothesis is that the two samples come from populations with the same median; that is, the two samples have equal medians. The alternative hypothesis is that the samples do not come from populations with the same median. The data should be at least ordinal (ordered) values.

Figures5.13 and 5.14 illustrate the Median test. The data are ratings on oral socialization anxiety in children (Rating), classifications of each society by the presence or absence of oral explanations of illness (Absent.Present — 1 is absent, 2 is present), and counts of cases in each grouping (Count).[2] The null hypothesis is that the median oral socialization anxiety in societies that give oral explanations of illness is less than or equal to the median oral anxiety in societies that do not give oral explanations. The alternative hypothesis (the prediction) is that the median oral anxiety in societies that give oral explanations is greater than that in societies that do not give such explanations.

---

[2] This data set, shown on page 114 in the Siegel text, is reproduced from Table 4 of Whiting, J.W.M., and Child, I.L., 1953, *Child Training and Personality*. New Haven: Yale University Press, p. 156, with the kind permission of the authors and the publisher.

Figure 5.13 Data for Median Test

FILE S114:

Rating	Absent Present	Count
13	1	1
12	1	2
10	1	4
9	1	1
8	1	2
7	1	5
6	1	1
17	2	1
16	2	1
15	2	3
14	2	3
13	2	3
12	2	4
11	2	2
10	2	3
8	2	2
6	2	1

This command requests the Median test:

```
NP.TEST  S114,  MEDIAN,  GROUP  Absent.Present,  WEIGHT  Count  $
```

The GROUP identifier specifies the grouping variable, which has values of 1 and 2. The WEIGHT identifier is used because each case is more than one observation. It is followed by the name of the variable whose values are the number of observations in that group with that score. The Median test is calculated for each variable in the input file, excluding the grouping and weight variables.

The NP.TEST command calculates the median of the combined groups and then classifies the scores in each group by whether they are above or below the median. If both groups are from the same population, half of each group's scores should be above the median and half below. The output, in Figure 5.14, shows this two-by-two classification table, as well as the smallest of the four expected values for each of the four cells in the table.

A chi-square statistic is calculated for all data and shown in the output. It incorporates a correction for continuity when the number of groups equals two (degrees of freedom equal one). The approximate probability associated with this chi-square statistic is obtained from the chi-square distribution. This is a two-tailed probability. When the number of observations is 50 or less, Fisher's exact test is also used to analyze the classification table. It computes the exact probability for a table with those cell counts and marginal totals. It is also a two-tailed probability. Either probability may be halved if a one-tailed probability is desired.

In Figure 5.14, the Fisher exact probability is given (the number of observations is 39). This two-tailed probability is halved to get a one-tailed one appropriate for the one-tailed hypotheses, and that value of .0049 is highly significant. The null hypothesis is rejected and the alternative one is accepted. The median oral anxiety in societies that give oral explanations is greater than that in societies that do not give such explanations.

Figure 5.14 Median Test of Equal Medians

```
      NP.TEST  S114,  MEDIAN,  GROUP  Absent.Present,  WEIGHT  Count  $


------ 2-sample MEDIAN test: file S114 ------
------ variable Rating

   gt median   le median   total  (weighted by Count)
           3          13      16  Absent.Present = 1
          15           8      23  Absent.Present = 2
        ----        ----    ----
          18          21      39  TOTAL

                     smallest                          chi     fisher
                     expected                chi   square      exact
   cases      median      value   DF       square      sig   2-tail P
      39      11.000       7.38    1       6.4350    .0112      .0098
```

If an arbitrary comparison value other than the median of the combined groups is desired, that value may be input as the argument to the MEDIAN identifier:

```
      NP.TEST  S114,  MEDIAN 14,  GROUP  Absent.Present,  WEIGHT  Count $
```

This command would produce the same Median test as the one in Figure 5.14, except that the number 14 would be used in place of the actual median to classify the observations. (Thus, the test statistic and its probability would be different.)

5.10 Mann-Whitney U Test

The Mann-Whitney U test (also called the Wilcoxon Rank Sum W test) calculates the probability that two independent samples come from the same population. It is one of the most powerful of the nonparametric tests that can be used when the data is at least ordinal measurements. The U statistic is calculated. The comparable W statistic for the Wilcoxon test may be computed from the sum of the ranks, given in the output. (When the two groups are the same size, W is the smaller of W_1 and W_2, where W_1 is the sum of the ranks of the smaller group and W_2 is $n_1(n_1 + n_2 + 1) - W_1$ Snedecor.) When there are more than 40 observations, the normal approximation to the sampling distribution is used to obtain the probability. When the number of observations is 40 or less, the exact probability is also computed.

The null hypothesis is that the two samples have the same distribution; that is, that the values in one sample are the same size as the values in the other sample. The alternative hypothesis is that one sample has larger values than the other. The data should be at least ordinal (ordered) values.

Figure 5.15 illustrates the Mann-Whitney test. The data are the same as those used in the Median test (the oral socialization anxiety data) and are shown in Figure 5.13. This command requests the Mann-Whitney U test:

```
      NP.TEST  S114,  MANN.WHITNEY,
         GROUP  Absent.Present,  WEIGHT  Count  $
```

In addition to the MANN.WHITNEY identifier, the required GROUP identifier and the optional WEIGHT identifier are used. The Mann-Whitney test is calculated for each variable in the input file, excluding the grouping and weight variables.

Figure 5.15 Mann-Whitney Test of Equal Distributions

```
      NP.TEST  S114,  MANN.WHITNEY,
        GROUP  Absent.Present,  WEIGHT  Count  $


----- Mann-Whitney U test: file S114 -----
----- variable Rating

  SUM OF RANKS   MEAN RANK   CASES  (weighted by Count)
        200.00       12.50      16  Absent.Present = 1
        580.00       25.22      23  Absent.Present = 2
                                -----
                                   39  TOTAL

                                normal approximation
                    exact       corrected  for  ties    tied
          U   2-tailed P             Z   2-tailed P   cases
      64.00       0.0004       -3.4510       0.0006      36
```

The data from both groups are combined and ranked. The number of scores from one group that precede scores from the other group is counted. This total is the U statistic. (The U statistic is actually the smaller of the two numbers that are totals for each of the groups — that is, the number of the first group that precede the second group and vice-versa.) Alternatively, U may be computed by a formula that incorporates the counts for each group and the sum of the ranks in one group.

The output, shown in Figure 5.15, gives the sum of the ranks and the mean rank of each group and the U statistic. The exact two-tailed probability associated with a U statistic of 64 is .0004. The corresponding one-tailed probability is .0002. This is highly significant or small enough to reject the null hypothesis and accept the alternative one. The Median test resulted in this same conclusion — the median oral anxiety in societies that give oral explanations is greater than that in societies that do not give such explanations.

The Mann-Whitney test is more powerful than the Median test because it deals with ranks or relative order rather than just with position above and below the median. Thus, it uses more of the information inherent in the data. This is illustrated by the probability of .0002 (one-tail) obtained in the Mann-Whitney test, which is much less than the probability of .0049 (also one-tail) obtained in the Median test. In this example, either test result is sufficient for rejecting the null hypothesis. However, with another data set, the Mann-Whitney test might lead to rejection of the null hypothesis and the Median test might not. Both tests are suitable for use with the same type of data — ordinal values. (Siegel)

The output from the Mann-Whitney test (Figure 5.15) also gives a Z value and its associated two-tailed probability of .0006 (one-tailed value of .0003). The formula for Z incorporates a correction for ties. As the sample size increases, the sampling distribution of U approaches the normal distribution. With a sample size of 39, the exact and normal approximation probabilities are extremely close. When the sample size is greater than 40, only the probability from the normal approximation is given in the output.

5.11 Kolmogorov-Smirnov Two-Sample Test

The Kolmogorov-Smirnov test calculates the probability that two independent samples come from the same population. Unlike the Median and the Mann-Whitney tests that discern differences between central location, the Kolmogorov-Smirnov test discerns differences in central location, variability or skewness. It compares the empirical cumulative distribution functions (the functions of observed values) from both the samples.

The Kolmogorov-Smirnov test statistic, for a two-tailed test, is the largest vertical distance between the two distribution functions. (For one-tailed tests, the test statistic is the largest vertical distance between the hypothesized smaller distribution and the hypothesized larger distribution. Thus, there are different test statistics for the two-tailed test and each of the one-tailed tests.) The approximate two-tailed probability of obtaining a difference this large with random samples from the same population is calculated using the Smirnov formula (1948).

The null hypothesis is that the two samples have the same distribution; that is, that the values in one sample are the same size as those in the other sample and have the same variability. The alternative hypothesis is that the two samples have different distributions, and that either the values in one distribution are larger than those in the other or they vary more. The data should be at least ordinal (ordered) values from distributions assumed to be continuous. (The test is more conservative when the values are discrete.)

Figure 5.16 Data for Kolmogorov-Smirnov Test

FILE S133:

Photos	Authoritarianism	Subjects
0	1	11
3	1	7
6	1	8
9	1	3
12	1	5
15	1	5
18	1	5
0	2	1
3	2	3
6	2	6
9	2	12
12	2	12
15	2	14
18	2	6

Figure 5.16 shows the data used in illustrating the Kolmogorov-Smirnov test. Subjects, who had been classified as either low or high (1 is low, 2 is high) on a measure of authoritarianism, identified the nationality of people in photos. Since the choices of nationalities did not include the actual nationality of the people, identification indicated stereotyping. The null hypothesis is that subjects low in authoritarianism would identify (stereotype) as many or more photos than those high in authoritarianism. The prediction or alternative hypothesis is that subjects low in authoritarianism would identify fewer photos. (Siegel)

This command requests the Kolmogorov-Smirnov test:

```
NP.TEST  S133,  KS2,  GROUP  Authoritarianism,  WEIGHT  Subjects  $
```

If Authoritarianism did not have values of 1 and 2, the two values that define the groups would be given as arguments of the GROUP identifier after the grouping variable name. This is the case here:

```
NP.TEST  S130,  KS2,  GROUP  Class  7  11  $
```

The WEIGHT identifier is used in the file S133 example because each case represents more than one subject. The Kolmogorov-Smirnov test is calculated for each variable in the input file, excluding the grouping and weight variables.

The output is shown in Figure 5.17. The largest differences between the distributions are given. The "absolute" difference (.4057) is appropriate for a two-tail test, and it is used in computing the two-tailed probability given at the right of the output (.001). The "positive" difference (.4057) could be used to test that the distribution of low authoritarianism subjects is larger, and the "negative" difference (-.0025) could test that the distribution of low authoritarianism subjects is smaller. (The absolute value of the largest negative difference is the largest difference when the first distribution — low authoritarianism — is subtracted from the second — high authoritarianism, the opposite of what is occurring here.) For situations when the probability is very small, the one-tailed probability is typically half the two-tailed one. (Conover)

The two-tailed probability of .001 (in Figure 5.17) is very small and thus may be halved to obtain a one-tail value of .0005. This value is highly significant, and therefore the null hypothesis is rejected and the alternative one accepted. Subjects low in authoritarianism identify fewer photos than do subjects high in authoritarianism.

The Kolmogorov-Smirnov test is more powerful than the Median test, and, for small samples, slightly more powerful than the Mann-Whitney test. For larger samples, the Mann-Whitney test is more powerful.

Figure 5.17 Kolmogorov-Smirnov Test of Equal Distributions

```
      NP.TEST  S133,  KS2,  GROUP  Authoritarianism,  WEIGHT  Subjects  $


 ---- Kolmogorov-Smirnov two sample test: file S133
 ---- variable Photos

      CASES  (weighted by Subjects)
         44  Authoritarianism = 1
         54  Authoritarianism = 2
      -----
         98  TOTAL

     .....LARGEST DIFFERENCES......
     absolute   positive   negative      K-S Z    2-tailed P
       0.4057     0.4057    -0.0025      1.998         0.001
```

5.12 Wald-Wolfowitz Runs Test

The Wald-Wolfowitz Runs test calculates the probability that two independent samples come from the same population. Like the Kolmogorov-Smirnov test, it discerns differences in central location, variability or skewness, as well as other possible differences. The Runs test ranks the combined values from both groups, and the number of runs (sequences from a given group) are counted. If the two groups come from the same population, there should be many runs — the data values from one group should be interspersed among those from the other group. Similarly, if the groups come from different populations, there should be few runs. An exact probability is calculated when the total number of values is 40 or fewer. The normal approximation is used when there are more than 40 values, with a correction for continuity when the total is less than 50.

The null hypothesis is that the two samples have the same distribution; that is, that the values in one sample are the same size as those in the other sample, have the same variability and are similar in other respects as well. The alternative hypothesis is that the two samples have different distributions, and that the values in one distribution differ in some way from those in the other. The data should be at least ordinal (ordered) values from distributions assumed to be continuous.

Figure 5.18 Data for Wald-Wolfowitz Runs Test

FILE S139:

Boys	Girls
86	55
69	40
72	22
65	58
113	16
65	7
118	9
45	16
141	26
104	36
41	20
50	15

Figures 5.18 and 5.19 illustrate the Runs test. The data in Figure 5.18 are scores on aggression observed in boys and girls during two 15-minute play periods. The null hypothesis is that the two groups come from the same population (four-year-olds), that there are no sex differences in aggression. The alternative hypothesis is that the two groups come from different populations — they differ in aggression. (Siegel)

Figure 5.19 shows the command requesting the Runs test:

```
NP.TEST  S139

[ SPLIT 2,  CREATE  Aggression  Boys  Girls,  INDEX  Boys.Girls ],

 RUNS,  GROUP  Boys.Girls  $
```

and the output. The SPLIT instruction is used to rearrange the data (the values of Boys and Girls are split and become the values of Aggression) and to create an index (the variable Boys.Girls with values of 1 for boys and 2 for girls) as the file is input to NP.TEST. The required RUNS and GROUP identifiers are included in the command, and the optional WEIGHT identifier could be used if each case represented more than a single observation. The Runs test is calculated for each variable in the input file, other than the grouping and weight variables.

The output contains the number of runs, the test statistic Z and the associated one-tail probability. Since the total number of values is 40 or fewer, this is an exact probability. (When there are values in both samples that are tied, the maximum and minimum number of runs is also reported in the output.) The two-tailed probability, appropriate for the two-tailed hypotheses, is twice the one-tailed one or .0002. This very small probability of obtaining such a low number of runs permits rejecting the null hypothesis and accepting the alternative one. There is a difference in observed aggression between the four-year-old boys and girls.

The Runs test is less powerful than the Mann-Whitney U test in testing specifically that two samples come from populations with the same central location. However, the Runs test is able to test the null hypothesis against many other alternatives in addition to differences in central location.

Figure 5.19 Wald-Wolfowitz Runs Test of Equal Distributions

```
   NP.TEST  S139

     [ SPLIT 2,  CREATE  Aggression  Boys  Girls,  INDEX  Boys.Girls],

      RUNS,  GROUP  Boys.Girls  $


------ Wald-Wolfowitz Runs test: file S139
------ variable Aggression

     CASES
        12  Boys.Girls = 1
        12  Boys.Girls = 2
     -----
        24  TOTAL

      runs         Z    1-tailed P
         4   -3.5481        0.0001
```

5.13 Squared Ranks Test

The Squared Ranks test calculates the probability that the variances of two samples are equal — that the samples come from either the same population or from different populations with identical distributions except for the central locations. (The Squared Ranks test may be generalized to more than two independent samples. See the subsequent section “K-INDEPENDENT-SAMPLE TESTS.”)

Figure 5.20 Data for Squared Ranks Test

FILE C243:

Amount In Box	Present New
10.8	1
11.1	1
10.4	1
10.1	1
11.3	1
10.8	2
10.5	2
11.0	2
10.9	2
10.8	2
10.7	2
10.8	2

The absolute deviation of each value from the mean of its sample is computed. The combined deviations from both samples are ranked. The test statistic (T1) is the sum of the squares of the ranks in the first of the samples, with a correction when there are ties. The normal approximation to the sampling distribution of the test statistic is used to obtain the probability.

The null hypothesis is that the two samples come from the same population or from different populations having the same variability. The alternative hypothesis is that the two samples come from different populations that do not have the same variability. The data should be at least interval values (ordered values with equal intervals between them).

Figures 5.20 and 5.21 illustrate the Squared Ranks test. The data are measurements of the amount of dry cereal in boxes filled using the present machine and using a new machine (1 is present, 2 is new). The null hypothesis is that the new machine has the same or more variability than the old machine with regard to the amount of cereal put in the boxes. The alternative hypothesis of interest is that new machine has less variability. (Conover)

Figure 5.21 Squared Ranks Test of Equal Variances

```
      NP.TEST  C243,  SQUARED.RANKS,  GROUP  Present.New  $


 ---- Squared ranks test for equal variances: file C243
 ---- variable Amount.In.Box

              mean of the
            squared ranks   cases
                   92.40       5  Present.New = 1
                   26.57       7  Present.New = 2
                  ------   -----
                   54.00      12

                        2-tailed P
             T1     (normal approx)
          2.3273              0.0199
```

The command requesting the Squared Ranks test:

```
      NP.TEST  C243,  SQUARED.RANKS,  GROUP  Present.New  $
```

and the output are shown in Figure 5.21. The SQUARED.RANKS and GROUP identifiers are required. The WEIGHT identifier is optional. The Squared Ranks test is calculated for each variable in the input file, excluding the grouping and weighting variables.

The output shows the mean squared rank of each sample, the test statistic T1 and the corresponding two-tailed probability. Halving this probability for a one-tailed value appropriate for the directional hypotheses yields .0099. This small a probability that the variance of the machines is equal permits rejecting the null hypothesis and accepting the alternative one. The new machine has less variability than the old one. The mean of its squared ranks (of deviations from its mean) is less than the mean of the old machine's squared ranks, which means that the new machine has smaller deviations or less variability.

5.14 TWO-PAIRED-SAMPLE TESTS

Paired samples occur when subjects selected from two samples are matched with regard to one or more background variables, so that any observed differences are due to the treatment or analysis variable (the one of interest). Paired samples also arise when a single subject is used for "before" and "after" measurements — in effect, the subject is perfectly matched with himself. Observed differences are attributed to the treatment since the subject matches (is the same) on all background variables.

The nonparametric tests for two paired samples are the Sign test, the McNemar test and the Wilcoxon Matched-Pairs Signed-Ranks test. These tests calculate the probability that the values observed in each sample are statistically the same. The comparable parametric test is the PAIRED.TTEST command.

The NP.TEST command requires only an input file and the identifier that specifies the desired nonparametric test: SIGN, MCNEMAR or WILCOXON. The input file should contain an even number of variables and, if necessary, an optional weighting variable. NP.TEST assumes that the first half of the variables are "pre" values and the second half are "post" values (the weighting variable is extra); that is, the first half are paired with the second half. Thus, if there are four variables in the file, excluding the weighting variable, the first is paired with the third and the second with the fourth. The output includes two tests, one for each pairing of variables.

The WEIGHT identifier is used when each case represents more than one observation. The values of the weighting variable should be positive integers. Cases without positive integer values or with missing values of the WEIGHT variable are skipped. The WEIGHT variable itself is not analyzed.

Figure 5.22 Data for Sign Test

FILE S70:

Father	Mother
4	2
4	3
5	3
5	3
3	3
2	3
5	3
3	3
1	2
5	3
5	2
5	2
4	5
5	2
5	5
5	3
5	1

5.15 Sign Test

The Sign test is one of the oldest nonparametric tests. It tests the relative sizes of paired variables for equality. The Sign test can use minus and plus signs (entered in the P-STAT system file as zeros and ones, for example), as well as actual values for data. Thus, it is useful when two paired values for a case can be ranked with respect to each other, even though the values cannot be quantified. It is a special case of the Binomial test, with the proportion expected in the either category in the population equal to .05. The Sign test calculates an exact binomial probability that the values of one variable in a pair are the same size as those of the other variable in the pair.

The null hypothesis is that the proportion expected in the plus category (a positive difference between paired values) equals the proportion expected in the minus category (a negative difference) equals .5. In other words, the median difference between the pairs is zero. The alternative hypothesis is that the proportion expected in each category is not .05. The data should be at least ordinal (ordered) values, with the underlying distribution of values assumed to be continuous. The measurement pairs need not all come from the same population.

