Behavior Research Methods & Instrumentation
1981, Vol. 13 (2), 290-293

An example of cooperating compact data analysis programs

GARY PERLMAN
Cognitive Science Laboratory, University of California at San Diego, La Jolla, California 92093

Some user-oriented compact data analysis programs are described. One program is useful for transforming and reformatting data, and the others perform analysis of variance and multiple regression. Along with other programs not described here, these form an adequate statistical package without sacrificing ease of use or computational power.

In this paper, I will demonstrate some programs I have written for analyzing experimental data. The programs have been designed to be easy to use and compact enough to fit on most small computers. I will first describe the optimum storage format for data. Then I will describe three programs: one for transforming and reformatting data, an analysis of variance program, and a multiple linear regression program. Finally, I will give an example showing how these programs can be used together to do an analysis of covariance.

DATA FORMATS

The idea behind the programs is to keep all the data from a study in a master data file and use a reformatting program (to be described later) to put data in the correct format for input to analysis programs. A master data file consists of a series of lines, each with the same number of alphanumeric fields, generally containing a description of the data collected on one trial of a study. For example, each line might contain a subject identification, a description of a stimulus, and a description of the response. With a series of lines like these, design information can be determined from the relation of the column holding subject identifications to those holding stimulus descriptions.

Consider a hypothetical experiment investigating the utility of indenting computer programs (most teachers of "structured programming" promote this practice to help produce more legible programs). In the experiment, programmers attempt to modify programs that are indented for one group and not indented for another. The number of minutes to modify each of three programs is the dependent measure. Because high programmer variability is expected, a programming ability score is obtained for 12 programmers, to be used later as a covariate. Fictitious data for all programmers are shown in Table 1.

In the first column are codes identifying programmers. Whether a program was indented is indicated in Column 2. Three programs were presented to all programmers: a sorting program, a searching program, and a statistical program. The third column tells which program was presented. The fourth column indicates the number of minutes it took to modify the program described by Columns 2 and 3. The final column contains programming ability scores. The columns are referred to by the mnemonics:

PROGRAMMER INDENT PROGRAM TIME ABILITY

From the format of these data, it can be determined that INDENT (Column 2) is a between-subjects factor, because the indexes in Column 2 are constant for each

Table 1
Programmers' Data

pgmr1	yes	sort	35	19
pgmr1	yes	search	27	19
pgmr1	yes	stat	32	19
pgmr2	yes	sort	41	18
pgmr2	yes	search	32	18
pgmr2	yes	stat	29	18
pgmr3	yes	sort	29	20
pgmr3	yes	search	35	20
pgmr3	yes	stat	38	20
pgmr4	yes	sort	39	19
pgmr4	yes	search	26	19
pgmr4	yes	stat	35	19
pgmr5	yes	sort	29	20
pgmr5	yes	search	34	20
pgmr5	yes	stat	41	20
pgmr6	yes	sort	46	16
pgmr6	yes	search	33	16
pgmr6	yes	stat	24	16
pgmr7	no	sort	47	8
pgmr7	no	search	46	8
pgmr7	no	stat	41	8
pgmr8	no	sort	57	7
pgmr8	no	search	33	7
pgmr8	no	stat	43	7
pgmr9	no	sort	56	4
pgmr9	no	search	45	4
pgmr9	no	stat	40	4
pgmr10	no	sort	48	13
pgmr10	no	search	32	13
pgmr10	no	stat	38	13
pgmr11	no	sort	57	5
pgmr11	no	search	46	5
pgmr11	no	stat	33	5
pgmr12	no	sort	50	4
pgmr12	no	search	45	4
pgmr12	no	stat	47	4

This paper was written while I was supported by a Natural Science and Engineering Research Council of Canada postgraduate scholarship. Requests for reprints of this paper and source listings and documentation of the programs described can be obtained from Gary Perlman, Department of Psychology, C009, University of California at San Diego, La Jolla, California 92093.

  0005-7878/81/020290-04$00.65/0

programmer. For example, the first programmer (pgmr1) modified only indented programs, whereas the seventh programmer modified only nonindented programs. It can also be inferred that the same programs were presented to all programmers, because each programmer has data for all levels of PROGRAM in Column 3. Thus, PROGRAM is a within-subjects factor. Before analyzing these data, the programs necessary for the analysis will be described.

PROGRAM DESCRIPTION

The descriptions of the following programs are cursory, but later examples refer to their use. More detailed descriptions can be found in Perlman (1980) and in the documentation that accompanies the programs.

DM–A Column-Oriented Data Manipulator

DM interprets a series of expressions involving the columns of its input and, for each line of the input, reevaluates and prints the values of the expressions. Usually, DM is used to extract columns from a master data file, but it will be used in a later example to transform data. Numerical values of columns can be accessed with xn, where n is the desired column number. Strings can be accessed analogously with sn. In addition to the uses made in later examples, DM offers a full set of comparison, algebraic, and logical operators, as well as some special variables to control output.

