A prototype system for interactive data analysis

by GERALD LEVITT, DAVID H. STEWART and BEATRICE YORMARK

The Rand Corporation
Santa Monica, California

INTRODUCTION

The analysis of small and simple data collections is commonly accomplished through the application of “canned” statistical analysis programs. For larger more complex data collections, however, such programs often do not satisfy a researcher’s needs. In these cases, the additional use of specially developed computer programs may be necessary. These programs frequently require modification and reformatting of data to meet their input requirements. These additional complexities are compounded when a researcher attempts to investigate alternative hypotheses or pursue hunches requiring further transformations or restructuring of the original data collection. Often, this process involves the services of a professional programmer making repeated program modifications and computer runs.

To study these difficulties, a prototype computer system called the Data Analysis System (DAS) was developed which aids researchers in accessing their data and assisting them in interactively applying a variety of standard analytic procedures in a unified and consistent manner. Through a comprehensive graphical terminal, researchers are able to: review data in tabular or graphical form, subset and restructure the data for hypothesis testing and formulation, and apply many standard statistical tests.

The DAS, although similar to other systems addressing a data analysis capability, differs from those systems in one or two basic respects: (1) the availability of a natural definitional language to manage and restructure data collections; and (2) the ability to easily and quickly form graphical presentations of data both in their collected and restructured forms.

This paper presents both the basic design notions used to develop the system and a functional description of the facilities it provides.

DESIGN CONCEPTS

Data analysis concepts

The Data Analysis System (DAS) design goals have drawn heavily upon Tukey’s characterization of the data analysis process. Five of these characteristics important in the design of the DAS are discussed below.

1. Summarization

Summarization is viewed as the process of formal statistical description. It consists of using statistical models to test for hypothesized relationships. Applications of summarization techniques include the ability to use the residuals of summarization processes as data. An example of this in the DAS is the ability to display the results of a linear regression and a plot of its residual against explanatory variables.

2. Exposure

Tukey defines Exposure as the “... effective laying open of data to explore the unanticipated”. This can be accomplished by using standard statistical methods on the data in a flexible way to elucidate new hypotheses and reveal possibly unknown relationships. It is the ingredient that has been missing in computer-assisted data analysis because it requires a level of informality in the use of techniques not normally considered within the domain of formal statistics.

3. Iterative Nature

The data analysis process is characterized by the repeated restructuring of data collections and multiple applications of summarization and exposure techniques. This process is intrinsically iterative—no step is clearly the last before it is taken. Human judgment is needed at almost every step. The analyst must be allowed to flexibly choose a model for summarization and apply it to any set of data. An important goal of DAS is to facilitate the interplay between exposure and summarization.

4. Scaling of Data

One of the goals of data analysis is the search for simplicity in the description or explanations of relationships between variables.

To permit the use of simpler analysis models there must be a facility for easy transformation of variables.

5. Missing Data

Often data used in the analysis process suffer from missing observations. Dealing with missing values, particularly in
multivariate forms of analysis, can be a problem. The patterns of missing observations often vary from variable to variable. Data in this form can result in a loss of predictive strength due to the loss of numerous observations.

An important aspect of the DAS approach is to provide a means of easily identifying, manipulating and managing missing observations.

**Information language**

One of the important goals of the DAS effort was to produce a system that could be used naturally by an analyst. In order to represent the system to these users and to allow them to use it in a natural way, we formalized an information language for data analysis. This information language presents terms and constructs for dealing with the data and analysis process. Several of the important aspects of this information language are discussed below.

1. **World View**

The world view captured by the information language of the DAS sees data bases and operations on them as functions of the analysis process rather than as an information retrieval system providing file retrieval. The DAS language therefore allows the user to define, label and indicate how data groups are to be used in the analysis process without requiring the user to either understand or deal with the underlying files and data manipulation mechanisms.

2. **Nature of the Data**

A great deal of the data for analysis is collected and stored in discretely identifiable units called cases. Cases may be in part an artifact of the data collection process or in part a predetermined structure specified by the analyst.