The data for the Sign test example are shown in Figure 5.22. Fathers, separated from their first born child because of war, and mothers were rated on their insight regarding paternal discipline. (A value of 1 is high insight and a 5 is low insight.) The null hypothesis was that fathers would have as much or more insight than mothers. The alternative hypothesis, predicted by the researcher, was that fathers would have less insight than mothers regarding paternal discipline (possibly because of their briefer association with the child). (Siegel)

The Sign test calculates whether the difference between each pair of measurements is positive or negative. Ties or zero differences are not used in the analysis. This command:

```
NP.TEST  S70,  SIGN  $
```

requests the Sign test. The output, shown in Figure 5.23, reports the number of negative and positive differences, as well as the number of ties (no difference). The probability calculated is that of obtaining 3 negative changes (differences) out of a total of 14 changes (ties are excluded), when the expected proportion is .5. This is an exact binomial probability.

Figure 5.23 Sign Test of Differences Between Paired Values

```
      NP.TEST  S70,  SIGN  $


--------- SIGN test: file S70
--------- variables Father and Mother

          CASES
              3 - changes (Father lt Mother)
             11 + changes (Father gt Mother)
              3   ties
           ----
             17
                  exact 2-tailed P = 0.0574
```

The probability shown in Figure 5.23 is a two-tailed value of .0574. This value should be halved to obtain the one-tailed value of .0287, which is appropriate for the one-tailed hypotheses. This small a probability of obtaining 3 negative differences out of a total of 14 differences, when the expected number of negative differences is 7, permits rejecting the null hypothesis and accepting the alternative one. Fathers, separated from their first born child because of war, have less insight than mothers regarding paternal discipline. (The small number of negative differences implies fathers score higher than mothers — higher numbers on the rating scale represent less insight.)

Figure 5.24 Data (+ and -) for Sign Test

Punishment Favored After Film

Punishment Favored Before Film		Less (-)	More (+)
	More (+)	59	7
	Less (-)	8	26

FILE S73:

Before	After	Count
1	0	59
1	1	7
0	0	8
0	1	26

Figure 5.24 shows another data set for the Sign test. The hypothetical data is shown both in tabular form and the way it is entered in a P-STAT system file. Adults gave their position on whether more or less punishment is necessary for juvenile delinquents both before and after viewing a film about juvenile delinquency. It is only necessary that categorical data, such as this, be orderable. Actual values are not required. The "less" values are entered in the file as "zeros" and the "more" values are entered as "ones". (Any two numbers could be used.) The numbers in each category are the values of the variable Count.

The null hypothesis is that the film does not affect opinions for more or less punishment for juvenile delinquents in a systematic way. Any changes after seeing the film are random, and the same number of negative as positive changes are expected. The alternative hypothesis is that the film does affect opinions, and either more or less negative than positive changes are expected. (Siegel)

Figure 5.25 Sign Test of Differences (+ and - Data)

```
          NP.TEST  S73,  SIGN,  WEIGHT  Count  $


--------- SIGN test: File S73 ---------
--------- variables Before and After

          CASES           (weighted by Count)
             26 - changes (Before lt After)
             59 + changes (Before gt After)
             15   ties
           ----
            100
                 exact 2-tailed P = 0.0004
```

Figure 5.25 shows the command requesting the Sign test and the output. The WEIGHT identifier is used because each case represents multiple observations. The small two-tailed probability (.0004) of obtaining 26 negative changes out of a total of 85 changes, when 42.5 changes are expected, permits rejecting the null hypothesis and accepting the alternative one. The film does affect opinions for more or less punishment for juvenile delinquents (in this hypothetical situation).

The Sign test is used to analyze this data because the values are not independent (they are paired) and because they are merely counts of differences, not actual scores. If the values are independent and actual ranks, the Mann-Whitney test is a more powerful test to use. The Sign test is more powerful for small sample sizes than for large ones.

5.16 McNemar Test for Significance of Changes

The McNemar test is a variant of the Sign test that is suitable for paired data that is merely nominal — two unordered categories. It calculates an exact binomial probability that a change has not occurred, and it gives the chi-square statistic. The null hypothesis is that the proportion of positive changes equals the proportion of negative changes equals .5. The alternative hypothesis is that the proportion of negative and positive changes do not equal .5.

Figure 5.26 Data for McNemar Test

FILE S66:

Day1	Day30	WWW
1	1	4
1	0	14
0	1	4
0	0	3

The data values must be either zeros or ones for the McNemar test. Although they are typically arranged in a two-by-two table, the data is input to NP.TEST as individual cases or as four variable pairs. The latter organization is shown in Figure 5.26. This data set shows counts (variable WWW) of nursery school youngsters by categories of social interaction for the first and thirtieth days of nursery school. A code of 1 indicates an adult-initiated contact and a 0 indicates a child-initiated one. Thus, 14 youngsters changed their mode of social interaction from adult-initiated contacts to child-initiated ones, and four youngsters changed in the opposite direction. Seven youngsters did not change. The null hypothesis was that youngsters would increase the number of child-initiated interactions with experience in nursery school. The alternative hypothesis was that they would not change. (Siegel)

This command requests the McNemar test:

```
NP.TEST  S66,  MCNEMAR,  WEIGHT  WWW  $
```

A WEIGHT variable is specified when each case represents multiple observations. Pairwise deletion of cases occurs when one or two values in a pair of cases is missing or when the value of the weighting variable is either missing or not a positive integer. The McNemar test is computed for all pairs of variables in the input file, excluding the weight variable. (The first half of the variables are paired with the second half.)

The output, shown in Figure 5.27, gives the number of changes (non-changes or ties do not figure in the calculations), a chi-square statistic (not corrected for continuity), and the binomial probability. This is the exact probability of obtaining the number of positive changes observed out of the number of total changes observed, in a population where the proportion of positive and negative changes are each .5. The binomial distribution is suitable for both very small and large data sets. The chi-square distribution is not suitable when the expected frequency in any category is less than 5 (Siegel) or when the number of changes is less than or equal to 20 (Conover).

Figure 5.27 McNemar Test of Significant Change

```
      NP.TEST  S66,  MCNEMAR,  WEIGHT  WWW  $


------ McNemar significance of change test: file S66 ------
------ variables Day1 and Day30
       (weighted by variable WWW)

                            Day30
                          zero     one
                       -----------------
                 zero  |     3 |     4 |
           Day1        |-------|-------|
                  one  |    14 |     4 |
                       -----------------

                                      binomial
    cases    changes     chi square   2-tailed P
       25         18         5.5556        .0309
```

The two-tailed probability value .0309 may be halved to obtain a one-tailed value of .0155, suitable for testing the directional hypotheses. This value is sufficiently small enough to reject the null hypothesis and accept the alternative one. With experience in nursery school, youngsters change their mode of social interaction from adult-initiated contacts to child-initiated ones.

5.17 Wilcoxon Matched-Pairs Test

The Wilcoxon Matched-Pairs Signed-Ranks test calculates the probability that the values of one variable in a pair are statistically the same size as the values of the other variable in the pair. If the data are a random sample, it may also be used to calculate the probability that a sample comes from a population with a particular mean or median. The normal approximation to the sampling distribution of the Wilcoxon Matched-Pairs test statistic is used to obtain the probability.

The Wilcoxon test uses the sizes of the differences between paired values in addition to their directions, unlike the Sign test which uses only the directions of the differences. Thus, the Wilcoxon test is more powerful than the Sign test for data that meet the test's criteria. The data should be at least interval values (ordered values with equal intervals between them), and the distribution of the differences between paired values should be symmetric.

The null hypothesis is that the values in each paired sample are the same size — that the median of the difference ranks is zero. That is, some of the differences should favor one sample and some should favor the other. The alternative hypothesis is that the values in each sample are not the same size, or the median of the difference ranks is not zero. If the data are random values, the hypotheses test means as well as medians. (Conover)

The Wilcoxon test ranks the absolute sizes of the differences between paired values. Tied differences (excluding zero differences) are given the average of the ranks they would have gotten if they were not tied. Then, the sign of each difference is reassociated with its rank. The Wilcoxon test statistic is the smaller sum of similarly signed ranks. This is appropriate only for small samples with no ties. The corresponding test statistic Z, for use in the normal approximation, is given in the Wilcoxon output. This is appropriate for all samples with as few as eight or so data pairs.

Figure 5.28 Data for Wilcoxon Test

FILE S79:

School	Home
82	63
69	42
73	74
43	37
58	51
56	43
76	80
85	82

Figure 5.28 shows the data used in illustrating the Wilcoxon test. The data are scores on a measure of social perceptiveness of twins, one of whom attended nursery school and the other of whom remained home. The scores are interval, rather than ratio, measurements because there is no absolute zero — no complete lack of social perceptiveness. The differences between scores can be ranked, but one difference cannot be thought of as twice the size of another, for example. The null hypothesis is that the social perceptiveness of children attending nursery school is the same as that of children not attending. The alternative hypothesis is that the social perceptiveness of the children differs. (Siegel)

This command:

```
NP.TEST  S79,  WILCOXON   $
```

requests the Wilcoxon Matched-Pairs Signed-Ranks test. The optional WEIGHT identifier may be used. The Wilcoxon test is computed for all pairs of variables, excluding a possible weighting variable. The “pre” variables should precede the “post” variables. The first variable will be paired with the third and the second with the fourth, for example, when there are four variables.

Figure 5.29 Wilcoxon Test of Differences or Central Location

```
NP.TEST  S79,  WILCOXON   $


--- WILCOXON matched-pairs signed-rank test: file S79 ---
--- variable School  with  variable  Home

    MEAN RANK   CASES
         2.00       2  - RANKS (School lt Home)
         5.33       6  + RANKS (School gt Home)
                    0    TIES
                  ----
                    8    TOTAL

      Z =   -1.9604          2-tailed P = 0.0499
```

Figure 5.29 shows the output from the Wilcoxon test. The mean rank of each group is given, as well the test statistic Z and the associated two-tailed probability. The small probability (.0499) of the same size values of social perceptiveness in both groups permits rejecting the null hypothesis and accepting the alternative one. The social perceptiveness of children attending nursery school differs from that of children not attending.

Figure 5.30 Data for Wilcoxon Test

FILE C284:

Xi	X2	X3	X4	X5	X6
23.8	26.0	26.9	27.4	28.0	30.3
30.7	31.2	31.3	32.8	33.2	33.9
34.3	34.9	35.0	35.9	36.1	36.4
36.6	37.2	37.3	37.9	38.2	39.6
40.6	41.1	42.3	42.8	44.0	45.8

WILCOXON may to used to test if the difference between the mean of a single random sample and a population mean (or an arbitrary constant) is zero — that is, if the sample mean equals a known value. Similarly, it may be used for the same test regarding medians. Figure 5.30 shows thirty ordered observations of Xi, a random variable. The null hypothesis is that the mean of the observations is 30 or less. The alternative hypothesis is that the mean is greater than 30. (Conover)

Figure 5.31 Wilcoxon Test of Population Mean

```
      NP.TEST  C284  [ SPLIT  6;  GEN  Mean  =  30 ],  WILCOXON  $


 --- WILCOXON matched-pairs signed-rank test: file C284 ---
 --- variable Xi  with  variable  Mean

     MEAN RANK   CASES
          9.40       5  - RANKS (Xi lt Mean)
         16.72      25  + RANKS (Xi gt Mean)
                     0    TIES
                   ----
                    30    TOTAL

       Z =   -3.8154          2-tailed P = 0.0001
```

Figure 5.31 shows the Wilcoxon test of these hypotheses. As the input file is read by NP.TEST, the data is rearranged and a variable is generated equal to the population mean or median:

```
NP.TEST  C284  [ SPLIT  6;  GEN  Mean  =  30 ],  WILCOXON  $
```

(SPLIT 6 breaks each case into six cases.) The very small probability of .0001, which may be halved for a one-tailed probability of .00005, permits rejecting the null hypothesis that the mean of the sample is 30 or less and ac-

cepting the alternative one that the mean is greater than 30. (There are 25 cases where Xi is greater than the mean and only 5 where Xi is less.)

A file containing just the differences between paired values may be input for the Wilcoxon test. A variable equal to zero should be generated:

```
NP.TEST  S82  [ KEEP  Differences;  GEN  Zero  =  0 ],  WILCOXON  $
```

The variable Zero serves as the comparison for the variable Differences.

5.18 K-INDEPENDENT-SAMPLE TESTS

"K" (meaning two or more) sample nonparametric tests compare a number of independent groups. They test for equality of central location or of variability. The comparable parametric test is analysis of variance.

The tests for k independent samples that may be requested in NP.TEST are the Median test, the Kruskal-Wallis one-way analysis of variance, and the Squared Ranks test. The Median and Squared Ranks tests are extensions of the same tests for two groups to more than two groups. The NP.TEST command requires one identifier to indicate which test is desired — MEDIAN, KRUSKAL.WALLIS and SQUARED.RANKS, the GROUP identifier and, when there are more than two groups, the NG (number of groups) identifier. Optional identifiers are WEIGHT and CONTRASTS (CONTRASTS is used only with SQUARED.RANKS).

The GROUP identifier specifies the variable whose values define membership in the different groups:

```
NP.TEST  S182,  MEDIAN,  NG  6,  GROUP  Mothers.Ed  $
```

The NG identifier specifies the number of groups. (If NG is omitted, it is assumed that there are only two groups.) The values of the grouping variable (Mothers.Ed) are assumed to be 1 through 6 when NG specifies that there are 6 groups. If the values defining group membership are different, they should be specified after GROUP and the variable name:

```
NP.TEST  S182,  MEDIAN,  NG  6,  GROUP  Mothers.Ed  0 1 2 3 4 5  $
```

If only 5 values are specified when NG is 6, all non-missing values of the grouping variable define the sixth group.

The WEIGHT identifier is used when each case represents more than one observation. It is followed by the name of the weighting variable whose values give the number of observations each case represents. The weight for a case should be a positive integer, or that case will be skipped. When WEIGHT is not used, each case in the input file counts as one observation.

5.19 Median Test

The Median test for more than two independent groups is almost identical to the same test for two groups. (See the earlier section on the Median test in "TWO-INDEPENDENT-SAMPLE TESTS" for background information.) The Median test calculates the probability that the medians of the k independent samples are statistically the same. This is the null hypothesis. The alternative hypothesis is that at least one median is not the same. It is not possible to say which sample medians are not the same without additional tests.

The Median test is calculated for each variable in the input file, with the exception of the grouping and weighting variables. The probability is obtained from the chi-square approximation to the sampling distribution. With more than two groups, there is no correction for continuity.

The data for the Median test with k groups is shown in Figure 5.32. The data are the number of voluntary visits made by mothers to their child's school, grouped by the highest level of education completed by the mother. The null hypothesis is that there is no difference in the number of visits (an indication of the interest the mother has in the schooling of her child) made by mothers with different levels of education. The alternative or predicted hypothesis is that the number of visits varies with the educational level of the mother. (Siegel)

Figure 5.32 Data for Median Test with K Groups

FILE S182:

Eight Grade	Tenth Grade	High School	Some College	College Grad	Grad School
4	2	2	9	2	2
3	4	0	4	4	6
0	1	4	2	5	-
7	6	3	3	2	-
1	3	8	-	-	-
2	0	0	-	-	-
0	2	5	-	-	-
3	5	2	-	-	-
5	1	1	-	-	-
1	2	7	-	-	-
-	1	6	-	-	-
-	-	5	-	-	-
-	-	1	-	-	-

The command requesting the Median test and the output are shown in Figure 5.33. The data are rearranged and an index is created as the file is input to the NP.TEST command:

```
NP.TEST  S182
[ SPLIT  6,  CREATE  Visits  V(1)  TO  V(6),  INDEX  Mothers.Ed ],
MEDIAN,  NG  6,
GROUP  Mothers.Ed  $
```

The MEDIAN identifier requests a median test for each variable in the input file except the GROUP variable and the WEIGHT variable, if one has been named. An arbitrary value, to be used in place of the actual median to dichotomize the values in each group, may follow MEDIAN. The required NG and GROUP identifiers give the number of groups and specify the variable that defines group membership.

The output gives the overall median and the number of cases in each group that are greater than and less than that median. A chi-square and its associated “two-tailed” (nondirectional) probability are computed — they are 1.8951 and .8635. This is a nonsignificant chi square value.

The smallest of the expected frequencies for the cells or categories above and below the median in each group is 1. There are some restrictions on the size of the expected frequencies for each category when using a chi-square statistic. (See the discussion of these restrictions in the earlier section “Chi-Square Test.”) A smallest expected value of 1 just meets part of these restrictions. However, in addition, no more than 20 percent of the expected frequencies should be less than 5. The expected frequencies above and below the median in any group are merely half the observed frequency in a group. Since three groups (Mothers.Ed values 4, 5 and 6) have total counts of only 4, 4 and 2, respectively, 50 percent of the expected frequencies are below 5. Combining groups that may meaningfully merged solves the problem of too small expected frequencies:

Figure 5.33 Median Test of Equal Medians with K Groups

```
      NP.TEST  S182

       [ SPLIT  6,  CREATE  Visits  V(1)  TO  V(6),  INDEX  Mothers.Ed] ,

       MEDIAN,  NG  6,  GROUP  Mothers.Ed  $


------ 6-sample MEDIAN test: file S182 ------
------ variable Visits

   gt median   le median   total
           5           5      10  Mothers.Ed = 1
           4           7      11  Mothers.Ed = 2
           7           6      13  Mothers.Ed = 3
           3           1       4  Mothers.Ed = 4
           2           2       4  Mothers.Ed = 5
           1           1       2  Mothers.Ed = 6
         ----        ----    ----
          22          22      44  TOTAL

 6 cells have an expected value less than five.

                      smallest
                      expected
   cases      median      value   DF   chi-square   significance

      44       2.500       1.00    5       1.8951          .8635
```

```
      NP.TEST  S182

       [ SPLIT  6,  CREATE  Visits  V(1)  TO  V(6),  INDEX  Mothers.Ed;
         SET  Mothers.Ed  =  RECODE  (Mothers.Ed,  4  TO  6  =  4) ],

       MEDIAN,  NG  6,  GROUP  Mothers.Ed  $
```

All of the college groups are combined to form one group as the file is input to NP.TEST for the Median test. The new chi-square statistic is 1.2951 and its associated probability is .7303. This chi square is still not significant, and so the null hypothesis is accepted. There is no difference (in this fictitious data) in the number of visits, made by mothers with different levels of education, to their child's school.

If the null hypothesis had been rejected and the alternative one accepted (that is, there is a difference in the number of visits made by mothers with varying educational levels), it would not be possible to conclude which groups of mothers differ without additional testing. The Median test could be used on subsets of any two or more groups to find the groups that differ. However, the significance levels of subsequent tests can not be interpreted like that of the first test. (Conover)

5.20 Kruskal-Wallis One-Way ANOVA

The Kruskal-Wallis one-way analysis of variance by ranks is an extension of the Mann-Whitney U test to more than two groups. Kruskal-Wallis calculates the probability that k independent groups come from the same population or from populations with identical means. It is one of the most powerful of the nonparametric tests for multiple independent groups. The test statistic is distributed as chi square with k - 1 degrees of freedom. The comparable parametric procedure is the F test.

The null hypothesis is that the k groups have the same distributions — that they have equal means. The alternative hypothesis is that at least one of the groups has a different distribution — that it has a larger or smaller mean. The data should be at least ordinal (ordered) measurements and the underlying distribution should be continuous.

Figures 5.34 and 5.35 illustrate the Kruskal-Wallis test. The data are hypothetical measurements of authoritarianism for educators in three positions: teachers (position 1), aspiring administrators (position 2) and administrators (position 3). The null hypothesis is that there is no difference in authoritarianism among the three positions. The alternative hypothesis is that there is a difference. (Siegel)

Figure 5.34 Data for Kruskal-Wallis One-Way ANOVA

FILE S187:

Authoritarianism	Position
96	1
128	1
83	1
61	1
101	1
82	2
124	2
132	2
135	2
109	2
115	3
149	3
166	3
147	3

The measurements are pooled and then ranked from the smallest through the largest. The ranks in each group are summed and the mean rank of each group is computed. If the groups have equal means, they should have approximately equal mean ranks. Kruskal-Wallis calculates the probability that the calculated mean ranks come from samples from the same population.

