Report on UNIX|STAT release 5.1: Data analysis programs for UNIX and MSDOS

GARY PERLMAN

Wang Institute of Graduate Studies, Tyngsboro, Massachusetts

and

FREDERICK L. HORAN

Cornell University, Ithaca, New York

UNIX|STAT is a statistical package developed at the University of California, San Diego, and at the Wang Institute of Graduate Studies. Over 20 programs allow the manipulation and analysis of data and are complemented by a detailed tutorial, a manual entry for each program, and a quick reference sheet. The package was first introduced as 4 programs (Perlman, 1980). Since then, many programs have been added, and since 1980, the package has been distributed to over 400 UNIX sites. The portability of the package, written in C, was demonstrated when it was ported from UNIX to MSDOS at Cornell University on an IBM PC using the Lattice C compiler. The added capabilities of the package, combined with the availability on the popular MSDOS, used on IBM, AT&T, Wang, and other personal computers, make the programs even more useful for psychological research and instruction.

UNIX|STAT is a set of data analysis programs developed on the UNIX operating system (Ritchie & Thompson, 1974) using the C programming language (Kernighan & Ritchie, 1979). UNIX|STAT was first announced at the 1979 meeting of the Society for Computers in Psychology, then the Conference for the Use of On-Line Computers in Psychology (Perlman, 1980). Since then, many new programs have been added, and the package has been made more consistent, faster, more robust, and better documented. The programs have enjoyed use in hundreds of government, research, educational, and industrial sites running UNIX. UNIX|STAT has been ported to the similar, but simpler, MSDOS operating system that runs on many personal computers with at least 96K of memory. In this paper, we describe the UNIX|STAT package running on the MSDOS system. Because the versions on UNIX and MSDOS are similar, this paper also serves as a status report on the continuing development of UNIX|STAT.

MSDOS is an interactive program-execution environ-

ment that runs on many personal computers, notably the IBM PC family (PC, XT, and AT) and IBM-compatible systems like the Wang and AT&T personal computers. Being interactive, MSDOS is an ideal environment for data analysis. Because UNIX|STAT runs on MSDOS, it can be used on many personal computers used in research and educational laboratories.

Before describing the programs, we will provide an overview of the MSDOS environment, which is similar to the UNIX environment. This will help explain how several simple programs work together to provide a flexible environment for data manipulation and analysis.

The UNIX|STAT programs were designed with the UNIX philosophy that programs should be designed to do one task well and to produce output suitable for input to other programs. By following this design philosophy, the UNIX|STAT programs can use existing utilities. This philosophy extends well to the MSDOS environment where *pipes* and *filters* are supported. A pipe allows the output from one program to be made the input to another without requiring the user to create a temporary file. A filter is a program designed to be used in pipelines. An example of a filter program on MSDOS is the **sort** utility that can be used with UNIX|STAT programs.

Using filter programs affects how data files are used. UNIX|STAT users typically keep their data in what we call a master data file. Using data manipulation programs, extractions from the master data file are transformed into a format suitable for input to an analysis program. Note that the original data do not change, but copies are made for transformations and analysis. Thus, an analysis consists of an extraction of data, optional transformations, and some analysis. Pictorially, this can be shown as:

Many people have contributed to the development of UNIX|STAT. Jay McClelland was instrumental in his influence on the user interface. Mark Wallen was an indispensable programming consultant. Caroline Palmer helped with enhancements to the regress program and with the port to MSDOS. Dozens of useful suggestions have come from too many people to name.

People on computer networks can send electronic mail to: **decvax!wanginst!perlman** (for UUCP users), **sdcs!perlman@nprdc** (for ARPA users), or **perlman@wanginst** (for CSNET users).

MSDOS is a trademark of Microsoft Corporation. UNIX is a trademark of AT&T Bell Laboratories.

Requests for reprints, and inquiries about UNIX|STAT, should be directed to Gary Perlman, School of Information Technology, Wang Institute of Graduate Studies, Tyngsboro, MA 01879.

< data extract | transform | analysis

where a copy of the data has some subset extracted by an extraction program, transformed by a transformation program, and analyzed by an analysis program. The use of pipelines, signaled with the pipe symbol, |, is the reason for the name UNIX|STAT.

