VAX DATATRIEVE

Guide to Using Graphics

Order No. AA-L631D-TE

Guide to Using Graphics

digital™
software

VAX DATATRIEVE
Guide to Using Graphics

Order No. AA-L631D-TE

November 1987

This manual introduces VAX DATATRIEVE graphics and contains the reference material for creating plots.

OPERATING SYSTEM:	VMS MicroVMS
SOFTWARE VERSION:	VAX DATATRIEVE V4.1

digital equipment corporation, maynard, massachusetts

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The postage-paid Reader's Comments forms at the end of this document request your critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

ACMS
CDD
DATATRIEVE
DEC
DECnet
DECUS
MicroVAX
MicroVMS
PDP
RALLY
Rdb/ELN
Rdb/VMS
ReGIS
TDMS
TEAMDATA
UNIBUS
VAX
VAXcluster
VAXinfo
VAX Information Architecture
VAX/VMS
VIDA
VMS
VT

digital™

IBM® is a registered trademark of International Business Machines Corporation.

Contents

4 DATATRIEVE Plot Statements

Tables

How to Use This Manual

This manual explains how to use VAX DATATRIEVE to represent data in graphic format.

Intended Audience

This manual is intended for all VAX DATATRIEVE users.

To use this manual, it is helpful to have a basic understanding of DATATRIEVE concepts.

Operating System Information

Information about the versions of the operating system and related software that are compatible with this version of VAX DATATRIEVE is included in the VAX DATATRIEVE media kit, in either the Installation Guide or the Before You Install letter.

Contact your DIGITAL representative if you have questions about the compatibility of other software products with this version of VAX DATATRIEVE. You can request the most recent copy of the *VAX System Software Order Table/Optional Software Cross Reference Table,* SPD 28.98.xx, which will verify which versions of your operating system are compatible with this version of VAX DATATRIEVE.

Structure

This manual contains five chapters, one appendix and an index:

Chapter 1	Tells you how to get started using DATATRIVE plots. Discusses the required and optional equipment you need and how to set CDD defaults for plots. Covers similarities between the PRINT and plot statements.

Chapter 2 Outlines the five basic types of DATATRIEVE plots: bar charts, scattergraphs, line graphs, pie charts, and utility plots.

Chapter 3 Explains how to determine the purpose of your plot and use that information to design and improve your plot.

Chapter 4 Presents, in alphabetical order, full descriptions of the plot statements. The chapter describes the function, syntax, arguments, restrictions, results, and use of each statement, and provides examples and results of typical operations.

Chapter 5 Demonstrates the use of DATATRIEVE plots with other database products.

Appendix A Contains record definitions used in this manual.

Related Documents

For further information on the topics covered in this manual, you can refer to:

- *VAX DATATRIEVE Release Notes*

  Includes specific information about the current DATATRIEVE release and contains material added too late for publication in the other DATATRIEVE documentation.

- *VAX DATATRIEVE Installation Guide*

  Describes the installation procedure for VAX DATATRIEVE. The manual also explains how to run User Environment Test Packages (UETPs), which test DATATRIEVE product interfaces, such as the interface between DATATRIEVE and Rdb/VMS.

- *VAX DATATRIEVE Handbook*

  Describes how to create DATATRIEVE applications. The manual includes some tutorial information on describing data and creating procedures.

- *VAX DATATRIEVE Guide to Writing Reports*

  Explains how to use the DATATRIEVE Report Writer.

- *VAX DATATRIEVE User's Guide*

  Describes how to use DATATRIEVE interactively. The manual includes information on using DATATRIEVE to manipulate data and on using DATATRIEVE with forms, relational databases, and database management systems. It also describes how to improve performance and how to work with remote data.

- *VAX DATATRIEVE Reference Manual*

  Contains reference information for DATATRIEVE.

- *VAX DATATRIEVE Pocket Guide*

  Contains quick-reference information about using DATATRIEVE.

- *VAX DATATRIEVE Guide to Programming and Customizing*

  Explains how to use the DATATRIEVE Call Interface. The manual also describes how to create user-defined keywords and user-defined functions to customize DATATRIEVE and how to customize DATATRIEVE help and message texts.

Conventions

This section explains the conventions for the syntax and symbols used in this manual:

WORD	An uppercase word in a syntax format is a keyword. You must include it in the statement if the clause is used.
word	A lowercase word in a syntax format indicates a syntax element that you supply.
[ ]	Square brackets enclose optional clauses from which you can choose one or none.
{ }	Braces enclose clauses from which you must choose one alternative.
<RET>	This symbol indicates the RETURN key. Unless otherwise indicated, end all user input lines in examples by pressing the RETURN key.
<TAB>	This symbol indicates the TAB key.
<CTRL/x>	This symbol tells you to press the CTRL (control) key and hold it down while pressing a letter key. If you press CTRL/Z, the word Exit appears in reverse video; if you press CTRL/Y, the word Interrupt appears in reverse video. Examples of video output in this book do not include either word; instead the conventions ^Z and ^Y are used.
<GOLD-x>	This symbol indicates that you press the GOLD key and then a specified letter key consecutively.
" "	These are double quotation marks.

' ' These are single quotation marks.

... A horizontal ellipsis in syntax formats means you can repeat the previous item.

A horizontal ellipsis in examples means that information not directly related to the example has been omitted.

.
.
. A vertical ellipsis in syntax formats means you can repeat the syntax element from the preceding line.

A vertical ellipsis in examples means that information not directly related to the example has been omitted.

Color Color in examples shows user input.

Since CDD Version 3.1, CDD path names include a leading underscore. For example:

```
DTR> SHOW DICTIONARY
The default dictionary is _CDD$TOP.DTR32.WEAGER
```

Examples of the output in DATATRIEVE manuals do not reflect this change. You do not need to enter CDD path names with the leading underscore.

References to Products

VAX DATATRIEVE is a member of the VAX Information Architecture, a group of products that work with each other and with VAX languages conforming to the VAX calling standard to provide flexible solutions for information management problems.

VAX Information Architecture documentation explaining how these products interrelate is included with the VAX CDD documentation. VAX Information Architecture documentation is also available separately. Contact your DIGITAL representative.

The VAX DATATRIEVE documentation to which this manual belongs often refers to products that are part of the VAX Information Architecture by their abbreviated names:

- VAX CDD software is referred to as CDD.
- VAX DATATRIEVE software is referred to as DATATRIEVE.
- VAX DBMS software is referred to as DBMS.
- VAX Rdb/ELN software is referred to as Rdb/ELN.
- VAX Rdb/VMS software is referred to as Rdb/VMS.
- VAX TDMS software is referred to as TDMS.
- VIDA software is referred to as VIDA.

This manual uses the terms relational database or relational source to refer to all three of these products:

- VAX Rdb/ELN
- VAX Rdb/VMS
- VIDA

Technical Changes and New Features

This section describes the new features for VAX DATATRIEVE documented in this manual.

Version 4.1

This documentation includes the following new plot statements:

- PLOT LIMITS_X
- PLOT LIMITS_Y
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT TITLE

These plot statements were new with Version 4.0. They were documented in the Version 4.0 Release Notes and online help. With Version 4.1, they are now included in Chapter 4 of this manual. Refer to Chapter 4 for further information.

New features since Version 3.0 are described in online help. To read a description of these new features from within DATATRIEVE, refer to the online help for Version 4.0:

```
DTR> HELP New_Features
```

Getting Started with DATATRIEVE Plots 1

This chapter introduces using DATATRIEVE graphics to present data in a visually informative format. (DATATRIEVE graphics are also referred to as DATATRIEVE plots.) The chapter describes:

- Prerequisite and optional equipment
- Steps to take before using DATATRIEVE plots
- Changing from the PRINT statement to the plot statement

The examples in this chapter use DATATRIEVE record selection expressions (RSEs) and FIND and PRINT statements. See the *VAX DATATRIEVE Reference Manual* for more information about the FIND and PRINT statements. See the *VAX DATATRIEVE User's Guide* for more information about RSEs.

1.1 Required and Optional Equipment

The following sections discuss the required and optional equipment to produce DATATRIEVE plots:

- Video terminals (required equipment)
- Hardcopy output devices (optional equipment)

This equipment must support the Remote Graphics Instruction Set (ReGIS).

1.1.1 Video Terminals

To use DATATRIEVE plots, you must have a DIGITAL terminal that supports ReGIS graphics. Check your owner's manual to determine if your terminal supports ReGIS graphics.

Terminals are available in two basic types: those with a monochrome monitor and those with a color monitor.

If you have a monochrome monitor, you should set it so it displays light characters on a dark background screen. All plots will then be completely visible.

You can also use a color monitor for a color representation of the plots. The owner's manual for your color monitor tells you how to connect the terminal.

After you have connected the monitor and performed any other setup operations necessary to activate the monitor, you must perform several steps. First, log in to your VMS system, invoke VAX DATATRIEVE, and enter the following DATATRIEVE command:

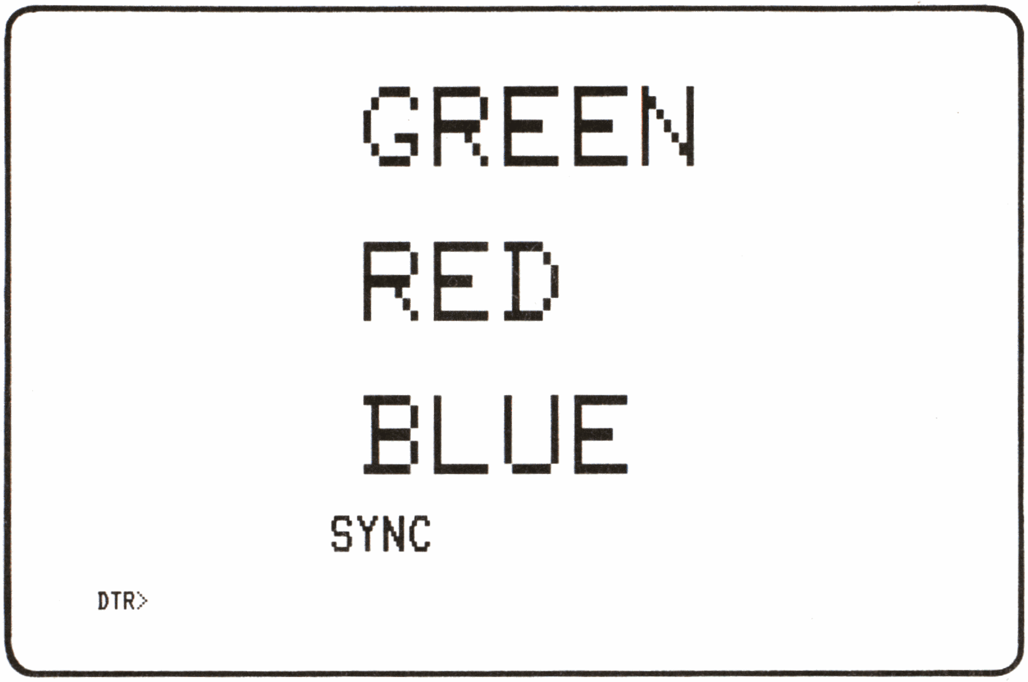
```
DTR> SET PLOTS CDD$TOP.DTR$LIB.PLOTS
```

The SET PLOTS command points to the VAX Common Data Dictionary (CDD) directory that contains the plot definitions. DATATRIEVE uses these plot definitions to produce plots, graphs, and charts from your data. You must issue the SET PLOTS command before using any DATATRIEVE plot statements.

Next, enter the PLOT MONITOR statement:

```
DTR> PLOT MONITOR
```

The PLOT MONITOR statement displays the word GREEN in green, RED in red, and BLUE in blue. The word SYNC appears on the screen to indicate that DATATRIEVE has reset the terminal to its default color settings after the red, green, and blue have been displayed. The following figure shows these words on the color monitor screen:



```
GREEN

RED

BLUE

SYNC

DTR>
```

ZK-00282-00

If the names of the colors do not correspond to the color in which they are displayed (for example, if the word RED is displayed in the color green), you have incorrectly attached the cables. Refer to the owner's manual for your monitor to make sure the cables from the video terminal match the proper connectors on the monitor.

1.1.2 Hardcopy Output Devices

To produce hardcopy output of DATATRIEVE plots, use a DIGITAL hardcopy device that supports ReGIS graphics.

See the owner's manual for information on ReGIS printers and how to attach the printer to your terminal. The owner's manual documents typical operations, such as connecting cables and checking the status of communication switches, baud rate (speed), and parity value for the printer.

1.2 Steps to Take Before Using DATATRIEVE Plots

To use DATATRIEVE plots, invoke DATATRIEVE and specify the CDD directories that contain the domains and record definitions you want to use and the DATATRIEVE plots:

```
DTR> SET DICTIONARY CDD$TOP.DTR$LIB.DEMO
DTR> SET PLOTS CDD$TOP.DTR$LIB.PLOTS
```

The DATATRIEVE installation procedure contains questions about storing the necessary information to produce plots. If your site uses DATATRIEVE plots, your installer responded to these questions to store the information in the CDD directory CDD$TOP.DTR$LIB.PLOTS. See your system manager if you cannot access the plot information in CDD$TOP.DTR$LIB.PLOTS.

If you use the same CDD directory for plots each time you enter DATATRIEVE, you can include a SET PLOTS command in your DATATRIEVE startup command file. (See the *VAX DATATRIEVE User's Guide* for more information on startup command files.) After you have issued the SET PLOTS command, you can access any of the plots regardless of the CDD dictionary you specify in the SET DICTIONARY command.

Note

Do not delete any of the plots in CDD$TOP.DTR$LIB.PLOTS.

All of the plots described in this manual call one or more plots to perform functions such as clearing the screen and labeling axes. If you delete any of the plots from the CDD directory, other plots that call the deleted plots will not work.

1.3 Changing from a PRINT Statement to a Plot Statement

The DATATRIEVE PRINT statement allows you to display data in a tabular format. Plots produced with the DATATRIEVE plot statements help you present the same data in a more manageable and accessible format. This format can clarify the relationship among data, enhancing your reports and presentations. The following sections discuss how you can develop raw data into a plot.

The sample domain called ANNUAL_REPORT is installed during the DATATRIEVE installation procedure. This domain contains the annual report data for Digital Equipment Corporation for 1971 through 1980. This data includes figures for equipment sales, services, revenue, net income, net income per share, research, inventories, and employees. This section uses ANNUAL_REPORT to show the change from a PRINT to a plot statement.

If you have not already done so, set your dictionary default to the DEMO directory in the CDD. Ready the ANNUAL_REPORT domain:

```
DTR> SET DICTIONARY CDD$TOP.DTR$LIB.DEMO
DTR> READY ANNUAL_REPORT
```

You can use the PRINT statement to specify a subset of fields. For example, you might not want to print the data for all the fields. Instead, you may be concerned only with the total revenue growth by year for equipment sales and services. You can limit the display by specifying particular fields in the PRINT statement:

```
DTR> PRINT DATE,
CON> REVENUE,
CON> EQUIPMENT_SALES,
CON> SERVICES OF ANNUAL_REPORT

               EQUIPMENT
DATE   REVENUE   SALES   SERVICES

1980   2368.0   1779.4    588.6
1979   1804.1   1381.8    422.3
1978   1436.6   1128.1    308.5
1977   1058.6    847.5    211.1
1976    736.3    586.7    149.6
1975    533.8    433.2    100.6
1974    421.9    360.8     61.1
1973    265.5    229.1     36.4
1972    187.6    166.3     21.3
1971    146.8    133.0     13.8
```

You can easily edit the preceding PRINT statement, changing it to a plot statement.

You must first decide how you will specify the data that the plot statement will use. You have two options:

- Form a collection using the FIND statement.
- Specify the record stream using a record selection expression in the plot statement.

The following two sections show how you can specify the same data using the FIND statement or an RSE.

1.3.1 Plot Statement Using Data from a Collection

The plot statement consists of the keyword PLOT, the name of a plot, and the optional word USING. The word ALL is required when you plot data contained in the current collection.

The following example shows a finished plot statement for a shaded line graph. The statement works on the data in the current collection formed by the FIND statement:

```
DTR> FIND ANNUAL_REPORT SORTED BY DATE
DTR> PLOT MULTI_SHADE USING ALL FORMAT
CON> DATE USING Y(4),
CON> REVENUE ("Gross Revenue"),
CON> EQUIPMENT_SALES ("Equipment Sales"),
CON> SERVICES ("Software and Support") THEN
CON> PLOT CROSS_HATCH
```

1.3.2 Plot Statement Using Data from Record Selection Expression

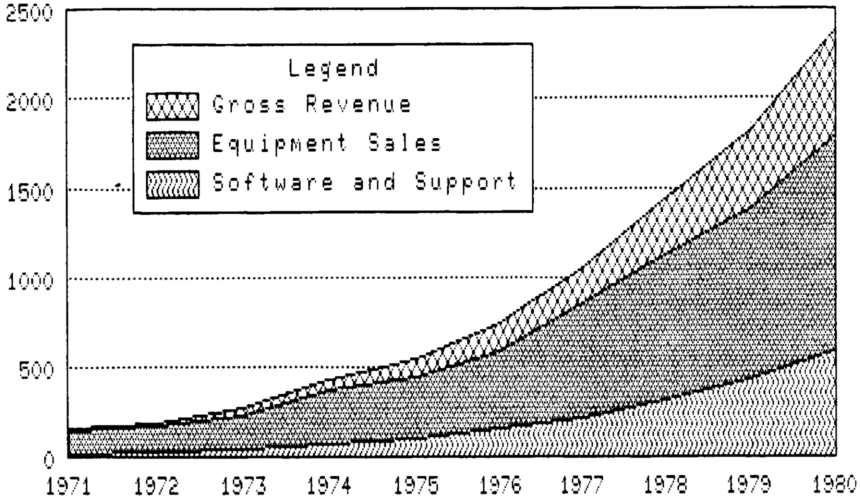
The following plot statement consists of the keyword PLOT, the name of a plot, and the optional word USING. The word ALL is *optional* because you are using the RSE, not data from the current collection.

This plot statement uses the identical data used in the previous plot statement. Instead of using the data from the current collection, however, this statement uses the data formed by the RSE:

```
DTR> PLOT MULTI_SHADE USING ALL FORMAT
CON> DATE USING Y(4),
CON> REVENUE ("Gross Revenue"),
CON> EQUIPMENT_SALES ("Equipment Sales"),
CON> SERVICES ("Software and Support") OF
CON> ANNUAL_REPORT SORTED BY DATE THEN
CON> PLOT CROSS_HATCH
```

1.3.3 Same Plot Produced by FIND Statement and RSE

The preceding two sections showed plot statements that used the same data; one formed the data using the FIND statement to form a current collection, the other used an RSE. Both plot statements produce the following plot.



ZK-00283-00

The plot allows you to demonstrate trends that are difficult to visualize from a tabular format. You can also see whether these trends are consistent over time.

Note

Some plot statements, such as PLOT STACKED_BAR and PLOT MULTI_SHADE, require relative shading of various fields. Printers cannot reproduce all the different shades produced by your terminal. The PLOT CROSS_HATCH statement converts the different gray levels into crosshatched shadings.

See the PLOT CROSS_HATCH section in the reference chapter for a list of all plots that require crosshatching.

Basic Types of DATATRIEVE Plots 2

There are five basic types of DATATRIEVE plots. You should be familiar with these types and your options before choosing a plot to present your data.

The five basic types of plots are:

- Bar charts
- Scattergraphs
- Line graphs
- Pie charts
- Utility plots

The following sections discuss these five basic types.

Note

Chapter 3 builds on the information from this chapter; it discusses choosing, designing, and improving your plot.

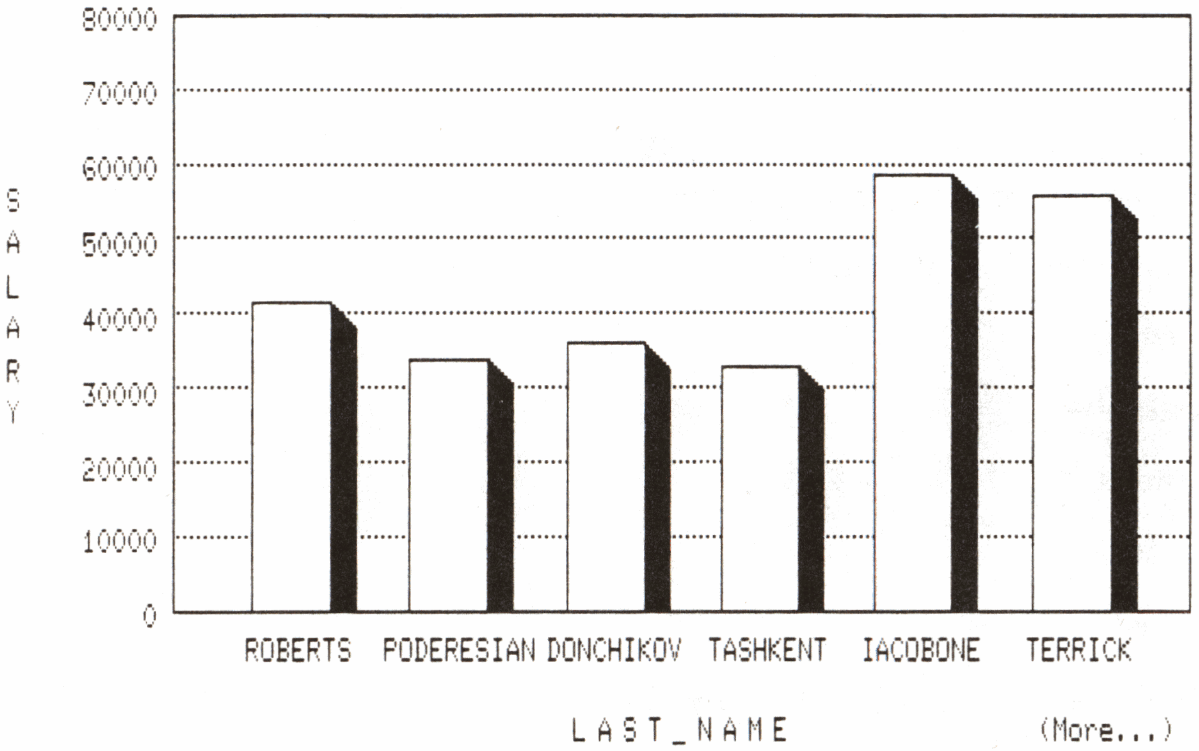
The beginning of the reference chapter includes a table of DATATRIEVE plot statements. The table categorizes the plot statements by types; it also includes a brief description of each plot. The table is a handy guide when you are choosing plots.