This command requests the Kruskal-Wallis test:

```
NP.TEST  S187,  KRUSKAL.WALLIS,  NG  3,  GROUP  Position  $
```

The NG identifier gives the number of groups (two are assumed when it is omitted) and the GROUP identifier gives the name of the variable whose values define group membership. These values should go from 1 through 3 since three groups are specified. If they do not, the values should be given after the grouping variable:

```
NP.TEST  S187,  KRUSKAL.WALLIS,  NG  3,  GROUP  Position  2 4 5  $
```

The identifier KW is a synonym for KRUSKAL.WALLIS. The Kruskal-Wallis test is calculated for each variable in the input file except the grouping and weight variables.

The output, shown in Figure 5.35, gives the mean rank and the number of cases for each group, the test statistic chi-square and the significance (probability) of a chi-square of that size (with degrees of freedom equal to the number of groups minus one). When there are ties in the ranks, the chi-square is corrected for ties and both the uncorrected and the corrected chi-squares and their significances are shown. The chi-square probability is appropriate for testing non-directional hypotheses; that is, it is "two-tailed."

The significance value .0406 (in Figure 5.35) is sufficiently small enough to reject the null hypothesis and accept the alternative one. There is a difference (in this fictitious data) in authoritarianism among the three educator positions. It is not possible to say which positions differ, however, without further testing.

The Kruskal-Wallis and the Median tests for k-independent groups are generally both suitable for the same types of data. However, the Kruskal-Wallis test is more powerful than the Median test because it uses the relative order of the scores rather than just their position above or below the median.

Figure 5.35 Kruskal-Wallis Test of Equal Distributions

```
    NP.TEST  S187,  KRUSKAL.WALLIS,  NG  3,  GROUP  Position  $


------ Kruskal-Wallis 1-way anova: file S187 ------
------ variable Authoritarianism

  MEAN RANK    CASES
       4.40        5  Position = 1
       7.40        5  Position = 2
      11.50        4  Position = 3
                -----
                   14  TOTAL

   cases    chi-square  significance
      14        6.4057        0.0406
```

5.21 Squared Ranks Test

The Squared Ranks test for k-independent groups is an extension of the same test for two groups. (See the earlier section on the Squared Ranks test in "TWO-INDEPENDENT-SAMPLE TESTS" for background information.) The Squared Ranks test calculates the probability that the variances of k groups are equal — that the groups come from the same population or from populations with identical distributions except for the central locations.

The multi-group test statistic (T2) is calculated in the same way as is the two-group test statistic (T1), and there is a similar correction for tied ranks. However, with more than two groups, the chi-square distribution with k - 1 degrees of freedom is used to obtain the corresponding probability and contrasts of all pairs of groups may be requested.

The null hypothesis is that all samples come from the same population or from different populations with equal variances. The alternative hypothesis is that one or more of the samples come from populations with different variances. When the null hypothesis is rejected, contrasts may be evaluated to calculate which sample pairs come from different populations. The data should be at least interval values.

Figure 5.36 Data for Squared Ranks Test

FILE C249:

Method	Inc .1	Inc .2	Inc .3	Inc .4	Inc .5	Inc .6	Inc .7	Inc .8	Inc .9	Inc .10	Inc .11
1	0.7	1.0	2.0	1.4	0.5	0.8	1.0	1.1	1.9	1.2	1.5
2	1.7	2.1	-0.4	0.	1.0	1.1	0.9	2.3	1.3	0.4	0.5
3	0.9	0.9	1.0	0.	0.1	-0.6	2.2	-0.3	0.6	2.4	2.5

Figure 5.36 shows the data used in illustrating the Squared Ranks test. Three methods of instruction for fifth grade students are compared by measuring the increase in performance over the course of the school year. The null hypothesis is that there is no difference in the variability of the three methods. The alternative hypothesis is that some of the methods have more or less variability than others. (Conover)

This command requests the Squared Ranks test for multiple groups:

```
NP.TEST  C249

 [ SPLIT 11,  CARRY  Method,  CREATE  Increase  Inc? ],

 SQUARED.RANKS,  NG 3,  GROUP Method,  CONTRASTS $
```

Each case of eleven measurements is split into separate cases as the file is input to the NP.TEST command. The relevant Method information is carried with each measurement and the measurements are named “Increase”. The SQUARED.RANKS identifier requests this particular test, and the NG and GROUP identifiers specify the number of groups and the variable whose values define group membership. CONTRASTS requests pairwise comparisons of the groups. The Squared Ranks test is computed for all variables in the file, except for the grouping and weight (if used) variables.

The output, in Figure 5.37, shows the mean squared rank for each group and for the combined groups. The test statistic (T2) and its significance are also given. This probability value is appropriate for testing non-directional (“two-tailed”) hypotheses. A value of .0769 is not sufficiently small enough to reject the null hypothesis. There is statistically no difference in variability among the three methods of instruction. Therefore, pairwise contrasts of the methods are not justified with this data. CONTRASTS were requested in the initial Squared Ranks analysis in case the null hypothesis was rejected. (The seemingly significant difference between groups 1 and 3 capitalizes on chance. Multiple pairwise tests increase the probability of falsely rejecting the null hypothesis, and thus they are not justified when the overall test is not significant.)

When the null hypothesis is rejected, the pairwise contrasts indicate which groups have different variances. The t distribution provides the probability, which is two-tailed. Generally, the level of significance deemed sufficient to reject the null hypotheses is used to evaluate the contrasts. (Conover)

Figure 5.37 Squared Ranks Test for Equal Variances

```
     NP.TEST  C249

        [ SPLIT 11,  CARRY  Method,  CREATE  Increase  Inc? ],

        SQUARED.RANKS,  NG 3,  GROUP Method,  CONTRASTS $


 ---- Squared ranks test for equal variances: file C249 ----
 ---- variable Increase

              mean of the
             squared ranks   cases
                   212.59      11  Method = 1
                   386.64      11  Method = 2
                   539.68      11  Method = 3
                   ------   -----
                   379.64      33

                      significance
              T2     (chi-sq approx)
          5.1300              0.0769

          groups           t    2-tailed P
           1   2     -1.2726        0.2129
           1   3     -2.2492        0.0320
           2   3     -1.1175        0.2726
```

5.22 K-PAIRED-SAMPLE TESTS

"K" (meaning two or more) sample nonparametric tests compare a number of "paired" groups — matching or blocking more accurately describes the relationship among more than two groups. Typically, the data is a tabular array with the variables (the columns) representing the treatment groups and the cases (the rows) representing groups of matched subjects. (See the earlier section, "TWO-PAIRED-SAMPLE TESTS," for a brief discussion of the reasons for pairing or matching subjects.) These procedures test that the treatment groups come from the same population; that is, that the different treatments yield equal results. The comparable parametric test is analysis of variance (an F test).

The tests for k paired samples that may be requested in NP.TEST are the Cochran Q test, the Friedman two-way analysis of variance and the Kendall Coefficient of Concordance W. The NP.TEST command requires an input file and the identifier that specifies the desired nonparametric test: COCHRAN, FRIEDMAN or CONCORDANCE. The input file should contain only the data for a single analysis (one test) and possibly a weighting variable. Cases with missing data in any variable are deleted from the analysis.

The WEIGHT identifier should be used when each case represents multiple observations. The values of the weighting variable should be positive integers. Cases without positive integer values or with missing values of the WEIGHT variable are skipped. The WEIGHT variable itself is not analyzed.

5.23 Cochran Q Test

The Cochran Q test is an extension of the two-paired-sample McNemar test to more than two groups. It calculates the probability that k treatment groups are equally effective. The Q statistic is distributed approximately as chi-square with k - 1 degrees of freedom.

The Cochran Q test is suitable for nominal data or dichotomized ordinal data — that is, data that has only two values. The values typically represent categories such as “yes” and “no”, “pass” and “fail”, and “change” and “no change”. The actual values in the input file must be zeros and ones, and the data should be organized so that the variables (columns) represent the different treatment groups and the cases (rows) represent the different subject groups or blocks. One analysis is calculated for the input file.

Figure 5.38 Data for Cochran Q Test

FILE S164:

Int1	Int2	Int3
0	0	0
1	1	0
0	1	0
0	0	0
1	0	0
1	1	0
1	1	0
0	1	0
1	0	0
0	0	0
1	1	1
1	1	1
1	1	0
1	1	0
1	1	0
1	1	1
1	1	0
1	1	0

Figure 5.38 shows the input for the Cochran Q test. The three variables are types of interviewing style: friendly, formal and harsh. The cases are the responses of groups of housewives, matched on background variables, to a particular question: 0 is a negative response and 1 is a positive one. The null hypothesis is that the probability of a positive response is the same for all styles of interviewing. The alternative hypothesis is that the probabilities are different. (Siegel)

This command inputs the data file and requests the Cochran Q test:

```
NP.TEST  S164,  COCHRAN  $
```

The WEIGHT identifier could be used if the input file contained counts observed in each of the eight possible blocks (2 x 2 x 2 = 8, when the number of treatment groups is three), or if the data was arbitrarily weighted. (See the earlier section on the McNemar test for an example of an input file with counts observed in each block.)

Figure 5.39 Cochran Q Test for Equal Frequencies

```
   NP.TEST  S164,  COCHRAN  $


------ Cochran Q test: file S164 ------

     ---count of---
     zeros     ones  variable
         5       13  Int1
         5       13  Int2
        15        3  Int3

     cases        Cochran Q     DF     significance
        18          16.6667      2             .0002
```

Figure 5.39 shows the output produced by the Cochran Q test. Counts of the zeros and ones observed in each treatment group are given. The test statistic is Cochran's Q and the significance is the chi-square approximation of the probability of obtaining that value in a population where equal effects are expected for each treatment group. This probability is appropriate for testing non-directional or "two-tailed" hypotheses, and the value .0002 is sufficiently small to reject the null hypothesis and accept the alternative one. Different interviewing styles have different effects in this artificial data set. The Cochran Q test is not recommended when the number of subject groups or blocks (the number of cases) is too small.

5.24 Friedman Two-Way ANOVA

The Friedman two-way analysis of variance by ranks (devised by economist Milton Friedman) is an extension of the two-paired-sample Sign test. It calculates the probability that k samples come from the same population. The test statistic is distributed as chi-square with k - 1 degrees of freedom.

The null hypothesis is that the k treatments yield identical effects. The alternative hypothesis is that the treatments have different effects. The data measurements should be at least ordinal values. The data in each of the matched groups (the cases) is ranked separately. The mean rank of each of the treatment groups (the variables) is calculated. If the treatments do not have differential effects, the mean ranks should be the same.

Figure 5.40 shows the data set used to illustrate the Friedman test. The subjects were 54 rats, in 18 matched groups of littermates, that were trained to run to white using three different reinforcement treatments. The first treatment group (RR) received reinforcement (reward) after each learning trial, the second group (RU) received partial reinforcement with the last trial in a session unreinforced, and the third group (UR) received partial reinforcement with the last trial reinforced.

The data are rankings of the number of errors made by rats in learning opposing behavior (unlearning the initial behavior of running to white and learning to run to black). The rats with the strongest learning should make the most errors unlearning. Thus, a rank of 1 means highest initial learning, as measured by the most errors unlearning. The null hypothesis is that the type of reinforcement has no effect on the strength of learning. The alternative hypothesis is that the type of reinforcement does affect learning strength. (Siegel)

Figure 5.40 Data for Friedman Two-Way ANOVA

```
FILE  S171:

  RR      RU     UR

 1.0     3.0     2
 2.0     3.0     1
 1.0     3.0     2
 1.0     2.0     3
 3.0     1.0     2
 2.0     3.0     1

 3.0     2.0     1
 1.0     3.0     2
 3.0     1.0     2
 3.0     1.0     2
 2.0     3.0     1
 2.0     3.0     1

 3.0     2.0     1
 2.0     3.0     1
 2.5     2.5     1
 3.0     2.0     1
 3.0     2.0     1
 2.0     3.0     1
```

Figure 5.41 shows the output from the Friedman two-way ANOVA. The mean rank of each treatment group (each variable) is given, as well as the chi-square and its significance. This probability is appropriate for testing nondirectional ("two-tailed") hypotheses. The value .0137 is sufficient small to reject the null hypothesis and accept the alternative one. Differential reinforcement affects the strength of learning in rats, as measured by errors made in unlearning an original behavior and learning a competitive one.

Figure 5.41 Friedman Test of Equal Distributions

```
     NP.TEST  S171,  FRIEDMAN  $


------ Friedman two-way anova: file S171 ------

  MEAN RANK    VARIABLE
       2.19    RR
       2.36    RU
       1.44    UR

    cases       chi-square     DF     significance
       18           8.5833      2            .0137
```

The Friedman test is more powerful than the Cochran Q test. It is approximately equal in power to the comparable parametric procedure, the F test, and it is suitable for both small and large size samples. (Siegel)

5.25 Kendall Coefficient of Concordance W

The Coefficient of Concordance W is a modification of the Friedman test statistic, and it is used in the same situations that Friedman is used. W measures the degree of association or agreement among k sets of rankings, much as the Spearman and Kendall correlation coefficients measure association between two sets of rankings. W may be thought of as the average of the various pairwise correlations. The chi-square distribution, with degrees of freedom N - 1, is used to obtain the significance of the coefficient of concordance.

The measurements should be at least ordinal values. The data in each of the matched groups (the cases) is ranked separately. The sum of the ranks in each of the treatment groups (the variables) and the overall mean sum of the ranks is computed. If there is low association among the rankings, the group sums should not be very different from the mean sum — that is, each group would have a mixture of rankings and not solely the highest or lowest rankings. The squares of the deviations of each group sum of ranks from the mean sum of ranks is used to compute W. There is a correction for tied rankings.

Figure 5.42 Data for Kendall Coefficient of Concordance

FILE S234:

Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8	Var9	Var10
1.0	4.5	2.0	4.5	3.0	7.5	6	9.0	7.5	10.0
2.5	1.0	2.5	4.5	4.5	8.0	9	6.5	10.0	6.5
2.0	1.0	4.5	4.5	4.5	4.5	8	8.0	8.0	10.0

Figures 5.42 and 5.43 show the data set and the Kendall test of concordance. The data are rankings given ten objects on three different aspects. The null hypothesis is that the k sets of rankings are independent; that is, there is little or no association among the rankings. The alternative hypothesis is that the rankings are not independent; there is agreement among the rankings given the ten objects. (Siegel)

This command:

```
NP.TEST  S234,  CONCORDANCE  $
```

requests the Kendall test. The WEIGHT identifier may be used when the input file contains counts of observations. Any cases with missing data are not included. One analysis is computed for the whole file.

Figure 5.43 shows the output produced by the prior command. The mean rank of each treatment group (this is a better comparison statistic than the sum of the ranks), the coefficient of concordance W, the chi-square statistic and its significance are given. A value of .0078 is sufficiently small to reject the null hypothesis and accept the alternative one — there is agreement on the rankings given the ten objects.

The significance of the Kendall coefficient of concordance W, obtained from the chi-square distribution, is suitable for testing non-directional (“two-tailed”) hypotheses. It is not appropriate when the total number of observations is less than seven. (Siegel)

Figure 5.43 Kendall Test of Concordance in Rankings

```
     NP.TEST  S234,  CONCORDANCE  $


 --- Kendall coefficient of concordance (W): file S234 ---

   MEAN RANK    VARIABLE
        1.83    Var1
        2.17    Var2
        3.00    Var3
        4.50    Var4
        4.00    Var5
        6.67    Var6
        7.67    Var7
        7.83    Var8
        8.50    Var9
        8.83    Var10

    CASES       W   CHI-SQUARE   D.F.   SIGNIFICANCE
        3  0.8277      22.3487      9         0.0078
```

5.26 RANK CORRELATION TESTS

Nonparametric correlation tests compute the degree of association between two groups of ordinal measurements and the corresponding significance level. The comparable parametric procedure is the Pearson product-moment correlation coefficient r. (See the CORRELATE chapter for additional background information on correlation in general and on various other nonparametric correlations.)

NP.COR is the nonparametric correlation command. The correlation coefficients and significances that may be computed are the Spearman Rank Correlation Coefficient rho and the Kendall Rank Correlation Coefficient tau. The NP.COR command requires an input file and the identifier OUT, which should be followed by a name for the output file that is to contain the correlation coefficients and significances. The Spearman correlation coefficient is computed, unless the identifier KENDALL specifies that the Kendall correlation coefficient is desired.

The output files produced by either the Spearman or Kendall tests are correlation matrices of all variable pairings. They contain the correlation coefficients, their significances, and the number of good pairs of data used in the computations. There is pairwise deletion of missing data values, and both the Spearman and Kendall statistics are adjusted for tied rankings. The LIST command is used to print the correlation matrices.

5.27 Spearman Rank Correlation

The Spearman correlation coefficient is actually the same as the Pearson product-moment correlation coefficient computed on ranked data. The correlation coefficient rho has values from -1 to +1 for negative and positive associations and values close to zero for little or no association. The null hypothesis is that the pair of variables is independent — there is no correlation between them. The alternative hypothesis is that the variables are not independent, but positively or negatively correlated. The data should be at least ordinal measurements. The probability of obtaining specific values of rho is obtained from the t distribution.

Figure 5.44 Data for Spearman and Kendall Rank Correlation

FILE S205:

Authoritarianism	Social Status
82	42
98	46
87	39
40	37
116	65
113	88
111	86
83	56
85	62
126	92
106	54
117	81

Figure 5.44 shows the data set used in illustrating both the Spearman and Kendall tests. The data are measures of authoritarianism and social status strivings for 12 college students. The null hypothesis is that these measures are independent; the alternative hypothesis is that they are associated. (Siegel) This command:

```
NP.COR  S205,  OUT  S205.Sp  $
```

requests the Spearman test and provides a name for the output file. The values for each measure are ranked separately, and the correlation between the rankings is computed.

Figure 5.45 shows a listing of the output correlation matrix. The LIST command is used to print the matrix:

```
LIST  S205.Sp,  STUB  Variable  Statistic,  SKIP 3   $
```

The identifiers STUB and SKIP make the listing easier to read. STUB positions the variables VARIABLE and STATISTIC on the left of each page of the listing and SKIP 3 skips a line after every three lines, thus separating the variables.

The output matrix is symmetrical about the diagonal. The value of rho for Authoritarianism and Social.Status is .8182, and its significance is .0006. The significance is a one-tailed value, which may be doubled to obtain the two-tailed value .0012 that is appropriate for testing the non-directional hypotheses. (If, for example, the alternative hypothesis predicted a positive correlation between Authoritarianism and Social.Status, the one-tailed significance value would be used.) A value of .0012 is more than sufficient to reject the null hypothesis and accept the alternative one: authoritarianism and social status strivings are associated.

The Spearman rank correlation is only slightly less efficient than the Pearson parametric correlation. The calculated significance levels are appropriate for all but very small data sets (less than ten data pairs). (Siegel)

Figure 5.45 Spearman Test of Rank Correlation

```
NP.COR  S205,  OUT  S205.Sp  $

SPEARMAN rank correlation completed.

LIST  S205.Sp,  STUB  Variable  Statistic,  SKIP 3   $


FILE  S205.Sp

SPEARMAN rank correlations from input file S205

                                                    Social
VARIABLE          STATISTIC  Authoritarianism       Status

Authoritarianism  cor                 1.0000       0.8182
                  n                  12.0000      12.0000
                  sig(1)              0.          0.0006

Social.Status     cor                 0.8182       1.0000
                  n                  12.0000      12.0000
                  sig(1)              0.0006       0.
```

Figure 5.46 Kendall Test of Rank Correlation

```
NP.COR  S205,  KENDALL,  OUT  S205.Ken  $

KENDALL rank correlation completed.

LIST  S205.Ken,  STUB  Variable  Statistic,  SKIP  3  $


FILE  S205.Ken

KENDALL rank correlations from input file S205

                                                    Social
VARIABLE          STATISTIC  Authoritarianism       Status

Authoritarianism  cor                 1.0000       0.6667
                  n                  12.0000      12.0000
                  sig(1)              0.          0.0013

Social.Status     cor                 0.6667       1.0000
                  n                  12.0000      12.0000
                  sig(1)              0.0013       0.
```

5.28 Kendall Rank Correlation

NThe Kendall correlation coefficient is appropriate for the same type of data and tests as the Spearman coefficient. The probability of obtaining a given value of tau is exact when the number of data pairs is ten or fewer and is approximated by the normal distribution when the number of pairs is greater than ten.