PROGRAM DESCRIPTIONS

There are three types of programs: those for data manipulation and transformations, those for analyses, and special programs, some of which are for mathematical psychology. These distinctions correspond to the arrangement of the programs on floppy diskettes and in this section. Some programs in UNIX|STAT are not supported on MSDOS: **io**, to control and monitor input and output flow, and **manstat**, to print the on-line manual.

For each of the programs in the following sections, a brief description of the program is followed by its prototypical use, a summary of available options, and the program limits. First, we will discuss features in all of the programs.

Command Format

Programs are run on UNIX and MSDOS by typing the name of the command, followed by program options and program operands, such as expressions or file names. Program names, options, and operands are separated or delimited by blank space. On UNIX, program names tend to be lowercase, whereas on the case-insensitive MSDOS, they are always uppercase, although users can type the names in lowercase. Program options and operands are sometimes complex, and it is sometimes useful to insert spaces into an option value or an operand, either to improve the output or to make the command line more readable. On UNIX, this is done by quoting the parts that should be kept together, but on MSDOS, there is no quoting supported. Fortunately, it is not often necessary to have nondelimiting spaces. A similar problem exists on MSDOS, because there is no way to pass the special characters for file redirection and piping to a program.

Program Options

Program options conform to the standard UNIX option-parsing convention (Hemenway & Armitage, 1984). In this standard, all program options are single characters preceded by a minus sign. For example, **-a** and **-X** are both options. All program options must precede operands (such as file names, variable names, or expressions). Some options require values, and these should follow the option. For example, the **pair** plotting function allows setting the height of the plot with the **-h** option: **-h 30** sets the plot height to 30 lines. There should be a space between an option and its value. Options that do not take values (logical options) can be "bundled" or grouped to save typing. For example, the descriptive statistics program, **desc**, has options for requesting a histogram, a table of frequencies, and a table of proportions. These can

be requested with the bundle of options, **-hfp**, instead of the longer **-h -f -p**.

File Inputs and Outputs

Most of the UNIX|STAT programs are filters, meaning that they can read from the standard input and write to the standard output. By default, the standard input is the keyboard, and the standard output is the screen. The standard input and output can independently be redirected using special characters: **<**, to redirect the standard input from an immediately following file name; **>**, to redirect the standard output to a file; or **|**, to connect the output from one program to the input of another. Some of these features are not available on early versions of MSDOS (before Version 2.0). The following command says for the **anova** program to read from the file **anova.in** and write to the file **anova.out**:

```
anova < anova.in > anova.out
```

If a program is expecting input from the keyboard, a prompt will be printed on the screen. Often, input from the keyboard is a mistake; most people do not type directly into an analysis program, but edit a file to be used as input. Another example: a pipe and output redirection could help get statistics on a series of numbers from 100 to 200:

```
series 100 200 | desc > results
```

In English, this is read, "A series of the numbers from 100 to 200 is created and sent to the **desc** descriptive statistics program, whose output is placed in the file **results**."

If a program reads from files directly, then the special name **-** (minus or dash) by itself says to insert the standard input. For example, the **abut** program can read several files with the standard input inserted with the following command line:

```
series 1 20 | abut file1 file2 - file3
```

The output would be four columns, the third of which would be the series 1-20.

Although pipes are supported on MSDOS, they are not efficient and they require that there be enough space for temporary files to hold the contents of the pipes (temporary files with names like PIPE%1.\$\$\$). This can make input and output redirection without pipes a better choice for speed, especially in command scripts, called "batch files" on MSDOS.

Input Formats

UNIX|STAT programs have simple input formats. Program input is read until the end of file, **EOF**, is found. On UNIX, end of input from the keyboard is signaled by typing **CONTROL + d**, or **^D**. MSDOS users type **^Z** to signal the end of input. End of file in disk files is done by the system; no special marking characters are needed or allowed.

Input fields (visibly distinguishable words) are separated by white space (blank spaces, tabs, new lines). For some programs, fields in files with embedded spaces can be en-

closed by single or double quotes. Most of the programs, especially analysis programs, ignore blank input lines used to improve the human readability of the data.

Program Limits

There is a system-dependent limit on the count of characters in an input line: on small systems, 512 characters, and on large ones, 1024. Many programs use dynamic memory allocation so that the memory available on a machine will determine the size of data sets that can be analyzed. Integer overflow is not checked, so numbers like data counts are limited on 16-bit machines to 32767; in practice, this has not presented problems. All calculations are done with double-precision floating-point numbers.