2.1 Bar Charts

DATATRIEVE bar charts represent quantities as rectangular bars with heights proportional to the values of the fields or expressions represented. You can use a bar chart to compare values of fields or expressions.

The following PLOT BAR statement creates a simple bar chart that shows the salary for each employee. It uses the sample domain PERSONNEL. (Appendix A shows all the sample domains used in this manual.)

```
DTR> FIND PERSONNEL SORTED BY DEPT
DTR> PLOT BAR ALL LAST_NAME, SALARY
```



ZK-00284-00

The value of the first argument you specified, LAST_NAME, is plotted on the horizontal (X) axis. The vertical (Y) axis tells you the value of the second argument, SALARY. The label (More...) in the lower right corner indicates there are more than six bars to be charted.

The following list shows all the bar chart statements:

- PLOT BAR
- PLOT BAR_AVERAGE
- PLOT HISTO
- PLOT MULTI_BAR

- PLOT MULTI_BAR_GROUP
- PLOT NEXT_BAR
- PLOT RAW_BAR
- PLOT STACKED_BAR

See the reference chapter for complete descriptions, syntax formats, and examples of these bar charts.

2.2 Scattergraphs and Line Graphs

Scattergraphs plot points based on their horizontal and vertical coordinates. Line graphs plot points and also connect the points with lines. You can use a scattergraph or line graph to compare values in fields or expressions.

Note

X coordinates are plotted according to values on the horizontal, or X axis. Y coordinates are plotted according to values on the vertical, or Y axis.

Scattergraphs and line graphs are useful in plotting trends. For example, the horizontal axis often represents some increment of time, such as the year in the following example:

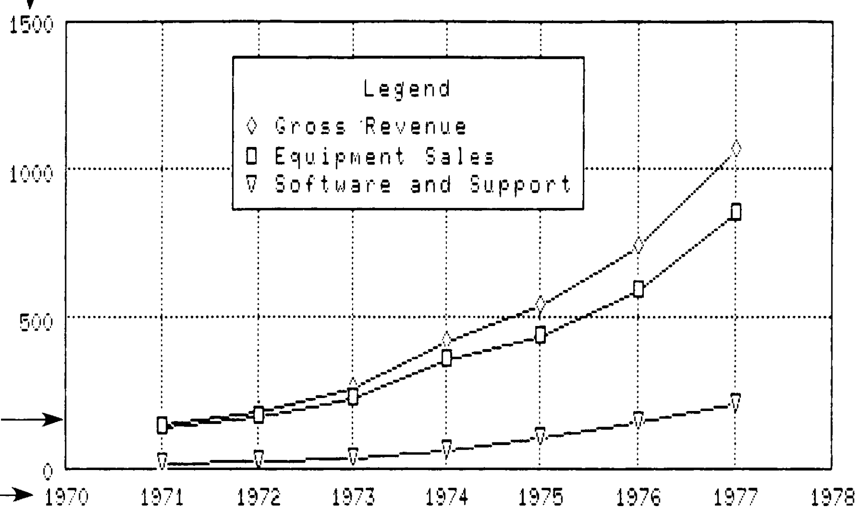
```
DTR> PLOT MULTI_LINE FORMAT DATE USING Y(4),
CON> REVENUE ("Gross Revenue"),
CON> EQUIPMENT_SALES ("Equipment Sales"),
CON> SERVICES ("Software and Support") OF
CON> ANNUAL_REPORT WITH FN$YEAR (DATE) LT 1978-
CON> SORTED BY DATE
```

The plot statement includes:

- An argument specifying the year portion of DATE for the horizontal (X) axis.
- Three additional arguments (REVENUE, EQUIPMENT_SALES, and SERVICES) represented by uniquely marked lines.

The resulting multiple line graph includes a vertical (Y) axis that shows the numeric value of each field represented by a line. The upper line of the Y axis specifies the maximum dollar boundary used in the plot.

The vertical (Y) axis specifies the maximum dollar boundary used in the plot.



ZK-00285-00

These lines represent the fields that you specified.

The horizontal (X) axis represents the relationship of time to REVENUE, EQUIPMENT_SALES, and SERVICES.

The following list shows all the scattergraph statements:

- PLOT DATE_LOGY
- PLOT DATE_Y
- PLOT LOGX_LOGY
- PLOT LOGX_Y
- PLOT X_LOGY
- PLOT X_Y

The following list shows all the line graph statements:

- PLOT MULTI_LINE
- PLOT MULTI_LR
- PLOT MULTI_SHADE

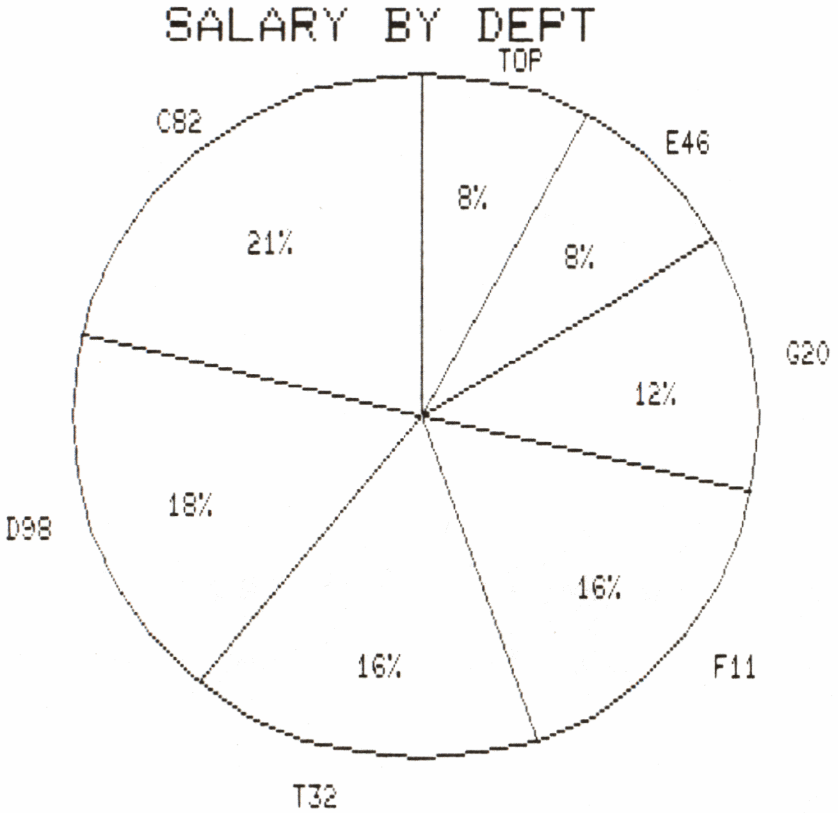
See the reference chapter for complete descriptions, syntax formats, and examples of these scattergraphs and line graphs.

2.3 Pie Charts

DATATRIEVE pie charts represent quantities as slices of a pie proportional to the whole pie. You can use a pie chart to compare values of fields or expressions.

The following statements use the PLOT VALUE_PIE statement with the PERSONNEL domain. The resulting chart displays the percent of total salary that each department represents:

```
DTR> READY PERSONNEL
DTR> PLOT VALUE_PIE ALL DEPT, SALARY OF PERSONNEL
```



ZK-00286-00

The following list shows all the pie chart statements:

- PLOT PIE
- PLOT RAW_PIE
- PLOT VALUE_PIE

See the reference chapter for complete descriptions, syntax formats, and examples of these charts.

2.4 Utility Plots

DATATRIEVE utility plot statements let you enhance and manipulate other plots. For example, the PLOT TITLE statement lets you add a title for your plot. The PLOT HARDCOPY statement produces hardcopy output of a plot you have already produced (if your terminal is connected to a DIGITAL-supported ReGIS hardcopy output device). The PLOT BIG statement also lets you produce hardcopy output, but the output is four times larger than PLOT HARDCOPY output.

The following list shows all the utility plot statements:

- PLOT BAR_ASCENDING
- PLOT BIG
- PLOT CONNECT
- PLOT CROSS_HATCH
- PLOT HARDCOPY
- PLOT LEGEND
- PLOT LIMITS_X
- PLOT LIMITS_Y
- PLOT LR
- PLOT MONITOR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT SHADE
- PLOT SORT_BAR
- PLOT TITLE

See the reference chapter for complete descriptions, syntax formats, and examples of these plot statements.

Choosing, Designing, and Improving Your Plot 3

Plots show various types of relationships. To correctly choose and design your plot, first decide your plot's purpose and what relationship you want to illustrate.

The following sections discuss choosing, designing, and improving your plot.

3.1 Determining the Purpose of Your Plot

To help you decide your plot's purpose, write a sentence or two explaining exactly what you want your plot to show. For example, you may want to:

- Compare the revenue and sales from an annual report for a five year period
- Determine the distribution of yachts for each beam size
- Compare the percentages of the total salaries of each department in a corporation

Using your written description of the relationship you want to plot, follow the guidelines in the next section to choose which type of plot to use.

3.2 Five Types of Relationships

There are five basic types of plot relationships. The recommended type of plot for each relationship is listed first:

Type of Relationship	Suggested Plots
Time comparisons	Line, Scatter, Bar, Histogram
Parts of the whole	Pie, Bar
Comparison of several items	Bar, Pie
Comparison of two sets of values	Line, Scatter, Bar, Log Scale
Number of times a value occurs across a range of possibilities	Histogram

The following sections discuss each of these relationships and the recommended plots to use.

Note

> The beginning of the reference chapter includes a table of DATATRIEVE plot statements. The table categorizes the plot statements by types and includes a brief description of each plot.
>
> The reference chapter also has detailed articles on each plot statement showing syntax, usage notes, results, restrictions, and examples.

3.2.1 Time Comparisons (Line Graphs, Scattergraphs, Bar Charts)

Time comparisons illustrate the relationship between a period of time and changes in your data. These changes include:

- Increases
- Decreases
- Trends

Use a line graph or bar chart to illustrate time comparisons. For example, these sentences illustrate relationships best plotted in time comparison plots:

- Annual figures for revenue and inventory have increased for the past ten years.
- The number of employees in our company has decreased during the past six months.

3.2.1.1 Line Graphs and Scattergraphs for Time Comparisons — Use PLOT MULTI_LINE to portray trends of up to three sets of values over time.

If you want to emphasize trends, use PLOT MULTI_SHADE, which shades the areas beneath the lines with different shades or patterns.

The PLOT DATE_Y statement creates a scattergraph that shows a chronological trend. The PLOT DATE_LOGY statement also creates a time-related scattergraph, but uses a logarithmic scale for the vertical (Y) axis.

3.2.1.2 Bar Charts for Time Comparisons — Bar charts are also useful to illustrate time comparisons. Use the following types of charts for the listed applications:

Type of Chart	Illustration
Simple Bar	Trend of one factor over time
Clustered Bars	Trends of two factors over time
Stacked Bar	Totals or sums of totals (not percentages)
Histogram	Changes over time

3.2.2 Parts of the Whole (Pie Chart, Bar Chart)

The relationship of two or more parts to one whole entity is known as parts of the whole. A plot showing parts of a whole may indicate:

- Percentages
- Portions
- Market shares

Use a pie or bar chart to illustrate the relationship of parts to a whole. For example, these sentences describe relationships illustrating parts of the whole:

- What percentage do trainees and experienced employees represent in department C82?
- What percentage of total employees earn less than $30,000 annually? Between $30,000 and $45,000? Greater than $45,000?

3.2.3 Comparing Several Items (Bar Chart, Pie Chart)

You can compare several items to illustrate the differences within a particular group. For example, you might want to compare differences within each of the following topics:

- Corporations
- People
- Projections

This type of comparative relationship does not include the element of time.

Use a bar chart or pie chart to illustrate the comparison of several items. For example, these sentences describe the comparison of several items:

- What is the salary of each employee, by employee name, in department T32?
- What are the average prices of the yachts that each manufacturer builds?
- What are the total salaries of all employees in each department, by department name?

3.2.4 Comparing Multiple Values (Line Graph, Scattergraph, Bar Chart)

Comparing two or more sets of values illustrates how the changes in one set are related to the changes in the other set.

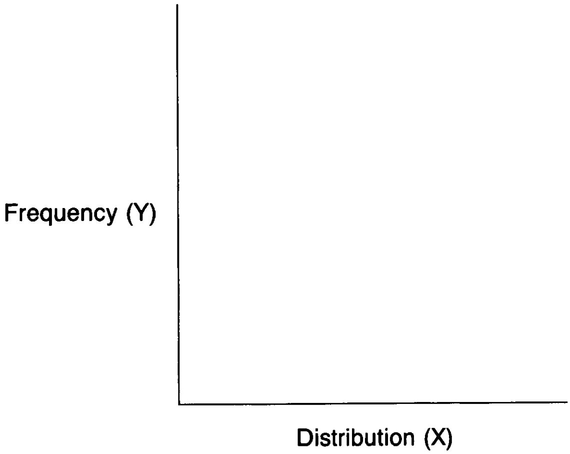
Use the line graph, scattergraph, or bar chart to portray the relationship between values. For example, these sentences illustrate the relationship between two sets of values:

- What is the displacement of each yacht relative to its price?
- What is the overall length of each yacht relative to its beam?
- What are the annual report figures for research relative to those for revenue?

If the relationship shows relative changes that are drastically different, use the logarithmically scaled scattergraphs.

3.2.5 Frequency Distribution (Histogram)

Frequency distribution shows the number of times a value occurs across a range of possibilities. The Y axis totals the number of times a value occurs (the frequency). The X axis lists the range of possibilities.



Frequency represents the number of occurrences of related values within a specified range. Distribution is a range of continuous variables, such as salaries, ages, or types of yachts.

Use the PLOT HISTO statement to plot frequency distribution. For example, these sentences illustrate frequency distribution:

- What is the distribution of annual salaries among the number of employees working in department T32?
- What is the distribution of yachts in each length-over-all category?

3.3 Designing and Improving Plots

Once you choose which type of plot is appropriate for your purpose, you can improve your plots by correctly designing and controlling the appearance of the final display.

The following sections discuss these topics.

3.3.1 Guidelines for Designing Plots

The following guidelines give you the basic framework for a successful design and presentation. These guidelines apply to both oral and written presentations:

- Define your audience.

  The purpose of any presentation, including one using plots, is to convey information clearly and accurately to a specific audience. You should first define the audience who will be using your plot. Consider the following questions:

  - What is the technical level of the people reading the plots? Does your audience understand the terminology of the subject matter? Should you use more common synonyms?

  - Does your audience understand logarithms in plot scales? Avoid logarithmic scales if you are not sure of your audience. If you feel you must use logarithmic scales, plan to explain how they function.

- Be explicit.

  Each plot should be self-explanatory. Provide explicit labels and titles to describe the information displayed in the plot.

- Keep the format simple.

  Convey your graphic message simply and clearly. Choose the important points you want to make and emphasize only those key points in your plots.

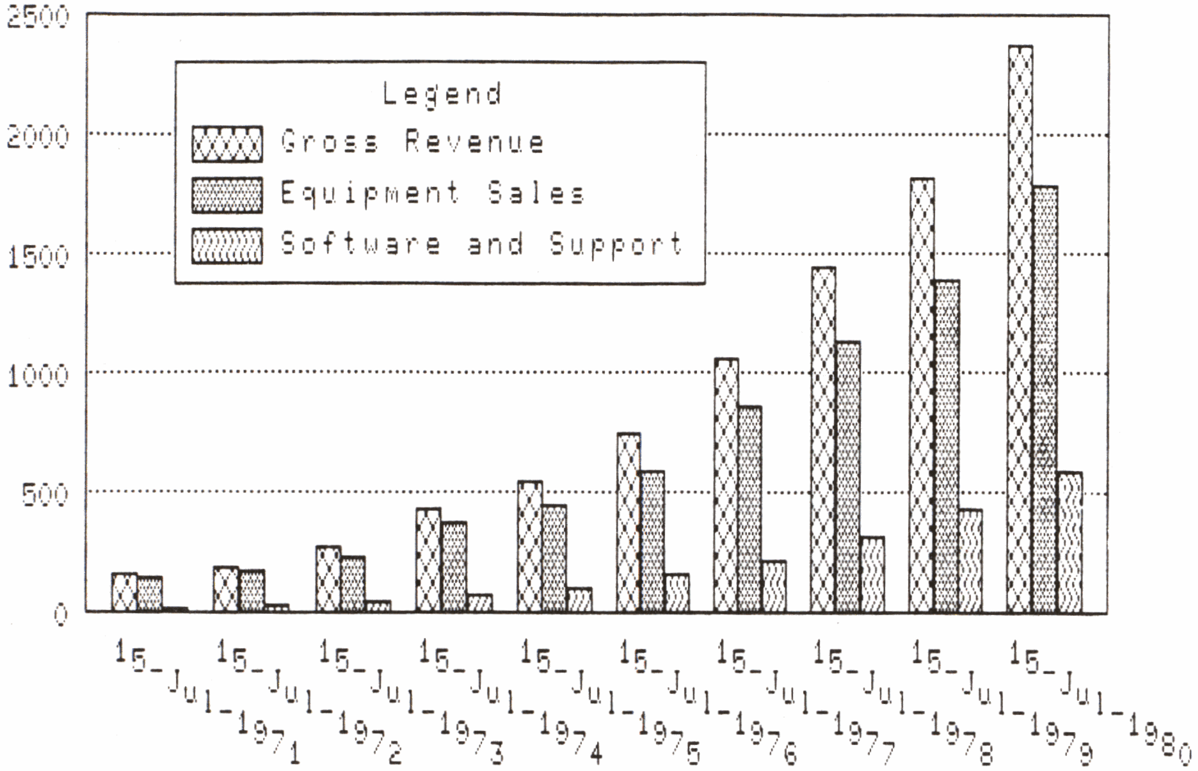
- Follow the standards of your business or audience.

  Be aware of the standards used by your profession or audience and follow them.

3.3.2 Improving Plots

You can improve your plots by removing redundant information. In the following example, each annual report is issued on July 15th. The repetition of the day and month in the plot does not contribute any new information to the plot:

```
DTR> FIND ANNUAL_REPORT SORTED BY DATE
DTR> PLOT MULTI_BAR USING ALL DATE,
CON> REVENUE ("Gross Revenue"),
CON> EQUIPMENT_SALES ("Equipment Sales"),
CON> SERVICES ("Software and Support") THEN
CON> PLOT CROSS_HATCH
```

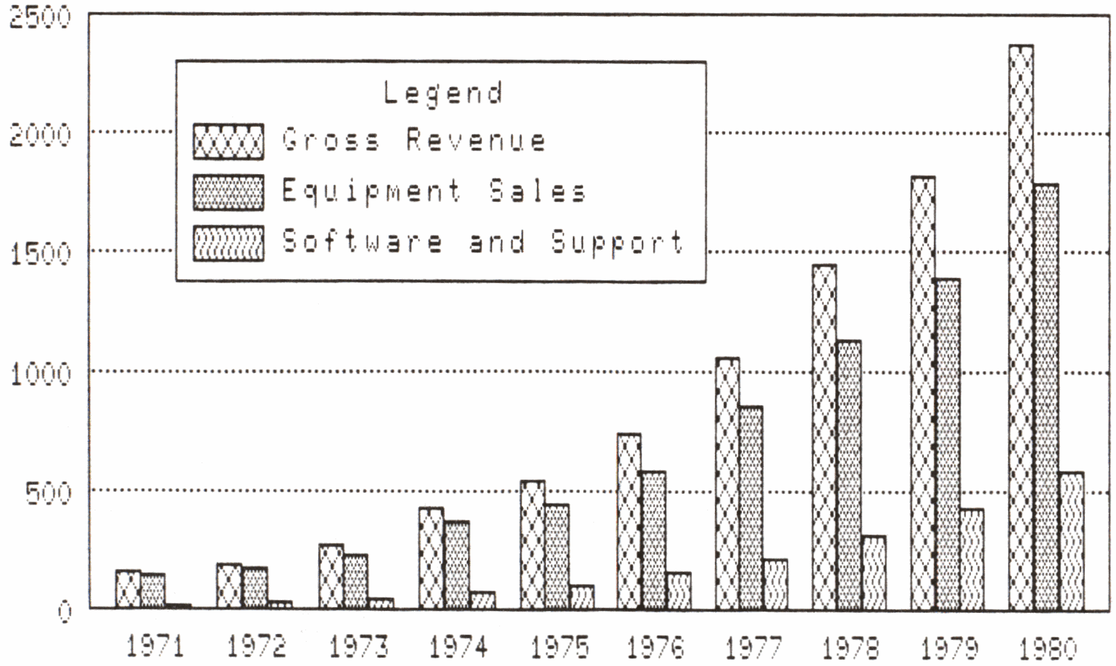


ZK-00287-00

Use a FORMAT clause to strip off everything in the DATE field except the year:

```
DTR> PLOT MULTI_BAR USING ALL FORMAT DATE USING Y(4),
CON> REVENUE ("Gross Revenue"),
CON> EQUIPMENT_SALES ("Equipment Sales"),
CON> SERVICES ("Software and Support") THEN
CON> PLOT CROSS_HATCH
```

The resulting plot shows the year as 1971, 1972, 1973, and so on, rather than 15-JUL-1971, 15-JUL-1972, and 15-JUL-1973.



ZK-00288-00

You can also improve your plots by ordering the data for each field within each year according to the general trend of increasing values.

In general, the sequence of the fields that you plot is important, especially with line graphs and bar charts:

- Line graphs

  Because of the way DATATRIEVE "paints" the screen with PLOT MULTI_SHADE, you should place the field with the highest values first, the field with the second highest values next, and the field with the lowest values last. Such sequencing allows DATATRIEVE to paint the screen without overwriting any values.