The same data set (shown in Figure 5.44) is used to illustrate the Kendall test. The KENDALL identifier must be included in the command:

```
NP.COR  S205,  KENDALL,  OUT  S205.Ken  $
```

to specify Kendall rank correlation and not Spearman. Figure 5.46 shows the listing of the output matrix. The value of tau is .6667, and it has a one-tailed significance of .0013. Doubled, this value is .0026, which is more than significant to reject the null hypothesis and accept the alternative one. Thus, the Kendall test results in the same decision as the Spearman test, even though the correlation coefficients are not directly comparable. The two tests are equally powerful. (Siegel)

REFERENCES

1. Conover, W.J. (1980). *Practical Nonparametric Statistics,* 2nd ed. John Wiley & Sons, New York.
2. Siegel, Sidney. (1956). *Nonparametric Statistics for the Behavioral Sciences,* McGraw-Hill Book Company, Inc., New York.
3. Siegel, Sidney and Castellan, N. John, Jr. (1988). *Nonparametric Statistics for the Behavioral Sciences,* 2nd edition, McGraw-Hill Book Company, Inc., New York.
4. Snedecor, George W., and Cochran, William G. (1980). *Statistical Methods,* 7th ed. The Iowa State University Press, Ames, p. 145.
5. Sprent, P. (1989). *Applied Nonparametric Statistical Methods,* Chapman and Hill, London.

SUMMARY

NP.TEST

One-Sample Tests: BINOMIAL

```
NP.TEST,  BINOMIAL,  COUNTS  16  2  $

NP.TEST   DieRolls,  BINOMIAL .17,  EQ  1  $

NP.TEST   S40,  BINOMIAL,  COUNTS  $
```

The binomial test calculates the probability of k observations in a one of *two categories* out of N total observations in a sample, using the expected probability of the population. The data need be only nominal (categorical) values. The expected probability of an observation in the category of interest is assumed to be .5, unless a different probability follows the BINOMIAL identifier.

With BINOMIAL, the NP.TEST command expects either: 1) no input file and the observed frequencies for each of two categories provided as arguments for the COUNTS identifier, 2) an input file with each case a single observation and a boundary-defining identifier (EQ, GE, GT, LE, LT or NE), or 3) an input file with two cases containing the observed frequencies and the COUNTS identifier without an argument. These three options are illustrated above.

Required:

NP.TEST fn

specifies the optional input file. When a file name does not follow NP.TEST, the identifier COUNTS should be used to give the number of cases in each category.

Required Identifiers:

BINOMIAL nn

requests a binomial test of the observations in the input file or of the observed counts (in the file or given as arguments after the COUNTS identifier). The probability of being in the category of interest follows the BINOMIAL identifier. If no argument follows, the probability is assumed to be .5.

An exact probability is calculated using the binomial distribution function. It is a one-tailed probability, unless otherwise noted in the output.

Optional Identifiers:

COUNTS nn nn

specifies the two counts observed in each of the two categories. The sum of the two counts is N, the total number of observations. An input file name should not follow NP.TEST when COUNTS is used with two arguments.

If COUNTS is used without any arguments, an input file should be specified after NP.TEST. That file should have only two cases; the first case should be "k" (the number of observations in the category of interest) and the second case should be "N-k" (the number in the other category).

EQ nn

specifies the category boundary. All values equal to the specified value are in one category; all other values are in the other category. Missing values are not in either category. Either COUNTS or one of these six boundary-defining identifiers must be used.

GE nn

specifies the category boundary. All values greater than or equal to the specified value are in one category; all other values are in the other category.

GT nn

specifies the category boundary. All values greater than the specified value are in one category; all other values are in the other category.

LE nn

specifies the category boundary. All values less than or equal to the specified value are in one category; all other values are in the other category.

LT nn

specifies the category boundary. All values less than the specified value are in one category; all other values are in the other category.

NE nn

specifies the category boundary. All values not equal to the specified value are in one category; all other values are in the other category.

WEIGHT vn

provides the name of the variable whose values are the counts observed in the corresponding category. The weight for each case should be a positive integer, or the case will be skipped. WEIGHT is typically used with an input file of single observations, although it may be used with COUNTS. When WEIGHT is not used, each case in the input file is counted as one observation (unless COUNTS is used).

One-Sample Tests: CHI.SQUARE

```
NP.TEST  S45,  CHI.SQUARE,  WEIGHT  Wins  $
NP.TEST  S45,  CHI  1  8,
               EXPECTED  22  20  18  18  18  18  16  14  $
```

The chi-square test computes a chi-square statistic and the associated probability that the frequencies observed in *multiple categories* in a sample are statistically the same as the frequencies expected for those categories in the population. The data need be only nominal (categorical) values.

Each case in the input file is assumed to be a single observation, unless the WEIGHT identifier names a variable whose values are the total observations in a category. The expected frequencies in the population are assumed to be the total number of observations divided by the number of categories, unless the EXPECTED identifier gives other expected values for each category.

Required:

NP.TEST fn

specifies the required input file. The chi-square statistic is computed for each variable in the file except the WEIGHT variable (if one has been designated).

Required Identifiers:

CHI nn nn

requests a chi-Square test. Two optional arguments specify the range of categories. When a range is specified, any empty categories are included in the calculations. When a range is not specified, only categories actually in the data are included. CHISQUARE and CHI.SQUARE are synonyms for CHI.

The chi-square distribution is used to obtain the probability, unless the degrees of freedom (number of categories less one) equal 500 or above, when a normal approximation to the chi-square distribution is used.

Optional Identifiers:

EXPECTED nn nn nn

specifies the expected frequency for each category. There should be as many numeric arguments as there are categories. If the sum of the expected frequencies does not equal the total number of observations, the frequency divided by the total is the proportion of observations expected for that category.

When EXPECTED is not used, the expected number for each category is assumed to be the total number of observations divided by the number of categories.

WEIGHT vn

provides the name of the variable whose values are the counts observed in each category. The weight for each case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

One-Sample Tests: KS1

```
NP.TEST  Test,  KS1,  NORMAL  $
```

The Kolmogorov-Smirnov goodness-of-fit test calculates the probability that an observed distribution is not different from an hypothesized distribution. Possible test distributions are the normal, uniform and Poisson distributions. The data should be at least ordinal (ordered) values.

The observed cumulative frequency distribution from the sample is compared with that from the test population, and the maximum vertical deviation between the distributions is found. The approximate two-tailed probability of obtaining a difference this large is calculated using the Smirnov formula (1948). The KS1 test is considered more powerful than the Chi-Square test.

Required:

NP.TEST fn

specifies the required input file. The KS test statistic is computed for each variable in the file except the WEIGHT variable (if one has been designated). When POISSON is used, the data must be positive integer values.

Required Identifiers:

KS1 nn nn

requests the Kolmogorov-Smirnov one-sample test. Either NORMAL, UNIFORM or POISSON must also be used with KS1.

Optional Identifiers:

DISCRETE

specifies that the hypothesized normal distribution is discrete, rather than continuous. DISCRETE may only be used with UNIFORM. When UNIFORM is used without DISCRETE, a continuous uniform distribution is assumed.

NORMAL nn nn

specifies the normal distribution as the hypothesized population distribution. The mean and standard deviation are optional arguments that may follow NORMAL. When they are not given, the observed mean and standard deviation in the sample are used in generating the comparison normal distribution.

POISSON nn

specifies the Poisson distribution as the hypothesized population distribution. The mean is an optional argument that may follow POISSON. When it is not given, the observed sample mean is used in generating the comparison Poisson distribution.

UNIFORM nn nn

specifies the uniform distribution as the hypothesized population distribution. Low and high values, giving the range, are optional arguments that may follow UNIFORM. When they are not given, the observed sample range is used in generating the comparison uniform distribution. A continuous uniform distribution is assumed unless DISCRETE is also used.

WEIGHT vn

provides the name of the variable whose values are weights. The weight for each case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

Two-Independent Sample Tests: MEDIAN

```
NP.TEST  S114,  MEDIAN,  GROUP  Absent.Present  $
```

The Median test calculates the probability that the medians of two independent samples are statistically the same size — that the samples come from populations with the same median. The data should be at least ordinal (ordered) values.

The median for the combined groups is calculated or an arbitrary value may be input. The data are grouped both by sample and by whether or not the value is above or below the median.

Required:

NP.TEST fn

specifies the required input file. The file should contain one or more variables whose values are scores and an additional variable whose values denote group membership. A weighting variable may also be included in the file.

Required Identifiers:

MEDIAN nn

requests a Median test for all variables in the input file except the GROUP variable and the WEIGHT variable (if present). An optional "median" (an arbitrary value) may be given as the argument for the identifier MEDIAN. It will be used to dichotomize the data into groups above and below the median. When there is no argument after MEDIAN, the median of the combined groups is computed.

The chi-square test, with a correction for continuity, is used to obtain an approximate probability when the sample size is above 50. When the sample size is 50 and below, Fisher's exact test is also used to compute the probability.

GROUP vn nn nn

specifies the variable whose values define group membership and, optionally, specifies one or two numeric values. When just the variable name is given, that variable should have only values of 1 and 2 for the Median test of two independent groups.

If the variable name is followed by one number, values equal to that number are one group and all other non-missing values are the second group. If the variable name is followed by two numbers, values equal to that number are one group and values equal to the second number are the second group.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

Two-Independent Sample Tests: MANN.WHITNEY

```
NP.TEST  S114,  MANN.WHITNEY,
                GROUP  Absent.Present,  WEIGHT  Count  $
```

The Mann-Whitney U test calculates the probability that two independent samples have the same sized values — that the two samples come from the same population. The data should be at least ordinal (ordered) values.

The values from both samples are combined and ranked. The U statistic is the number of times a score in one group precedes a score from the other group in the ranking. (The W statistic, in the equivalent Wilcoxon Rank Sum W test, may be computed from the sum of the ranks given in the output.) The Mann-Whitney test is more powerful than the Median test because it deals with ranks (relative order) rather than just position above and below the median.

Required:

NP.TEST fn

specifies the required input file. The file should contain one or more variables whose values are scores and an additional variable whose values denote group membership. A weighting variable may also be included in the file.

Required Identifiers:

MANN.WHITNEY

requests a Mann-Whitney U test for all variables in the input file except the GROUP variable and the WEIGHT variable (if present). The identifier U is a synonym for MANN.WHITNEY. When the total number of observations is 40 or less, the sampling distribution of the Mann-Whitney U Statistic is used and an exact probability is calculated. When the total is greater than 40, the normal approximation (with a correction for ties) is used to obtain the probability.

GROUP vn nn nn

specifies the variable whose values define group membership and, optionally, specifies one or two numeric values. When just the variable name is given, it should have only values of 1 and 2 for the two independent groups.

If the variable name is followed by one number, values equal to that number are one group and all other non-missing values are the second group. If the variable name is followed by two numbers, values equal to that number are one group and values equal to the second number are the second group.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

Two-Independent Sample Tests: KS2

```
NP.TEST  S130,  KS2,  GROUP  Class  7  11  $
```

The Kolmogorov-Smirnov two-sample test (KS2) calculates the probability that the values in two independent samples are not statistically different — that the two samples come from the same population. The test discerns differences in central location, variability or in skewness. The data should be at least ordinal (ordered) values from a distribution assumed to be continuous.

The observed cumulative frequency distributions from both samples are compared and the maximum vertical deviation between the distributions is found. The approximate two-tailed probability of obtaining a difference this large is calculated using the Smirnov formula (1948). The KS2 test is more powerful than the Median test.

Required:

NP.TEST fn

specifies the required input file. The file should contain one or more variables whose values are scores and an additional variable whose values denote group membership. A weighting variable may also be included in the file.

Required Identifiers:

KS2

requests the Kolmogorov-Smirnov two-sample test for all variables in the input file except the GROUP variable and the WEIGHT variable (if present). A two-tailed probability is calculated.

GROUP vn nn nn

specifies the variable whose values define group membership and, optionally, specifies one or two numeric values. When just the variable name is given, it should have only values of 1 and 2 for two independent groups.

If the variable name is followed by one number, values equal to that number are one group and all other non-missing values are the second group. If the variable name is followed by two numbers, values equal to that number are one group and values equal to the second number are the second group.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

Two-Independent Sample Tests: RUNS

```
NP.TEST  S139,  RUNS,  GROUP  Boys.Girls  $
```

The Wald-Wolfowitz Runs test calculates the probability that two independent samples do not differ in any aspect — that they come from the same population. The test discerns differences in central location, variability, skewness or any other aspect of the population distribution. The data should be at least ordinal (ordered) values with an underlying distribution that is continuous.

The values from both samples are ranked together and the number of runs (sequences of scores from the same sample) is counted. A relatively small number of runs suggests that the samples are from different populations. The RUNS test is less powerful that the Mann-Whitney U test when just differences in central location are of interest.

Required:

NP.TEST fn

specifies the required input file. The file should contain one or more variables whose values are scores and an additional variable whose values denote group membership. A weighting variable may also be included in the file.

Required Identifiers:

RUNS

requests the Wald-Wolfowitz two-sample test for all variables in the input file except the GROUP variable and the WEIGHT variable (if present). An exact probability is calculated when the total number of values is 40 or less. The normal approximation is used when the number of values is above 40, with a correction for continuity when the total is less than 50. The probability is one-tailed.

GROUP vn nn nn

specifies the variable whose values define group membership and, optionally, specifies one or two numeric values. When just the variable name is given, it should have only values of 1 and 2 for two independent groups.

If the variable name is followed by one number, values equal to that number are one group and all other non-missing values are the second group. If the variable name is followed by two numbers, values equal to that number are one group and values equal to the second number are the second group.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

Two-Indep Sample Tests: SQUARED.RANKS

```
NP.TEST  C243,  SQUARED.RANKS,  GROUP  Present.New  $
```

The Squared Ranks test for equal variances computes the probability that the variances of two independent samples are equal — that the samples come from populations with identical distributions except for possibly different means. The data should be at least interval values (ordered values with equal intervals between them).

The absolute deviation of each value from the mean of its sample is computed. The combined deviations from both samples are ranked. The test statistic (T1) is the sum of the squares of the ranks in one sample, with a correction for ties.

Required:

NP.TEST fn

specifies the required input file. The file should contain one or more variables whose values are scores and an additional variable whose values denote group membership. A weighting variable may also be included in the file.

Required Identifiers:

SQUARED.RANKS

requests the Squared Ranks two-sample test for all variables in the input file except the GROUP variable and the WEIGHT variable (if present). The normal approximation to the distribution of the test statistic T1 is used. A two-tailed probability is calculated.

GROUP vn nn nn

specifies the variable whose values define group membership and, optionally, specifies one or two numeric values. When just the variable name is given, it should have only values of 1 and 2 for two independent groups.

If the variable name is followed by one number, values equal to that number are one group and all other non-missing values are the second group. If the variable name is followed by two numbers, values equal to that number are one group and values equal to the second number are the second group.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

Two-Paired Sample Tests: SIGN

```
NP.TEST  S73,  SIGN,  WEIGHT  Cases  $
```

The Sign test calculates the probability that the observed values in two paired samples are statistically the same. The samples need not come from the same population. The data should be at least ordinal (ordered) values because the Sign test deals with whether the difference between the values for a given pair of measurements is positive or negative (zero differences do not figure in the calculation). It is assumed that the underlying distribution of the values is continuous.

Required:

NP.TEST fn

specifies the required input file. The file should contain an even number of variables and, if desired, a weighting variable. The first half of the variables are paired with the second half — the first with the third and the second with the fourth, for example, when there are a total of four variables. A two-tailed probability is computed for each pair of variables, except the WEIGHT variable (if one has been used).

Required Identifiers:

SIGN

requests the Sign test. An exact probability is calculated using the binomial distribution function.

Optional Identifiers:

ONE

specifies a one-tailed significance test.

TWO

requests a two-tailed significance test. This is assumed by default.

WEIGHT vn

provides the name of the variable whose values are the counts observed for each data pair. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

Two-Paired Sample Tests: MCNEMAR

```
NP.TEST  S66,  MCNEMAR,  WEIGHT  WWW  $
```

The McNemar test calculates the probability of obtaining the observed number of positive changes in a two-categoried population, where the probability of either a negative or a positive change is .5. The data need be only nominal values — the two categories should be coded zero and one.

Required:

NP.TEST fn

specifies the required input file. The file should contain an even number of variables and, if desired, a weighting variable. The first half of the variables are paired with the second half — the first with the third and the second with the fourth, for example, when there are a total of four variables. A two-tailed probability is computed for each pair of variables, except the WEIGHT variable (if one has been used).

Required Identifiers:

MCNEMAR

requests the McNemar test. An exact probability is calculated using the binomial distribution function. A chi-square statistic is also computed.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each data pair. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

Two-Paired Sample Tests: WILCOXON

```
NP.TEST  S79,  WILCOXON  $

NP.TEST  C284  [ SPLIT  6;  GEN  Mean  =  30 ],  WILCOXON  $

NP.TEST  S82   [ KEEP  Differences;  GEN  Zero = 0 ],  WILCOXON  $
```

The Wilcoxon Matched-Pairs Signed-Ranks test calculates the probability that the observed values in two paired samples are statistically the same. The data should be at least interval values (ordered values with equal intervals between them). It is assumed that the distribution of differences is symmetric.

The Wilcoxon test deals with both the direction of the differences between paired values (positive or negative — zero differences are not included in the calculation) and the relative size of the differences. Thus, for data that meet these criteria, it is more powerful than the Sign test.

Required:

NP.TEST fn

specifies the required input file. The file should contain pairs of variables. The input file may contain differences between pairs, rather than the paired values themselves, but a second variable should be generated with all values equal to zero (as in the second example above). Similarly, the values in the input file may be compared with a population mean or any arbitrary value — generate a variable equal to that value (as in the third example).

A two-tailed probability is computed for each pair of variables in the file except the WEIGHT variable (if one has been designated).

Required Identifiers:

WILCOXON

requests the Wilcoxon test. The normal approximation to the sampling distribution of the Wilcoxon Signed-Ranks test statistic is used. This is appropriate for all but very small (below eight data pairs) sample sizes.

Optional Identifiers:

ONE

specifies a one-tailed significance test.

TWO

requests a two-tailed significance test. This is assumed by default.

WEIGHT vn

provides the name of the variable whose values are the counts observed for each data pair. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

K-Independent Sample Tests: MEDIAN

```
NP.TEST  S182,  MEDIAN,  NG 6,  GROUP  Mothers.Ed  $
```

The Median test calculates the probability that the medians in k-independent samples are not statistically different — that the samples come from populations with the same median. The data should be at least ordinal (ordered) values.

The median for the combined groups is calculated or an arbitrary value may be input. The data are grouped both by sample and by whether or not the value is above or below the median.

Required:

NP.TEST fn

specifies the required input file. The file should contain one or more variables whose values are scores and an additional variable whose values denote group membership. A weighting variable may also be included in the file.

Required Identifiers:

MEDIAN nn

requests a Median test for all variables in the input file except the GROUP variable and the WEIGHT variable (if present). An optional “median” (an arbitrary value) may be given as the argument for the identifier MEDIAN. It will be used to dichotomize the data into groups above and below the median. When there is no argument after MEDIAN, the median of the combined groups is computed.

The probability is calculated using the approximation to the chi-square distribution.

GROUP vn nn nn

specifies the variable whose values define group membership and, optionally, specifies k-1 or k numeric values. When just the variable name is given, that variable should have only values of 1 through k for the Median test of k-independent groups.

If the variable name is followed by k-1 numbers, values equal to those numbers define the first through k-1 groups and all other non-missing values are the k^{th} group. If the variable name is followed by k numbers, values equal to those numbers define the k groups.

Optional Identifiers:

NG nn

specifies the number of groups. When NG is not used, two groups are assumed.