3. **Attributes**

The data of a case can be conceptually divided into different categories called *attribute classes*. The term *variable* is often used to connote the same property as an attribute. An attribute class is defined by its name and associated *set of values* which may be: empty; consist of only one datum; or consist of a large class of related data. For example, in hospital data the attribute "patient name" would be associated with one datum, the attribute "patient temperatures" with series of data and the attribute "patient age" null if that datum was unobtainable. A particular case, in turn, is completely defined by the enumeration of its contribution to attribute class values.

4. **Sets**

Cases that are described by the same attribute classes may be collected to form a group called a *set*. Each case in a set is assigned a unique identifier used thereafter to refer to the case or a particular attribute value of a case. Conceptually, a set may be thought of as a matrix whose rows are labeled by case identifiers, and columns labeled by attribute names. Sets play a very important role in the process of data analysis. Many of the activities of data analysis specify, construct and evaluate sets.

**User interface**

Of primary concern in the design and implementation of the prototype system has been the interface between the user and the system. It is of utmost importance that the medium of communication be convenient and appropriate to the data analysis context, resulting in users feeling as if they were dealing directly with their problems.

Because the analysis environment requires facilities as broad in scope as data management, statistical analysis and graphing, the tools the analyst must use to carry out these functions can become exceedingly complex and require many steps or procedures to accomplish a single task. It is with a respect for the power of graphic presentation and its relationship to other analytic tasks that we have placed a primary emphasis on user control of the system through a graphic medium.

The methodology employed in creating the interface utilizes a display tube with a sensitive surface at which the user can point to invoke system responses. In addition to graphs and diagrams, the objects displayed on the screen include menus of functional options of the system, tutorials, and requests by the system for action from the user.

A general theme throughout this communication is that the user should be informed of what is expected of him, whether or not he has control, and, what the system is doing. Also incorporated in the user interface is an extensive subsystem of tutorials used to further clarify the user-system status. Used in another mode, tutorials are presented to explain the state of the system and the standard conventions of the hardware and data analysis.

The interface assists the user in selecting and using the functions of the system. In turn, it elicits the specific functions requested and re-receives control of the system when a particular function has terminated its activities. Since functional modules can elicit modules subordinate to them, another task of the physical user interface is context management. By context management is meant performing the housekeeping to determine how a function was entered and, as a result, determine the alternatives and actions necessary to return to a previous context, go to a new context, or return to the initial context. An important task of the user interface is to make this complexity transparent to the user.

**FUNCTIONAL CHARACTERISTICS**

**The initial display context**

The DAS provides its users with two basic classes of capabilities. The first class consists of operations which are executed on sets; they include the creation of sets, the deletion of sets and the display of set data and associated...
data summary statistics (e.g., max., min., avg.). The second class consists of a variety of statistical methods which can be applied to the data contained in sets; these include the histogram, scattergram, plot, barchart, cross-tabulations (2-way) and stepwise linear regression.

These capabilities are presented to the user for their selection as options in our initial display context. The options are organized in two display menus as pictured in Figure 1. A third display menu provides the user with an added set of options which when selected will generate tutorial information about the system itself, its operation and each of its set manipulation and analysis capabilities.

Normally, a user invokes the desired capability by touching the appropriate option name in one of these menus; when this happens the display context immediately becomes that of the option selected. If the user first selects OPTION in the tutorial menu, however, the context does not change. Instead, the user may then point to an item in any one of the other menus to display tutorial information about it. The user may easily switch from the invoke mode to tutorial mode and back again by touching the appropriate sensitive areas.

This initial display context is presented when the user first logs on to the DAS and when he returns to it from some other context.

Selection of sets and attributes

All system analysis capabilities require the selection of a set and one or more of its attributes before that capability can be executed. The process involves two menus. The first is a menu of set names. This menu is displayed automatically whenever it is needed. On selecting a set from this menu (using the pointing device), a second menu containing the attributes of that set is then also displayed.

Loading the data analysis data base

Data are entered into the DAS through a batch component called the Data Base Loader. The main function of this component is to transform the input data collection into an output data file organized to facilitate interactive data manipulation and display in the on-line mode. This latter data file is referred to as the Data Analysis Data Base (DADB).