Error Messages

All error and warning messages identify the program detecting the problem (useful when pipelines are used), print diagnostic information, sound a bell, and, for errors, cause an exit.

Batch Files and Shell Scripts

Because the UNIX|STAT programs work well together, and because most data analysis is routine, it is often advantageous to save a series of commands in a file for later analyses. Both UNIX and MSDOS support this, MSDOS with *batch* files and UNIX with *shell* scripts. Batch files and shell scripts also support variables, some set by command line calls and some set internally. They provide UNIX|STAT with a simple but effective programming facility.

DATA MANIPULATION AND TRANSFORMATIONS

The UNIX|STAT data manipulation programs allow reformatting and conditional algebraic and string transformations of data. Many of the programs are useful outside the statistical domain because they are compatible with MSDOS and UNIX filter utilities.

abut: formatted printing of corresponding lines from files

abut combines corresponding lines from files so that, pictorially, the files are pasted beside each other. Typically, **abut** prepares input files for analysis programs that require multicolumn inputs. Optionally, the printing format can be adjusted, lines numbered, and input files re-wound and repeated to match the length of the longest file. The number of files that can be open at one time is system dependent, but is usually around 16.

colex: column/field extraction

colex efficiently extracts the requested whitespace-separated columns from its input. Typically, it sends a part of a larger file to another program. Optionally, missing columns can be replaced or ignored, and columns can

be surrounded by quotes. Ranges of columns, either increasing or decreasing, help abbreviate requests. At most, 100 input columns are allowed.

dm: formatting and conditional transformations

dm is the most important program for data manipulation. It has column-extraction capabilities like **colex**, string-manipulation and comparison operations, arithmetic and standard mathematical functions, and several special control features. Typically, it prepares some data for input to an analysis program. It can be used interactively, or in batch mode with algebraic and string expressions supplied on the command line. It can work with 100 input columns, each at most 15 characters wide, and can produce as many as 100 columns of output, one for each expression. There is a detailed tutorial and reference manual for this program.

maketrix: build matrix format file

maketrix reformats its free format input so that each line of output has the same number of columns. Typically, it converts its input into matrix format for input to an analysis program. By default, a two-column matrix is formed, but any number of columns can be requested. Unless asked to be silent, **maketrix** complains about last lines with missing columns.

perm: randomly permute lines

perm produces a random permutation of its input lines. Typically, it randomizes items for experiments or simulations. A random seed for the random number generator can be supplied on the command line, allowing for replications of random orderings. Otherwise, the MSDOS program user is prompted for the seed, and the UNIX version of **perm** uses the time and process identification number to form the seed. At most, 1000 lines can be permuted.

repeat: repeat a string or file

repeat repeats its input file or argument string a specified number of times. Typically, it duplicates its input or produces a fixed number of input lines. When reading the standard input, **repeat** stores its input in a special file so that it can be repeated, so there must be space on the disk to hold the input.

reverse: reverse lines, fields, or characters

reverse reverses its input, either by lines, by columns, or by characters. Typically, it reverses columns to make the last columns in a file more accessible. It has options to request reversal by lines, fields, characters, or any combination. When lines are being reversed, there is a limit of 1000 lines. When fields are being reversed, there is a limit of 100 fields.

series: create a series of numbers

series creates a series of numbers inclusively between a minimum and a maximum. It can also produce a decreas-

ing series. Typically, it creates a vector of numbers that may be randomly permuted or used for some calculation. By default, series produces numbers changing by ones from the start to the end of the series. Optionally, the increment (or decrement) of the series can be supplied, making it possible, for example, to count backward from 10 to 1 by halves:

```
series 10 1 -0.5
```

transpose: transpose matrix type file of strings

transpose flips the rows and columns of a matrix format file. Typically, it reformats a row-oriented file for a column-oriented program. It has an option to control the width of the output columns. By default, the output columns are separated by tabs. It is limited to 100 input columns and 100 output columns (i.e., input lines).

validata: validate columnar file for consistency

validata reports on the data types, counts, minimum and maximum values, and regularity of the number of columns in its input. It can find an irregularity in a data file that produced an error message from another program. At most, 100 input columns are allowed.