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

K-Independent Sample Tests: KRUSKAL.WALLIS

```
NP.TEST  S187,  KRUSKAL.WALLIS,  NG 3,  GROUP  Position  $
```

The Kruskal-Wallis one-way analysis of variance by ranks computes the probability that k-independent samples come from the same population or from identical populations with regard to their means. The data must be at least k-paired ordinal (ordered) values whose underlying distribution is continuous.

The Kruskal-Wallis test is more efficient than the Median test because it makes use of the relative order of the scores rather than just their position above or below the median.

Required:

NP.TEST fn

specifies the required input file. The file should contain one or more variables whose values are scores and an additional variable whose values denote group membership. A weighting variable may also be included in the file.

Required Identifiers:

KRUSKAL.WALLIS

requests the Kruskal-Wallis k-sample test for all variables in the input file except the GROUP variable and the WEIGHT variable (if present). KW is a synonym for this identifier. The chi-square distribution is used to calculate the probability.

GROUP vn nn nn

specifies the variable whose values define group membership and, optionally, specifies k-1 or k numeric values. When just the variable name is given, that variable should have only values of 1 through k for the Median test of k-independent groups.

If the variable name is followed by k-1 numbers, values equal to those numbers define the first through k-1 groups and all other non-missing values are the k^{th} group. If the variable name is followed by k numbers, values equal to those numbers define the k groups.

Optional Identifiers:

NG nn

specifies the number of groups. When NG is not used, two groups are assumed.

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

K-Independent Sample Tests: SQUARED.RANKS

```
NP.TEST  C249,  SQUARED.RANKS,  NG 3,  GROUP  Method,  CONTRASTS  $
```

The Squared Ranks test for equal variances computes the probability that the variances of k-independent samples are equal — that the samples come from populations with identical distributions except for possibly different means. The data should be at least interval values (ordered values with equal intervals between them).

The absolute deviation of each value from the mean of its sample is computed. The combined deviations from all samples are ranked. The test statistic (T2) is the sum of the squares of the ranks in all samples, with a correction for ties.

Required:

NP.TEST fn

specifies the required input file. The file should contain one or more variables whose values are scores and an additional variable whose values denote group membership. A weighting variable may also be included in the file.

Required Identifiers:

SQUARED.RANKS

requests the Squared Ranks k-sample test for all variables in the input file except the GROUP variable and the WEIGHT variable (if present). The chi-square approximation to the distribution of the test statistic T2 is used to calculate the probability.

GROUP vn nn nn

specifies the variable whose values define group membership and, optionally, specifies k-1 or k numeric values. When just the variable name is given, that variable should have only values of 1 through k for the Median test of k-independent groups.

If the variable name is followed by k-1 numbers, values equal to those numbers define the first through k-1 groups and all other non-missing values are the k^{th} group. If the variable name is followed by k numbers, values equal to those numbers define the k groups.

Optional Identifiers:

CONTRASTS

requests all pairwise contrasts of the k groups. The t distribution provides the significance level of each contrast.

NG nn

specifies the number of groups. When NG is not used, two groups are assumed.

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

K-Paired Sample Tests: COCHRAN

```
NP.TEST  S164,  COCHRAN  $
```

The Cochran Q test computes the probability that k treatments are equally effective. The data may be nominal or dichotomized ordinal values that are matched or blocked. The variables should be the treatments and the cases should be the blocks. The actual data values should be only zeros and ones. Any cases with missing data are dropped.

Required:

NP.TEST fn

specifies the required input file. The file should contain k-paired variables. A weighting variable may also be included in the file. One analysis is computed for the whole file.

Required Identifiers:

COCHRAN

requests the Cochran Q test. The chi-square distribution is used to approximate the distribution of Cochran's Q statistic and obtain the significance level.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each block or case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

K-Paired Sample Tests: FRIEDMAN

```
NP.TEST  S171,  FRIEDMAN  $
```

The Friedman two-way analysis of variance by ranks computes the probability that k samples come from the same population. The data must be at least k-paired ordinal (ordered) values. Any cases with missing data are dropped.

Required:

NP.TEST fn

specifies the required input file. The file should contain two to k-paired variables. A weighting variable may also be included in the file.

Required Identifiers:

FRIEDMAN

requests the Friedman two-way analysis of variance by ranks test. The chi-square distribution is used to approximate the test statistic and obtain the probability.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

K-Paired Sample Tests: CONCORDANCE

```
NP.TEST  S234,  CONCORDANCE  $
```

The Kendall coefficient of concordance W measures the degree of association between k rankings of values. The significance level associated with the test statistic gives the probability that the k rankings are independent. Any cases with missing data are dropped.

Required:

NP.TEST fn

specifies the required input file. The file should contain two to k-paired variables. A weighting variable may also be included in the file.

Required Identifiers:

CONCORDANCE

requests the Kendall coefficient of concordance. An approximation to the chi-square distribution is used to calculate the significance level.

Optional Identifiers:

WEIGHT vn

provides the name of the variable whose values are the counts observed for each case. The weight for a case should be a positive integer, or the case will be skipped. When WEIGHT is not used, each case is counted as one observation.

NP.COR

Rank Correlation:

```
NP.COR  S205,  OUT  S205.Sp  $

NP.COR  S208,  KENDALL,  OUT  S208.Ken  $
```

The NP.COR command computes the Spearman (rho) and Kendall (tau) test statistics measuring the degree of association or correlation between pairs of variables. The Spearman statistic is computed unless KENDALL is specified. The data must be at least paired ordinal (ordered) values so that they can be ranked.

The significance level associated with the test statistic is the probability that two paired variables are mutually independent. An approximation to the t distribution is used for the Spearman statistic and an approximation to the normal distribution is used for the Kendall statistic (unless the count of paired values is 10 or less when an exact probability is computed). The probability is one-tailed.

Required:

NP.COR fn

specifies the required input file. Correlation coefficients are computed for each variable paired with every other variable in the file, unless the count of non-missing values for a pair is less than three or all values of one of the variables are ties.

Required Identifiers:

OUT fn

provides the required name for the output file containing the correlation coefficients.

Optional Identifiers:

KENDALL

requests that the Kendall statistic (tau) be calculated. When KENDALL is not specified, the Spearman statistic (rho) is calculated.

ONE

specifies a one-tailed significance test. This is assumed by default.

TWO

requests a two-tailed significance test.

6
EDA:
Exploratory Data Analysis

Exploratory data analysis uncovers patterns hidden within the numbers that comprise a collection of data. It produces displays that invite new assumptions and choices of models for analysis.

EDA was implemented on the P-STAT system using the programs of P. Velleman and D. Hoaglin in *ABCs of EDA*, Duxbury Press, 1981. The EDA concepts originated with John W. Tukey, *Exploratory Data Analysis*, Addison-Wesley Publishing Company, 1977.

The EDA procedures are:

1. BOX.PLOT — graphically pictures where the middle of a batch of data is, its spread, and its tails;
2. CODED.TABLE — summarizes the patterns of data in a table format and uses coded symbols to represent the location of the data with respect to the median;
3. LETTER.VALUES — summarize data in terms of medians, hinges (quarters), and eighths;
4. MEDIAN.POLISH — fits an additive model, similar to the one two-way ANOVA uses, but which finds row and column medians rather than means;
5. RESISTANT.LINE — fits a linear model that is more resistant to stray points than that fitted by least-squares regression;
6. RESISTANT.SMOOTH — smooths data curves by using averages to summarize overlapping curve segments;
7. ROOTOGRAM — is a square-root reexpression of a data histogram;
8. STEM.AND.LEAF — displays the range and symmetry of the whole data file, places of concentration, and gaps or stray values;
9. XY.PLOT — plots data with coded symbols.

For a thorough explanation of these EDA procedures and algorithms, see the Velleman and Hoaglin book. Because this book is both complete and readily available, no attempt is made to provide exhaustive explanations in this manual. The examples in this chapter are taken from the Velleman and Hoaglin book unless otherwise noted.[1]

6.1 HOW TO USE EDA

Only an input file is required for the EDA command. The identifiers LABELS and OUT are optional:

```
EDA  Filename ;
EDA  Filename,  OUT  OutFile,  LABELS  'LabFile' ;
```

If the identifier OUT is used, an output file is created containing the original input variables plus any variables created by the command. LABELS is used to supply value labels for box plots.

[1] There may be differences between the results in the book and results from the P-STAT programs. Some of these are typographical errors in early editions. Others are due to the better accuracy of the algorithms used in these programs.

EDA procedures and their options are entered as *subcommands*. Subcommand text is processed until a *semicolon* signals that the procedure be performed. In EDA, convenient abbreviations may and should be used for both subcommands and variable names. The user need only supply enough information to make those names unique.

6.2 EDA Procedures

The following is a list of EDA procedures and suggested abbreviations:

BOX.PLOT	BOX	RESISTANT.SMOOTH	R.SM
CODED.TABLE	C.T	ROOTOGRAM	ROOT
LETTER.VALUES	LET	STEM.AND.LEAF	STEM
MEDIAN.POLISH	M.P	XY.PLOT	PLOT
RESISTANT.LINE	R.L		

The procedure name may be followed directly by its options. For example, when requesting a stem-and-leaf display, either of the following subcommands is acceptable. STEM.AND.LEAF may be followed by the identifier VAR and the name of the variable to be displayed:

```
STEM.AND.LEAF,  VAR  Income ;
```

Or as a shortcut, the user may supply the variable name immediately after STEM:

```
STEM  Income ;
```

Both produce a stem-and-leaf display of the variable Income.

Procedures and options remain in effect until they are specifically changed or cancelled. Income is now the dependent or "Y" variable and it is used if a second EDA procedure is requested without specifying the variable name. These subcommands produce a stem-and-leaf of Income, followed by a letter-values display of Income:

```
STEM  Income ;
LET ;
```

If STEM is used by itself, either the previous "Y" variable is used or, if no "Y" variable has been specified, *all* the variables in the file are used in turn. Unique abbreviations are allowed. This produces a stem-and-leaf of Income if there are no other variables in the file beginning with the letter "I":

```
ST  I ;
```

Some of the procedures, such as RESISTANT.LINE, RESISTANT.SMOOTH and MEDIAN.POLISH, create new variables. Supplying names for these variables is optional. If the user does not specify particular variable names, P-STAT generates names such as "Smooth...", "Residuals...", and so on.

6.3 Transformations

Reexpressions may be used to change the shape of a batch of data that is asymmetrical or skewed. Transformations are applied to the specified variable. If a new EDA procedure is not specified, the previous EDA procedure is re-executed using the transformed data. In this example, the log to the base e of each value of the variable Precipitation is requested:

```
LOG  Precipitation ;
```

Transformations may also be used without specifying a variable name. In such cases, the values of the currently displayed variable are transformed.

Optional reexpressions that may be used are:

Power	Name
3	CUBE
2	SQUARE

```
 1                  NO TRANSFORMATIONS
 1/2                SQUARE.ROOT
 1/3                CUBE.ROOT

 0                  LOG, LOG10

-1/3                POWER ** -.3333
-1/2                RECIP.ROOT
-1                  RECIPROCAL
-2                  RECIP.SQUARE
```

POWER requests any arbitrary transformation (any transformation that is not a specific subcommand option). POWER requires a number to which the data values are raised. This example produces the reciprocal cube root:

```
STEM  Rate,  POWER  Rate ** -.3333 ;
```

(Note that ** is optional.)

Raising values to powers *less than 1* corrects a *positive skew* in the data distribution. Raising values to powers *greater than 1* corrects a *negative skew*. The further you go from a power of 1, the greater the effect on the shape of the data. PRESERVE.ORDER can be used to preserve the order of data values after reexpression. With PRESERVE.ORDER, negative numbers (in addition to positive numbers) retain their original sign after the transformation is complete. (P.O is an abbreviation for PRESERVE.ORDER.) NO TRANSFORMATIONS or POWER 1 may be used to repeat the current EDA procedure, using the original (untransformed) values.

6.4 Interactive Behavior

On-line help is available when using EDA interactively. For information about a given procedure, use HELP followed by the name of the procedure. This command gives HELP on producing stem-and-leaf displays:

```
HELP  STEM ;
```

Other useful HELP options include:

`HELP EDA ;`	produces a generalized EDA HELP text.
`HELP TRANSFORM ;`	gives HELP on the use of transformations.
`SHOW VARIABLES ;`	lists existing variables.

The semicolon signals that the execution of an EDA procedure should begin. A procedure may be re-executed by using:

```
AGAIN ;
```

If the user wishes to re-execute the procedure with a different option, the command AGAIN may be followed by the new option or the new option may be entered by itself. Either of these examples re-executes the previous EDA with a log transformation of the variable Precipitation:

```
AGAIN,  LOG  Precip ;   or
LOG  Precip ;
```

RESET may be used to reset all options to their original settings. A new EDA request, along with desired variables and transformations, is specified immediately after RESET:

```
RESET,  STEM  Income,  LOG ;
```

RESET may also be used to request the previous display with a new transformation. This subcommand re-executes the previous EDA procedure using the square root of the data:

```
RESET,  SQRT ;
```

NO can be used to turn off specific options within the individual EDA procedures. NO TRANSFORMATIONS turns off all previous transformations.

6.5 SHOWING DISTRIBUTIONS

Box plots, letter value displays, stem-and-leaf plots and rootogram histograms show the *distribution* of data values. The spread, extremes and center of the distribution may be seen. The terms hinges, inner fence and outer fence, which are specific to exploratory data analysis, are defined in this section.

6.6 BOX.PLOT

Box plots, sometimes called quantile plots, provide a convenient way to visualize the distribution of a variable. They summarize a batch of data by indicating the location of the median, the spread, the tails and outlying data points. Particular attention is given to stray values at the ends. The BOX.PLOT subcommand in EDA produces character plots. The BOX.PLOT command, which is documented in "P-STAT: Plots, Graphs and PostScript Support", produces high quality box plots for a PostScript printer.

The body of the box describes the data that falls within the *hinges*, which are analogous to quartiles. The hinges contain 50% of the data values. (This is comparable to .675 standard deviations about the mean, in non-EDA terminology.) The value for the median is indicated by a plus (+) sign.

Whiskers extend in either direction from the hinges up to the *inner fences*. The inner fences are defined as the upper or lower hinge plus or minus 1.5 times the spread between the hinges. (This is comparable to 1 standard deviation about the mean, which encloses 68% of the data.) The *outer fences* are the upper or lower hinges plus or minus 3 times the spread. (This is comparable to 2 standard deviations or 95% of the data.) Any data point outside of the inner fences is indicated by an asterisk (*). Any data point that lies outside of the outer fences is indicated by the letter O.

Figure 6.1 shows the BOX.PLOT subcommand. Exploratory data analysis is invoked by issuing the command EDA, followed by the name of the file to be analyzed. In Figure 6.1, the file is named Taxes:

```
EDA  Taxes,  LABELS  'TaxLab' ;
```

A label file may also be input to label grouped box plots — enclose the name in quotes. The semicolon indicates that EDA subcommands follow. The subcommand requests a box plot:

```
BOX.PLOT  Percent,  GROUP  Region ;
```

There are several options available when using the BOX.PLOT procedure. VAR specifies the name of the variable to be used for the BOX.PLOT display:

```
BOX.PLOT,  VAR  Percent,
```

The variable name may also follow directly after BOX.PLOT as in Figure 6.1:

```
BOX  Percent,
```

VAR is typically used to request the same procedure with a new variable. If a variable name is not supplied, the most recently referenced variable is used in the display.

GROUP, followed by the variable name that defines subgroups, requests *multiple* box plots by groups on the same axis. The values of the specified variable determine group membership. In Figure 6.1, multiple box plots for the variable Region are requested. LEVELS identifies which groups are printed in the box plot. LEVELS is followed by one or more integers enclosed in parentheses and separated by commas. In Figure 6.2 LEVELS 1 and 6 are requested along with NO AXIS:

```
LEVELS (1, 6),  NO AXIS ;
```

This causes box plots to be printed only for Region One (North Atlantic) and Region Six (South West). Ranges of LEVELS are permitted:

```
    LEVELS ( 1 TO 3, 6 ) ;
```

AXIS requests that the bottom axis be printed. This is assumed unless NO AXIS is used.

When box plots are created with subgroups, as in Figure 6.1, it is easy to visualize the differences in subgroup midpoints and distributions. However, it is not possible to tell if these differences are significant without using the NOTCH identifier. NOTCH defines an area (confidence interval) around each median based on the hinge spread for that group and the standard deviation for the entire population. This area is defined by the characters < and >. The second output in Figure 6.2 shows notched output.

Figure 6.1 Box Plots For Groups

----------The EDA Command and Subcommands-------------

```
EDA  Taxes,  LABELS  'TaxLab' ;
  BOX.PLOT  Percent,  GROUP  Region ;
$
```

----------The BOXPLOT Output---------------------------

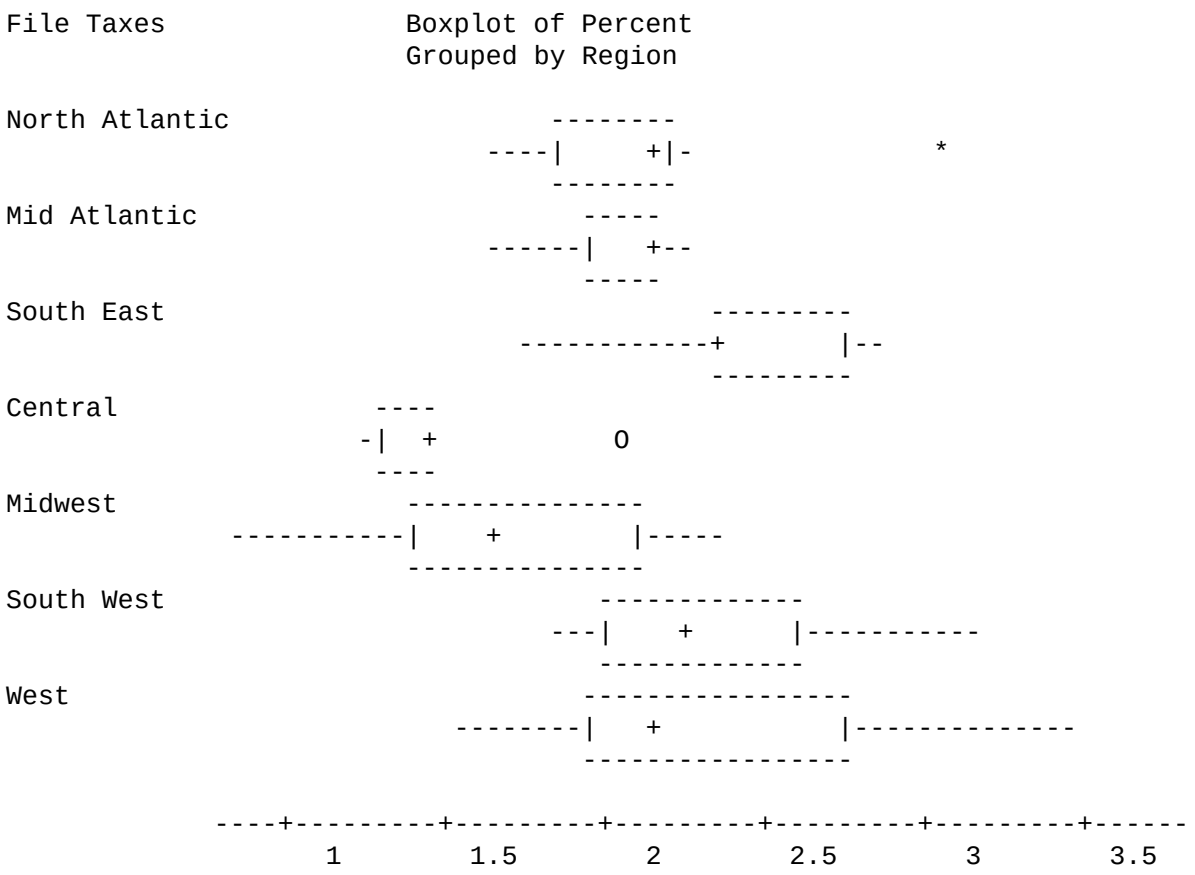


Figure 6.2 BOXPLOT with Groups and Levels

----------The EDA Command and Subcommands-----------

```
EDA  Taxes,  LABELS  'TaxLab' ;
  BOX.PLOT  Percent,
  GROUP  Region,
  LEVELS  ( 1, 6 ),
  NO AXIS ;
```

----------The BOXPLOT Output------------------------

```
File Taxes              Boxplot of Percent
                        Grouped by Region

North Atlantic                      --------
                                ----|     +|-              *
                                    --------
South West                             --------------
                                    ---|    +      |-----------
                                       -------------
```

----------Notched Output-----------------------------

```
AGAIN, NOTCH, AXIS;
```

```
North Atlantic        --------
                  ----|<    +|- >             *
                      --------
South West               --------------
                      --<|    +      >-----------
                         -------------
  ----+---------+---------+---------+---------+---------+---------+--
        1        1.5        2        2.5        3        3.5
```

Notches that do not overlap indicate groups that have significantly different medians at roughly the five percent level. (This is an individual five percent, with no allowance for the number of comparisons made.) Two groups are significantly different from each other if there is *no overlap* in the areas defined by the notches. NO NOTCH is assumed unless NOTCH is specified.