In addition to forming the DADB, the loader also constructs a dictionary defining set and attribute names, identifies and transforms missing values to a unique form, provides a facility for selecting random subsamples of cases, and stores a description of the data collection for eventual retrieval and display in the interactive mode. This added information is provided to the Loader in the form of a data definition directory.

Creation and description of sets

The set formation facility in the DAS provides a mechanism for naming and preserving data relationships in the system. As the analysts use of the data becomes more and more qualified, he requires a mechanism to express these qualifications and name them. The set formation facility also serves the purpose of allowing the analyst to delete data items and cases from the body of data he is working with to provide a simpler analysis or to concentrate on specific variables. Set formation provides a method for recombining data sets and attributes to produce aggregate sets.
Command Expression

<table>
<thead>
<tr>
<th>Command Expression</th>
<th>Set Specification Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE SET</td>
<td>A FROM S1, S2, S3, ..., Sn</td>
</tr>
</tbody>
</table>

WITH Attributes:

- A1 = f (s1, s2, s3, ..., sn)
- A2 = f (s1, s2, s3, ..., sn)
- ...
- AN = f (s1, s2, s3, ..., sn)

WHERE:

- s1 = s2 and s3 = s4

Figure 3—Set formation language

1. Set Formation Process

Set formation is achieved by the use of a language. This language, in turn, drives the data management system. The set formation command, although having a superficial appearance of a programming language, is a definitional command. It describes the characteristics of the resulting set.

In the present implementation of the Data Analysis System the user types into the terminal the command form to be executed. The system saves the command form for every set created and permits easy retrieval for editing and re-execution as well as set description. In addition, during the set creation process a set of basic summary statistics are computed for every attribute in the set created. These are also retrievable for review.

2. Syntax and Semantics of Set Creation Language

Every statement in the command language consists of four expressions:

- Command expression
- Set specification expression
- Attribute specification expression
- Membership specification expression

The manner in which these expressions fit together in a statement is represented in the paradigm in Figure 3.

Command Expressions

The command expression of the statement specifies that a set is to be created. This includes a variety of forms, i.e., FORM SET, FORM, CREATE, CREATE SET. The number of sets and their function in the statement is determined by the command scope.

Set Specification Expressions

Figure 3 is an illustration of the form of the set formation command. In the example, the “A” and “S1, S2, S3, ..., Sn” are sets in the set specification expressions. “A” is a character string chosen by the user to be used as a name for the set to be created. This name will be added in the dictionary of set names. “S1, S2, S3, ..., Sn” is a series of character strings representing names of sets already created and maintained by the system.

Attribute Specification Expressions

Attribute specification is accomplished by functional expressions involving attributes of the sets specified. The attribute specification expression of the command, that is, “A1 = f(S1, S2, S3, ..., Sn)” in Figure 3 defines a character string “A1” which is to be the name of an attribute of the set being created. It also specifies, from the source sets, which of their attributes are to be used in forming the object attribute.

The source set attributes to be used can be manipulated before being included in the new set. The mathematical functions allowed for this operation are the standard set available with FORTRAN compilers. In addition to these mathematical functions Boolean subexpressions are permitted to define alternative functional forms for an attribute. In this manner alternative functional forms can be used based upon the value set for specified attributes in each case.

Membership Specification Expressions

The membership specification in the command applies a restriction on membership. This specification limits membership by filing, in the object set, only those cases whose attribute values satisfy the Boolean expressions of membership.

Membership specification is accomplished by a Boolean expression of attribute values of the members of the sets specified. This expression follows a “WHERE” clause and results in the command being executed only on those cases of the specified sets where the conditions of the logical expression are found to be true.

3. Formation of Subsets

One of the uses of the set formation language is the creation of subsets. This is accomplished by the use of the membership specification expression. Cases for the subset are chosen on the basis of their attributes values satisfying the Boolean expression. Those cases whose values do not satisfy the expression are not included in the subset.

4. Formation of Sets by Union and Intersection

In addition to creating proper subsets the set formation language can be used to define sets that are formed from more than one parent set. This type of formation can be performed by either the union or intersection operation. When each case (or observation) of data is included in the master set (i.e., the DADB) at loading time a unique identifier is assigned. This identifier is perpetuated throughout all sets in which an observation may be defined as a member.