ANALYSIS PROGRAMS

The UNIX|STAT analysis programs have facilities for summary statistics, simple graphics, and inferential statistics. All of the programs have extensive input checking and diagnostic error messages. Because program outputs best convey their capabilities, we provide example inputs and outputs for the analysis programs.

anova: multifactor analysis of variance

anova performs an analysis of variance with one random factor and up to nine independent factors. Both within-subjects and unequal-cells between-subjects factors are supported and there can be up to 100 levels of any factor. The input format is simple: each datum is preceded by a description of the conditions under which the datum was obtained. For example, if Subject 3 took 325 msec to respond to a loud sound on the first trial, the input line to anova might be:

```
s3 loud 1 325
```

From input lines of this format, **anova** infers whether a factor is within- or between-subjects, prints cell means for all mean effects and interactions, and prints standard format *F* tables with probability levels. The computations done in **anova** are based on a method of analysis described by Keppel (1973); however, for unequal cell sizes on between-groups factors, the weighted-means solution is used instead of Keppel's preferred unweighted solution. The weighted-means solution requires that sample sizes must be in constant proportions across rows and columns in interactions of between-subjects factors or else the anal-

ysis may be invalid. The input to **anova** is read twice—the first time to infer the experimental design, and the second to do computations—so there must be adequate disk space to hold the input in a temporary file.

An example input to **anova** is shown below. The output from **anova** is shown in Table 1.

```
s1 loud 1 259
s1 loud 2 228
s2 soft 1 526
s2 soft 2 480
s3 loud 1 325
s3 loud 2 315
s4 soft 1 418
s4 soft 2 397
```

The call to **anova** gives meaningful names to the columns of its input.

desc: describing a single distribution

desc describes a single distribution. Summary statistics, modifiable format histograms and frequency tables, and single distribution *t* tests are supported. **desc** reads free format input, with numbers separated by any amount of whitespace (blank spaces, tabs, new lines). When order statistics are being printed, or when a histogram or frequency table is being printed, there is a limit to the number of input values that must be stored. Although system

Table 1
anova subject noise trial RT

SOURCE: grand mean					
noise	trial	N	MEAN	SD	SE
		8	368.5000	104.8713	37.0776
SOURCE: noise					
noise	trial	N	MEAN	SD	SE
loud		4	281.7500	46.1257	23.0629
soft		4	455.2500	58.8749	29.4374
SOURCE: trial					
noise	trial	N	MEAN	SD	SE
	1	4	382.0000	116.0603	58.0302
	2	4	355.0000	108.1943	54.0971
SOURCE: noise trial					
noise	trial	N	MEAN	SD	SE
loud	1	2	292.0000	46.6690	33.0000
loud	2	2	271.5000	61.5183	43.5000
soft	1	2	472.0000	76.3675	54.0000
soft	2	2	438.5000	58.6899	41.5000
FACTOR: subject noise trial RT					
LEVELS:		4	2	2	8
TYPE :		RANDOM	BETWEEN	WITHIN	DATA
SOURCE					
	SS	df	MS	F	p
=====					
mean	1086338.0000	1	1086338.0000	145.111	0.007 **
s/n	14972.5000	2	7486.2500		
noise	60204.5000	1	60204.5000	8.042	0.105
s/n	14972.5000	2	7486.2500		
trial	1458.0000	1	1458.0000	10.942	0.081
ts/n	266.5000	2	133.2500		
nt	84.5000	1	84.5000	0.634	0.509
ts/n	266.5000	2	133.2500		

dependent, usually several thousand values can be stored. For frequency tables and histograms, there is a limit of 500 bins.

An example input to **desc** is shown below. The output from **desc** is shown in Table 2.

```
3 3 4 4 7 7 7 7 8 9 1 2 3 4 5 6 7
8 9 9 8 7 6 5 4 3 2 4 5 6 1 2 3 4 3 1 7 7
```

oneway: one-way analysis of variance and *t* test

oneway performs a between-groups one-way analysis of variance. It is redundant with the **anova** program, but is provided to simplify analysis of this common experimental design. The input to **oneway** consists of each group's data, in free-input format, separated by a special value called the *splitter*. Group sizes can differ, and like **anova**, **oneway** uses a weighted-means solution. At most, 20 groups can be compared. When 2 groups are being compared, the *-t* option prints the significance test as a *t* test. The program is based on a method of analysis described by Guilford and Fruchter (1978).