[2] The precipitation data is taken from D. Hinkley. "On Quick Choice of Power Transformation," *Applied Statistics 26 (1977):67-69.* Reprinted by permission.

6.7 LETTER.VALUES

LETTER.VALUES summarize the size of numbers, indicate how spread out they are, and provide information on the shape and pattern of the values. This procedure shows the shape of the distribution of values in relation to the location of the median, hinges, eighths, and so on.

To produce a letter values display, either LETTER.VALUES, L.V or LET is issued. There are two options available when using LETTER.VALUES. VAR specifies the name of the variable to be used for the LETTER.VALUES display. The variable name may follow the identifier VAR or may appear directly after LETTER.VALUES:

```
LETTER.VALUES,  VAR  Square.Miles ;   or
LETTER.VALUES   Square.Miles ;
```

LET by itself, gives a letter value display of the current "Y" variable or of all the variables in the file if no "Y" variable is defined. GAUSSIAN requests a comparison with the standard Gaussian or normal distribution. A separate column appears giving the spreads at the various letter values for the Gaussian distribution.

Figures 5.3 and 5.4 illustrates the letter values display.[2] Figure 6.3 illustrates LETTER.VALUES with the untransformed values. Figure 6.4 illustrates LETTER VALUES with the square root transformation. Five columns are labeled Depth, Lower, Upper, Mid and Spread. Depth indicates the position of the value, that is, how far it is from the low or high end of the batch of data values. Lower gives the lower letter value; upper gives the upper letter value.

The remaining two columns provide information on the shape of the data values. Mid specifies the midsummary or average value of each pair of letter values. Spread gives information on how the tails look. This is computed by subtracting the lower letter value from the upper letter value.

Figure 6.3 Letter Values

```
----------The EDA Command and LETTER.VALUES Output----------

EDA  Precip ;
     LETTER.VALUES ;
$


File is Precip           Letter Values of Precipitation

    Depth       Lower        Upper          Mid        Spread

N=   30
M    15.5            1.470                1.470
H     8.0        0.900        2.100        1.500        1.200
E     4.5        0.680        2.905        1.792        2.225
D     2.5        0.495        3.230        1.862        2.735
C     1.5        0.395        4.060        2.227        3.665
      1          0.320        4.750        2.535        4.430
```

The letters to the far left of the display are the "letter values". "N" gives the count of cases in the file. "M" gives the location and the value of the median. The Precip file has 30 cases and the median lies between case 15 and 16 at 15.5. The median value is 1.47.

The third letter value, labeled H, specifies the values at the hinges (roughly the quartiles). In this example, the hinges are located 8 cases from either end. There are 7 cases with values that are more extreme than either hinge and 14 cases between. The lower hinge has a value of .900. The upper hinge has a value of 2.100. The spread between the hinges is 1.200 and the midpoint is 1.500. Thus, *half* of the cases lie between the hinges. Because the median is the midpoint of all the cases, it should be exactly the same as the midpoint of the hinges if the distribution is *symmetric*. The letter labeled E is eighths. D is sixteenths, and so on.

Comparing the midsummaries to the median value and examining trends in the midsummaries indicates whether or not a batch of values is symmetrical. If the values for the midsummaries are nearly equal, then they are considered symmetrical around the median. However, if the midsummaries are distributed toward either the high or the low side of the median, the values are skewed positively or negatively, respectively.

Figure 6.4 Letter Values: Square Root Transformation

```
----------The EDA Command and LETTER.VALUES Output----------

EDA  Precip ;
     LETTER.VALUES, SQRT ;
$


File is Precip          Letter Values of Square Root of Precipitation

    Depth       Lower        Upper          Mid        Spread

N=   30
M    15.5            1.212                1.212
H     8.0        0.949        1.449        1.199        0.500
E     4.5        0.823        1.704        1.263        0.881
D     2.5        0.703        1.797        1.250        1.093
C     1.5        0.626        2.008        1.317        1.382
      1          0.566        2.179        1.373        1.614
```

In Figure 6.3, the midpoints gradually *increase*, indicating a distribution that is *skewed to the right*. It is possible to produce a more symmetrical distribution by reexpressing the numbers. In Figure 6.4, the data is reexpressed with the square root transformation and the difference between the midsummaries becomes less. The lowest is 1.199 and the highest 1.373 compared with the pre-transformation values of 1.47 and 2.535.

6.8 STEM.AND.LEAF

A stem-and-leaf display shows the range and symmetry of the data values, where those data values are concentrated, and any gaps or stray values. It is like a histogram on its side, except that the actual digits in the data values are used in the display. Each data value is split into *leading* digits and *trailing* digits. For example, the number 426 is split into leading digits of 42 and a trailing digit of 6. The leading digits are the *stem* of the display and the trailing digit is a *leaf* of the display.

To invoke the Stem-and-Leaf display, either STEM.AND.LEAF, STEM or ST are used. There are several options available when using the STEM.AND.LEAF procedure. VAR specifies the name of the variable to be used in the Stem-and Leaf display:

```
STEM.AND.LEAF,  VAR  Precipitation ;
```

The use of the identifier VAR is optional. The name of the variable may immediately follow STEM:

```
  STEM  Precipitation ;
```

If a variable name is not specified, the most recently referenced variable is used. The use of STEM by itself gives stem-and-leaf displays for *all* variables if a previous single "Y" variable has not been defined.

Figure 6.5 Stem and Leaf

----------EDA Stem and Leaf: Command and Output------------

```
EDA  Precip ;
STEM ;

 Stem-and-Leaf Display of Variable Precipitation
 Leaf Digit Unit = .1;
 Thus     1 |2  represents    1.2

    2   +0* 34
    9   +0. 5578899
   15    1* 122334
   15    1. 567889
    9    2* 0124
    5    2. 8
    4    3* 003

   HI  47
```

---------Include Extreme Values in the Output-----------

```
 SCALE  EXTREMES ;

 Stem-and-Leaf Display of Variable Precipitation
 Leaf Digit Unit = .1;
 Thus     1 |2  represents    1.2

    2   +0* 34
    9   +0. 5578899
   15    1* 122334
   15    1. 567889
    9    2* 0124
    5    2. 8
    4    3* 003
    1    3.
    1    4*
    1    4. 7
```

SCALE FENCE requests that the display include *only* those data values within the inner fence limits. Any data values outside of the inner fences are given at the bottom of the display on special LO and HI stems. SCALE FENCE is assumed when the identifier SCALE is not used. (TRIM is a synonym for SCALE FENCE.) SCALE EXTREMES requests that the display include *all* data values of the specified variable, including any stray values.

(NO TRIM is a synonym for SCALE EXTREMES.)

Figure 6.5 illustrates STEM.AND.LEAF with some of these options. The EDA command is invoked by entering EDA followed by the name of the file:

```
EDA  Precip ;
```

The semicolon indicates that EDA subcommands follow. The first subcommand requests a stem and leaf display:

```
STEM ;
```

Since there is only one variable in this file, the variable name does not need to be provided and only one stem and leaf display is produced.

"Leaf Digit Unit" indicates where the decimal place falls for this group of data values. An example is given in the output to illustrate how to interpret the stem and leaf display with the appropriate decimal place:

```
Thus    1 |2  represents   1.2
```

A column of *depths* is located in the left-most column of the display. It indicates how many leaves have accumulated thus far in the batch of data values. The numbers increase from the top and bottom of the display toward the middle of the data values. For example, in the third line of the display:

```
15    1* 122334
```

the number 15 is the number of cases represented up through this line, that is, a cumulative count. This accumulation is done from both ends towards the middle. If the number of cases is even, no single value is located exactly in the middle. As a result, a pair of data values is given. The two rows have a cumulative distribution that is half of the total number of cases. If the number of cases is odd, the line containing the middle value shows a count of its number of leaves, enclosed in parentheses, in the depth columns.

The second column in the display contains the stem, which in this example (above) is the number 1. The * indicates that the stems are represented on multiple lines, and this is the first of a two line per stem display. (Please consult the Velleman and Hoaglin book for details on the definition of labels for multiple lines per stem.)

The third column contains the leaf digits, each one of which represents a single data point. By using the legend at the top indicating the placement of the decimal point, we can tell that this row represents data points 1.1, 1.2, 1.2, 1.3, 1.3 and 1.4. Notice that some information is lost because we cannot tell whether 1.1 represents a more precise number such as 1.11 or 1.19.

Values that stray noticeably from the rest of the data values are listed on a separate stem line at the bottom of the stem-and-leaf display. HI is used for high stray values and LO is used for low stray values. The display in Figure 6.5 shows one extreme value of 4.7 that is placed outside the body of the display.

When the stem-and-leaf is completed for variable Precipitation, a prompt is issued for another subcommand. The user requests the same variable and data values, but this time scaled to the extremes:

```
Begin subcommand, or type Q or H:
SCALE EXTREMES ;
```

SCALE EXTREMES requests that the display include all values rather than just those inside the inner fences.

6.9 ROOTOGRAM

Rootograms are a type of frequency distribution or histogram. The ROOTOGRAM procedure apportions measurements into *bins* and determines the *counts* in each bin. The bins are comparable to the bars of a histogram and the counts are comparable to the heights of the bars. The Gaussian (normal) distribution curve is fitted to the root-

ogram. Square-root reexpression of the data values stabilizes the variation from bin to bin. Raw residuals and double-root reexpressions of the residuals are calculated. A suspended rootogram display, showing coded double-root residuals, is produced.

The ROOTOGRAM display contains:

1. the bin number,
2. the observed count,
3. the raw residual,
4. the double-root residual and
5. the suspended rootogram display.

New variables containing bins, counts, fitted values and residuals are produced by ROOTOGRAM. The new variables, as well as the original ones, are placed in an output file if the OUT identifier is used at the EDA command level. The new variables have program-provided names unless the identifiers BINS, COUNTS, FITTED.VALUES and RESIDUALS are used to provide user-specified names.

There are several options available when using the ROOTOGRAM procedure. VAR specifies the name of the variable to be used for the ROOTOGRAM display. The variable name may also follow directly after ROOTOGRAM. If a variable name is not defined, the most recently referenced variable is used.

BINS specifies the variable whose values are the bin boundaries. A count is made of the data values that fall into each bin. If the variable whose values are the actual data values is not input, the BINS variable should be an existing variable to be used to provide bin boundaries. If the actual data values are also input (after ROOT):

```
ROOT  Weight,  BINS  Weight.Gps,  COUNT  Number ;
```

the BINS variable should be a unique name for the values giving bin boundaries to be calculated by the program. If BINS and COUNT are *not* used, the variable whose values are the actual data values must be specified either after ROOTOGRAM or VAR. Also, the new variables giving bin boundaries and bin counts are named “Bins...” and “Counts...”.

COUNT specifies the variable whose values are the counts for the corresponding bins. If the variable whose values are the actual data values is not input, the COUNT variable should be an existing variable to be used to provide counts per bin. If the actual data values are also input, the COUNT variable should be a unique name for the values giving bin counts to be calculated by the program. If BINS and COUNT are not used, the variable whose values are the actual data values must be specified either after ROOTOGRAM or VAR.

When the identifiers BINS and COUNT have as their respective arguments the variables that give the bin boundaries and the counts per bin, it is *not* necessary to specify the variable with the actual data values.

RESIDUALS provides a variable name for the residuals. If the RESIDUALS option is not used, the residuals are named “Residuals...”. FITTED.VALUES provides a variable name for the fitted values. If the FITTED.VALUES option is not specified, the fitted values are named “Fitted.Values...”.

In Figure 6.6, the bins and counts of the ROOTOGRAM display are in the two left-most columns. The third column, “Rawres”, gives the values of the residuals. The last two columns allow the user to determine if the data values are valid. “Drres” provides a way to compare the actual data values with the fitted data values. By examining these values closely, we see that -5.34 in Bin 12 represents a value that is clearly different from all other fitted values. The suspended rootogram display in the next column confirms this. The fit appears to be reasonable with the exception of this value. At this point, the original data values should be checked and this particular one, if it is an error, should be corrected.

Figure 6.6 Rootogram

```
---------Rootogram Subcommands and Output------------

EDA  Chest,  OUT  Rootout ;
ROOT,  BINS  Chest,  COUNT  Number ;


File Chest              Variable Number

Bin   Count   Rawres    Drres       Suspended Rootogram

  1     3.0     -2.7    -1.13        .    ------          .
  2    18.0     -2.6    -0.52        .       ---          .
  3    81.0      9.6     1.13        .          ++++++    .
  4   185.0    -11.2    -0.79        .      ----          .
  5   420.0     -7.7    -0.36        .        --          .
  6   749.0     10.1     0.38        .          ++        .
  7  1073.0     61.3     1.91        .          ++++++++++.
  8  1079.0    -21.4    -0.64        .      ----          .
  9   934.0    -13.3    -0.42        .       ---          .
 10   658.0     10.3     0.41        .          +++       .
 11   370.0     19.0     1.01        .          ++++++    .
 12    92.0    -58.7    -5.34   *----------------          .
 13    50.0     -1.3    -0.15        .          -          .
 14    21.0      7.2     1.76        .          +++++++++ .
 15     4.0      1.0     0.66        .          ++++      .
 16     1.0      0.4     0.63        .          ++++      .


In display, value of one character is .2       OO


Your new variables are:

              Chest
             Number
    Fitted.Values...
        Residuals...

Fitted means =39.3580    Fitted S.D. =2.05586


----------The Output File With Residuals----------

                Fitted
                Values   Residuals
Chest  Number     ...        ...

   33       3     5.691   -1.133179
   34      18    20.570   -0.523531
```

```
35      81    71.352    1.131851
36     185   196.224   -0.794174
37     420   427.745   -0.363892
38     749   738.853    0.381088
39    1073  1011.730    1.905350
40    1079  1100.376   -0.639877
41     934   947.279   -0.424728
42     658   647.746    0.410980
43     370   351.004    1.013214
44      92   150.729   -5.339279
45      50    51.310   -0.148424
46      21    13.848    1.764055
47       4     2.963    0.657567
48       1     0.578    0.629976
```

6.10 RELATIONSHIPS BETWEEN VARIABLES

XY plots, fitted lines and smoothed lines show the relationships between ordered pairs of variables. They permit inference or prediction of the dependent variable from the independent variable. Often the data values of the Y variable are ordered by time, the X variable. Then, the Y data values are thought of as time series.

6.11 XY.PLOT

An XY.PLOT is a *condensed, coded* scatter plot. Counts of observations at a given point are not shown on an XY.PLOT, as they are on traditional scatter plots. Ten lines or divisions of the traditional vertical (Y) axis are condensed into one line on the XY.PLOT. A coded symbol, numbered from 0 to 9, indicates the relative positions of the data values on these lines or divisions.

The use of numbers as characters in XY.PLOT is somewhat different from that in other plots. The numbers do not mean the number of data points falling at a particular location. Rather, the numbers indicate how far up the scale each point is located. In examining an XY.PLOT, we can see similarities between this plot and the output produced in stem-and-leaf displays. The legend at the left is analogous to the stem. The number in the plot is analogous to the leaf. In a ten character plot, which uses the numbers 0 to 9, the number 3 indicates a point that is three tenths of the way between the numbers on the axis.

There are several options available when using XY.PLOT. Y, followed by a variable name, specifies the Y variable name. X, followed by a variable name, specifies the X variable name. If Y is not given, the previous Y is used. If X is not given, the previous X is used. The variables to be plotted may be specified directly after XY.PLOT, or with the subcommand identifiers X and Y:

```
XY.PLOT   Birthrate  BY  Year ;
XY.PLOT,  X  Year,   Y   Birthrate ;
```

A range of values is supported for both X and Y values:

```
X.RANGE  1  TO  5,  Y.RANGE  6  TO  10 ;
```

In addition, a range may be supplied that pertains to both X and Y values :

```
RANGE  20  TO  50 ;
```

This subcommand requests a plot of Birthrate on the vertical axis by Year on the horizontal axis:

```
PLOT  Birthrate  BY  Year ;
```

To produce another plot with the same X variable but a new Y variable, this subcommand is entered:

```
      Y  Income ;
```

A plot of Income by Year is produced with the same number of lines and characters.

Figure 6.7 XY Plots

```
    ----------EDA: Command, Subcommand and XY Plot Output--------


    EDA  Births ;

      Exploratory Data Analysis
      Enter   HELP;  For Details on Usage

     PLOT  Birthrate  BY  Year ;

 File Births              9 Line,  10 Character Plot
 Birthrate by Year

 +   260 |                                        42241
 +   240 |                                     19     60
 +   220 |                                  158         4
 +   200 |                               50007            2
 +   180 |11  0                             4                  6  0
 +   160 |  19 638551                7                          99 2
 +   140 |           522           4 381                            9
 +   120 |              9524446768                                  831
 +   100 |                                                             9
         -+---------+---------+---------+---------+---------+---------+----
         1920      1930      1940      1950      1960      1970      1980



   ----------EDA:  Controls for XY Plots----------------------

   LINES  6, CHAR  4 ;


 File Births              6 Line,   4 Character Plot
 Birthrate by Year
 +   240 |                                        023333310
 +   210 |                               0   0123         2
 +   180 |00  0                           1 222              31  0
 +   150 |  13 3232210               3 0                        33 21
 +   120 |            3321011111123 3 3                              200
 +    90 |                                                           3
         -+---------+---------+---------+---------+---------+---------+----
         1920      1930      1940      1950      1960      1970      1980
```

In Figure 6.7, an XY.PLOT of Birthrate by Year is requested. At this point, a legend appears at the top indicating how many lines the plot requires and the number of characters:

```
9 Line,  10 Character Plot
```

A second subcommand requesting a plot with six lines and four characters is specified:

```
LINES 6,  CHAR 4 ;
```

LINES, followed by an integer between 5 and 40, specifies the number of lines to be used for the plot. If LINES is not specified, ten lines is assumed. CHARS, followed by an integer between 1 and 9, specifies the number of characters in the plot. When CHARS is not specified, ten characters are assumed, numbering 0 through 9.

AXIS or NO AXIS can be used to obtain or suppress the lower axis of the plot. AXIS is the assumed setting. YMIN, YMAX, XMIN, and XMAX can all be set to control the plot boundaries.

6.12 RESISTANT.LINE

The RESISTANT.LINE technique is similar to regression, but it is based on the median rather than the mean. The program produces an equation with the slope and intercept that define the line that best fits the data points. It also calculates residuals, which are the differences between the observed values and the fitted values.

RESISTANT.LINE fits a line to pairs of data values and calculates the equation of that line. This line is more resistant to the effect of outliers because the slope of the line is calculated from pairs of median values that summarize thirds of the data set.

There are several options available when using the RESISTANT.LINE procedure. Y specifies the name for the Y variable. X specifies the name for the X variable. These variables may be supplied in either of two ways:

```
RESISTANT.LINE,  Y  Age,  X  Salary ;   or
R.L  Age  BY  Salary ;
```

In addition, a new Y variable may be supplied without changing the X variable:

```
Y  Income.Bracket ;
```

RESIDUALS provides a variable name for the residuals to be calculated. When RESIDUALS is not used, the output residuals are named “Residuals...”. The following instruction:

```
R.LINE  Mpg    BY  Displacement,
        RECIP  Mpg,  POWER  Displacement ** -.3333 ;
```

produces the resistant line solution of the reciprocal of variable Mpg by the reciprocal cube root of Displacement. Since RESIDUALS is not used, the output residuals are named “Residuals...”.