ANOVA–Multivariate Analysis of Variance

The input to ANOVA consists of each datum on a separate line, preceded by a list of alphanumeric indexes, one for each factor, that specifies the level of each factor at which that datum was obtained. By convention, indexes for the one allowable random factor must be in the first column. With a series of lines like this, ANOVA determines design information that people using more conventional programs usually need to specify: the number of factors, the number and names of levels of each factor, and whether a factor is within or between subjects. In addition to the designs analyzed in later examples, ANOVA deals with replications and unequal cell sizes on between-subjects factors, all using the same simple notational scheme.

REGRESS–Multivariate Linear Regression

The input to REGRESS consists of a series of lines, each with the same number of numerical fields. From this input, REGRESS determines the number of variables and the number of points. The variable to be predicted need not be specified because REGRESS prints a regression equation for each variable.

AN EXAMPLE OF COOPERATING PROGRAMS

To analyze the data from the indentation experiment, first the data are analyzed without taking into account the ability covariate.

```
DM      s1            s2         s3        x4
ANOVA PROGRAMMER INDENT PROGRAM TIME
```

DM is used to extract the first four columns (Strings s1-s4) from the master data file, shown in Table 1. The four-column output from DM is in the correct format (the random factor is in the first column and the data are in the last) for input to ANOVA, which gives mnemonic names to the factors.

The output from this analysis, shown in Table 2, includes cell counts, means, standard deviations, the design information ANOVA determined, and an F table with significance tests for each systematic source. This analysis may be taken as evidence that programmers prefer to modify indented programs. The second F test in Table 2 indicates a significant facilitation from program indentation [$F(1,10) = 75.201$, $p = .000$], but the analysis that includes the covariate shows no such trend. With DM and REGRESS, it is a simple matter to find the regression equation predicting modification time with programmer ABILITY.

```
DM      x4   x5
REGRESS TIME ABILITY
```

First, DM extracts the desired columns (Columns 4 and 5) from the master data file. Then REGRESS is called, assigning mnemonic names to the two variables extracted. The output from this analysis, shown in Table 3, includes means and standard deviations for each variable, correlations, and a set of regression equations predicting each variable with every other. The slope (–.9014) and intercept (50.6319) are obtained from

Table 2
Analysis of Variance of Raw Modification Times

```
SOURCE: grand mean
INDEN   PROGR      N        MEAN          SD
                  36     39.1389      8.7543

SOURCE: INDENT
INDEN   PROGR      N        MEAN          SD
yes               18     33.6111      5.8424
no                18     44.6667      7.6773

SOURCE: PROGRAM
INDEN   PROGR      N        MEAN          SD
        sort      12     44.5000     10.0408
        searc     12     36.1667      7.3711
        stat      12     36.7500      6.4403

SOURCE: INDENT PROGRAM
INDEN   PROGR      N        MEAN          SD
yes     sort       6     36.5000      6.8044
yes     searc      6     31.1667      3.7639
yes     stat       6     33.1667      6.1779
no      sort       6     52.5000      4.6797
no      searc      6     41.1667      6.7355
no      stat       6     40.3333      4.7188

FACTOR: PROGRAMMER      INDENT     PROGRAM        TIME
LEVELS:         12           2           3          36
TYPE  :     RANDOM     BETWEEN      WITHIN        DATA

SOURCE                  SS      df              MS          F        p
=========================================================================
mean            55146.6944       1      55146.6944   3769.998    .000 ***
P/I               146.2778      10         14.6278

I                1100.0278       1       1100.0278     75.201    .000 ***
P/I               146.2778      10         14.6278

P                 519.3889       2        259.6944      6.537    .007 **
PP/I              794.5556      20         39.7278

IP                122.0556       2         61.0278      1.536    .239
PP/I              794.5556      20         39.7278
```

the column labeled "TIME" in Table 3. To remove any linear effects on time attributable to ability, the TIME data are transformed, using DM to subtract the ability covariate weighted by the slope obtained from REGRESS. With the effects attributable to group differences in ABILITY factored out, ANOVA is called on the four-column output from DM, once again assigning mnemonic names for the factors.

```
DM    s1        s2     s3     (x4+.9*x5-50.6)

ANOVA PROGRAMMER INDENT PROGRAM TIME'
```

From this analysis, shown in Table 4, the significant differences in modification time shown in Table 2 can be attributed to group differences in ABILITY.[1] An analysis of variance comparing the two groups on programming ability shows that the group modifying indented programs had much higher ability scores than the nonindented group.

```
DM    s1        s2     x5
ANOVA PROGRAMMER INDENT ABILITY
```

The three columns for this analysis are extracted from the data in Table 1 by DM. ANOVA is called to analyze these data, and the results of the analysis can be seen in Table 5.

GENERAL CHARACTERISTICS OF THE PROGRAMS

The programs described are written in C (Kernighan & Richie, 1978), the systems programming language of the UNIX[2] operating system (Richie & Thompson, 1974). The programs have been designed with one overriding philosophy: to simplify the task of their users as much as possible, without sacrificing computational power.