An example input to **oneway** is shown below. The output from **oneway** is shown in Table 3.

```
1 2 3 4 5 4 3 2 1
999
3 4 5 4 3 4 5 4 3
999
7 6 5 7 6 5
```

pair: paired points analysis and plots

pair analyzes paired data by printing summary statistics and significance tests for two input variables in

Table 2
desc -o -hcfp -m 0.5 -M 8 -i 1 -t 5

Under Range	In Range	Over Range	Sum	
0	35	3	164.000	
Mean	Median	Midpoint	Geometric	Harmonic
4.686	4.000	4.500	4.055	3.296
SD	Quart Dev	Range	SE mean	
2.193	2.000	7.000	0.371	
Minimum	Quartile 1	Quartile 2	Quartile 3	Maximum
1.000	3.000	4.000	7.000	8.000
Skew	Kurtosis			
-0.064	1.679			
Null Mean	t	prob (t)	F	prob (F)
5.000	-0.848	0.402	0.719	0.402
Midpt	Freq	Cum	Prop	Cum
1.000	3	3	0.086	0.086 ***
2.000	3	6	0.086	0.171 ***
3.000	6	12	0.171	0.343 *****
4.000	6	18	0.171	0.514 *****
5.000	3	21	0.086	0.600 ***
6.000	3	24	0.086	0.686 ***
7.000	8	32	0.229	0.914 *****
8.000	3	35	0.086	1.000 ***

Note—The call to **desc** includes many options: *-o* for order statistics; *-hcfp* for a histogram and a table with cumulative frequencies and proportions; *-m 0.5* to set the minimum allowed value to 0.5; *-M 8* to set the maximum allowed value to 8; *-i 1* to set the interval width in the histogram and table to 1; and *-t 5* to request a *t* test with null mean equal to 5.

Table 3
oneway -s 999 less equal more

Name	N	Mean	SD	Min	Max
less	9	2.778	1.394	1.000	5.000
equal	9	3.889	0.782	3.000	5.000
more	6	6.000	0.894	5.000	7.000
Source	SS	df	MS	F	p
Between	37.556	2	18.778	16.132	0.000 ***
Within	24.444	21	1.164		

Note—The call to **oneway** includes the *-s* option with 999 as the value of the splitting value between groups. Meaningful names are given to the groups.

columns and their difference, correlation and simple linear regression, and plots. The input consists of a series of lines, each with two paired points. When data are being stored for a plot, at most 1000 points can be processed. The maximum plot dimensions are 100×100.

An example input to **pair** is generated using the series and **dm** programs connected by pipes. The input to **pair** are the numbers 1–100 in Column 1, and the square roots of those numbers in Column 2. The **dm** built-in variable **INLINE** is used in a condition to switch the sign of the second column for the second half of the data. The output from **pair** is shown in Table 4. **pair** reads X-Y points and predicts Y (in Column 2) with X (in Column 1). The significance test of the difference of the columns against 0.0 is equivalent to a paired-groups *t* test.

```
series 1 100 | dm x1 "(INLINE>50?-1:1)*x1.5"
```

The call to **pair** includes several options: *-sp* requests statistics and a plot, *-w 40* sets the width of the plot to 40 characters, and *-h 15* sets the height of the plot to 15 characters.

regress: multivariate linear regression and correlation

regress performs a multivariate linear correlation and regression analysis. Its input consists of a series of lines, each with an equal number of columns, one column per variable. In the regression analysis, the first column is predicted with all the others. There are options to print the matrix of sums of squares and the covariance matrix. There is also an option to perform a partial correlation analysis to see the contribution of individual variables to the whole regression equation. The program is based on a method of analysis described by Kerlinger and Pedhazur (1973). Nonlinear regression models are possible using transformations with UNIX/STAT utilities, such as **dm**. The program can handle up to 20 input columns, but the width of the output for more than 10 is awkward.

The following artificial example predicts a straight line with a log function, a quadratic, and some random numbers. The input to **regress** is created with **series** and **dm**. The output from **regress** is shown in Table 5.

```
series 1 20 | dm x1 log(x1) x1*x1 floor(RAND*10)
```

The call to **regress** includes the *-p* option to request a

Table 4
pair -sp -w 40 -h 15

	Column 1	Column 2	Difference
Minimums	1.0000	-10.0000	0.0000
Maximums	100.0000	7.0711	110.0000
Sums	5050.0000	-193.3913	5243.3913
SumSquares	338350.0000	5049.9989	395407.6303
Means	50.5000	-1.9339	52.4339
SDs	29.0115	6.8726	34.8845
t(99)	17.4069	-2.8140	15.0307
p	0.0000	0.0059	0.0000
Correlation	r-squared	t(98)	p
-0.8226	0.6767	-14.3219	0.0000
Intercept	Slope		
7.9070	-0.1949