When the set specification expression contains more than one set name for the source set, a union or intersection
specification is included. These processes are performed on the identifier attributes in the source sets and are used to reassemble data groupings in a building block fashion or to locate those observations contained in sets of data representing different characteristics.

5. Set Summary Display

As discussed earlier, when a set is created a group of summary statistics are computed for each attribute. The set summary display option permits the user to retrieve these statistics as shown in Figure 4.

By changing set names the user can selectively recall related attributes from other sets and easily compare the statistics. Figure 4 shows a case where the user has called up 2 attributes (CPUSCNDS, ELAPTIME) for 2 different sets (SAWTELLE, FINNEGAN). In this case FINNEGAN is a proper subset of SAWTELLE and the set summary facility is being used to compare the two sets.

Analysis of sets

The Data Analysis System allows the user flexibility in analyzing and viewing his data and its interrelationships by providing the following analytical packages:

- Two variable plots with curve fitting
- histograms
- barcharts
- cross tabulations
- stepwise regression

The following is a review of the above tools and their associated features.

Two Variable Plots

To obtain a two variable plot the user requests the plot option and chooses the two attributes for the respective axes of the graph (as shown in Figure 1, 2). The system automatically computes the grid using the minimum and maximum values of the chosen attributes. When the computed scattergram is displayed, the user receives, in addition to the graph, a set of statistics for each attribute. The statistics displayed include: minimum data value, maximum data value, sample size, mean, standard deviation, variance.

With the plot displayed, the user may rescale the grid to focus attention on different sets of ranges. This option is used, for example, when there are outliers or clustered data values.

Also, after the plot appears on the screen the user may choose to fit the data with a line. By choosing this option, the system computes the best fit of the data in the form:

\[ y = ax + b \]

The computed line is displayed along with values for \( a \) and \( b \), and the statistics indicating "goodness of fit" (i.e., correlation coefficient, \( r^2 \)).

Histogram

To compute and display the frequency distribution for an attribute having numeric values, the user indicates this to the system by choosing the histogram option and the attribute for which the histogram is to be drawn. The histogram is computed by using the maximum and minimum attribute values and the sample size. The computed histogram is displayed along with relevant statistics (see Figure 5).

When the frequency distribution has been displayed, the user has the option of rescaling the grid (if, for example, he detects an outlier) and redisplaying the histogram using these new values. As in the plot option, the user may recompute and display the histogram as often as necessary to achieve desired results.
Barchart

The barchart option allows the user to obtain a frequency distribution display for an attribute having discrete numeric or alphanumeric values. Along with the displayed barchart, the values for relative and absolute frequency are displayed similarly to the histogram. If the data is numeric, the following statistics are also displayed: minimum value, maximum value, mean, number of observation, variance, standard deviation.

Crosstabs

The crosstab option of the DAS gives the user the ability to display a 2-way crosstab. After indicating to the system that he wants the crosstab display, he chooses the two attributes which are to be cross tabulated. At this point the user has two options. He can let the system automatically generate the crosstab by computing the row and column values from the attribute data values or he can choose the values for the rows and columns explicitly.

Along with displaying the absolute frequency count for each cell, the user can indicate that he wishes the system to compute and display one of the following relative frequency percentages:

- percent of row total
- percent of column total
- percent of total sample size

The crosstab is displayed along with row and column totals, and relevant statistics for each attribute.

After the crosstab has been displayed, the user has the option of changing the row and/or column values, choosing a different relative frequency count to be displayed and redisplaying the crosstab with these new values. As in the other packages, new crosstabs may be computed and displayed as often as the user desires.

Stepwise Regression

The stepwise regression option is entered when the user indicates to the context manager to perform a regression analysis. Upon entering the package, he chooses a dependent variable and up to 29 independent variables to be included in the analysis. After the attributes have been chosen, the user has the option of setting his own tolerance limits for the regression (i.e., F-to-enter, F-to-remove, tolerance level & maximum number of steps), or of letting the system use its default values. The regression computation may proceed in two different ways: (1) The user can direct the system to perform the regression automatically letting the system choose the variables to be entered or removed at each step until the analysis is complete (i.e., one of the tolerance limits has been reached) or; (2) He can explicitly choose the variables to be entered or removed at each step, thus allowing him to have control over the variables to be entered or removed regardless of the tolerance levels.