STEPS specifies the number of iterations to be done. SUMMARY requests that only summary information be printed. The step information is omitted. NO SUMMARY is assumed and the slope step information typically prints. REPORT (REP) is a synonym for NO SUMMARY and requests the *full* step report. NO REPORT (NO REP) is a synonym for SUMMARY.

In Figure 6.8, an output file named Mean2 is requested when the EDA command is given:

```
EDA  MeanTemp,  OUT  Mean2 ;
```

The RESISTANT.LINE procedure of Mortality by MeanTemp is then specified:

```
R.LINE  Mortality  BY  MeanTemp ;
```

Summary points are used to calculate the slope of the residuals. The residuals from the line with this slope and the original intercept are “second residuals”. Their slope is calculated and is used to correct the initial slope estimate, and so on. The closer the slope of the residuals is to zero, the better the line fits the data. The list of slope estimates, shown in the output, converges toward the true slope (unless convergence is not possible). In Figure 6.8, five slope estimates are calculated to determine what is the best fitted line. Once the fitted line equation is determined, P-STAT issues a message that a new variable named “Residuals...” has been created.

Figure 6.8 Resistant Line

```
----------Resistant Line: Subcommands and Output---------

EDA  MeanTemp,  OUT  Mean2 ;

R.LINE  Mortality  BY  MeanTemp ;


Resistant Line of Mortality
               by MeanTemp

 Straightness check.
   Left  half-slope =    3.245464
   Right half-slope =    3.630941
   Ratio            =    1.118773

Slope   1=     3.412373
Slope   2=     2.539375
Slope   3=     2.903832
Slope   4=     2.890175
Slope   5=     2.890230

Fitted Line: Y = -45.9058 + 2.89017 * X

Your new variable is:
        Residuals...

---------Examine the Residuals With Stem and Leaf------

STEM  Res ;


Stem-and-Leaf Display of Variable Residuals...
Leaf Digit Unit = 1 ;
Thus    1  |2  represents   12


    1  -1*  1
    2  -0.  9
    4  -0S  77
    4  -0F
    6  -0T  22
    8  -0*  00
    8  +0*  001
    5  +0T  2
    4  +0F  4
    3  +0S  6
    2  +0.  8

        HI  21
```

In order to examine these residuals more closely, a stem-and-leaf display is requested:

```
STEM  Res ;
```

Note that the high outlier (HI 21) has not twisted the resistant line. Finally, in order to visualize all three of these variables, the output file, Mean2, is listed. When an output file is requested using the OUT identifier, the output file contains all the original variables in the file, as well as any that are created as subcommands are executed. As illustrated in Figure 6.9, the new variable is named “Residuals...”.

The straightness of the line describing the relationship between X and Y may be determined from the ratio of the left and right half-slopes. (See Figure 6.8). If it is close to 1, the line is nearly straight. If not, reexpressing X or Y or both may help. (When the half slope is negative, indicating *non*linearity, reexpression does not straighten the line.)

Figure 6.9 Output File from Resistant Line

```
---------File Mean2: Output from Resistant Line--------
```

Mean.Temp	Mortality	Residuals ...
51.3	102.5	0.13989
49.9	104.5	6.18613
50.0	100.4	1.79707
49.2	95.9	-0.39078
48.5	87.0	-7.26765
47.8	95.0	2.75551
47.3	88.6	-2.19942
45.1	89.2	4.75897
46.3	78.9	-9.00923
42.1	84.6	8.82948
44.2	81.7	-0.13989
43.5	72.2	-7.61678
42.3	65.1	-11.24853
40.2	68.1	-2.17920
31.8	67.3	21.29829
34.0	52.5	0.13988

6.13 RESISTANT.SMOOTH

Smoothing is based on the technique of using running medians to “smooth” out irregularities in data. This makes it possible to more easily visualize trends in the values. Typically, the X values are equally spaced measurements (like time series data) and therefore the Y values should change smoothly across each X interval. Running medians are computed and actually replace each Y-value with the value that is closest to those of its neighbors. There are two combinations of smoothers available when using P-STAT. One is the 4253H combination; the other is the 3RSSH combination. 4253H is assumed when neither is specified.

Figure 6.10 illustrates the output from RESISTANT.SMOOTH. In this figure, the body temperature of a cow is measured on successive days. The file CowTemp is input to EDA:

```
EDA  CowTemp ;
```

Figure 6.10 Resistant Smooth

```
----------RESISTANT.SMOOTH: Subcommands and Report----------

EDA  CowTemp ;
RESISTANT.SMOOTH  Temperature ;

 4253H Twice Smoothing completed for variable temperature

 Your new variables are:
            Smooth...
             Rough...


-----------Plot the Unsmoothed and Smoothed Values---------

PLOT  Day  BY  Temp ;


 File CowTemp            8 Line,  10 Character Plot
Day by Temperature

+ 28|             0                   5
+ 24| 2                   0      5
+ 20|           2                     0  5                        7
+ 16|      7    2                        0
+ 12|                                    5                        0
+  8|                          7                              5  2     R
+  4|                   7      0                         5         2
+   )                     7              2                        5
    ---+---------+---------+---------+---------+---------+---------+-----
       48        52        56        60        64        68        72


    PLOT,  Y  Day,  X  Smooth ;

 File CowTemp            8 Line,  10 Character Plot
Day by Smooth...

+    28|              02  5
+    24|              75  2   0
+    20|       0           2    75
+    16| 5    2            0
+    12|                                  7          5       2       0
+     8|                                                              0725
+     4|                                      0   2        5         7
+      )                                    25
       -+---------+---------+---------+---------+---------+---------+-----
       51        54        57        60        63        66        69
```

Smoothing of the data for the variable Temperature is then requested. RESISTANT.SMOOTH may be abbreviated to SMOOTH or R.SMOOTH if desired, and thus any of these subcommands may be used to produce the same smoothing procedure:

```
RESISTANT.SMOOTH,  VAR  Temperature ;
R.SMOOTH  Temp ;
SMOOTH    Temp ;
```

A message appears that the 4253H combination of smoothing has been completed for the variable Temperature and that new variables named “Smooth...” and “Rough...” have been created.

Plots of Day by Temperature and of Day by Smooth (smoothed Temperature) are then requested to illustrate the results of smoothing. It is difficult to see a pattern in the high and low temperature readings in the original data. Smoothing the data values, however, causes such patterns to become apparent. Plotting the variable Day by the new variable, “Smooth...”:

```
PLOT,  Y  Day,  X  Smooth ;
```

produces a nicely smoothed 9 line, 10 character plot.

There are several options available for RESISTANT.SMOOTH. The USING option requests that either the 4235H or 3RSSH smoothing procedure be used. The argument for USING must either be the character string 3RSSH or 4253H, enclosed in single or double quotes. 4253H is assumed and thus smoothing is done using the 4353H version unless 3RSSH is specified:

```
SMOOTH  Temperature,  USING  '3RSSH' ;
```

The smoothing options are a short-hand notation describing the smoothing procedure used. The procedures are running medians of various spans, multiplication of data values by weights (“hanning”), resmoothing and reroughing. (The Velleman and Hoaglin book describes these two smoothing options precisely.)

SMOOTH.OUT provides a variable name for the smoothed values that are created. If SMOOTH.OUT is not used, the new variable is called “Smooth...”. ROUGH.OUT provides a variable name for the residual or rough values that are created. If ROUGH.OUT is not used, the new variable is named “Rough...”.

6.14 DIFFERENCES AMONG GROUPS

Coded tables and median polish show differences among multiple groups. The groups are defined by categorical variables. The dependent or analysis variable is typically continuous; it is organized in a table defined by the groups. When the columns or rows of the table have an order, patterns and trends may be observed.

6.15 CODED.TABLE

CODED.TABLE displays the overall patterns and trends of an analysis variable in the rows and columns of a table. The rows and columns of a coded table are similar to the levels of a two-factor analysis of variance design. Typically, the row and column variables are ordered or unordered *categories*. The analysis variable is a *continuous* response measurement.

The data in a coded table are replaced with one-character codes that indicate the relative position of each data item in the data set. Data coded with a dot (.) are located within the hinges (the middle 50 percent of the data). Data coded with a plus (+) sign or a minus (-) sign are above or below the hinges, but within the fences. Those coded with a pound (#) sign or an equal (=) sign are outside the inner fences. Finally, data coded P (for Plus) or M (for Minus) are considered far outside on the high side and the low side respectively. If a cell is empty, it is coded with a blank. When there is *more than one* data value in a cell, the program must be told whether to code the minimum or maximum (extreme) value.

Figure 6.11 illustrates a coded table. In this example, test animals were given one of three poisons and treated in one of four different ways. The number of hours each animal lived appears in the table. A possible trend toward increased hours of life for combinations of higher treatment values with lower poison values is shown.

There are several options available when using the CODED.TABLE procedure. VAR specifies the name of the variable whose values are displayed in the cells of the table. CODED.TABLE may be followed directly by the name of the variable to be analyzed, or the identifier VAR may be used:

```
CODED.TABLE  Times ;   or
C.T,  VAR    Times ;
```

ROW.VARIABLE specifies the name of the row variable whose values define the number of rows in the table. COLUMN.VARIABLE specifies the name of the column variable whose values define the number of columns in the table. I n Figure 6.11, the ROW.VARIABLE is specified as Poison and the COLUMN.VARIABLE is specified as Treatment. LEGEND requests that an explanation of the symbol codes be provided at the bottom of the display.

Figure 6.11 Coded Tables

```
----------EDA: Command and Subcommands for CODED.TABLE-----

EDA  Survival ;

SHOW ;

Your variables are:
             Times
            Poison
         Treatment

CODED.TABLE  Times,  ROW.VARIABLE  Poison,
COLUMN.VARIABLE  Treatment,
LEGEND ;


----------CODED.TABLE Output-----------------------------

Coded Table of Times
Rows are:      Poison
Columns are:   Treatment

     1 2 3 4

   1 . + + +
   2 . # . +
   3 - . - .

-------------------------------------------

M   Far outside low
=   Below low inner fence (outside)
-   Below lower hinge but within inner fence
.   Between hinges
+   Above upper hinge but within inner fence
#   Above high inner fence (outside)
P   Far outside high
```

The PATTERN option is used when separate variables specifying the row and column cell associated with each response or "Y" measurement do *not* exist. PATTERN specifies the number of rows and number of columns in which the table is organized. The PATTERN chosen must be a multiple of the number of Y values. The data values are placed in the table according to the specified pattern. This subcommand defines a table with 8 rows and 4 columns:

```
PATTERN  8  BY  4 ;
```

The first data value is positioned in row 1, column 1; the second in row 1, column 2; the third in row 1, column 3, and so on. In other words, the Y values are positioned in the table *row-by-row*. If PATTERN is not specified, there must be a row variable and column variable whose values define the rows and columns of the table.

CODED.TABLE assumes that there is one Y value in each cell of the table. When there are multiple values in the cells, additional options specify which value to use. EXTREMES requests that the most extreme or maximum value be coded when more than one value falls in a cell. MAXIMUMS is a synonym for EXTREMES. MINIMUMS requests that the minimum value be coded when more than one value falls in a cell.

In summary, CODED.TABLE requires that either PATTERN or ROW.VARIABLE and COLUMN.VARIABLE be used. If PATTERN is used, the data values are placed in the table according to the specified pattern. If PATTERN is not used, then ROW.VARIABLE and COL.VARIABLE contain values that identify the rows and columns to which the data items belong. If there are multiple entries for a cell, the EXTREME entries are used unless MINIMUMS is specified.

6.16 MEDIAN.POLISH

MEDIAN.POLISH uses an additive model, which represents each cell of a table as the sum of a common value (the average Y or response measurement), a row effect and a column effect. This model is similar to the one used by ANOVA, except that MEDIAN.POLISH uses medians rather than means to summarize each of the effects and significance is not considered. Tables of residuals and comparison values are produced, unless subcommand identifiers specify otherwise.

The table of *residuals* shows the effect of each row and column on a cell in the row and column labeled "Effect". The residuals replace the original data in each cell. The common value is in the lower right corner of the table. The common value plus the row effect plus the column effect plus the residual in a given cell yields the original data value in that cell. Residuals close to *zero* are desirable.

The table of *comparison values* shows the Y values that are expected in each cell, given the row and column effects of that cell and the common value. The row effect times the column effect divided by the common value yields the comparison value. (This is similar to an "expected value".) Figure 6.12 illustrates the output from MEDIAN.POLISH.

A plot of residual values by comparison values may indicate a need for reexpression of the data values. If the slope of the line in the plot is zero (the residuals are constant across the comparison values), no reexpression is necessary. If the slope is approximately 1 (the residuals increase as the comparison values increase), then power 0 (a log transformation) is suggested. The goal is to reduce the size of the residuals to as close to zero as possible.

The MEDIAN.POLISH procedure can handle sparse data and data with multiple entries in the cells. However, two of the features, SORT and DISPLAY, work when there is *one and only one* entry for each cell.

There are several options available when using MEDIAN.POLISH. VAR specifies that name of the variable whose polished values are displayed in the cells of the table. MEDIAN.POLISH is followed directly by the name of the variable to be analyzed, or the identifier VAR may be used:

```
MEDIAN.POLISH  Scores ;
M.P,  VAR  Scores ;
```

STEPS specifies the number of polishing steps (sweeps). START, followed by ROW or COLUMN, specifies the order in which the polish is performed. START.ROW causes the rows to be polished before the columns and is assumed. START.COL causes the columns to be polished first.

Figure 6.12 Median Polish

```
----------EDA: Command and Subcommands for MEDIAN.POLISH----

 EDA  Winning ;
 M.P  Time,  ROW  Year,  COL  Distance ;


----------MEDIAN.POLISH Output----------------------------

 Residuals of Time
 Rows are:        Year
 Columns are:     Distance

                                      Eff
          100   200   400   800  1500   ect

 1948      -8    -3     0    18   106    11
 1952      -4    -4     0    21    63     8
 1956      -9   -11     2     0    17    14
 1960       2     2    -2     0   -25     0
 1964       0     0     0   -12     0     0
 1968      12     8     0    -7   -19   -13
 1972       5     1     0     0   -14    -4

 Effect  -351  -248     0   612  1730   451

 Comparison Values of Times
 Rows are:        Year
 Columns are:     Distance

             100      200      400      800     1500   Effect

 1948      -8.56    -6.05     0.0     14.93    42.20    11.00
 1952      -6.23    -4.40     0.0     10.86    30.69     8.00
 1956     -10.90    -7.70     0.0     19.00    53.70    14.00
 1960       0.0      0.0      0.0      0.0      0.0      0.0
 1964       0.0      0.0      0.0      0.0      0.0      0.0
 1968      10.12     7.15     0.0    -17.64   -49.87   -13.00
 1972       3.11     2.20     0.0     -5.43   -15.34    -4.00

 Effect  -351.00  -248.00     0.0    612.00  1730.00   451.00

 Your new variables are:
         Residuals...
     Compar.Values...
```

ROW.VARIABLE specifies the name of the row variable whose values define the number of rows in the table. COLUMN.VARIABLE specifies the name of the column variable whose values define the number of columns in the table.

PATTERN is used when a ROW.VARIABLE and COLUMN.VARIABLE are not specified. PATTERN specifies the number of rows and columns in which the table is organized. The data values are positioned according to the specified pattern. The first data value is positioned in row 1, column 1; the second in row 1, column 2; the third in row 1, column 3, and so on.

COMPARISON.VALUES provides a new variable name for the comparison values. RESIDUALS specifies a new variable name for the residuals.

SORT, followed by ROWS.EFFECTS, specifies that the rows should be reordered before they are displayed. SORT, followed by COLUMNS.EFFECTS, specifies that the columns should be reordered before they are displayed. SORT works *only* when there is one entry per cell. Sorting orders either the row or column effects so that a trend is discernible.

DISPLAY specifies what table should be displayed. DISPLAY may be followed by RESIDUALS, COMPARISON.VALUES, NONE or BOTH, indicating that residuals, comparison values, both (residuals and comparison values) or neither be displayed. DISPLAY BOTH is assumed. DISPLAY works *only* when there is one entry per cell.

SUMMARY

EDA

```
EDA  TestFile,  LABELS  'LabFile' ;
  BOX.PLOT  Income,  GROUP  Region;
  VAR       Savings;
  $

EDA  TestFile ,  OUT  OutFile ;
  LETTER.VALUES  MPG;
  SQUARE.ROOT;
  RESISTANT.SMOOTH;
  $
```

EDA, the Exploratory Data Analysis command, produces displays that describe a set of measurements and suggest models for further analysis. Transformations of the data set may be made, and the results may be observed in subsequent displays. Subcommands and subcommand identifiers specify the desired display and any options. Subcommands may be abbreviated, as long as the shortened names are sufficient to distinguish between the procedures.

Required:

EDA fn

provides the name of the required input P-STAT system file.

Optional Identifiers:

LABELS 'fn'

specifies an external file of value labels to be used for labeling grouped box plots. Enclose the file name in quotes.

OUT fn

provides a name for an optional output P-STAT system file that contains the input variables plus any variables generated during the course of the EDA command.

Optional Subcommands:

At least *one* of the EDA options must be specified. Each EDA procedure, such as BOX.PLOT, is followed by the specific subcommand options that apply to the procedure. The subcommands — AGAIN, HELP, PRESERVE.ORDER, RESET and SHOW and any transformations, may be used with all the EDA procedures.

AGAIN

re-executes the previous procedure.

HELP sub

requests help on the specified EDA subcommand procedure. HELP TRANSFORMATIONS is also supported.

PRESERVE.ORDER

requests that the order of data values be preserved, despite their transformation. Powers of positive numbers preserve the relative order of the transformations. However, powers of negative numbers may change the relative order, making PRESERVE.ORDER desirable.

RESET

resets all options to their original settings.

SHOW.VARIABLES

lists the current variables in the file. SHOW is an abbreviation for SHOW.VARIABLES.

Optional Subcommands: Transformations

Transformations are applied to the variable specified, *and* the previous EDA procedure is repeated when a new procedure is not specified. Transformations may also be used without specifying a variable name. In such cases, the values of the variable currently being displayed are transformed. NO TRANSFORMATIONS or POWER 1 repeats the previous EDA procedure, using the *original* (untransformed) values.

CUBE vn

requests that each data value be cubed.

CUBE.ROOT vn

requests that the cube root of each data value be used.

LOG vn

requests that the log, to the base e, of each data value be used.

LOG10 vn

requests that the log, to the base 10, of each data value be used.

POWER vn nn

requests that the data values of the specified variable be raised to the specified power:

```
POWER  Displacement  .25
```

All data values of Displacement are raised to the .25 power. POWER 1 repeats the current EDA procedure, using the *original* values.

RECIPROCAL vn

requests that the reciprocal or inverse of each data value be used.

RECIP.ROOT vn

requests that the reciprocal or inverse of the square root of each data value be used.

RECIP.SQUARE vn

requests that the reciprocal or inverse of the square of each data value be used.

SQUARE vn

requests that each data value be squared.

SQUARE.ROOT vn

requests that the square root of each data value be used. SQRT is an abbreviation for SQUARE.ROOT.

Optional Subcommands: Procedures

BOX.PLOT

```
BOX.PLOT,     VAR    Income ;
BOX  Income,  GROUP  Region ;
```

Box plots graphically summarize a batch of data by showing the median, the spread, and the tails of the data. Outliers are indicated. Multiple box plots on a single axis, showing values by groups, may be requested. BOX is an abbreviation for BOX.PLOT.

VAR vn

specifies the name of the variable to be used for the BOX.PLOT display. The variable name may also follow directly after BOX.PLOT or BOX:

```
BOX  Height ;
```

VAR is typically used to request the current procedure with a new variable:

```
VAR  Weight ;
```

VAR is an abbreviation for VARIABLE, which may also be used. If a variable is not specified, the most recently referenced variable is used for the box plot display.

AXIS

requests that the bottom axis be printed. AXIS is assumed unless NO AXIS is used.