Note - The output from pair is shown in Table 4. Pair reads X-Y points and predicts Y (in Column 2) with X (in Column 1). The significance test of the difference of the columns against 0.0 is equivalent to a paired-groups t test.

partial correlation analysis and meaningful names for the variables in the analysis.

stats: print summary statistics

stats prints summary statistics for its input. Its input is a free format series of strings, from which it extracts numbers for analysis. When a full analysis is not needed, the following options request specific statistics:

n min max sum ss mean var sd skew kurt

ts: time series analysis and plots

ts performs simple analyses and plots for time series data. Its input is a free format stream of at most 1000 numbers. It prints summary statistics for the time series, allows rescaling of the size of the time series so that time series of different lengths can be compared, and optionally computes autocorrelations of the series for different lags. Autocorrelation analysis detects recurring trends in data. For example, an autocorrelation of Lag 1 of a time series pairs each value with the next in the series. ts is best demonstrated on an oscillating sequence, the output from which is shown in Table 6.

1 2 3 4 5 4 3 2 1 2 3 4 5 4 3 2 1 2 3 4 5

The call to ts includes several options: -c 5 requests autocorrelations for lags of 1 to 5, the -ps options request

a time-series plot and statistics, and the -w 40 option sets the width of the plot to 40 characters.

SPECIAL PROGRAMS

The remaining programs include general utilities and programs supporting some areas of mathematical psychology.

calc: algebraic modeling calculator

calc does algebraic calculations with operations for ordinary arithmetic, trigonometric functions, and other transcendental functions. Variables can be assigned expressions using other variables and algebra. calc is unusual in that variables retain their definitions so that if x depends on y, and y changes, then so does x. Definitions of variables describing complicated computations can be read from or written to files. The program is designed to be robust against illegal operations, such as using undefined variables or division by zero. When an operation is undefined, it contaminates all expressions and variables based on it, so that they are all undefined. There is a tutorial and manual for this program. The following example shows some of the flexibility of calc in solving quadratic roots (note that conventional operator precedences are used and that checks for negative input to the sqrt function are not necessary because of calc's robustness feature):

```
y      = a*x^2 + b*x + c
radical = sqrt(b*b - 4*a*c)
root1   = (-b - radical) / (2*a)
root2   = (-b + radical) / (2*a)
```

critf: probability to F ratio conversion function

critf prints the critical F ratio required to attain a significance level given the number of degrees of freedom. It saves time looking up values in tables.

```
> critf .05 3 27
F(3,27) = 2.960, p = 0.050
```

dataplot: more flexible scatterplots

dataplot provides more flexible plots than those found in the plotting option of the pair paired data analysis program. dataplot adds horizontal and vertical axis labels, and better control over the absolute width and height of the plot, independent of the range of data. pair always uses the full scatterplot region to plot data; dataplot lets you set the extreme axis values so that plots of different data sets are more easily compared.

dprime: compute d' and beta for signal detection data

dprime computes d' (a measure of discrimination of stimuli) and beta (a measure of response bias) using a method of analysis discussed in Coombs, Dawes, and Tversky (1970). The input to dprime can be a series of lines, each with two paired indicators: the first tells whether a signal

Table 5
regress -p linear log quadratic

Analysis for 20 points of 4 variables:						
Variable	linear	log	quadratic	C		
Min	1.0000	0.0000	1.0000	0.0000		
Max	20.0000	2.9957	400.0000	9.0000		
Sum	210.0000	42.3356	2870.0000	78.0000		
Mean	10.5000	2.1168	143.5000	3.9000		
SD	5.9161	0.8127	127.9023	2.9001		
Correlation Matrix:						
linear	1.0000					
log	0.9313	1.0000				
quadratic	0.9713	0.8280	1.0000			
C	0.1442	0.1961	0.1307	1.0000		
Variable	linear	log	quadratic	C		
Regression Equation for linear:						
linear = 2.98 log + 0.02937 quadratic + -0.0389 C + 0.130368						
Significance test for prediction of linear						
Mult-R	R-Squared	F(3,16)	prob (F)			
0.9976	0.9952	1098.0676	0.0000			
Significance test(s) for predictor(s) of linear						
Predictor	beta	b	Rsq	t(16)	F(1,16)	p
log	0.4093	2.9796	0.6935	13.0372	169.9681	0.0000
quadratic	0.6349	0.0294	0.6867	20.4463	418.0513	0.0000
.C	-0.0191	-0.0389	0.0417	1.0740	1.1535	0.2988