  See the section on the PLOT MULTI_SHADE statement in the reference chapter for more information.

- Bar charts

  Unlike line graph values, the values in bar charts are not overwritten. You may, however, want to sort the bars of the chart.

  If your bar chart was produced by the PLOT BAR, PLOT BAR_AVERAGE, PLOT RAW_BAR, or PLOT HISTO statement:

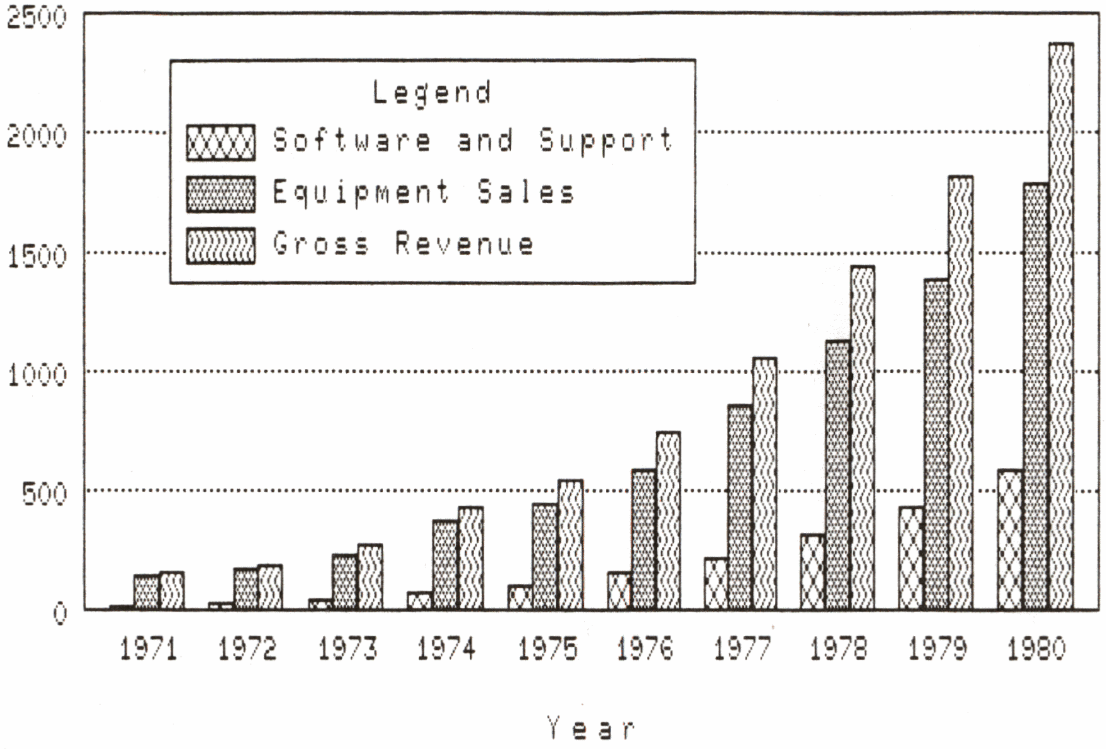
  - Use the PLOT BAR_ASCENDING statement to sort and replot the bars in ascending order
  - Use the PLOT SORT_BAR statement to sort and replot the bars in descending order

  If your bar chart was produced using the PLOT MULTI_BAR statement, you may want to place the field with the lowest value first. Edit the statement in the former example to reorder the fields and to insert a PLOT CROSS_HATCH statement:

```
DTR> PLOT MULTI_BAR USING ALL FORMAT
CON> DATE USING Y(4) ("Year"),
CON> SERVICES ("Software and Support"),
CON> EQUIPMENT_SALES ("Equipment Sales"),
CON> REVENUE ("Gross Revenue") THEN
CON> PLOT CROSS_HATCH
```

  Remember to correctly place the commas when resequencing lines, or you will get a syntax error.

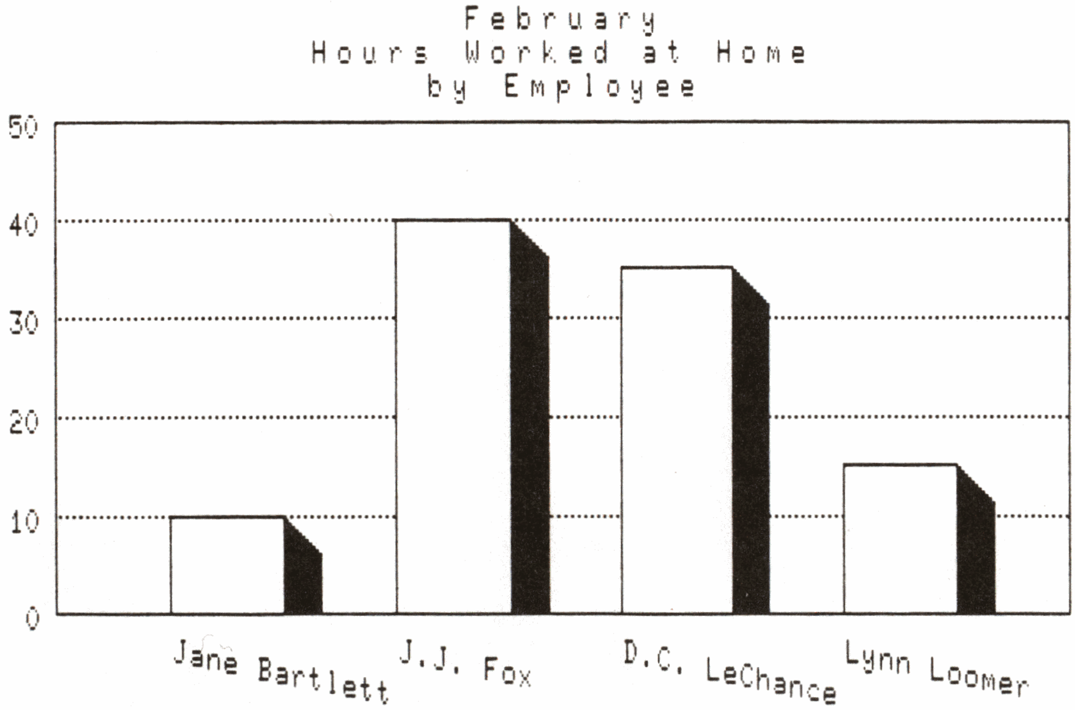
  The chart shows the bars of each cluster arranged in ascending order.



ZK-00289-00

You can also improve your plot by adding a title. This helps clarify the purpose of the plot for your audience:

```
DTR> PLOT RAW_BAR 10 ("Jane Bartlett"),
CON> 40 ("J.J. Fox"),
CON> 35 ("D.C. LeChance"),
CON> 15 ("Lynn Loomer") THEN
CON> PLOT TITLE "February"/"Hours Worked at Home"/"by Employee"
```



ZK-00290-00

Reference Section

Replace this page with heavy page entitled Reference Section located at the end of this package.

DATATRIEVE Plot Statements 4

This chapter describes the DATATRIEVE plot statements. The sections that follow include an explanation of syntax common to all plots and descriptions of each of the DATATRIEVE plot statements. Individual plot statements are arranged in alphabetical order. The function of each plot statement is described immediately after the name of the plot statement. Each section then divides its presentation into the following categories of information.

- **Format**

  The format of the statement includes the spelling and placement of keywords and the placement of required and optional syntax elements. As a rule, statement names and other keywords cannot be abbreviated. The sequence of statement elements is also critical. You must follow the sequence shown in the format. If you omit an optional element, proceed to the remaining elements in a left-to-right order.

- **Results**

  Results tell you what action DATATRIEVE takes when you use the statement and its various options.

- **Usage Notes**

  Usage Notes present some common uses of the statement and indicate what other statements you can use in conjunction with the statement.

- **Restrictions**

  The restrictions tell you what requirements and limits there are on the use and action of a statement. Some sections do not include a restrictions category.

- **Examples**

  The examples show the use of representative sequences of statements. They refer to the ANNUAL_REPORT, PERSONNEL, and YACHTS domains and record definitions found in the CDD$TOP.DTR$LIB.DEMO dictionary. To access the plot definitions for each of the plot statements, enter the following command:

  ```
  DTR> SET PLOTS CDD$TOP.DTR$LIB.PLOTS
  ```

Table 4-1 lists various ways of presenting data in DATATRIEVE plots and will help you choose the best plot for the data you want to display. Look in the first column for the type of plot you want. Then look in the next column for the name of the DATATRIEVE plot statement. Follow the row across for the results of the plot statement.

Table 4-1: Summary of Plot Statements

Type	Statement	Result
Bar Charts	PLOT BAR	Total of values contained in one field
	PLOT BAR_AVERAGE	Average of values contained in one field
	PLOT HISTO	Frequency of occurrence of values in a field
	PLOT MULTI_BAR	Bar or cluster of bars for each unique record
	PLOT MULTI_BAR_GROUP	Bar or cluster of bars representing the totals from a number of records
	PLOT NEXT_BAR	Next screen with overflow bars
	PLOT RAW_BAR	Relative height of values supplied as arguments
	PLOT STACKED_BAR	Y values stacked in one bar for each X value

(continued on next page)

Table 4-1: Summary of Plot Statements (Cont.)

Type	Statement	Result
Line Graphs	PLOT MULTI_LINE	One value plotted along X axis, up to three values plotted along Y axis (unshaded)
	PLOT MULTI_LR	Linear regression of 1 value plotted along X axis and up to three values plotted along Y axis (unshaded)
	PLOT MULTI_SHADE	One value plotted along X axis, up to three values plotted along Y axis (shaded)
Pie Charts	PLOT PIE	Number of occurrences for each unique field value plotted as percentage of whole
	PLOT RAW_PIE	Simple pie chart of data you supply
	PLOT VALUE_PIE	Total of values in a field plotted as percentage of whole
Scattergraphs	PLOT DATE_LOGY	Time line for X axis and logarithmic scaling for Y axis
	PLOT DATE_Y	Time line for X axis and decimal scaling for Y axis
	PLOT LOGX_LOGY	Fully logarithmic scaling for values on X and Y axes
	PLOT LOGX_Y	Logarithmic scaling for values on X axis and decimal scaling for Y axis
	PLOT X_LOGY	Logarithmic scaling for values on Y axis and decimal scaling for X axis
	PLOT X_Y	Decimal scaling for values on X and Y axes

(continued on next page)

Table 4-1: Summary of Plot Statements (Cont.)

Type	Statement	Result
Utility	PLOT BAR_ASCENDING	Values represented by bars, sorted in ascending order
	PLOT BIG	Hardcopy output (4 times larger)
	PLOT CONNECT	Points of scattergraph connected
	PLOT CROSS_HATCH	Shading changed to crosshatching
	PLOT HARDCOPY	Normal size plot printed on attached printer
	PLOT LEGEND	Display of legend
	PLOT LIMITS_X	Minimum and maximum limits on the horizontal (X) axis specified
	PLOT LIMITS_Y	Minimum and maximum limits on the vertical (Y) axis specified
	PLOT LR	Linear regression for scattergraphs
	PLOT MONITOR	Checks color of video output
	PLOT PAUSE	Causes a pause between plots
	PLOT REFERENCE_X	Solid reference line plotted on the horizontal (X) axis
	PLOT REFERENCE_Y	Solid reference line plotted on the vertical (Y) axis
	PLOT RE_PAINT	Displays most recent plot
	PLOT SHADE	Shading for scattergraph
	PLOT SORT_BAR	Values represented by bars, sorted in descending order
	PLOT TITLE	Adds a title to the most recent plot
Miscellaneous	PLOT WOMBAT	Displays a marsupial

4.1 Syntax Common to All Plot Statements

Each DATATRIEVE plot statement uses the following general syntax.

Format

```
PLOT plotname   [USING]   [ALL]

         [arg [,   arg]...]   [OF rse]

              [   ( file-spec  )  ]
              [ON {            }  ]
              [   ( *.prompt   )  ]
```

Arguments

plotname

Is the name of the plot, for example:

```
DTR> PLOT BAR
```

BAR is the name of the plot.

USING

Is an optional keyword to make the syntax more like English. The keyword USING does not affect the plot statement.

ALL

Is a keyword that may be optional or required depending on the structure of the plot statement you use. It is:

- Optional if you use the OF rse clause
- Required if you use the current collection without any qualification

For example, the following two plot statements are the same:

```
DTR> PLOT BAR BUILDER, PRICE OF YACHTS WITH
CON> PRICE NE 0

DTR> FIND YACHTS WITH PRICE NE 0
DTR> PLOT BAR ALL BUILDER, PRICE
```

The first example uses an OF rse clause in which the keyword ALL is not required. In the second example, ALL is required because it refers to the collection created in the FIND statement.

In the syntax diagrams of the sections that follow, the keyword ALL is shown in brackets as an optional element. Note, however, that if you use the keyword ALL to refer to the contents of the current collection, ALL is required.

arg

Is a field name or other value expression. The argument can also contain an optional label string. To specify a label string, put a quoted string inside parentheses after the value expression. For example:

```
DTR> PLOT X_Y LOA,
CON> DISP / 2000 ("Weight in tons") OF YACHTS
```

OF rse

Specifies the record stream to be used in the plot. (See the chapter on record selection expressions in the *VAX DATATRIEVE Reference Manual* for more information on RSEs.)

If you do not specify an RSE, the plot statement will use data from the current collection.

ON {file-spec }
{*.prompt}

Specifies a device or file for output. If you do not specify a device or file specification, DATATRIEVE displays the plot on your terminal screen.

You can use a complete VMS file specification or simply a file name. DATATRIEVE uses .LIS as the default file type. You can also use a prompting value expression that prompts you for a file or device specification. (See the section on prompting value expressions in the *VAX DATATRIEVE Reference Manual* for more information.) For example:

```
DTR> PLOT PIE ALL RIG ON DRA0:[IACOBONE]RIGPLOT.LIS
DTR> PLOT PIE ALL RIG ON *."File Name or Device Specification"
```

You can use the ON clause with any of the plot statements to create a file containing the ReGIS graphics commands. Note the following rules and restrictions:

- You can use any terminal when you issue the plot statement that creates such a file. However, you need a DIGITAL supported ReGIS terminal to convert the code in that file into a graphic representation of data.

- You can display the plot file you create at your DIGITAL supported ReGIS terminal with the DCL TYPE command. If you examine the file with a text editor, it is meaningless unless you are familiar with the ReGIS graphics language.

- Sometimes the ReGIS code is not sufficient to draw the picture. For example, some plot statements (such as PLOT CROSS_HATCH after PLOT MULTI_SHADE) take the output of another plot, change the plot in some way, and send the output to a file. The ReGIS code stored in that file produces a crosshatched plot only if the multishaded plot is still in the graphics memory of your DIGITAL supported ReGIS terminal.

4.2 Labels with DATATRIEVE Plots

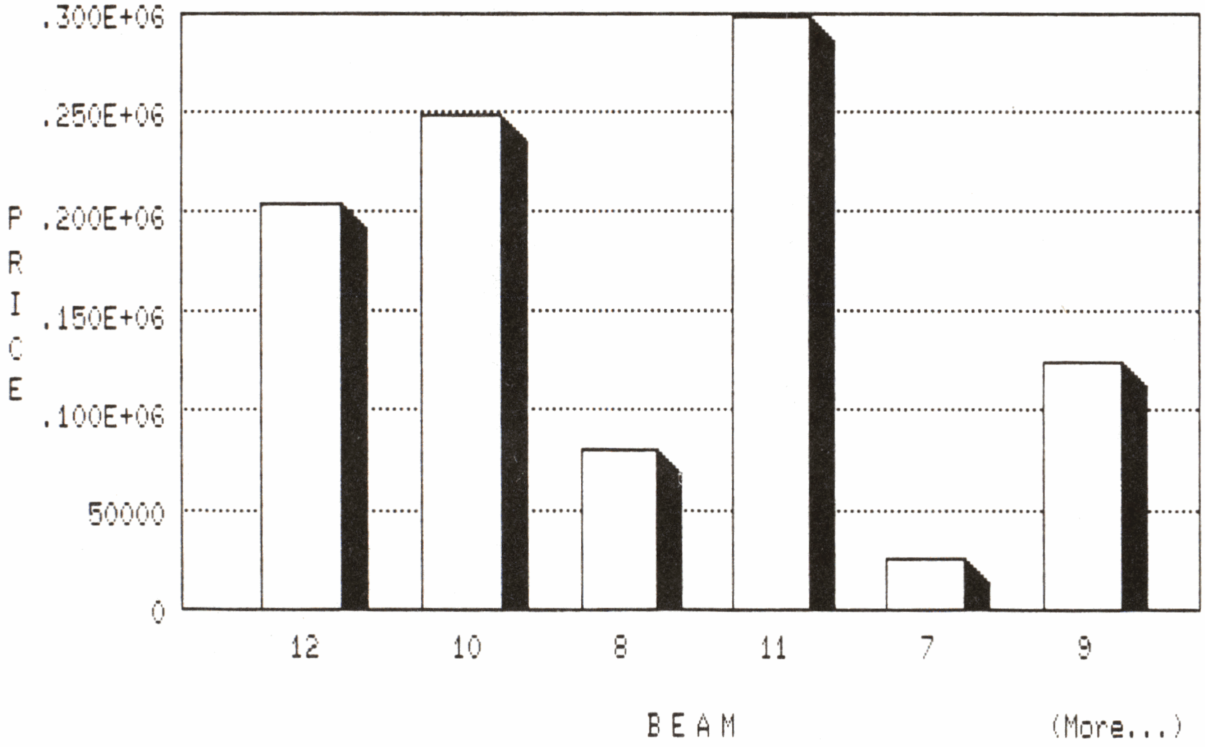
There are several different types of labels used with DATATRIEVE plots. DATATRIEVE produces default labels for plots; in addition, you can specify and alter particular labels. The following sections discuss default, specified, and altered labels.

4.2.1 Default Labels

DATATRIEVE produces two different types of default labels: those for plots with X and Y axes, and those for pie charts.

- For plots with X and Y axes, DATATRIEVE labels the axes using the field name or other value expression from the arguments. For example, the following PLOT BAR statement uses the arguments BEAM and PRICE. The resulting plot has the label BEAM on the X axis and the label PRICE on the Y axis.

```
DTR> PLOT BAR ALL BEAM, PRICE OF
CON> YACHTS WITH PRICE NE 0
```



ZK-00291-00

In addition to the labels that summarize the X and Y axes, DATATRIEVE supplies reference points in the plot:

- In the preceding plot, DATATRIEVE supplied reference points for the bars on the X axis. The reference points showed each bar's width in feet: 12, 10, 8, 11, 7, and 9.
- DATATRIEVE also supplied the reference points on the Y axis showing the increasing price: 50000, .100E+06, .150E+06, .200E+06, .250E+06, .300E+06.

  Notice that these prices use scientific notation. DATATRIEVE uses scientific notation to display reference points greater than 100,000. See the section on eliminating scientific notation for information on customizing these labels.

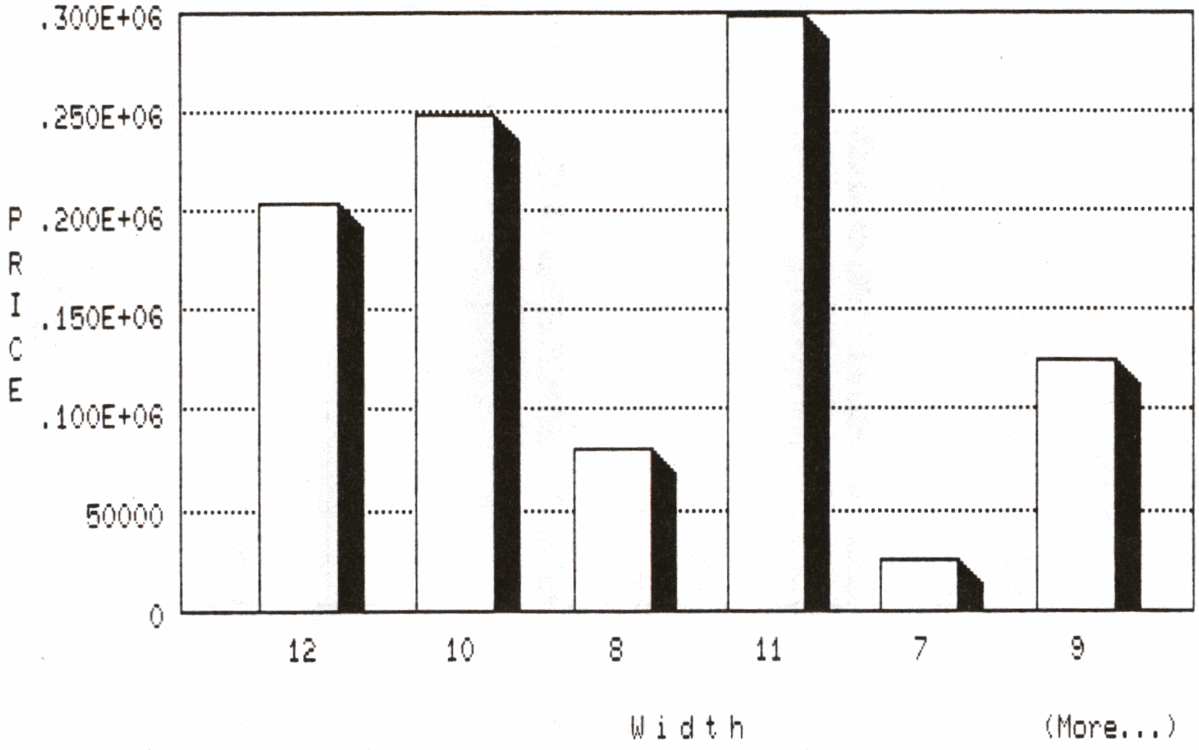
- DATATRIEVE produces various labels for pie charts:
  - For a PLOT PIE or PLOT VALUE_PIE statement, DATATRIEVE prints the percentage and a label for groups with a large enough slice to accommodate these labels. Groups with a small percentage (and thus a small slice) might not have a label or percentage displayed.
  - For the PLOT RAW_PIE statement, the field name or other value expressions you specified as arguments are used in the labels:
    - If you use value expressions for arguments, you specify the label by including a label string with the argument. (See the following section on specifying label strings for more information.)
    - If you use field names for the arguments, DATATRIEVE uses the field name for the label.