Table 3
Regression of TIME and ABILITY

```
Analysis for 36 points of 2 variables:
VARIABLE  :       TIME    ABILITY
MEAN      :    39.1389    12.7500
SD        :     8.7543     6.4824
CORRELATION MATRIX:
TIME      :     1.0000
ABILITY   :     -.6675     1.0000
VARIABLE  :       TIME    ABILITY
REGRESSION EQUATIONS:
SLOPES    :
TIME      :                -.4943
ABILITY   :     -.9014
INTERCEPT :    50.6319    32.0947
R-Squares :      .4455      .4455
F(1,34)   :    27.3197    27.3197
prob (F)  :      .0000      .0000
```

Note—Read the regression equation for a variable in the column under the predicted variable's name. In this analysis, TIME = 50.6319 – .9014 ABILITY.

Table 4
Analysis of Variance of Transformed Modification Times

```
SOURCE: grand mean
INDEN  PROGR     N      MEAN        SD
                36    -.0002    6.5187

SOURCE: INDENT
INDEN  PROGR     N      MEAN        SD
yes             18    -.1947    5.9503
no              18     .1943    7.2111

SOURCE: PROGRAM
INDEN  PROGR     N      MEAN        SD
       sort     12    5.3609    5.3648
       searc    12   -2.9724    4.3409
       stat     12   -2.3891    6.3532

SOURCE: INDENT PROGRAM
INDEN  PROGR     N      MEAN        SD
yes    sort      6    2.6942    5.5529
yes    searc     6   -2.6391    4.0160
yes    stat      6    -.6391    7.4829
no     sort      6    8.0277    3.9263
no     searc     6   -3.3057    5.0061
no     stat      6   -4.1390    5.0455

FACTOR: PROGRAMMER    INDENT    PROGRAM      TIME'
LEVELS:         12         2          3         36
TYPE  :     RANDOM   BETWEEN     WITHIN       DATA

SOURCE               SS     df              MS         F      p
===============================================================
mean              .0000      1           .0000      .000   .995
P/I             49.8983     10          4.9898

I                1.3618      1          1.3618      .273   .617
P/I             49.8983     10          4.9898

P              519.3889      2        259.6944     6.537   .007 **
PP/I           794.5556     20         39.7278

IP             122.0556      2         61.0278     1.536   .239
PP/I           794.5556     20         39.7278
```

Table 5
Analysis of Variance Comparing Groups for Ability

```
SOURCE: grand mean
INDEN    N      MEAN        SD
        12   12.7500    6.6759

SOURCE: INDENT
INDEN    N      MEAN        SD
yes      6   18.6667    1.5055
no       6    6.8333    3.4303

FACTOR: PROGRAMMER    INDENT    ABILITY
LEVELS:         12         2         12
TYPE  :     RANDOM   BETWEEN       DATA

SOURCE               SS     df              MS         F      p
===============================================================
mean          1950.7500      1       1950.7500   278.017   .000 ***
P/I             70.1667     10          7.0167

I              420.0833      1        420.0833    59.869   .000 ***
P/I             70.1667     10          7.0167
```

The programs have been written in a well commented, highly modularized style, in a structured programming language. Much of the software has been translated without much trouble to PASCAL (Jensen & Wirth, 1974), and translation to most structured programming languages is straightforward. Efficiency has sometimes been sacrificed so that the programs can more easily be modified and verified. Still, the programs usually have run times of only a few seconds, and the complete analysis presented here takes less than 1 min.

The programs use algorithms conducive to easy verification. DM uses an automatic parser generator (Johnson & Lesk, 1978), ANOVA uses a method of analysis based on Keppel (1973), and REGRESS uses a method based on Kerlinger and Pedhazur (1973). ANOVA and REGRESS have been tested against most of the examples in these sources and against outputs from BMD-P2V (Dixon, 1975).

REFERENCES

DIXON, W. J. *BMD-P biomedical computer programs.* Berkeley, Calif: University of California Press, 1975.

JENSEN, K., & WIRTH, N. *Pascal user manual and report.* New York: Springer-Verlag, 1974.

JOHNSON, S. C., & LESK, M. E. Language development tools. *Bell System Technical Journal,* 1978, 57, 2155-2175.

KEPPEL, G. *Design and analysis: A researcher's handbook.* Englewood Cliffs, N.J: Prentice-Hall, 1973.

KERLINGER, F. N., & PEDHAZUR, E. J. *Multiple regression in behavioral research.* New York: Holt, Rinehart, & Winston, 1973.

KERNIGHAN, B. W., & RICHIE, D. M. *The C programming language.* Englewood Cliffs, N.J: Prentice-Hall, 1978.

PERLMAN, G. Data analysis programs for the UNIX operating system. *Behavior Research Methods & Instrumentation,* 1980, 12, 554-558.

RICHIE, D. M., & THOMPSON, K. The UNIX time-sharing system. *Communications of the ACM,* 1974, 17, 365-375.

NOTES

1. This is technically not the correct analysis because a degree of freedom has not been removed for the regression, but the pattern of results is the same regardless.
2. UNIX is a trademark of Bell Laboratories.