was present and the second tells whether the observer detected a signal. From that, **dprime** computes the hit rate (the proportion of times the observer detected a signal that was present), and the false-alarm rate (the proportion of times the observer reported a signal that was not present). If the hit rate and the false-alarm rate are known, then they can be supplied directly to the program:

```
> dprime .7 .4
hr far dprime beta
0.70 0.40 0.78 0.90
```

pof: F ratio to probability conversion function

pof computes the probability of an *F* ratio, given some degrees of freedom. Probabilities of *F* ratios and *t* values are computed by most of the analysis programs, so using **pof** directly is uncommon.

```
> pof 4.5 2 16
F(2,16) = 4.500, p = 0.028
```

vincent: vincentizing of data from files

vincent allows the comparison of time series of different lengths. Each time series is kept in its own free format file. Each time series is rescaled by a linear interpolation method, attributed to Vincent (1912), into a fixed length so that trends in different time series can be compared. The method enjoyed popularity in mathematical psychology learning research in the 1960s.

EVALUATION OF UNIX|STAT

UNIX|STAT programs allow researchers to perform most of the analyses reported in the psychological research

literature. Although not a complete package, it is practical, especially for instruction in statistics and laboratory courses.

A common comment on the programs is that they are easy to use, especially compared to other statistical packages. This can be explained by the simplicity of the input formats, the ease of option selection, and the integration with other programs. Awkward syntax, the hallmark of many statistical packages, has been minimized or avoided. The input formats to the programs are simple, usually free formats with columns separated by white space for readability. The program options conform to the UNIX system command line option standard (Hemenway & Armistage, 1984), and so are uniform. The programs all read from the standard input and write to the standard output, and can be connected with pipes on UNIX and MSDOS.

It is informative to compare UNIX|STAT to other statistical packages, running on mainframe computers and on personal computers. Rather than mention specific statistical packages, we will discuss features in other packages that are lacking in UNIX|STAT. To understand the limitations of UNIX|STAT, it is necessary to understand how it came into being. The programs are not a comprehensive statistical package, but reflect the types of analyses done by the first author in his research. There are no guarantees of what may be added in the future, but some notes about current development are in this discussion. The MSDOS version of UNIX|STAT is expected to lag several months behind developments on UNIX, because all new development is on UNIX. For example, during the writing of this paper, new programs for computing probabilities and critical statistics for normal and chi-square distributions were added to the UNIX version.

Table 6
ts -c 5 -ps -w 40

n = 21
sum = 63
ss = 223
min = 1
max = 5
range = 4
midpt = 3
mean = 3
sd = 1.30384

Lag	r	r^2	n'	r	df	p
0	0.161	0.026	21	0.504	19	0.487
1	0.678	0.459	20	15.301	18	0.001
2	-0.018	0.000	19	0.006	17	0.940
3	-0.684	0.468	18	14.083	16	0.002
4	-1.000	1.000	17	0.000	15	0.000
5	-0.667	0.444	16	11.200	14	0.005

They probably will not appear in the MSDOS version for several months, mainly due to the awkwardness of the MSDOS program-development environment.

The only multivariate analyses in UNIX|STAT include the regression program and some bivariate plots. There is no factor analysis, cluster analysis, discriminant analysis, and no multidimensional scaling. Nonparametric analyses are lacking, although some rank-order analyses may be added to the paired data analysis (Spearman) and multivariate regression programs (Wilcoxon). The analysis of variance program supports between-subjects designs, but not nested independent factors. Also, there is little support for analysis of covariance, although it can be done by combining **anova** with the **regress** program. There are no cross-tabulation or chi-square programs, but development on these programs is active. Finally, the

graphics in UNIX|STAT are primitive, using only character matrix displays.

UNIX|STAT has many missing programs when compared to large statistical packages. Many of the better-known statistical packages are being ported to personal computers and, in time, the full capabilities of packages like SPSS, BMD, and SAS may be available. This brings up the question, "Why use UNIX|STAT?" The main reason is cost. The programs are inexpensive and can be copied freely, making them especially useful for classroom use. If UNIX|STAT has enough functionality, which is likely for instruction and probably for research, then it makes sense to get the programs. If critical functionality is missing from UNIX|STAT, then getting UNIX|STAT may make for future incompatibilities with the critical programs from another package.