    In both of the preceding cases, DATATRIEVE then prints the percentage and a label for arguments with a large enough slice to accommodate these labels. Arguments with a small percentage (and thus a small slice) might not have a label or percentage displayed.

4.2.2 Specifying Label Strings

Instead of using the default labels for plots with an X and Y axis, you can specify a label string with the argument of the plot statement. To specify a label string, put a quoted string inside parentheses after the field name or other value expression. For example, you can specify the label Width instead of BEAM:

```
DTR> PLOT BAR ALL BEAM ("Width"), PRICE OF
CON> YACHTS WITH PRICE NE 0
```



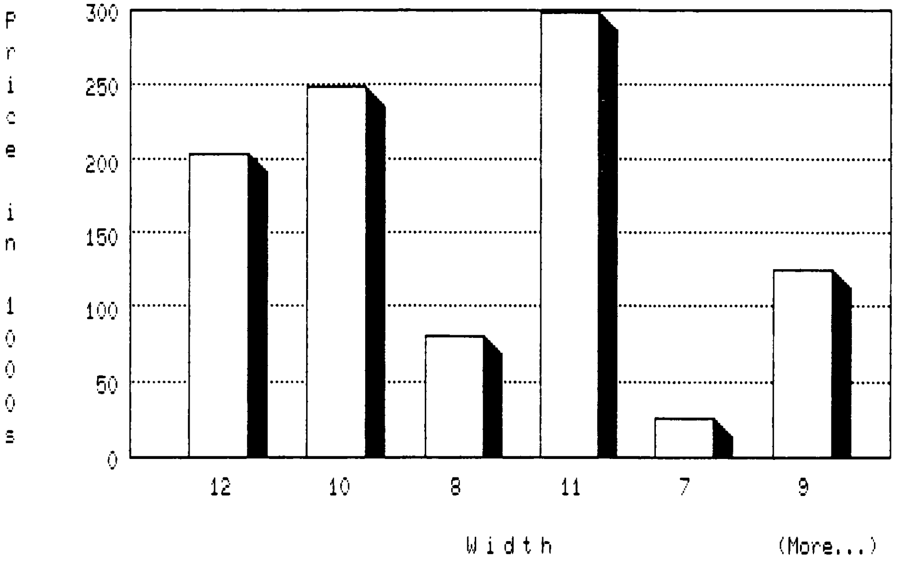
ZK-00292-00

4.2.3 Eliminating Scientific Notation

By default, DATATRIEVE uses scientific notation when labeling reference points greater than 100,000. To eliminate this scientific notation, divide the argument by either 100 or 1000. (Which number you divide by is your choice and depends on the situation. For example, prices are generally shown in thousands instead of hundreds. Follow the standards of your profession or business.)

Be sure to supply a label string showing what the new gradations represent:

```
DTR> PLOT BAR ALL BEAM ("Width"),
CON> PRICE/1000 ("Price in 1000s") OF
CON> YACHTS WITH PRICE NE 0
```



ZK-00293-00

4.3 PLOT BAR Statement

Creates a simple bar chart.

Format

```
                                            ┌    { file-spec }  ┐
PLOT BAR   [ALL]   arg,   arg   [OF rse]    │ON {            }  │
                                            └    { *.prompt  }  ┘
```

Arguments

arg

The PLOT BAR statement requires two arguments. The arguments can be either field names or other value expressions:

- The first argument provides the values for the horizontal (X) axis and groups all records that share the same value for the argument.
- The second argument provides the values for the vertical (Y) axis. Values are totaled for each group of records specified by the first argument.

Result

The PLOT BAR statement produces a simple bar chart of up to six bars. If there are more than six bars, DATATRIEVE breaks the data into separate charts containing a maximum of six bars. The notice (More...) appears at the bottom right of the screen if additional bars remain to be plotted. To display the additional charts, type PLOT NEXT_BAR.

Usage Note

You can use the following utility plots with the PLOT BAR statement:

- PLOT BAR_ASCENDING
- PLOT BIG
- PLOT HARDCOPY
- PLOT PAUSE

- PLOT RE_PAINT
- PLOT SORT_BAR
- PLOT TITLE

See the section on each of these utility plot statements for more information.

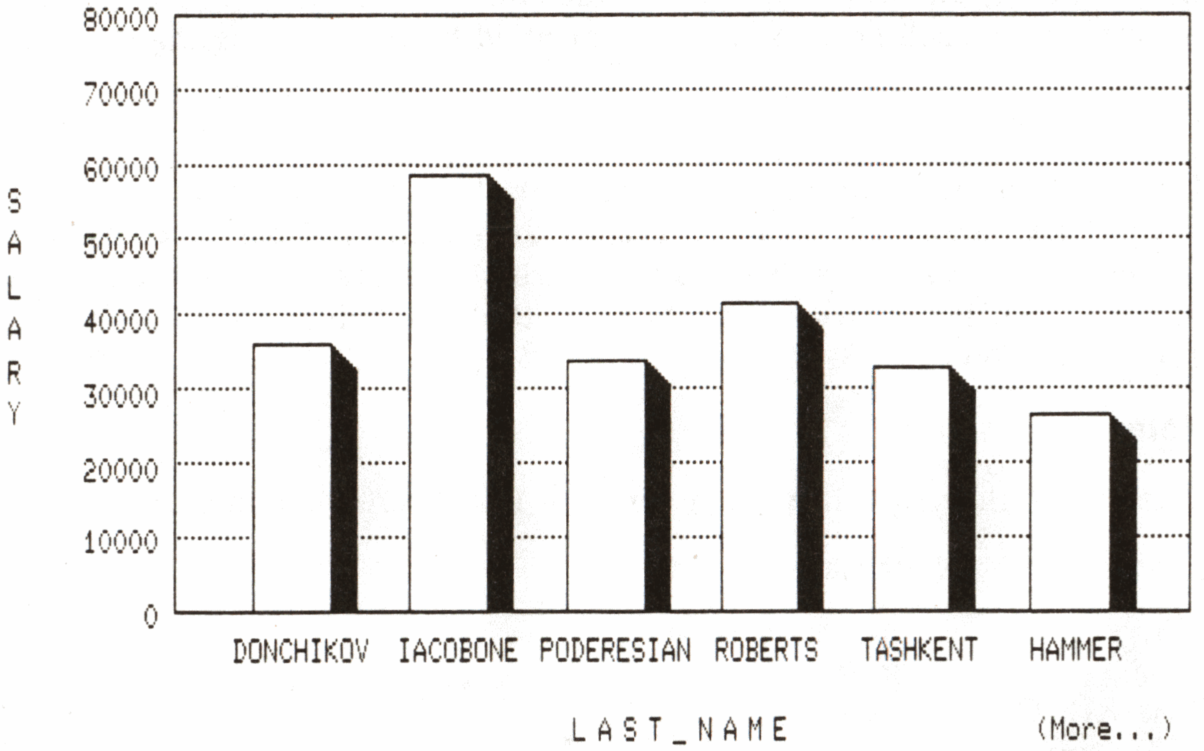
Examples

Example 1

Create a collection from the PERSONNEL domain. Then create a bar chart showing the salary of each employee. Note that the keyword ALL is required because the plot statement refers to the collection created by the first statement.

```
DTR> FIND PERSONNEL SORTED BY DEPT, LAST_NAME
DTR> PLOT BAR ALL LAST_NAME, SALARY
```

Plot for Example 1



ZK-00294-00

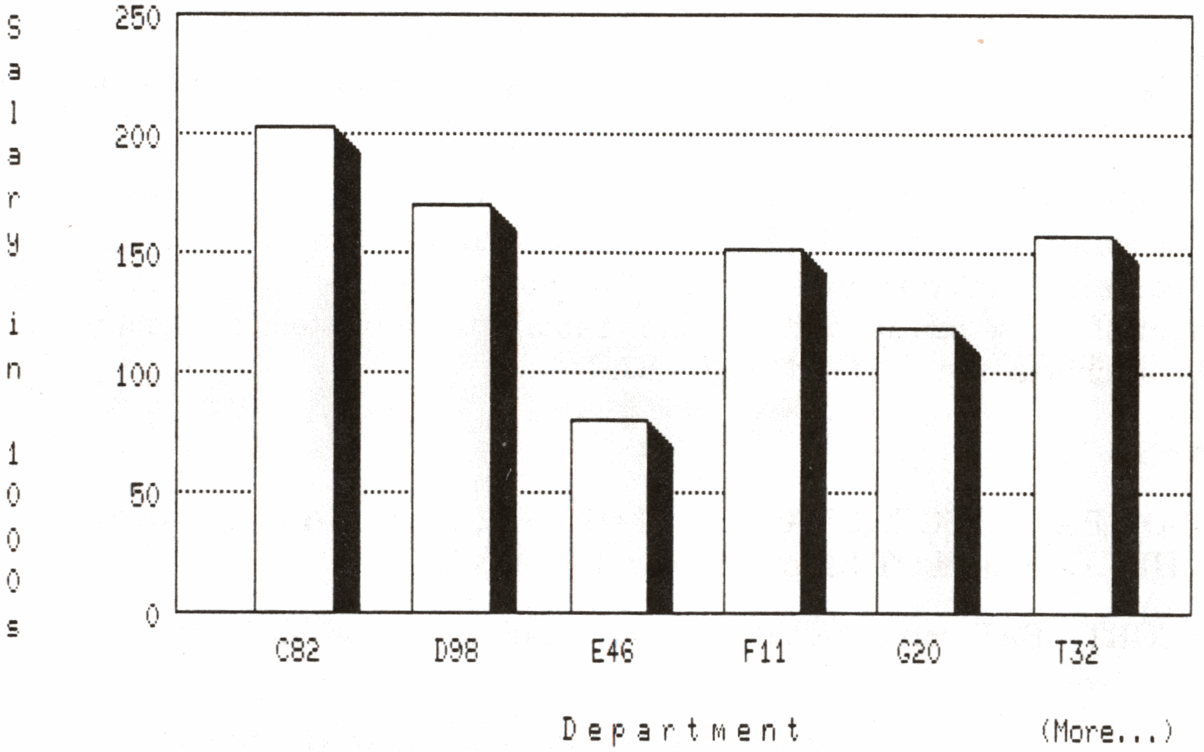
Example 2

Plot the total salary for each department in PERSONNEL.

Dividing the SALARY argument by 1000 eliminates the default scientific notation along the vertical (Y) axis. The quoted strings in parentheses after the arguments replace the default labels DEPT and SALARY.

```
DTR> PLOT BAR ALL DEPT ("Department"),
CON> SALARY/1000 ("Salary in 1000s") OF
CON> PERSONNEL SORTED BY DEPT, LAST_NAME
```

Plot for Example 2



ZK-00295-00

4.4 PLOT BAR_ASCENDING Statement

Sorts the bars in a bar chart or a histogram by ascending Y values.

Format

```
PLOT BAR_ASCENDING
```

Argument

The PLOT BAR_ASCENDING statement takes no argument.

Results

- PLOT BAR_ASCENDING uses the data from the bar chart you have just created and sorts bars in ascending order.

- PLOT BAR_ASCENDING plots up to six bars on a chart. If there are more than six bars, DATATRIEVE breaks the data into separate charts, each of which contains a maximum of six bars. The notice (More...) appears at the bottom right of the screen if additional bars remain to be plotted. To display the additional charts, type PLOT NEXT_BAR.

Usage Notes

- Use PLOT BAR_ASCENDING with PLOT BAR, PLOT BAR_AVERAGE, PLOT HISTO, and PLOT RAW_BAR statements.

- PLOT SORT_BAR sorts the bars in a bar chart or histogram in descending order.

- If you use the PLOT BAR_ASCENDING statement with the PLOT TITLE and PLOT NEXT_BAR statements, enter the statements in this sequence:

  1. PLOT BAR

  2. PLOT TITLE

  3. PLOT BAR_ASCENDING

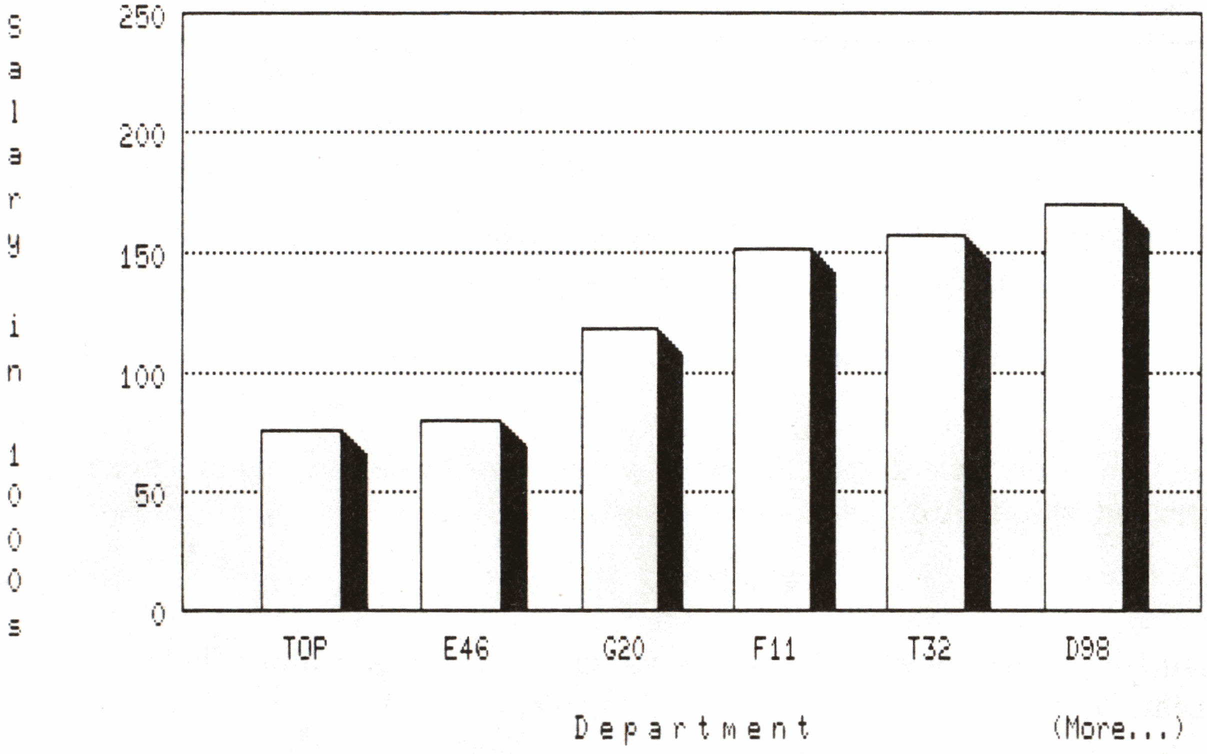
  4. PLOT NEXT_BAR

  If you enter the PLOT TITLE statement after PLOT BAR_ASCENDING, the sorting will not appear in the subsequent plots produced by PLOT NEXT_BAR.

Example

Using data from the PERSONNEL domain, use the PLOT BAR statement to create a bar chart showing the total salary for each department. In the statement, divide the salary by 1000 to eliminate the default scientific notation used for such large numbers.

```
DTR> PLOT BAR DEPT ("Department"),
CON> SALARY/1000 ("Salary in 1000s") OF PERSONNEL
```



ZK-00296-00

To reorder the bars in ascending value, enter the PLOT BAR_ASCENDING statement:

```
DTR> PLOT BAR_ASCENDING
```

4.5 PLOT BAR_AVERAGE Statement

Creates a simple bar chart showing the average value of a particular field or value expression.

Format

```
                                                  ┌    ┌ file-spec ┐ ┐
PLOT BAR_AVERAGE  [ALL]  arg,  arg  [OF rse]      │ ON ┤           ├ │
                                                  └    └ *.prompt  ┘ ┘
```

Arguments

arg

The PLOT BAR_AVERAGE statement requires two arguments. Arguments can be either field names or other value expressions:

- The first argument groups and labels the records for the horizontal (X) axis.
- The second argument specifies what values are to be averaged. These averaged values are plotted on the vertical (Y) axis.

Results

- The resulting chart shows the average value of each group identified on the horizontal (X) axis.
- The PLOT BAR_AVERAGE statement produces a bar chart of up to six bars. If there are more than six bars, DATATRIEVE breaks the data into separate charts, each of which contains a maximum of six bars. (More...) at the bottom right of the screen indicates additional charts. To display additional charts, type PLOT NEXT_BAR.

Usage Note

You can use the following utility plots with the PLOT BAR_AVERAGE statement:

- PLOT BAR_ASCENDING
- PLOT BIG
- PLOT HARDCOPY

- PLOT LIMITS_X
- PLOT LIMITS_Y
- PLOT NEXT_BAR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT SORT_BAR
- PLOT TITLE

See the section on each of these utility plot statements for more information.

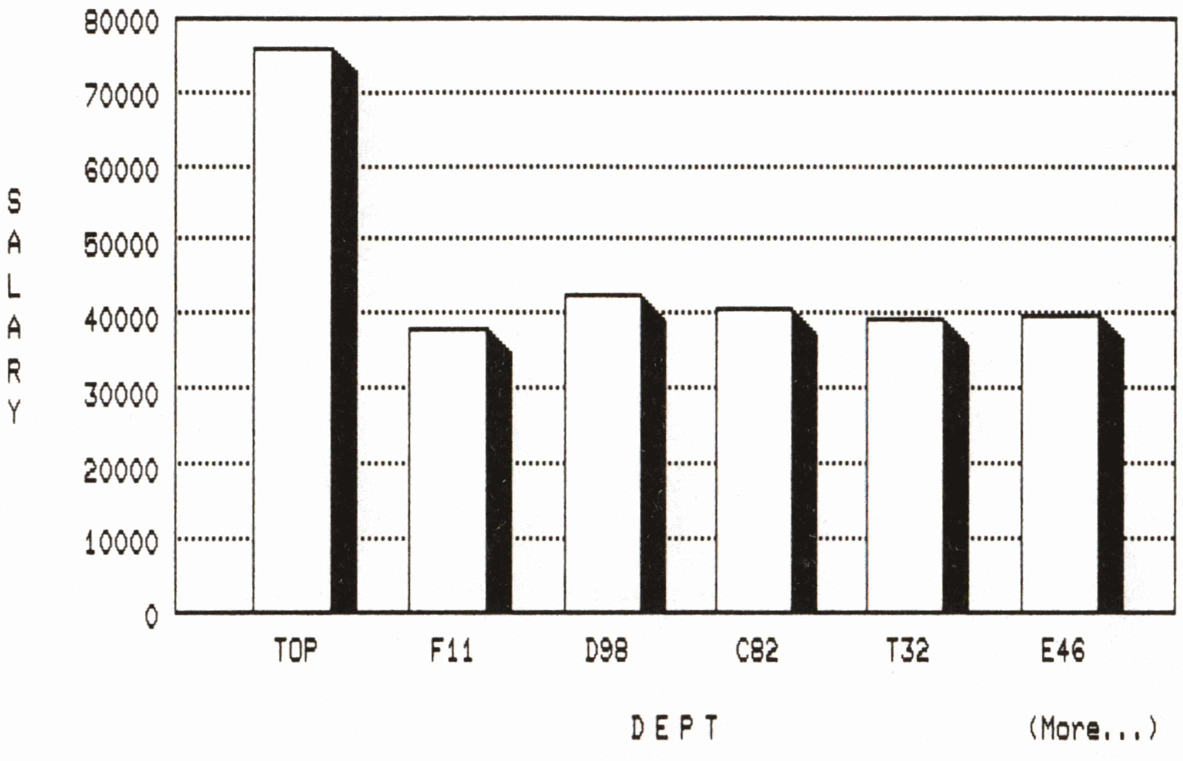
Examples

Example 1

Using the PERSONNEL domain, create a bar chart showing the average salary of each department.

```
DTR> PLOT BAR_AVERAGE ALL DEPT, SALARY OF PERSONNEL
```

Plot for Example 1



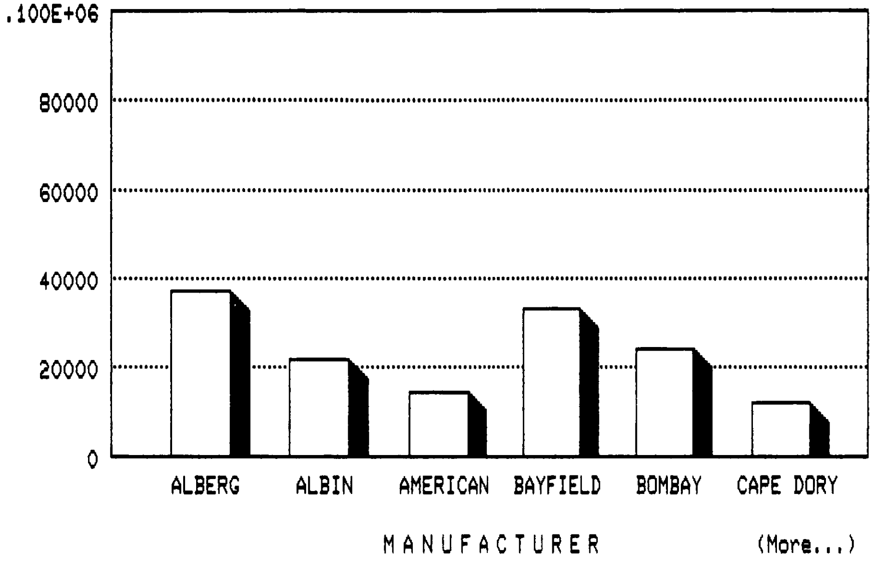
ZK-00297-00

Example 2

Create a chart showing the average price of each builder's yachts. This example uses the PLOT BAR_AVERAGE statement with an RSE. In this case, the keyword ALL is optional.

```
DTR> PLOT BAR_AVERAGE ALL BUILDER, PRICE OF
CON> YACHTS WITH PRICE NOT MISSING SORTED
CON> BY BUILDER
```

Plot for Example 2



ZK-00298-00

4.6 PLOT BIG Statement

Prints an enlarged version of your most recent plot on the graphics printer attached to your terminal.