GROUP vn

requests multiple box plots on the same axis, showing the data values by groups. The values of the specified variable determine group membership.

LEVELS nn

specifies integer levels of the groups to be displayed.

NOTCH

requests that the box plots be notched. Non-overlapping notches indicate groups that have significantly different medians at roughly the five percent level. (This is an individual five percent, with no allowance made for the number of comparisons made.) NO NOTCH is assumed unless NOTCH is specified.

CODED.TABLE

```
CODED.TABLE     SATMath,
  ROW.VARIABLE  Grade,   COLUMN.VARIABLE  Sex ;
```

```
C.T,  VAR  SAT.Math,   PATTERN  3  BY  4 ;
```

CODED.TABLE displays the patterns found in data in tables with coded symbols. T he symbols give the relative position of each data item in the data set. A LEGEND explaining the symbol codes may be requested. CODED.TABLE may be followed directly by the name of the variable to be analyzed, or the subcommand identifier VAR may be used. C.T is an abbreviation for CODED.TABLE.

VAR vn

specifies the name of the variable whose values are displayed in the cells of the table. This is the *dependent* or "Y" variable.

COLUMN.VARIABLE vn

specifies the name of the column variable. Its values define the number of columns in the table. This is one of the *independent* variables.

EXTREMES

requests that when *more than one* value falls in a cell, the most extreme value be coded. EXTREMES is assumed. EXTREMES is a synonym for MAXIMUMS.

LEGEND

requests that a legend giving the meaning of the symbol codes be displayed.

MAXIMUMS

requests that when more than one value falls in a cell, the maximum value be coded. This is assumed.

MINIMUMS

requests that when more than one value falls in a cell, the minimum value be coded. MAXIMUMS (EXTREMES) is assumed.

PATTERN nn BY nn

specifies the number of rows and the number of columns in which the table is organized. The data values are positioned in the table cells according to the specified pattern. For example, PATTERN 2 BY 3 defines a table with 2 rows and 3 columns. The first data value is positioned in row 1, column 1; the second in row 1, column 2; the third in row 1, column 3; the fourth in row 2, column 1; and so on going across the rows.

The pattern specified should accurately reflect the organization of the data values. If PATTERN is not used, there must be a variable whose values define the rows of the table and a variable whose values define the columns. These are the *independent* variables.

ROW.VARIABLE vn

specifies the name of the row variable. Its values defines the number of rows in the table. This is one of the *independent* variables.

LETTER.VALUES

```
LETTER.VALUES  Pulse ;
```

This EDA procedure shows the shape of the distribution in terms of the location of the median with respect to the location of the hinges (roughly quartiles), eighths, and so on. L.V is an abbreviation for LETTER.VALUES.

VAR vn

specifies the name of the variable to be used for the LETTER.VALUES display. The variable name may also follow directly after L.V or LETTER.VALUES. If a variable is not specified, the most recently referenced variable is used for the letter-values display.

GAUSSIAN

requests a comparison with the standard Gaussian or normal distribution (with a mean of zero and a standard deviation of 1). Spreads at the various letter values are given for the Gaussian distribution.

MEDIAN.POLISH

```
MEDIAN.POLISH  Tax ;
M.P ,   VAR    Tax ;
```

MEDIAN.POLISH uses an additive model, which represents each cell of a table as the sum of a common value of the dependent variable, a row effect and a column effect. This model is analogous to the one used by ANOVA, except that MEDIAN.POLISH uses medians rather than means to summarize each of the effects. Tables of residuals and comparison values are produced, unless subcommand identifiers specify otherwise. MEDIAN.POLISH may be followed directly by the name of the variable to be analyzed, or the subcommand identifier VAR may be used. M.P is the minimum abbreviation for MEDIAN.POLISH.

VAR vn

specifies the name of the variable whose polished values are displayed in the cells of the table. This is the *dependent* variable.

COLUMN.VARIABLE vn

specifies the name of the column variable. Its values define the number of columns in the table. This is one of the *independent* variables. If ROW.VARIABLE and COLUMN.VARIABLE are not used, PATTERN must be used.

COMPARISON.VALUES vn

provides a new variable name for the comparison values. COMP.VALUES is an abbreviation for COMPARISON.VALUES.

DISPLAY RESIDUALS

specifies that only the table of residuals be displayed. COMPARISON.VALUES or NONE or BOTH may also follow DISPLAY. BOTH (residuals and comparison values) is assumed. DISPLAY works only when there is *one* data value per cell.

PATTERN nn BY nn

specifies the number of rows and the number of columns in which the table is organized. The data values are positioned in the table cells according to the specified pattern. For example, PATTERN 2 BY 3 defines a table with 2 rows and 3 columns. The first data value is positioned in row 1, column 1; the second in row 1, column 2; the third in row 1, column 3; the fourth in row 2, column 1; and so on going across the rows.

The pattern specified should accurately reflect the organization of the data values. If PATTERN is not used, there must be a variable whose values define the rows of the table and a variable whose values define the columns. These are the independent variables.

RESIDUALS vn

provides a new variable name for the residuals.

ROW.VARIABLE vn

specifies the name of the row variable. Its values define the number of rows in the table. This is one of the *independent* variables.

SORT ROWS.EFFECTS

specifies that the rows should be reordered before they are displayed. SORT COLUMNS.EFFECTS reorders the columns. S ORT works only when there is *one* entry per cell.

START ROW

specifies the order in which the polish is performed. The rows are polished before the columns. START COL may be specified if the columns should be polished first. START ROW is assumed.

STEPS nn

specifies the number of polishing steps or sweeps.

RESISTANT.LINE

```
RESISTANT.LINE,  Y  Age,  X  Medical.Costs ;
R.L  Age  BY  Medical.Costs ;
```

RESISTANT.LINE fits a line to pairs of data values and calculates the equation of that line. This line is more resistant to the effects of extreme values because the slope of the line is calculated from pairs of median values that summarize thirds of the data set. R.L is an abbreviation for RESISTANT.LINE. The variables may be specified directly after the subcommand or with the subcommand identifiers Y and X.

Y vn

supplies the Y variable name. A new Y may be supplied without changing X. The variables may also be supplied directly after the subcommand name.

X vn

supplies the X variable name.

REPORT

requests that a report including the step information be printed. REPORT is assumed; NO REPORT may be used. NO SUMMARY and SUMMARY are respective synonyms for REPORT and NO REPORT.

RESIDUALS vn

provides a variable name for the residuals.

STEPS nn

specifies the number of iterations to be done.

SUMMARY

requests that only summary information is printed. The step information is omitted. NO SUMMARY is assumed. REPORT and NO REPORT are respective synonyms for NO SUMMARY and SUMMARY.

RESISTANT.SMOOTH

```
RESISTANT.SMOOTH  Temperature ;
SMOOTH  Temperature,  USING '3RSSH' ;
```

RESISTANT.SMOOTH smooths a sequence of data values so that each value is not as different from its neighbor as it was previously. Running medians replace each value, thereby smoothing out irregularities in the data. Two smoothing procedures are available. R.SMOOTH and SMOOTH are abbreviations for RESISTANT.SMOOTH.

VAR vn

specifies the name of the variable to be smoothed. Y followed by a variable name may also be used.

ROUGH.OUT vn

provides a variable name for the residual or rough values. If ROUGH.OUT is not used to provide a unique name, the new variable is called “Rough...”. Its values may replace the residuals of a previously smoothed variable.

SMOOTH.OUT vn

provides a variable name for the smoothed values. If SMOOTH.OUT is not used to provide a unique name, the new variable is called “Smooth...”. Its values may replace those of a previously smoothed variable.

USING 'cs'

specifies that RESISTANT.SMOOTH use the specified smoothing procedure. The argument for USING must be either 3RSSH or 4253H and it must be enclosed in single or double quotes. When a smoothing procedure is not specified, 4253H is used. (See *ABCs of EDA* by Paul F. Velleman and David C. Hoaglin, Duxbury Press, 1981, for an explanation of the algorithms used in these smoothing procedures.)

ROOTOGRAM

```
ROOTOGRAM  Weight ;
ROOT,      BINS  Age.Gps,  COUNT  Number ;
```

Rootograms are a type of frequency distribution or histogram. The ROOTOGRAM procedure apportions measurements into bins and determines the count in each bin.

VAR vn

specifies the name of the variable used in the ROOTOGRAM display. The variable name may also follow directly after ROOT or ROOTOGRAM. If a variable is not specified, the most recently referenced variable is used.

BINS vn

specifies the variable whose values are the bin boundaries. If the variable whose values are the actual data values is *not* input, the BINS variable should be an existing variable to provide bin boundaries. If the actual data values are also input:

```
ROOT  Weight,  BINS  Age.Gps,  COUNT  Number ;
```

the BINS variable should be a unique name for the values giving bin boundaries calculated by the program. If BINS and COUNT are not used, the variable whose values are the actual data values must be specified either after ROOTOGRAM or VAR. Also, the new variables giving bin boundaries and bin counts are named “Bins...” and “Counts...”.

COUNT vn

specifies the variable whose values are the counts for the corresponding bins. If the variable whose values are the actual data values is not input (as in the second example under the ROOTOGRAM command), the COUNT variable should be an existing variable to provide counts per bin. If the actual data values are also input, the COUNT variable should be a unique name for the values giving bin counts calculated by the program. If BINS and COUNT are not used, the variable whose values are the actual data values must be specified either after ROOTOGRAM or VAR. Also, the new variables giving bin boundaries and bin counts are named “Bins...” and “Counts...”.

FITTED.VALUES vn

provides a variable name for the fitted values. If this is not used, the residuals are named “Fitted.Values...”.

RESIDUALS vn

provides a variable name for the residuals. If this is not used, the residuals are named “Residuals...”.

STEM.AND.LEAF

```
STEM.AND.LEAF,  VAR  Temperature ;
STEM  Temperature ;
```

A stem-and-leaf display, a type of histogram on its side, uses the actual digits in the data values. It shows the range and symmetry of the data, unimodal or multimodal patterns, concentrations, gaps and stray values. ST is an abbreviation for STEM.AND.LEAF.

VAR vn

specifies the name of the variable to be used for the STEM.AND.LEAF display. The variable name may also follow directly after STEM.AND.LEAF or STEM. If a variable is not specified, the most recently referenced variable is used for the stem and leaf displays.

SCALE EXTREME

requests that the stem-and-leaf display include *all* data values of the specified variable. SCALE FENCE is the opposite of SCALE EXTREME. It requests that only data values within the inner fence limits be shown on the stem-and-leaf display. Data values outside the inner fences are shown on special LO and HI stems. SCALE FENCE is assumed when the identifier SCALE is not used.

TRIM

is a synonym for SCALE FENCE. NO TRIM is a synonym for SCALE EXTREME. (See SCALE.)

XY.PLOT

```
XY.PLOT   Height  BY  Weight ;
PLOT      Height  BY  Weight ;
XY,    Y  Height,  X  Weight ;
```

An XY.PLOT is a *condensed*, *coded* scatter plot. Counts of observations at a given point are not shown on an XY.PLOT, as they are on traditional scatter plots. Rather, ten lines or divisions of the traditional vertical (Y) axis are condensed into one line on the XY.PLOT. A coded symbol, a number from 0 to 9, indicates the relative positions of the data values. XY or PL are abbreviations for XY.PLOT. The variables to be plotted may be specified directly after XY.PLOT, or with the subcommand identifiers X and Y.

Y vn

specifies the Y variable name. If X is not given, the previous X is used. Y and X variables may be explicitly supplied, or the PLOT vn BY vn format may be used.

X vn

specifies the X variable name. If Y is not given, the previous Y is used. All values are plotted unless a RANGE is specified.

Y.RANGE nn TO nn

specifies a range of Y values to be plotted:

```
Y.RANGE  6  TO  10 ;
```

Y.R is an abbreviation for Y.RANGE. A range that pertains to both X and Y values may be supplied:

```
RANGE  20  TO  50 ;
```

RA is an abbreviation for RANGE.

X.RANGE nn TO nn

specifies a range of X values to be plotted.

CHARS nn

specifies the number of subdivisions (characters) of each line. The number must be between 1 and 9. When CHARS is not used, the line is subdivided into ten divisions and ten characters (0 through 9) are used.

LINES nn

specifies the number of lines to be used for the plot. The number must be between 5 and 40. If LINES is not used, ten lines are assumed.

7
TURF.SCORES

The TURF.SCORES command is used after a TURF run to create an output file with variables from the original input file plus two new variables: the reach score for the case, and whether this case was actually reached. If demographic variables are included in the TEMPLATE output file, they can be used to identify the subjects who contributed the most to the TURF results.

7.1 THE TEMPLATE FILE

The TURF.SCORES command needs to know the items and options to be used in the scoring. These can be supplied within the TURF.SCORES command itself. However, if you want to score the best result from a TURF run, it is easier to have TURF write a TEMPLATE file which TURF.SCORES can read directly. This option cannot be used when several sizes are being run.

Figure 7.1 Template File for Use in TURF.SCORES

```
   -----Creating the TEMPLATE file in the TURF Command-----

TURF ddd [ DROP case.id ],
           CASE.WEIGHT www,
           SIZE 3,
           TEMPLATE ttt,
           REACH.RESULTS rrr$


   -----File ttt-----

           item  case     response      reach
 items  weights  weights  weights   threshold

 v2           1  www      no               1
 v3           1  -        -                -
 v5           1  -        -                -
```

TEMPLATE ttt, This optional output P-STAT system file contains the names of the items that comprised the best combination. It also contains information about the options (like weighting) that were used. This file can be given to the TURF.SCORES command, to score the cases on the combination contained in the file. TURF.SCORES then writes an output file that has the reach score for each case on that combination. It is then quite easy to investigate the demographics of the reached cases.

Figure 7.1 shows a typical template file for a combination of 3 items, with variable www being used as a case-weight, and no other options in use.

Figure 7.2 A Typical Final Report From the TURF COmmand

```
------------TURF analysis for file ddd completed-------------
| OPTIONS: case.weights (in variable www)                    |
|                                                            |
|        5 items were used in the analysis.                  |
|                                                            |
|       16 cases were read and used; the sum                 |
|          of the weights for these cases was 20.            |
|                                                            |
|       16 cases had at least one positive response.         |
|          The sum of the weights for these cases was 20,    |
|          making that the maximum possible reach.           |
|                                                            |
| SIZE   3 evaluated 10 combinations:                        |
|       17 was the best REACH, found in 1 combination.       |
|       17 was the FREQ value in that combination.           |
|       19 was the best FREQ in any size 3 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 reached cases.       |
|                                                            |
| REACH.RESULTS file rrr has the 10                          |
| combinations with the highest reach scores.                |
| The items are ordered by their REACH contribution.         |
| Cumulative reach is shown.                                 |
|                                                            |
| TEMPLATE file ttt has the names of the items               |
| that made up the best combination.                         |
|                                                            |
| Time: less than 0.1 second.                                |
------------------------------------------------------------
```

7.2 TURF.SCORES

```
TURF.SCORES xin, template           carry case.num, out xout $
TURF.SCORES xin, items v2 v4 to v8, carry case.num, out xout $
```

TURF.SCORES computes the REACH score on a specified combination of items for each case of an input file. These scores are written to an output file. The calculations are identical to those in the TURF command. The output file will have the items used in the scoring, the reach score, and any “carried” variables. Carried variables are usually variables that identify the individual cases, facilitating demographic breakdowns of the reached cases.

The command must be given the names of the variables (items) them make up the combination to be scored. In addition, the defaults can be changed for various options. These are: case weighting, item weighting, response weighting and the response threshold. This information can be supplied in 2 ways:

1. by providing a TEMPLATE file (created in a TURF command).
2. by providing the controls as part of this command.

Figure 7.3 TURF.SCORES: An Example

------------------THE COMMAND----------------

```
TURF.SCORES ddd, ITEMS v1 v2 v5,
      RESPONSE.WEIGHTS,
      REACH.THRESHOLD 2,
      CARRY case.id,
      OUT sss $
```

--------------------THE REPORT----------------

```
--------------TURF.SCORES completed--------------
| The reach scoring was done using these items: |
| v1               v2               v5         |
|                                               |
| 16 cases were read from file ddd.             |
| The reach.threshold was 2.                    |
| The threshold was met by 8 cases.             |
| The RESPONSE.WEIGHTS option was in use.       |
| 6 variables were written to file sss.         |
-------------------------------------------------
```

-----------------THE OUTPUT FILE---------------

case id	v1	v2	v5	reach score	reach category
9001	1	1	0	2	1
9002	1	1	0	2	1
9003	1	1	0	2	1
9004	1	1	0	2	1
9005	1	1	0	2	1
9006	1	1	0	2	1
9007	0	1	0	1	0
9008	0	0	0	0	0
9009	0	0	0	0	0
9010	0	0	0	0	0
9011	0	0	0	0	0
9012	0	0	0	0	0
9013	0	0	0	0	0
9014	0	0	0	0	0
9015	0	0	2	2	1
9016	0	0	2	2	1

7.3 Identifiers When Using a TEMPLATE File

The input data file, the template file and the output file are required. The CARRY identifier is optional.

1. TURF.SCORES file, this supplies the input data file and is required.
2. TEMPLATE file, this is a file, created in a TURF run, that contains the settings that were used, and the names of the items that made up the best combination.
3. OUT file, name for the result file. Required.
4. CARRY vars, variables that should be carried over from the input file to the OUT file, even though they are not involved in the execution of the command. This is optional unless you wish to examine the demographics of the respondents.

7.4 TURF.SCORES Output File

The OUT file will have the following variables:

1. the items in the combination that was scored,
2. the case-weight variable (if there was one), and
3. any CARRY variables that were requested.

In addition two new variables are added:

1. REACH.SCORE, the score of each case for the combination. This is set to M1 if there are negative or missing values for the case in the combination items. It is also M1 if the case has a non-positive case-weight.
2. REACH.CATEGORY, was the case reached, given its score and the threshold in use? It is M1 when reach.score is M1. It is zero when reach.score is less than the threshold. It is one when reach.score satisfies the threshold.

7.5 Identifiers When Not Using a Template file

1. TURF.SCORES file, this supplies the input data file and is required.
2. ITEMS vars, names of the variables (items) that make up the combination to be scored. Ranges can be used: var2 to var5. This is required.
3. ITEM.WEIGHTS numbers, weights of the items. The default is to treat all items the same, i.e., with weights of 1. This is optional
4. CASE.WEIGHT var, name of a variable to be used for case weighting when computing the overall reach for the dataset. This is optional.
5. RESPONSE.WEIGHTS, causes the actual values of the items to be used in the scoring instead of just zero/one. This is optional.
6. REACH.THRESHOLD number, default one. the value that a case must achieve to be “reached”. This is optional.
7. OUT file, name for the result file. This is required.
8. CARRY vars, variables that should be carried over from the input file to the OUT file, even though they are not involved in the execution of the command. This is optional unlees you wish to examine the demographics of the respondents.

SUMMARY

TURF.SCORES

This command creates an output file which contains variables from the original input file plus 2 new variables: the reach score for the given case and a variable which tells whether that case was actually reached. The combination being scored is defined either by a TEMPLATE file or by the ITEMS identifier.

Required:

TURF.SCORES fn

provides the name of the input file.

Identifiers When Using a Template File:

TEMPLATE fn

provides the name of the TEMPLATE file created by a previous TURF command. Required.

OUT fn

provides the name for the output file. Required.

CARRY vn vn TO vn

provides the list of the variables in the input file that should be carried over to the output file. Optional but necessary if you wish to look at the demographics of the respondents that were reached.

Identifiers When NOT Using a Template File:

ITEMS vn vn TO vn

provides the names of the variables (items) that make up the combination to be scored. Required.

OUT fn

provides the name for the output file. Required.

ITEM.WEIGHTS nn nn nn

provides weights for the items. Optional.

CASE.WEIGHTS vn

provides the name of the variable to be used for case weighting when computing the overall reach for the dataset. Optional.

RESPONSE.WEIGHTS

causes the actual values of the items to be used in the scoring. Optional.

REACH.THRESHOLDnn

provides the value that a case must achieve to be ‘reached’. The default is one. Optional.

CARRY vn vn TO vn

variables to be carried over from the input file to the OUT file. Optional.