Another important concern is software support. For UNIX|STAT, there is no support for setting up the programs or using them. Almost all users have found the documentation adequate for installation and internal instruction.

Finally, there is the question of accuracy of the programs. The large established statistical packages pay close attention to problems like rounding errors and the correctness of the computations. Less effort is possible with UNIX|STAT, but the programs have been found accurate against book examples and inputs known to be stressful.

AVAILABILITY

The programs were developed on several versions of the UNIX system and have been ported to most UNIX look-alikes running on about 20 different hardware-operating system combinations. The most recent port to MSDOS makes the programs available on such personal computers as IBM, AT&T, Wang, and probably all MSDOS compatible systems. People also use the programs because UNIX|STAT is distributed for the cost of distribution: \$15 covers the cost of 5¼-in. (13.35-cm) floppy diskettes, packaging, and postage anywhere in the world. This price should not change for future releases of the programs that may require more diskettes for new programs. The UNIX ½-in. (1.27-cm) 9-track magnetic tape distribution costs \$20, and includes program source, on-line documentation, and test files. Requests should include return-mailing labels to simplify handling.

The price of the programs is not the only reason the programs are used, because they have enjoyed continued use running alongside most commercial statistical packages running on UNIX. Still, a one-time cost of pocket money compares favorably with several thousand dollars per year for licenses for several copies to run a research or educational laboratory.

UNIX system users interested in program development can request a magnetic tape with the source files for UNIX|STAT from the first author. A UNIX|STAT programming guide describes the program development

conventions to help researchers write their own programs that will integrate well with UNIX, MSDOS, and the UNIX|STAT programs. The source code and the programs may not be copied for commercial advantage, which makes development of commercial products with these tools unlikely.

REFERENCES

- COOMBS, C. H., DAWES, R. M., & TVERSKY, A. (1970). *Mathematical psychology: An elementary introduction*. Englewood Cliffs, NJ: Prentice-Hall.
- GUILFORD, J. P., & FRUCHTER, B. (1978). *Fundamental statistics in psychology and education* (6th ed.). New York: McGraw-Hill.
- HEMENWAY, K., & ARMITAGE, H. (1984). Proposed syntax standard for UNIX system commands. *Proceedings of the 1984 Summer Usenix Conference, Washington, DC*. El Cerrito, CA: Usenix Association.
- KEPPEL, G. (1973). *Design and analysis: A researcher's handbook*. Englewood Cliffs, NJ: Prentice-Hall.
- KERLINGER, F. N., & PEDHAZUR, E. J. (1973). *Multiple regression in behavioral research*. New York: Holt Rinehart Winston.
- KERNIGHAN, B. W., & RITCHIE, D. M. (1979). *The C programming language*. Englewood Cliffs, NJ: Prentice-Hall.
- PERLMAN, G. (1980). Data analysis programs for the UNIX operating system. *Behavior Research Methods & Instrumentation*, 12, 554-558.
- RITCHIE, D. M., & THOMPSON, K. (1974). The UNIX Time-Sharing system. *Communications of the Association for Computing Machinery*, 17, 365-375.
- VINCENT, S. B. (1912). The function of the vibrissae in the behavior of the white rat. *Behavior Monographs*, 1(5).

POSTSCRIPT

(Dated April 27, 1986)

Since this report was written, UNIX|STAT development has continued, and the 5.2 release is being distributed for both UNIX and MSDOS machines. The following changes supersede previous versions.

The **pof** and **critf** programs have been replaced by a more general probability distribution program: **probdist**. **probdist** allows probability to statistic conversion and random number generation for uniform, normal, chi-square, *t*, and *F* distributions.

The function of the **vincent** program has been incorporated into the **ts** program. The more flexible plotting in **dataplot** has been placed in the **pair** program. Thus, both **vincent** and **dataplot** are no longer distributed.

The **oneway** analysis of variance program now allows for unweighted means solutions to unequal cell size designs.

A new contingency table and chi-square program, **contab**, has been added. Its input format is similar to that of the **anova** program.

The online manual from UNIX is now distributed with the MSDOS version. Typeset copies of the manual entries are available for \$5 per set. A printed handbook with tutorial examples, conventions used in UNIX|STAT, and **dm** and **calc** manuals is now available for \$10.