Format

```
PLOT BIG
```

Argument

The PLOT BIG statement takes no argument.

Results

- The PLOT BIG statement prints an enlarged version of the most recent plot on the graphics printer attached to your terminal. This plot is four times the size of a plot printed with the PLOT HARDCOPY statement.
- DATATRIEVE prints a separate legend for each plot created with a PLOT BIG statement after it prints the plot. This occurs even if the plot already includes the legend.

Usage Notes

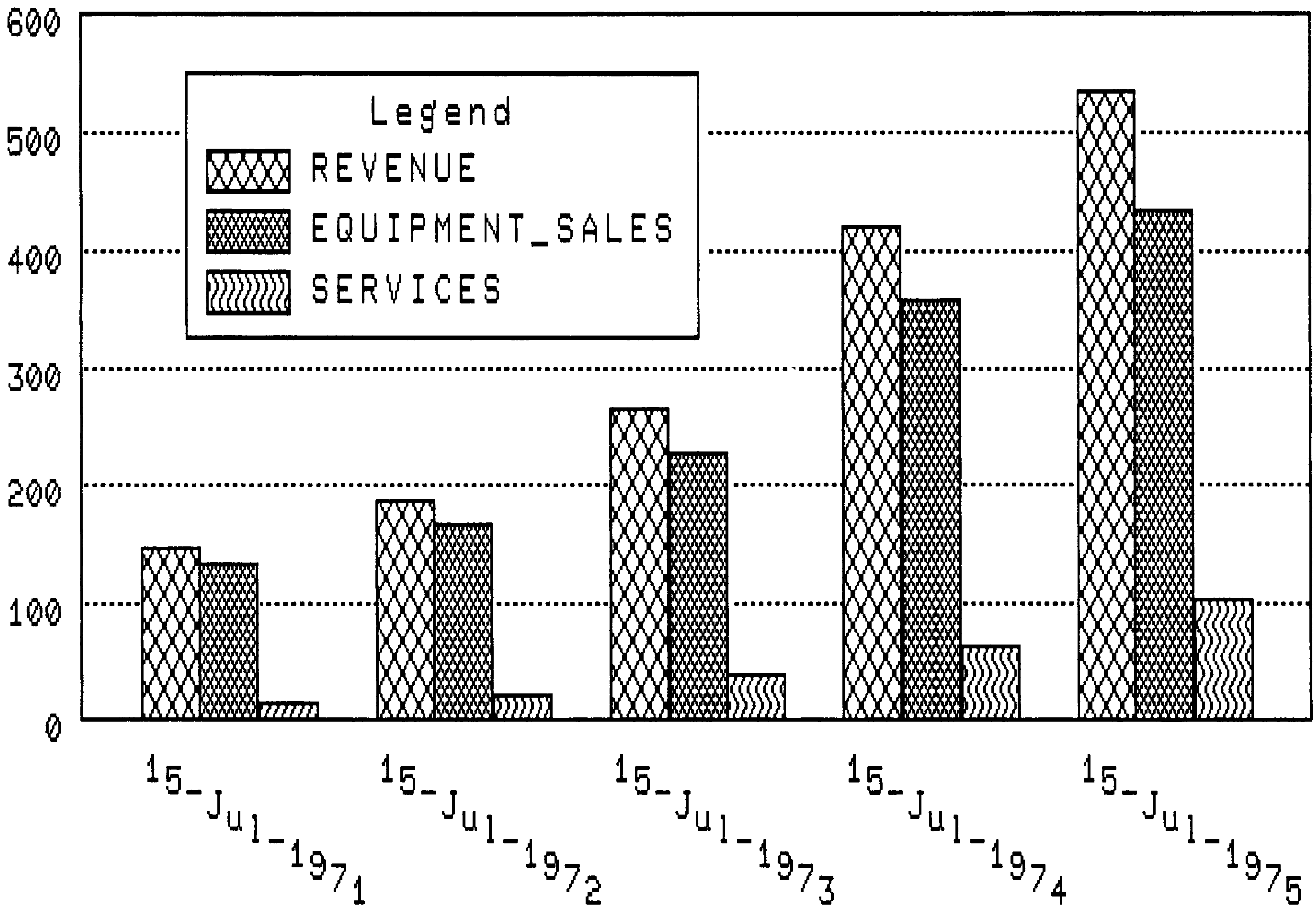
- If you plan to use a PLOT MULTI_BAR, PLOT MULTI_SHADE, or PLOT STACKED_BAR statement, then use the PLOT CROSS_HATCH statement before using the PLOT BIG statement. Otherwise, all plotted elements print out as solid black.
- If you are using a printer with connected pages of paper (not individual pages), PLOT BIG does not advance the paper after printing the plot. Before issuing another PLOT BIG statement, you should advance the paper feed of your printer.
- The PLOT BIG statement can be used only with a VT125 terminal connected to a compatible printer. You cannot use PLOT BIG with VT240 terminals. (To get large plots with VT240 terminals, change the graphics set-up Compressed Print option to Expanded Print and use the PLOT HARDCOPY statement.)
- Once you have entered the PLOT BIG statement, the terminal will not respond to any additional commands or statements until the printer is almost finished printing the plot.

Example

Use the PLOT MULTI_BAR statement to create a plot using information from the ANNUAL_REPORT domain. Add crosshatching to distinguish the various values being plotted.

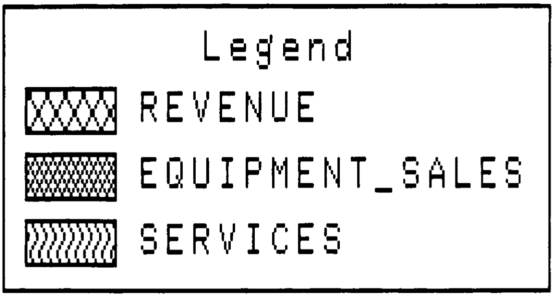
```
DTR> PLOT MULTI_BAR ALL DATE, REVENUE,
CON> EQUIPMENT_SALES, SERVICES OF
CON> FIRST 5 ANNUAL_REPORT SORTED BY DATE THEN
CON> PLOT CROSS_HATCH
DTR> PLOT BIG
```

The plot, showing the increase in total revenue and in the revenues from equipment sales and services over a five year period, is partially displayed on your screen. It is printed in its entirety on your graphics printer.


600
500
400
300
200
100
0
Legend
REVENUE
EQUIPMENT_SALES
SERVICES
15_Jul_1971
15_Jul_1972
15_Jul_1973
15_Jul_1974
15_Jul_1975

ZK-00299-00

PLOT BIG automatically prints out a separate copy of the legend, even if the plot already includes a legend.



ZK-00370-00

4.7 PLOT CONNECT Statement

Connects the points in the scattergraph that you have just plotted.

Format

```
PLOT CONNECT
```

Argument

The PLOT CONNECT statement takes no argument.

Results

- PLOT CONNECT takes the scattergraph you have just created and connects the points. Each pair of value expressions in the scattergraph is represented by a plus sign (+).
- PLOT CONNECT connects points in the order DATATRIEVE processed them.

Usage Notes

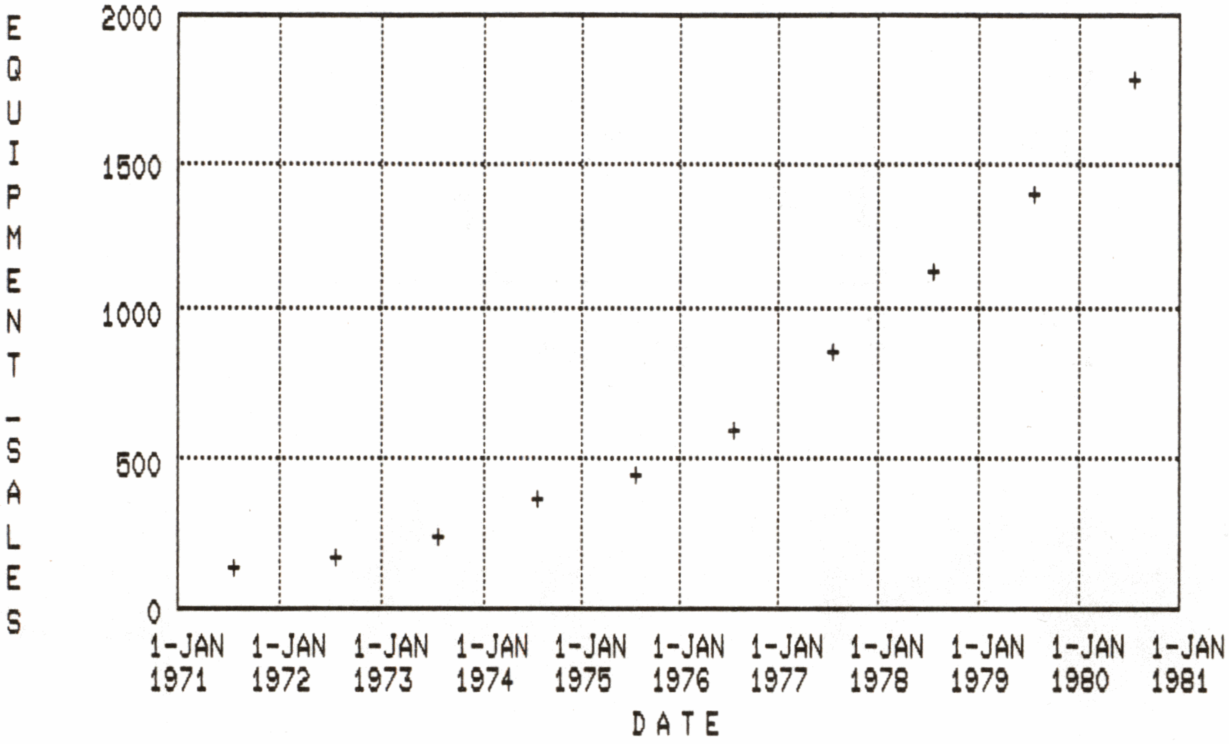
- You can use the PLOT CONNECT statement with the following plots:
  - PLOT DATE_LOGY
  - PLOT DATE_Y
  - PLOT LOGX_LOGY
  - PLOT LOGX_Y
  - PLOT X_LOGY
  - PLOT X_Y
- If you want to control the order in which the points are plotted, use a SORT statement or a SORTED BY expression in the RSE you use to form the collection of data to be plotted.
- Use the PLOT SHADE statement to shade the area below the points.

Example

1. Using the ANNUAL_REPORT domain, plot a scattergraph.

```
DTR> PLOT DATE_Y ALL DATE, EQUIPMENT_SALES OF ANNUAL_REPORT
```

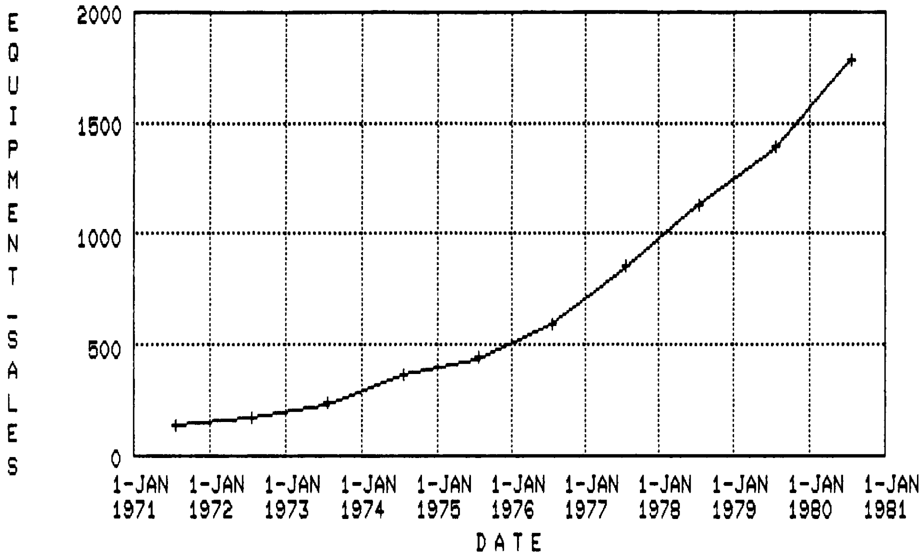
The following plot shows a scattergraph before the PLOT CONNECT statement:



ZK-00300-00

2. Enter the PLOT CONNECT statement to connect the points shown on the preceding graph. This emphasizes the trend, delineating the increase in equipment sales over time.

```
DTR> PLOT CONNECT
```


EQUIPMENT_SALES
2000
1500
1000
500
0
1-JAN 1971
1-JAN 1972
1-JAN 1973
1-JAN 1974
1-JAN 1975
1-JAN 1976
1-JAN 1977
1-JAN 1978
1-JAN 1979
1-JAN 1980
1-JAN 1981
DATE

ZK-00301-00

4.8 PLOT CROSS_HATCH Statement

Assigns a unique crosshatch pattern to each field name or other value expression plotted on the vertical (Y) axis of certain plots. Changes shades to crosshatched patterns if used following a shaded plot.

Format

```
PLOT CROSS_HATCH
```

Argument

The PLOT CROSS_HATCH statement takes no argument.

Results

- PLOT CROSS_HATCH changes each of the shading variations in multiple bar charts to a unique crosshatch pattern. The crosshatching enhances the distinction among the displays of different field names or other value expressions.
- The PLOT CROSS_HATCH statement assigns from one to three different crosshatch patterns to each of the arguments plotted against the vertical (Y) axis of a multiple bar chart. This plot statement is particularly useful with the following statements where distinguishing field names or other value expressions is important:
  - PLOT MULTI_BAR
  - PLOT MULTI_BAR_GROUP
  - PLOT MULTI_SHADE
  - PLOT STACKED_BAR
- The PLOT CROSS_HATCH statement is the only way to differentiate bar elements and shaded areas in hardcopy printouts of multiple bar charts.

Usage Note

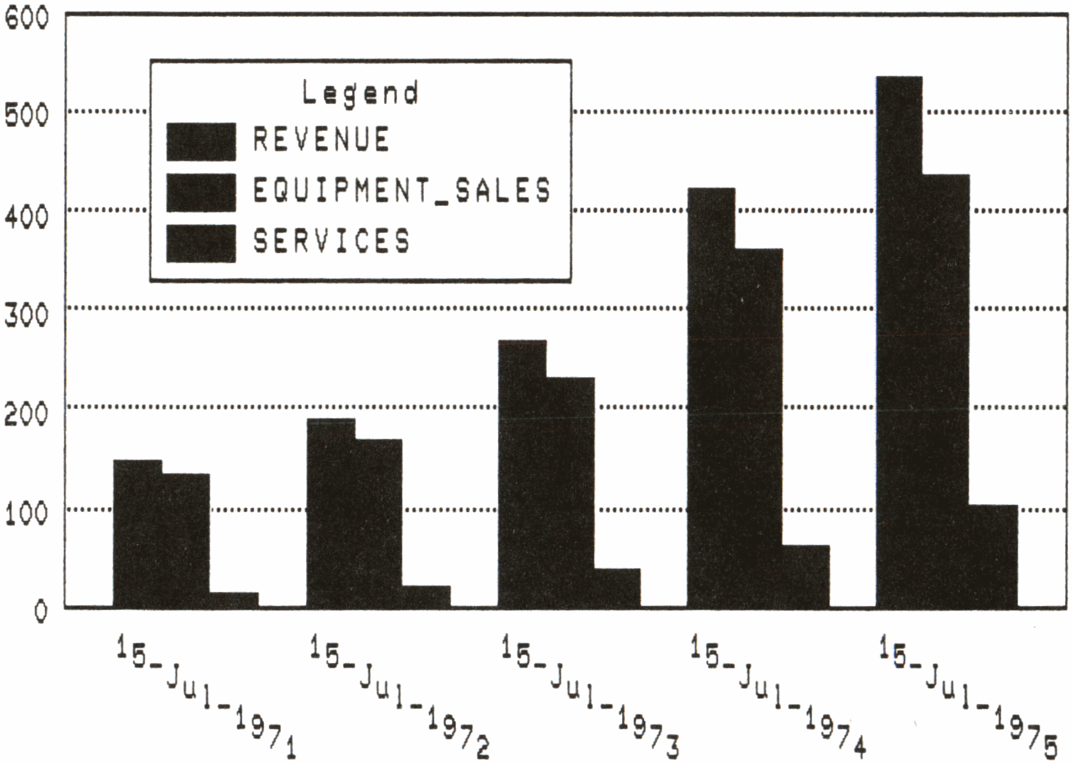
The PLOT CROSS_HATCH statement can assign no more than three pattern variations. It is therefore not appropriate for raw bar charts or pie charts, which can plot more than three elements.

Example

Create a collection from the ANNUAL_REPORT domain. Then create a multiple bar chart using the PLOT MULTI_BAR statement. If you display the chart on your monitor, each of the bars in a cluster of bars is distinguished by shading or color variations. If you print the chart on the graphics printer attached to your terminal, shading or color variations are lost.

```
DTR> FIND FIRST 5 ANNUAL_REPORT SORTED BY DATE
DTR> PLOT MULTI_BAR ALL DATE, REVENUE,
CON> EQUIPMENT_SALES, SERVICES THEN
CON> PLOT HARDCOPY
```

The following plot represents a hardcopy printout. Note how shading gradations are indistinguishable.

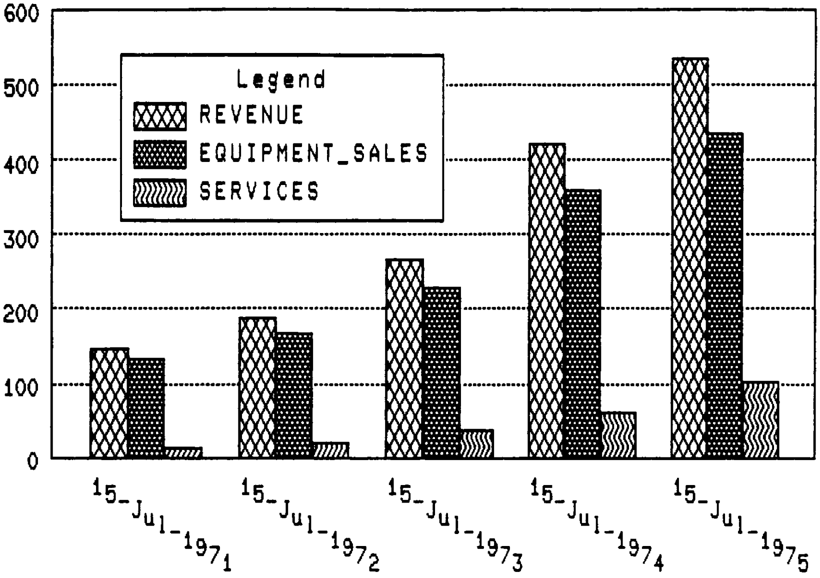


ZK-00302-00

To clearly define each bar in the group, use PLOT CROSS_HATCH to replace the shaded areas with unique crosshatch patterns.

```
DTR> PLOT CROSS_HATCH THEN
CON> PLOT HARDCOPY
```

The new hardcopy printout looks like this.



ZK-00303-00

Note that PLOT HARDCOPY produces the legend on a second page, even if it appears on the first page with the plot.

4.9 PLOT DATE_LOGY Statement

Creates a time-related scattergraph using a logarithmic scale for the vertical (Y) axis.

Format

```
PLOT DATE_LOGY  [USING]  [ALL]  arg,  arg  [OF rse]  [ON { file-spec } ]
                                                         { *.prompt  }
```

Arguments

arg

The PLOT DATE_LOGY statement takes two arguments. The arguments can be field names or other value expressions:

- The first argument is plotted on the horizontal (X) axis. It must contain a date value.
- The second argument is plotted on the vertical (Y) axis. It must contain a value expression greater than zero for each record in the record stream. DATATRIEVE uses a logarithmic scale to plot these values on the vertical (Y) axis.

Results

- The PLOT DATE_LOGY statement creates a scattergraph using date values on the horizontal (X) axis and a logarithmic scale on the vertical (Y) axis. Each pair of value expressions in the scattergraph is represented by a plus sign (+).
- With logarithmic scaling, each successive unit of length on the scaled axis covers a wider range of numbers. This helps to make trends more apparent.
- PLOT DATE_LOGY sorts the records in ascending order by date before plotting them.

Usage Notes

You can use the following utility plots with the PLOT DATE_LOGY statement:

- PLOT BIG
- PLOT CONNECT
- PLOT HARDCOPY
- PLOT LIMITS_X
- PLOT LIMITS_Y
- PLOT LR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT SHADE
- PLOT TITLE

See the section on each of these utility plot statements for more information.