This option is usually used when a definite model is being pursued.

At each step of the regression, or for the last step if the first mode is chosen, the following information is displayed:

- step number
- variable entered/removed
- standard error of the estimate
- multiple r
- analysis of variance table
- list of variables in the regression equation
- list of variables not in regression equation

When the regression is completed, a summary table is produced which summarizes the results of the regression at each step.

Also, at any point during the regression analysis, the user can view the following statistics are displayed for each variable in the regression:

mean, standard deviation, correlation with other variables, covariances with other variables

After the summary table has been produced, the user can display a tabular listing of the residuals and plot the residuals against any of the variables in the regression.

If the user discovers either at the end of the regression or during the regression that he has not chosen the correct variables, or that the tolerance limits should be changed, he can stop the regression, change the relevant information and restart the analysis. This can be done as often as is necessary to test the hypothesis.

IMPLEMENTATION

Reliance upon existing software

It was our intent, from the outset of the Data Analysis project, to capitalize upon as much existing software as possible. We felt that by doing this we could direct our efforts to those portions of the system which were unique and had not been addressed before.

The following are several highlights of this approach:

- Implementation Language

The major part of the Data Analysis System was written in standard FORTRAN IV. The decision was made to use FORTRAN since most computer systems support it and thus provided options for portability.

- DAS Language Processor

The command language developed for the Data Analysis System is translated using APAREL* (A Parse Request Language). APAREL is used as a series of commands in the standard PL/1 language. APAREL was chosen since it not only freed us from developing a
language parser of our own, but also avoided the development of a unique translator for the DAS language.

- **Graphics Software**

All of the graphics manipulation and display was accomplished using the Integrated Graphics System (IGS) developed at The Rand Corporation. IGS is a series of graphic routines callable from languages with standard OS/360 linkage.

- **Analysis Packages**

Whenever possible, the analysis packages provided by the system were taken from standardized, widely used packages and adapted to the interactive environment of the Data Analysis System. For example, this was done with the stepwise regression facility. For this we used the stepwise regression package of the BIOMED Library (BMD02R).

**Hardware and systems**

The prototype system has been implemented at The Rand Corporation on an IBM 360 model 65 computer under OS/MVT. The total amount of storage needed to run the system is 228K.

The hardware used for the graphical displays in the prototype system is the Rand Videographics System (VGS) and a pointing device. The VGS consists of an interactive graphics console comprised of a cathode ray tube (CRT) and a keyboard. The pointing device can be either a data tablet or a light pen.

**EXPERIENCE**

**Variety of users and responses**

The Data Analysis system has been exposed to a variety of data bases, researchers, and research methodologies. These applications have included: the analysis of medical research data, studies in computer performance analysis, studies in the production of software and use in econometric and management sciences.

The response of these users to the system has been diverse which, in part, can be explained by the diversity of the data analysis process itself, i.e. there does not seem to be a single or predominant approach to analyzing a body of data. Significant variance in response seems also linked to both the analysts expectations from the data and the pre-conditioning of previous analysis experiences.

Two possible appeals of system

Some found the system most useful for data description and general "getting acquainted" with the data before applying in depth analysis techniques not supported by the system. For these users, the histogram, barcharting and plotting capabilities were used extensively to gather insights before applying more robust tests to the data.

Others found the system more useful for searching for unanticipated relationships in the data. These users found the system useful in forming hypotheses about the data and the phenomena under investigation. In some cases they felt that the data description capabilities of the system were better served by a batch processing system. These users said they generally began their analysis by performing data description on all variables routinely and were not disposed to performing this activity in an interactive mode. They felt this process was too routine for them to be interacting directly with the computer.

**ACKNOWLEDGMENT**

The authors would like to extend special thanks to Tom Wisniewski of RAND who was an integral member of the team that designed and implemented the Data Analysis System. Thanks are also due Carol Johnson of RAND who programmed the cross tabulation module of the system.

**REFERENCES**