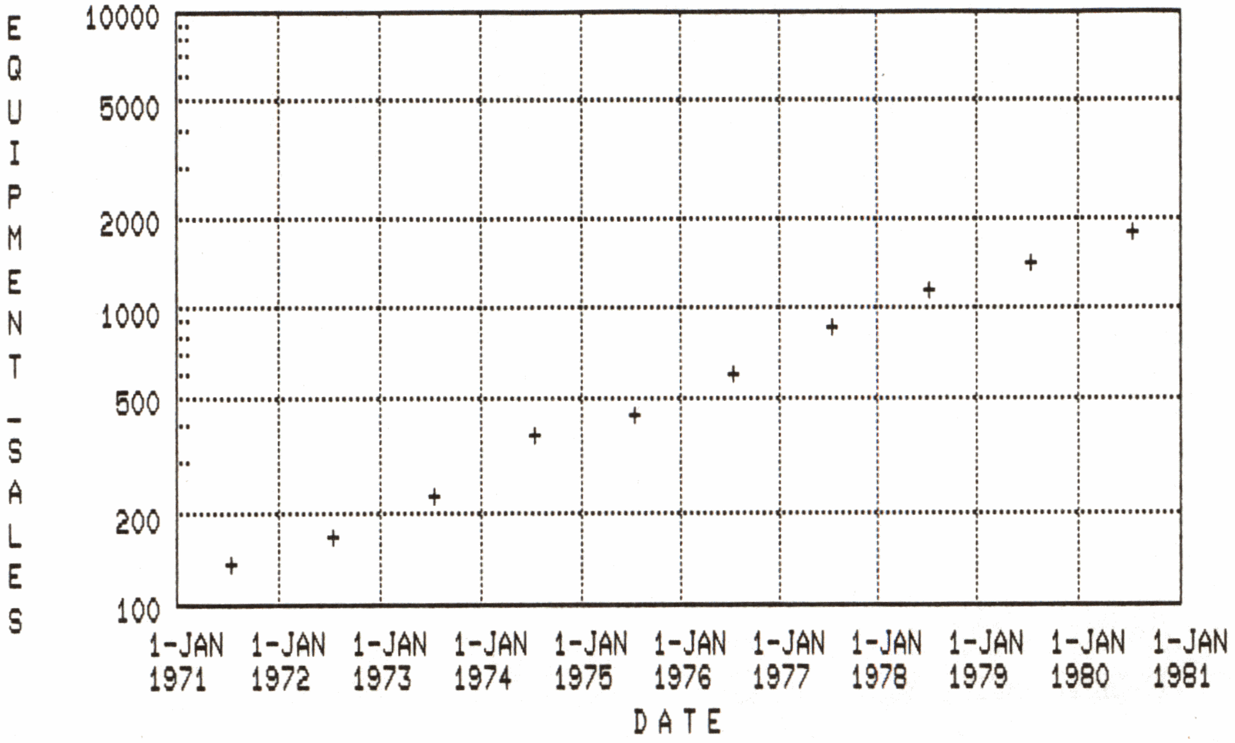
Examples

Example 1

Using the ANNUAL_REPORT domain and the PLOT DATE_LOGY statement, create a scattergraph that indicates the trends in equipment sales over the years.

```
DTR> PLOT DATE_LOGY ALL DATE, EQUIPMENT_SALES OF ANNUAL_REPORT
```

Plot for Example 1



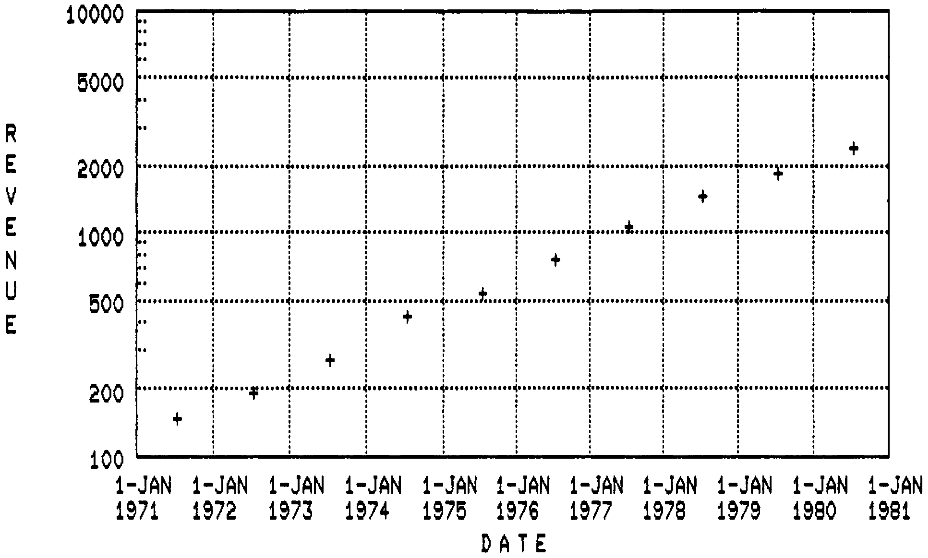
ZK-00304-00

Example 2

Use the PLOT DATE_LOGY statement with an RSE to create a scattergraph similar to the scattergraph used in Example 1. Example 2 uses the DATE and REVENUE fields from ANNUAL_REPORT to show the trend that total revenue has increased in the period from 1971 to 1980. In this case the keyword ALL is optional because the RSE is specified in the OF clause.

```
DTR> PLOT DATE_LOGY ALL DATE, REVENUE OF ANNUAL_REPORT
```

Plot for Example 2



ZK-00305-00

4.10 PLOT DATE_Y Statement

Creates a scattergraph that shows a chronological trend.

Format

```
                                                 ┌    ┌ file-spec ┐ ┐
PLOT DATE_Y  [USING]  [ALL]  arg,  arg  [OF rse] │ON  ┤           ├ │
                                                 └    └ *.prompt  ┘ ┘
```

Arguments

arg

The PLOT DATE_Y statement takes two arguments. The arguments can be field names or other value expressions:

- The first argument, which is plotted on the horizontal (X) axis, must contain a date value.
- The second argument, which is plotted on the vertical (Y) axis, must contain a value expression greater than zero for each record in the record stream or collection.

Results

- The PLOT DATE_Y statement creates a scattergraph that shows a chronological trend. Each pair of value expressions in the scattergraph is represented by a plus sign (+).
- PLOT DATE_Y sorts the records by date before plotting them.

Usage Note

You can use the following utility plots with the PLOT DATE_Y statement:

- PLOT BIG
- PLOT CONNECT
- PLOT HARDCOPY
- PLOT LIMITS_X
- PLOT LIMITS_Y

- PLOT LR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT SHADE
- PLOT TITLE

See the section on each of these utility plot statements for more information.

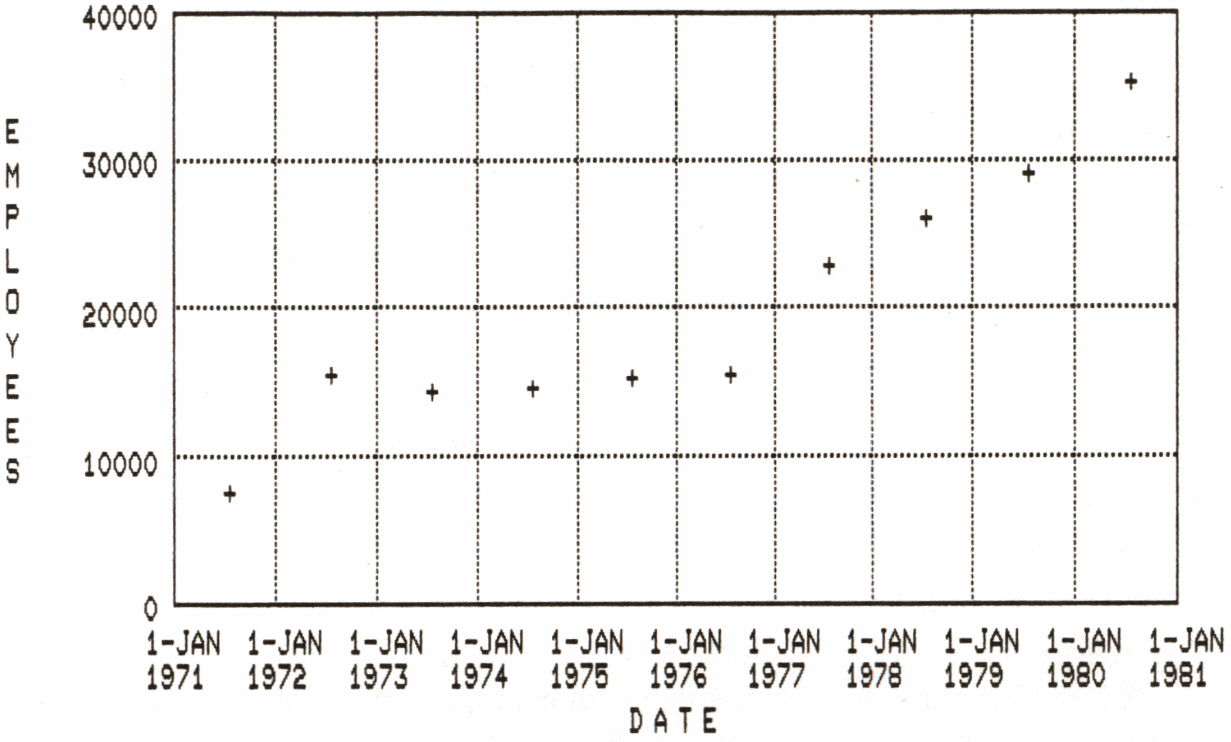
Examples

Example 1

Create a scattergraph using the PLOT DATE_Y statement and the ANNUAL_REPORT domain. The resulting plot emphasizes the trend that the number of employees has increased over a period of time.

```
DTR> PLOT DATE_Y ALL DATE, EMPLOYEES OF
CON> ANNUAL_REPORT WITH EMPLOYEES GT 0
```

Plot for Example 1



ZK-00306-00

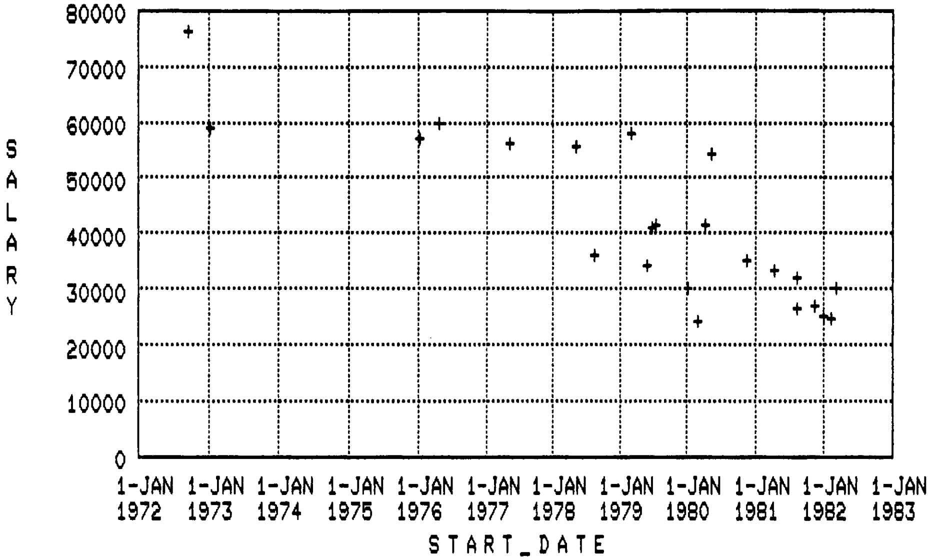
Example 2

Use the PLOT DATE_Y statement with an RSE to create a scattergraph using the START_DATE and SALARY fields from PERSONNEL. The resulting plot emphasizes the trend that employees who have been with the company for a longer period of time have higher salaries.

In this case the keyword ALL is optional because the RSE is specified in the OF clause.

```
DTR> PLOT DATE_Y ALL START_DATE,
CON> SALARY OF PERSONNEL WITH SALARY GT 0
```

Plot for Example 2



ZK-00307-00

4.11 PLOT HARDCOPY Statement

Sends the most recent plot you have created to the printer attached to your terminal.

Format

```
PLOT HARDCOPY
```

Argument

The PLOT HARDCOPY statement takes no argument.

Results

- The PLOT HARDCOPY statement produces a hardcopy printout of the most recently generated plot.
- PLOT HARDCOPY prints a separate legend for the plot after it prints the plot. This occurs even if the plot already includes the legend as output.

Usage Notes

- Your terminal does not respond to any commands or statements until the printer is almost finished printing the plot.
- If you are using a printer with connected pages of paper (not individual pages), PLOT HARDCOPY does not advance the paper after printing the plot. Before issuing another PLOT HARDCOPY statement, you should advance the paper feed of your printer.
- If you want to use PLOT HARDCOPY with one of the following plots, you should use PLOT CROSS_HATCH before issuing the PLOT HARDCOPY statement:
  - PLOT MULTI_BAR
  - PLOT MULTI_BAR_GROUP

  - PLOT MULTI_SHADE
  - PLOT STACKED_BAR

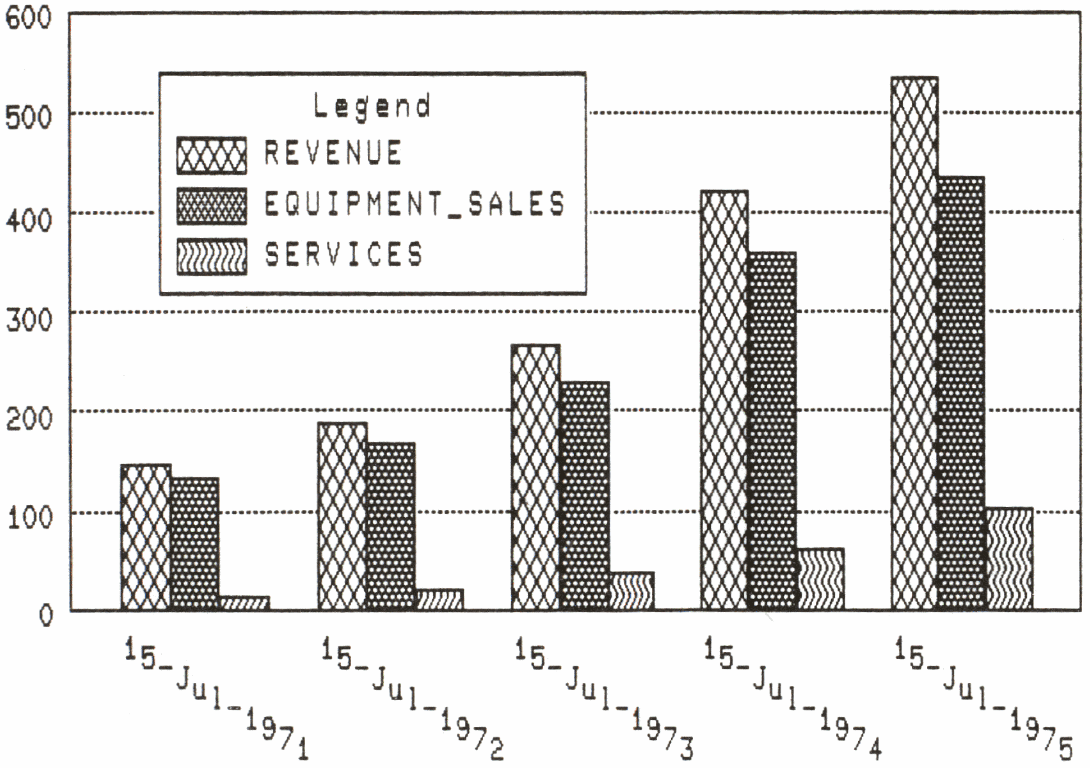
  PLOT CROSS_HATCH differentiates bars or shaded areas that are otherwise indistinguishable when printed.

- PLOT BIG produces a hardcopy printout with a plot four times the size of the plot produced using PLOT HARDCOPY.

Example

Create a multiple bar chart using data from the ANNUAL_REPORT domain. Note that on the terminal screen, the bars are differentiated by varying shades. This differentiation is not present in a hardcopy of the plot. Next, use the PLOT CROSS_HATCH statement to differentiate the bars for the hardcopy printout.

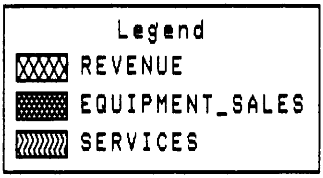
```
DTR> FIND FIRST 5 ANNUAL_REPORT SORTED BY DATE
DTR> PLOT MULTI_BAR ALL DATE, REVENUE,
CON> EQUIPMENT_SALES, SERVICES THEN
CON> PLOT CROSS_HATCH
DTR> PLOT HARDCOPY
```



ZK-00303-00

PLOT HARDCOPY

PLOT HARDCOPY automatically prints out a separate copy of the legend, even if the plot already includes a legend.



ZK-00308-00

4.12 PLOT HISTO Statement

Creates a simple histogram showing the frequency distribution (frequency of occurrence) of values in a given field or value expression.

Format

```
PLOT HISTO  [ALL]  arg  [OF rse]  [ ON { file-spec } ]
                                       { *.prompt  }
```

Argument

arg

The PLOT HISTO statement takes one argument. That argument can be a field name or other value expression. The argument provides the values for the horizontal (X) axis and groups the records according to the value or field specified by the argument.

Results

- PLOT HISTO produces a histogram showing the frequency distribution of values in a given field or value expression. The horizontal (X) axis identifies the different values found in the field or value expression for a group of records. The vertical (Y) axis shows the number of records that contain the value.

- If there are more than six bars in the plot, DATATRIEVE breaks the data into charts that contain a maximum of six bars. To indicate additional charts, DATATRIEVE prints (More...) at the bottom right of the screen. Type PLOT NEXT_BAR to see the next section of the chart.

Usage Note

You can use the following utility plots with the PLOT HISTO statement:

- PLOT BAR_ASCENDING

- PLOT BIG

- PLOT HARDCOPY

- PLOT LIMITS_X

- PLOT LIMITS_Y
- PLOT NEXT_BAR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT SORT_BAR
- PLOT TITLE

See the section on each of these utility plot statements for more information.

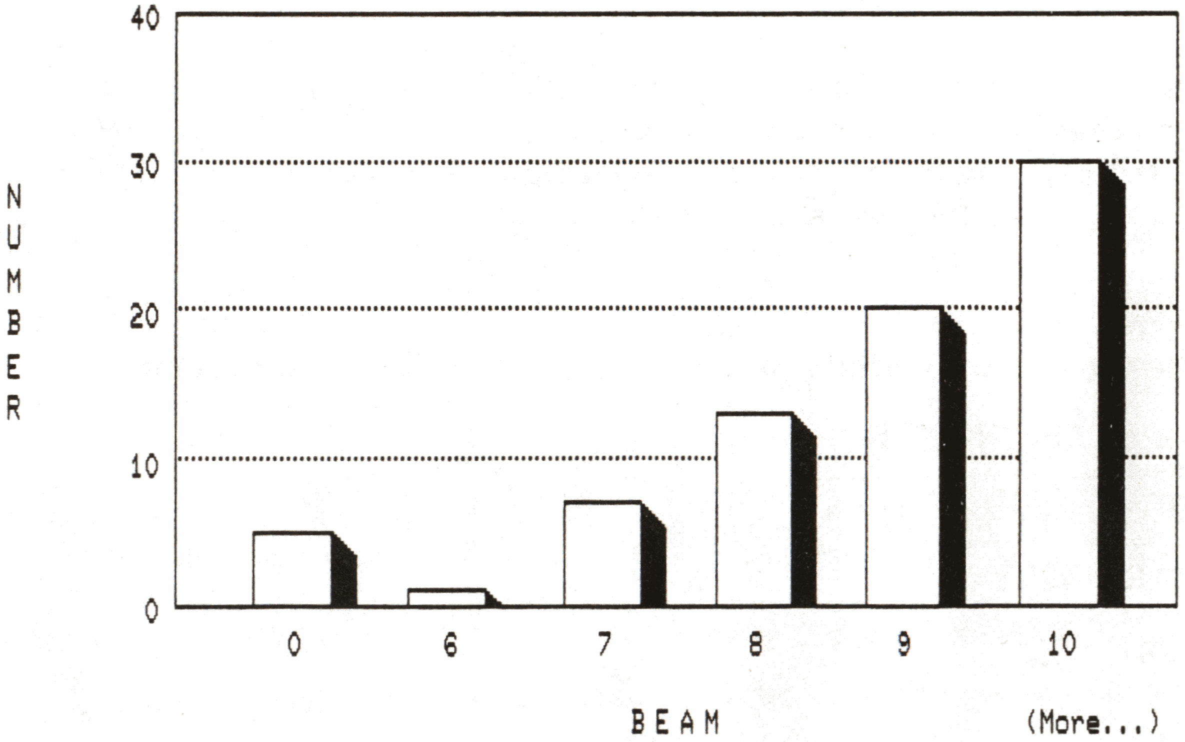
Examples

Example 1

Use data from the YACHTS domain to plot a histogram showing the total number of yachts for each beam size:

```
DTR> PLOT HISTO BEAM OF YACHTS SORTED BY BEAM
```

Plot for Example 1



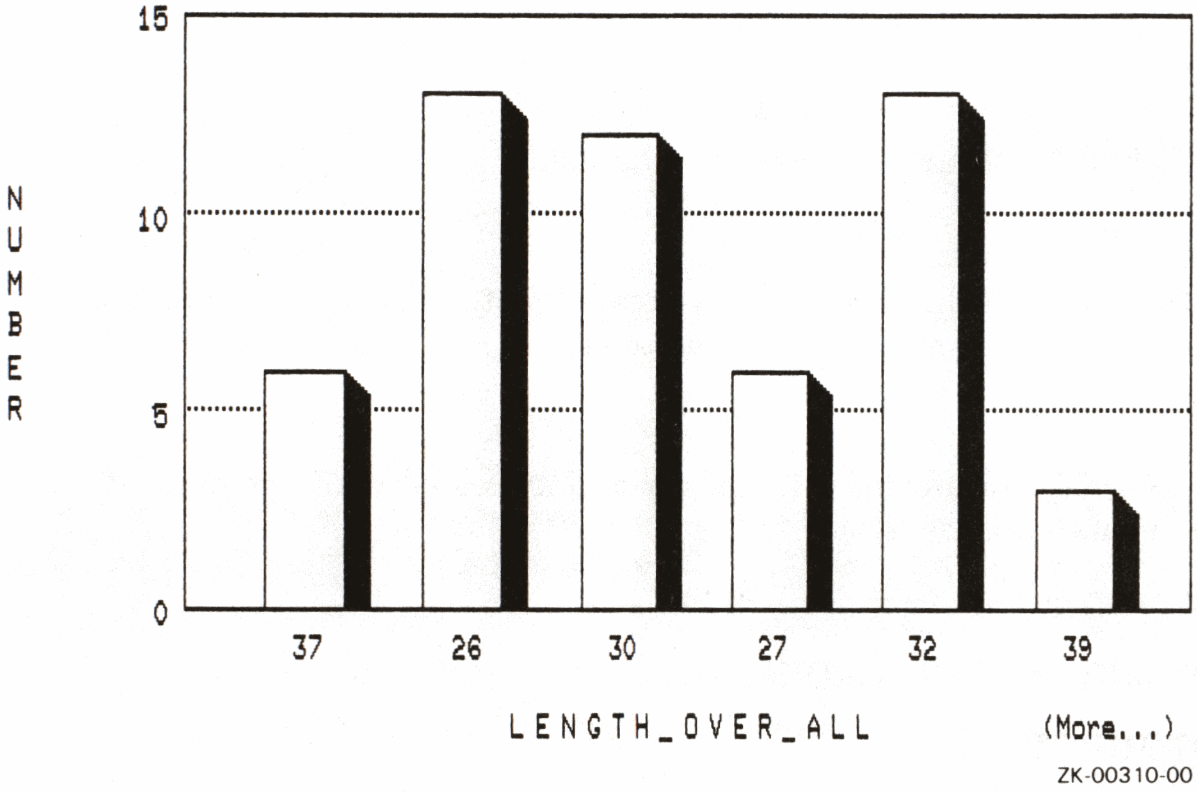
ZK-00309-00

Example 2

Plot a histogram showing the total number of yachts in each LOA category using the YACHTS domain.

```
DTR> PLOT HISTO ALL LOA OF YACHTS
```

Plot for Example 2



4.13 PLOT LEGEND Statement

Displays the legend for certain plots that do not have enough room for the legend.

Format

```
PLOT LEGEND
```

Argument

The PLOT LEGEND statement takes no argument.

Result

PLOT LEGEND takes the most recently created plot, creates a legend for the plot, and displays the legend on a cleared screen.

Usage Notes

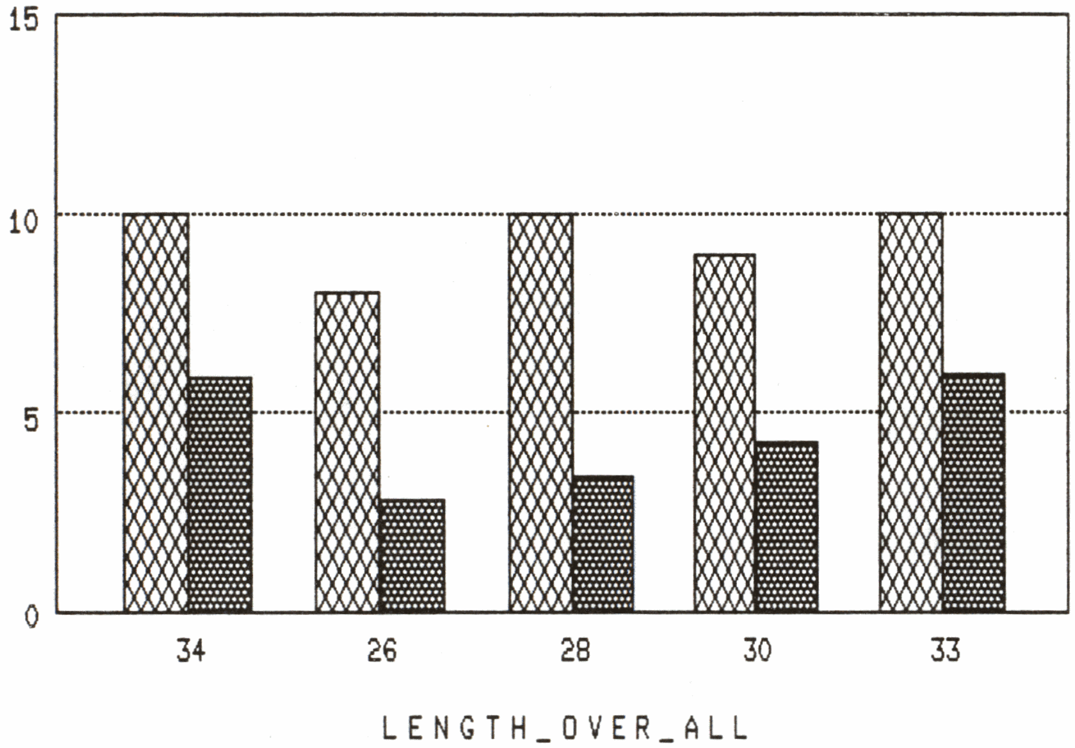
- PLOT LEGEND can provide a legend for a plot created with one of the following statements:
  - PLOT MULTI_BAR
  - PLOT MULTI_BAR_GROUP
  - PLOT MULTI_LINE
  - PLOT MULTI_LR
  - PLOT MULTI_SHADE
  - PLOT STACKED_BAR
- The PLOT BIG and PLOT HARDCOPY statements automatically send the plot and a separate legend to the printer as output.
- If you are creating a multiple bar, multishade, or stacked bar plot, use the PLOT CROSS_HATCH statement to provide unique crosshatch patterns to differentiate the elements in each plot. The crosshatch patterns appear in the legend, further enhancing the distinction among the plotted elements.

Example

Create a collection from the YACHTS domain consisting of boats by the builder Grampian. Then create a multiple bar chart showing the total displacement and beam size for all yachts in the collection, grouped by LOA value. Use PLOT CROSS_HATCH to highlight the different bars.

```
DTR> FIND YACHTS WITH BUILDER = "GRAMPIAN"
DTR> PLOT MULTI_BAR ALL LOA, BEAM,
CON> DISP/2000 ("Weight in Tons") THEN
CON> PLOT CROSS_HATCH
```

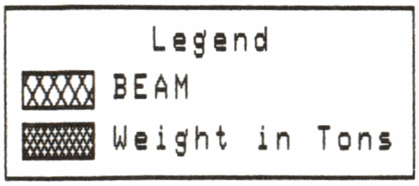
The following plot is shown without the legend because the plot does not allow enough room for the legend.



ZK-00311-00

When you issue a PLOT LEGEND statement, DATATRIEVE clears the screen to display the legend for the plot:

```
DTR> PLOT LEGEND
```



```
        Legend
▨ BEAM
▩ Weight in Tons
```

ZK-00312-00

4.14 PLOT LIMITS_X Statement

Lets you specify the minimum and maximum range limits of the data to be plotted on the horizontal (X) axis.

Format

```
                ┌ low-value,   high value ┐
PLOT LIMITS_X   ┤ low-value               ├
                └ ,high value             ┘
```

Arguments

low-value

Is the value expression, based on the values on the horizontal (X) axis of the base plot, that determines the lowest value of the data to be plotted.

high-value

Is the value expression, based on the values on the horizontal (X) axis of the base plot, that determines the highest value of the data to be plotted.

If you use zero (0) as either your low or high value, PLOT LIMITS_X returns the value plotted in the original base plot.

If you omit either argument, PLOT LIMITS_X uses the corresponding argument from the previous PLOT LIMITS_X statement.

Result

DATATRIEVE replots the most recent plot using a subset of the original data. This subset is determined by the low and high values specified in the plot statement.

Usage Notes

- By default, DATATRIEVE bases the scaling of plot values on the total range of the data values provided as input for the plot.
- If no data values fall within the limits you specify, DATATRIEVE creates a plot with no values plotted.

- The minimum and maximum values specified set the limits for filtering out the data used in the base plot. The range of values DATATRIEVE replots on the X axis reflects the actual high and low values of the data that get through the filter, not the high and low limits specified as parameters.

- You can use PLOT LIMITS_X repeatedly to vary the subset of data DATATRIEVE uses to replot the previous plot.

- The following examples show the syntax variations common to the PLOT LIMITS_X statement. Comments indicate the results of each PLOT LIMITS_X statement.

```
DTR> FIND YACHTS WITH PRICE NE 0
DTR> PLOT X_Y ALL LOA, PRICE
DTR> !
DTR> ! Original range limits for this example are 15 and 45.
DTR> ! Examples are cumulative.
DTR> !
DTR> PLOT LIMITS_X 20, 40
DTR> ! DATATRIEVE ignores LOA values less than 20 and greater
DTR> ! than 40 when it replots the base plot data.
DTR> !
DTR> PLOT LIMITS_X 25
DTR> ! The minimum LOA value for the data plotted is 25;
DTR> ! the maximum value stays the same (40).
DTR> !
DTR> PLOT LIMITS_X ,35
DTR> ! The minimum LOA value for the data plotted stays the
DTR> ! same (25); the maximum value is 35.
DTR> !
DTR> PLOT LIMITS_X
DTR> ! Both minimum and maximum values remain the same
DTR> ! (25, 35).
DTR> !
DTR> PLOT LIMITS_X 0,35
DTR> ! The minimum LOA value returns to the minimum value of
DTR> ! the base plot (15).  The maximum value remains 35.
DTR> !
DTR> PLOT LIMITS_X 0,0
DTR> ! Both minimum and maximum values are returned to the
DTR> ! values plotted by the original PLOT X_Y statement
DTR> ! (15,45).
```

Restrictions

- You cannot use PLOT LIMITS_X with pie charts.

- DATATRIEVE ignores date values unless they are expressed as a year both in the base plot and in the PLOT LIMITS_X statement. For example, you can use YYYY for a date field, but not DD_MMM_YYYY.

- You cannot use scientific notation for the low and high values.

- You must use PLOT LIMITS_X before you use any other utility plot statements or you will lose the characteristics added with those plot statements. PLOT LIMITS_X replots the previous plot using the data limits you specify. Any utility plots used to enhance the old base plot are ignored. The only exceptions are PLOT TITLE and PLOT LIMITS_Y. If you use PLOT LIMITS_X after PLOT TITLE, DATATRIEVE retains the title specified in the PLOT TITLE statement. If you use PLOT LIMITS_X after PLOT LIMITS_Y, the line specified in the PLOT LIMITS_Y statement remains in the final plot.

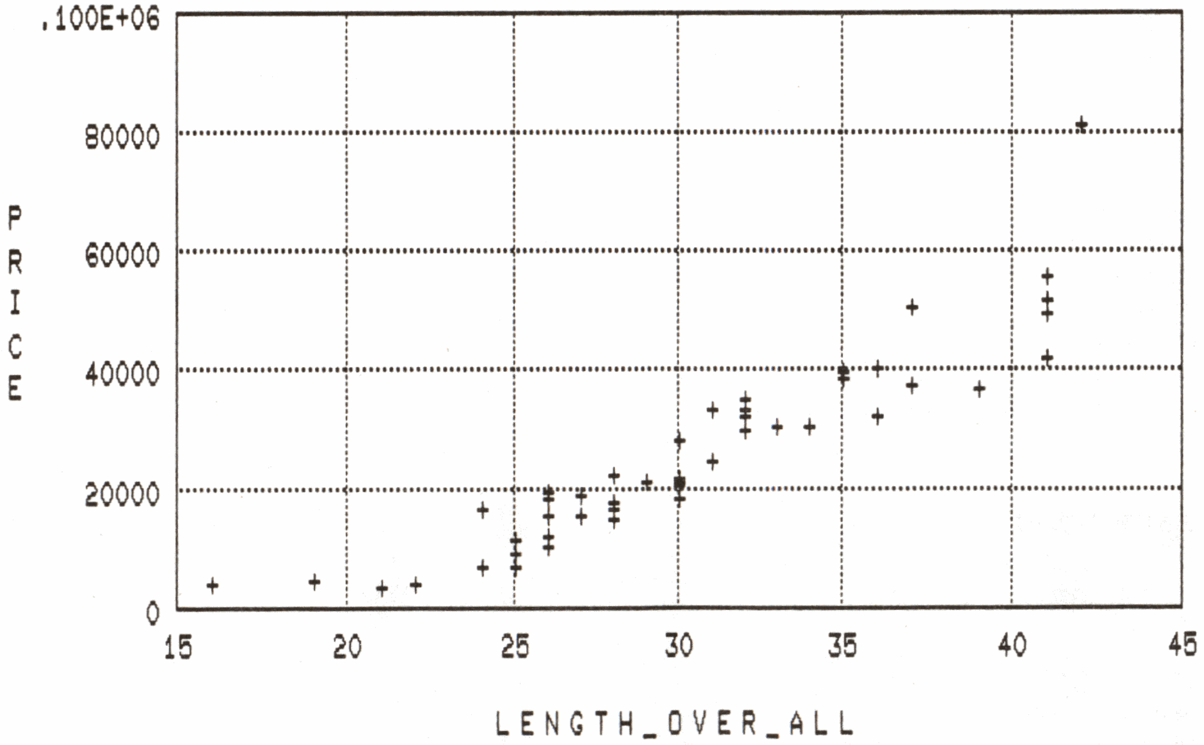
Examples

Example 1

1. Use the PLOT X_Y statement to create a chart showing the trend that as LOA of YACHTS increases, so does price.

```
DTR> PLOT X_Y ALL LOA, PRICE OF
CON> YACHTS WITH PRICE NE 0
```

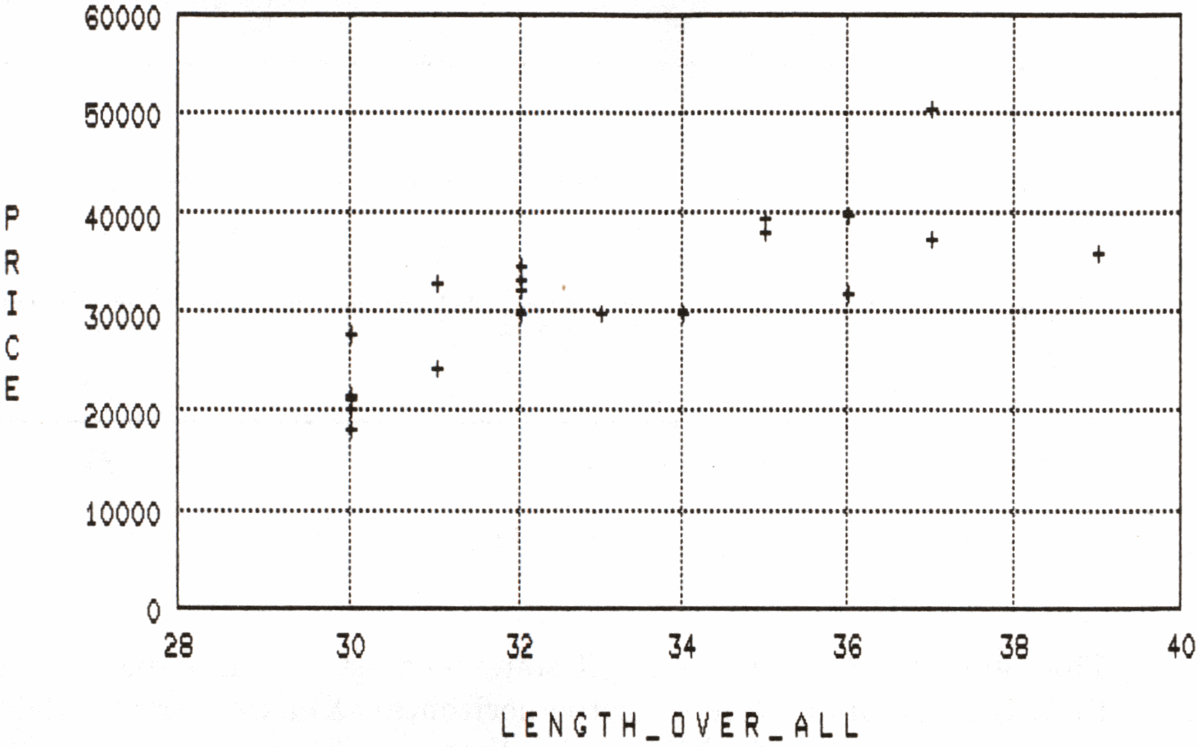
The resulting plot includes all values found in the RSE for the horizontal (X) axis.



ZK-00313-00

2. Then use the PLOT LIMITS_X statement to further limit the range of values plotted on the horizontal (X) axis.

```
DTR> PLOT LIMITS_X 30, 40
```



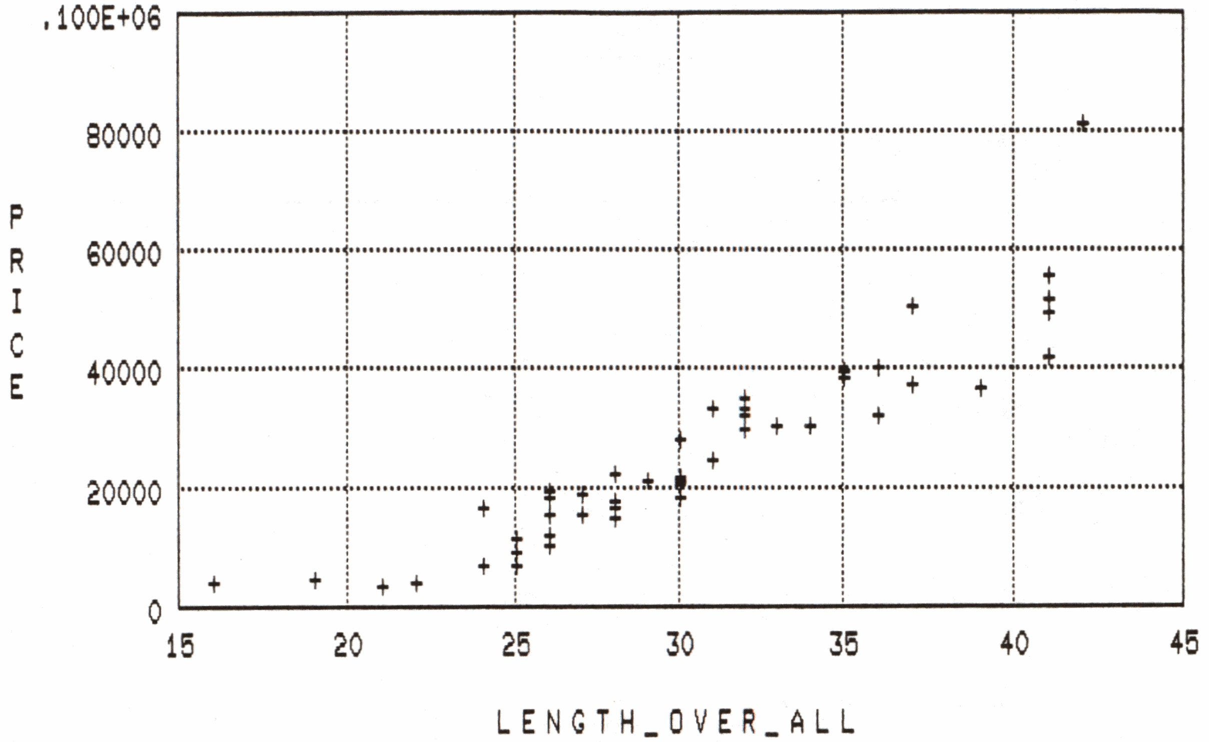
ZK-00314-00

Example 2

1. You can also use a value expression as an argument for PLOT LIMITS_X. The following example uses the same base plot as the preceding example:

```
DTR> PLOT X_Y ALL LOA, PRICE OF
CON> YACHTS WITH PRICE NE 0
```

The resulting plot includes all values found in the RSE for the horizontal (X) axis.

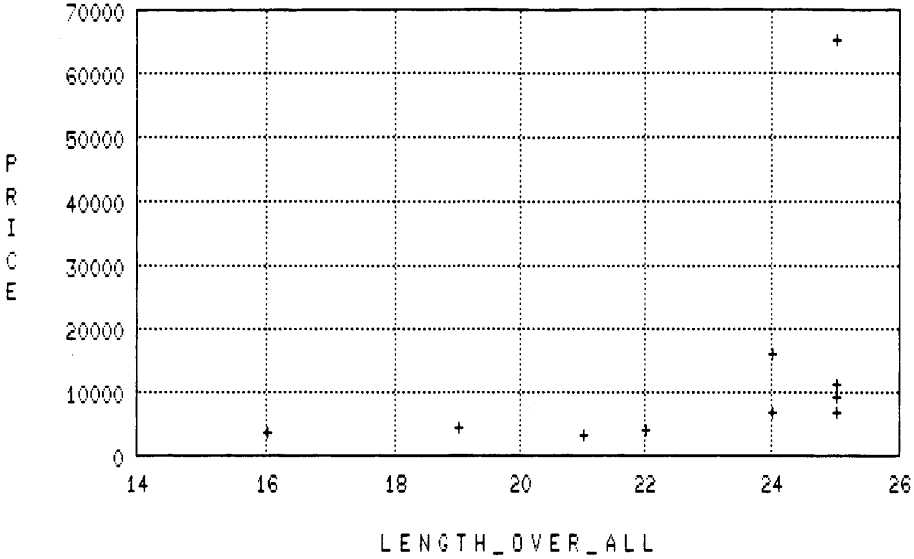


ZK-00313-00

2. The following PLOT LIMITS_X statement uses a value expression to limit the maximum values for the horizontal (X) axis. First, it determines the average length of all the yachts that cost less than $20,000. Because that average length is 25 feet, PLOT LIMITS_X limits the display of highest X values at 25 feet in length.

   Note that the resulting plot includes some yachts that cost more than $20,000. This is predictable because PLOT LIMITS_X limits only the display of the horizontal (X) axis, not the vertical (Y) axis. (The second example in the PLOT LIMITS_Y article shows this data with limits on the vertical (Y) axis.)

```
DTR> PLOT LIMITS_X 0,
CON> AVERAGE LOA OF YACHTS WITH PRICE LT 20000 AND
CON> PRICE NE 0
```


70000
60000
50000
40000
30000
20000
10000
0
PRICE
14
16
18
20
22
24
26
LENGTH_OVER_ALL

ZK-00347-00

4.15 PLOT LIMITS_Y Statement

Lets you specify the minimum and maximum range limits of the data to be plotted on the vertical (Y) axis.

Format

```
                 { low-value,   high value }
PLOT LIMITS_Y    { low-value               }
                 { ,high value             }
```

Arguments

low-value

Is the numeric value, based on the values on the vertical (Y) axis of the base plot, that determines the lowest value of the data to be plotted.

high-value

Is the numeric value, based on the values on the vertical (Y) axis of the base plot, that determines the highest value of the data to be plotted.

If you use zero (0) as either the low or high value, PLOT LIMITS_X will return the value plotted in the original base plot. If you do not provide a low or high value, the existing values stay the same.

Result

DATATRIEVE replots the most recent plot using a subset of the original data. This subset is determined by the low and high values specified in the plot statement.

Usage Notes

- By default, DATATRIEVE bases the scaling of plot values on the total range of the data values provided as input for the plot.
- If no data values fall within the limits you specify, DATATRIEVE creates a plot with no values plotted.

- The minimum and maximum values specified set the limits for filtering out the data used in the base plot. The range of values DATATRIEVE replots on the vertical (Y) axis reflects the actual high and low values of the data that gets through the filter, not the high and low limits specified as parameters.
- You can use PLOT LIMITS_Y repeatedly to vary the subset of data DATATRIEVE uses to replot the previous plot.
- You can use statistical value expressions to provide the low and high values.
- The following examples show the syntax variations common to the PLOT LIMITS_Y statement. Comments indicate the results of each PLOT LIMITS_Y statement.

```
DTR> FIND YACHTS WITH PRICE NE 0
DTR> PLOT X_Y ALL LOA, PRICE
DTR> !
DTR> ! Original range limits for this example are 0 and
DTR> ! 100000.
DTR> ! Examples are cumulative.
DTR> !
DTR> PLOT LIMITS_Y 10000,50000
DTR> ! DATATRIEVE ignores LOA values less than 10000 and
DTR> ! greater than 40000 when it replots the base plot data.
DTR> !
DTR> PLOT LIMITS_Y 15000
DTR> ! The minimum LOA value for the data plotted is 15000;
DTR> ! the maximum value stays the same (50000).
DTR> !
DTR> PLOT LIMITS_Y ,40000
DTR> ! The minimum LOA value for the data plotted stays the
DTR> ! same (15000); the maximum value is 40000.
DTR> !
DTR> PLOT LIMITS_Y 20000,0
DTR> ! The minimum LOA value is 20000; the maximum value
DTR> ! returns to the original value of the base plot
DTR> ! (100000).
DTR> !
DTR> PLOT LIMITS_Y 0,0
DTR> ! Both minimum and maximum values are returned to the
DTR> ! values plotted by the original PLOT X_Y statement
DTR> ! (0, 100000).
```

Restrictions

- You cannot use PLOT LIMITS_Y with pie charts.
- DATATRIEVE ignores date values unless they are expressed as a year both in the base plot and in the PLOT LIMITS_Y statement. For example, you can use YYYY for a date field, but not DD_MMM_YYYY.

- You cannot use scientific notation for the low and high values.
- You must use PLOT LIMITS_Y before you use any other utility plot statements or you will lose the characteristics added with those plot statements. DATATRIEVE replots the previous plot to use the data limits you specify. Any utility plots used to enhance the old base plot are ignored. The only exceptions are PLOT TITLE and PLOT LIMITS_X. If you use PLOT LIMITS_Y after PLOT TITLE, DATATRIEVE retains the title you specified in the PLOT TITLE statement. If you use PLOT LIMITS_Y after PLOT LIMITS_X, the line specified in the PLOT LIMITS_X statement remains in the final plot.

Examples

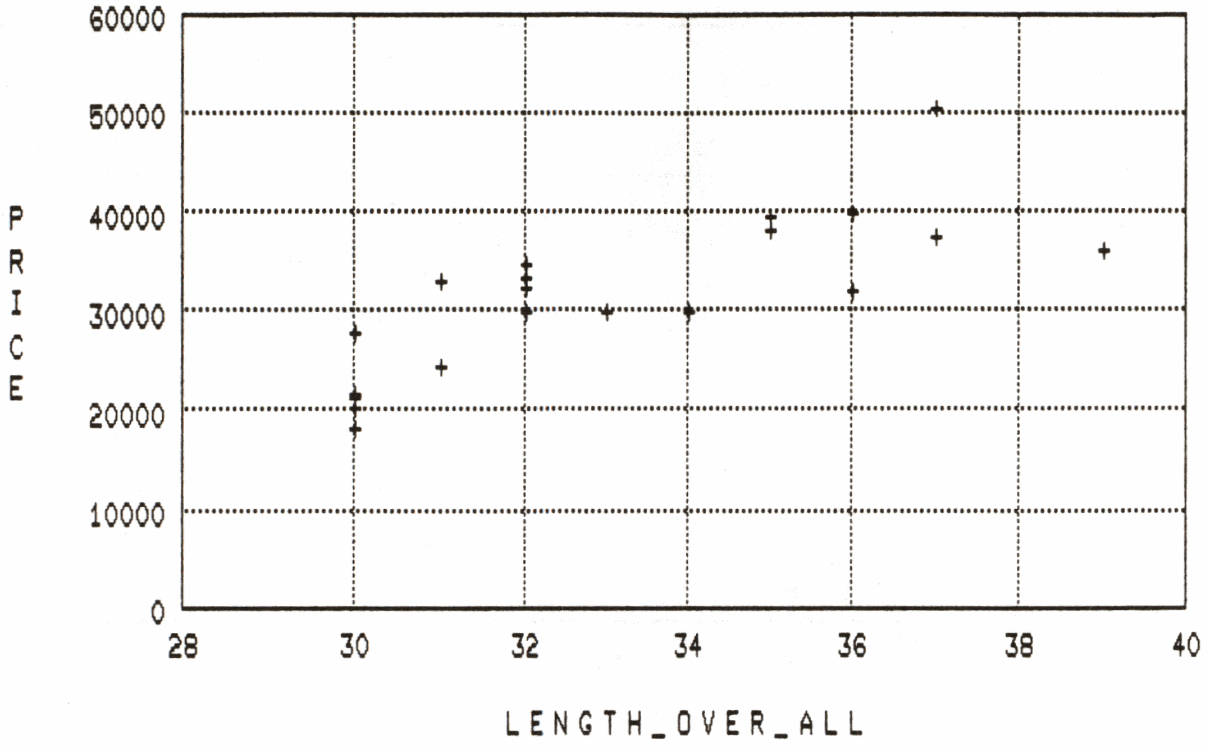
Example 1

1. This example shows how you can use PLOT LIMITS_Y with PLOT LIMITS_X to impose range limits on both axes.

   Use the PLOT X_Y statement to create a chart showing the trend that as LOA of YACHTS increases, so does price. Use PLOT LIMITS_X to limit the range of values on the horizontal (X) axis.

   ```
   DTR> PLOT X_Y ALL LOA, PRICE OF
   CON> YACHTS WITH PRICE NE 0 THEN
   CON> PLOT LIMITS_X 30, 40
   ```

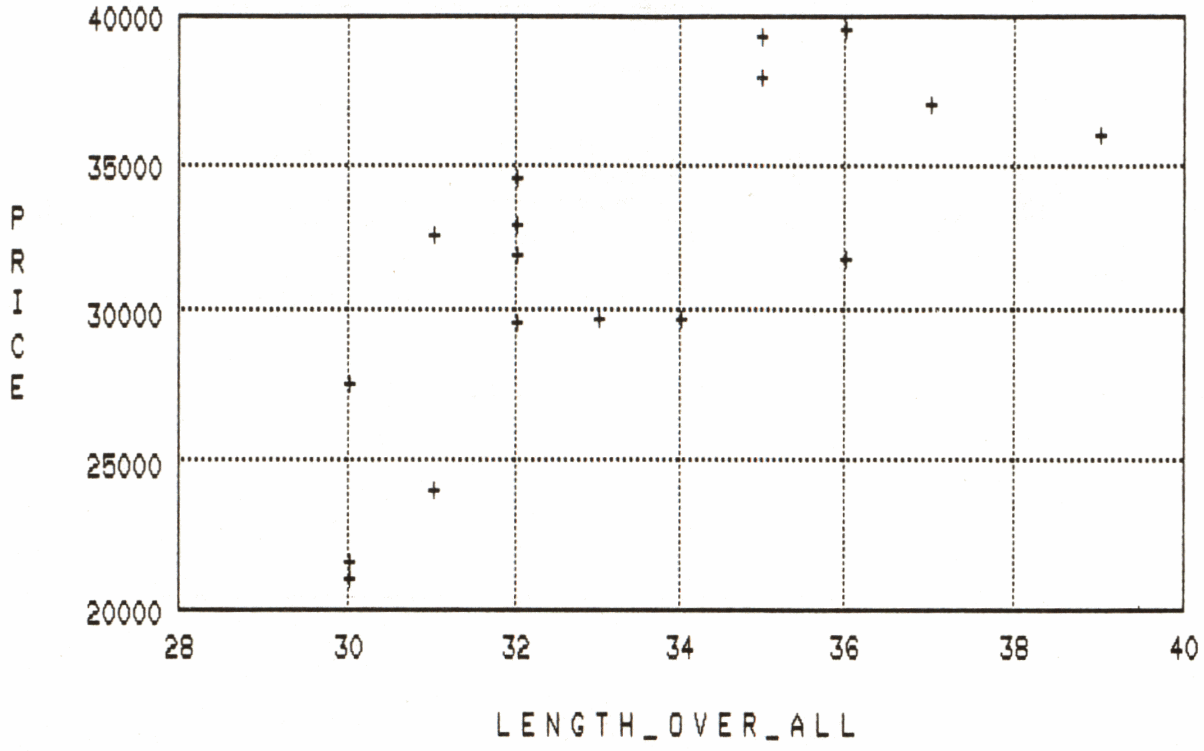
   The resulting plot includes all values found in the RSE for the prices on the vertical (Y) axis; however, the PLOT LIMITS_X statement has limited the range of values for length on the horizontal (X) axis.



ZK-00315-00

2. Then use the PLOT LIMITS_Y statement to limit the range of price values plotted on the vertical (Y) axis. Note that because of the PLOT LIMITS_X statement used in the previous example, the resulting plot provides limits for both axes. PLOT LIMITS_Y can be used without PLOT LIMITS_X, however.

```
DTR> PLOT LIMITS_Y 20000, 40000
```



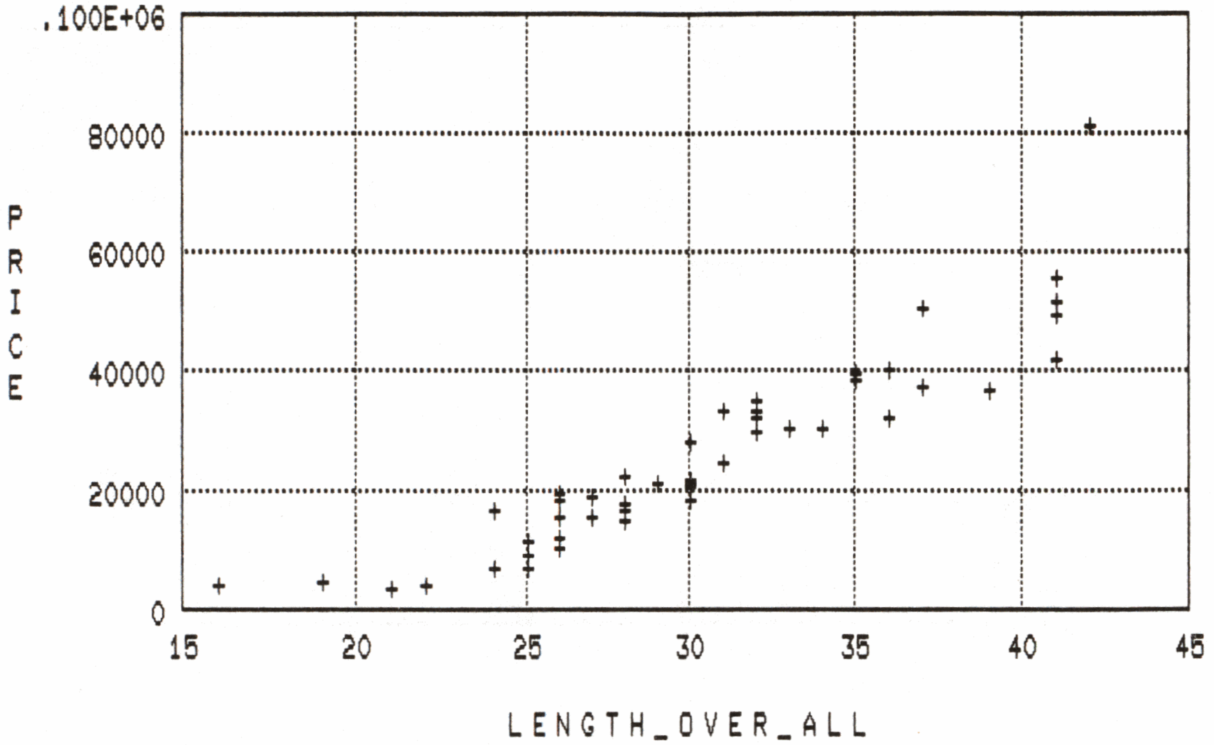
ZK-00316-00

Example 2

1. You can also use a value expression as an argument for PLOT LIMITS_Y. The following example is similar to the second example in the PLOT LIMITS_X section, except this example limits the data on the Y axis. The following statement and plot provide an initial plot:

   ```
   DTR> PLOT X_Y ALL LOA, PRICE OF
   CON> YACHTS WITH PRICE NE 0
   ```

   The resulting plot includes all values found in the RSE for the horizontal (X) axis.

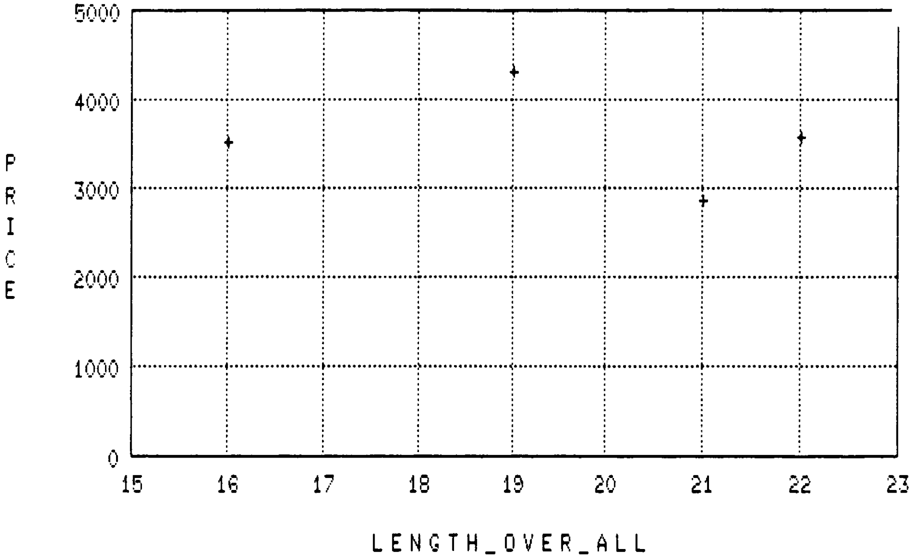


ZK-00313-00

2. The following PLOT LIMITS_Y statement uses a value expression to limit the maximum values for the vertical (Y) axis. First, it determines the average price of all the yachts that have a length less that 25 feet. Because that average is $6,097, LIMITS_Y limits the display of highest vertical (Y) values at $6,097.

   Note that the second example in the LIMITS_X article shows this data with limits on the horizontal (X) axis.

```
DTR> PLOT LIMITS_Y 0,
CON> AVERAGE PRICE OF YACHTS WITH LOA LT 25 AND
CON> LOA NE 0
```


5000
4000
3000
2000
1000
0
P
R
I
C
E
15
16
17
18
19
20
21
22
23
LENGTH_OVER_ALL

ZK-00348-00

4.16 PLOT LOGX_LOGY Statement

Creates a scattergraph that uses logarithmic scaling for both the horizontal (X) and vertical (Y) axes.

Format

```
                                                     ⎡    ⎧ file-spec ⎫ ⎤
PLOT LOGX_LOGY  [USING]  [ALL]  arg,  arg  [OF rse]  ⎢ ON ⎨           ⎬ ⎥
                                                     ⎣    ⎩ *.prompt  ⎭ ⎦
```

Arguments

arg

The PLOT LOGX_LOGY statement takes two arguments. The arguments may be field names or other value expressions and must contain numeric values greater than zero (0) for each record in the plot:

- The first argument provides the values for the horizontal (X) axis.
- The second argument provides the values for the vertical (Y) axis.

Results

- PLOT LOGX_LOGY produces a scattergraph that uses logarithmic scaling for both the horizontal (X) and vertical (Y) axes. Each pair of value expressions in the scattergraph is represented as a plus sign (+).
- With logarithmic scaling, each successive unit of length on a logarithmic scale covers a wider range of numbers. This helps to make trends more apparent.

Usage Note

You can use the following utility plots with the PLOT LOGX_LOGY statement:

- PLOT BIG
- PLOT CONNECT
- PLOT HARDCOPY
- PLOT LIMITS_X

- PLOT LIMITS_Y
- PLOT LR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT SHADE
- PLOT TITLE

See the section on each of these utility plot statements for more information.

Examples

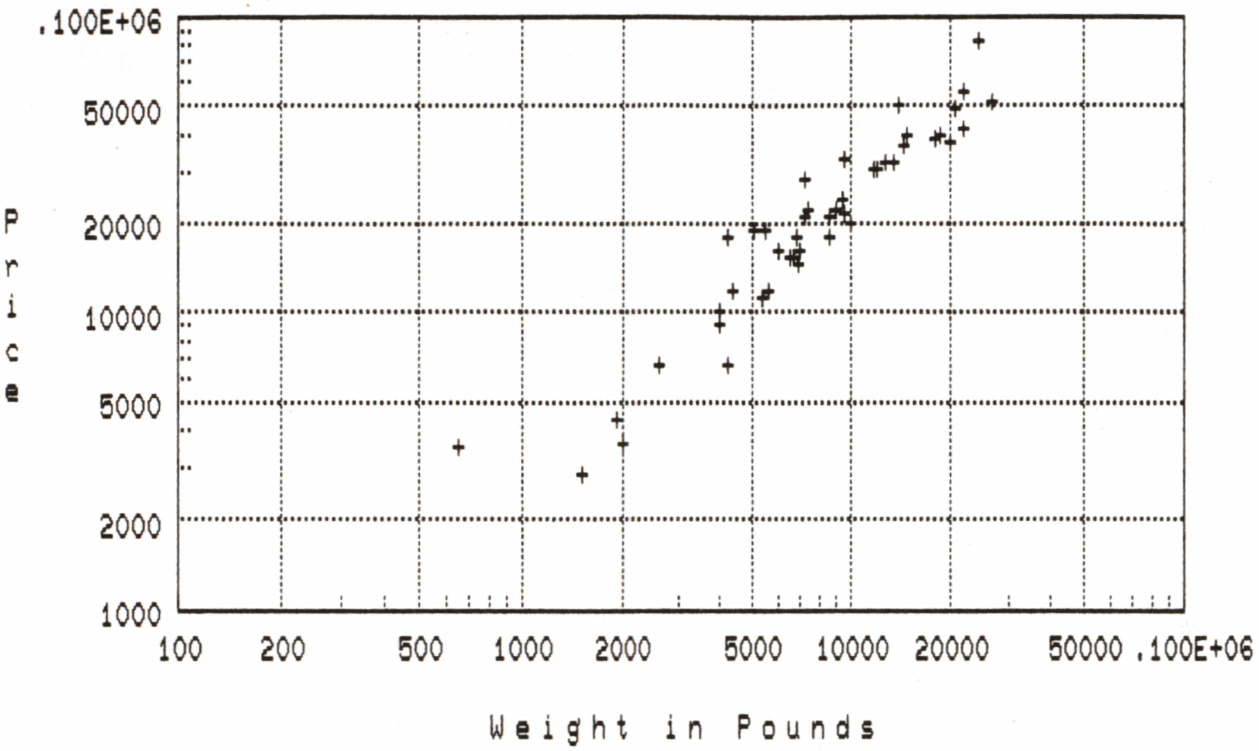
Example 1

Using a collection from the YACHTS domain, create a scattergraph that shows the relationship between weight and cost for all yachts. The resulting scattergraph demonstrates the trend that heavy boats cost more than light ones.

The keyword ALL is required because the plot statement refers to the collection created by the first statement. Note that you can ensure that values for the plot arguments are greater than zero (0) by including appropriate relational operators in the FIND statement.

```
DTR> FIND YACHTS WITH PRICE GT 0 AND DISP GT 0
DTR> PLOT LOGX_LOGY ALL DISP -
CON> ("Weight in Pounds"), PRICE ("Price")
```

Plot for Example 1



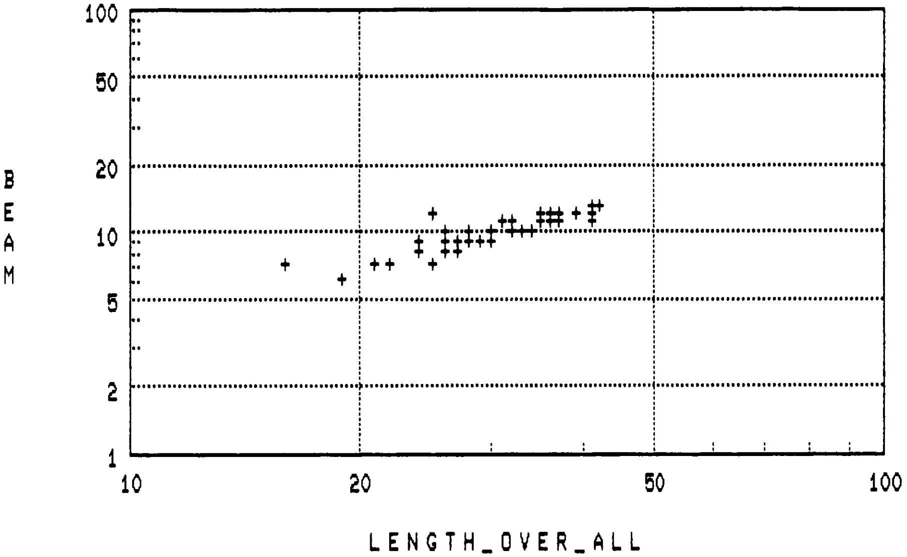
ZK-00317-00

Example 2

Using YACHTS, create a scattergraph that shows the relationship between length and width (beam size) for all yachts in the collection. The keyword ALL is required because the plot statement refers to the collection created by the first statement.

```
DTR> FIND YACHTS WITH PRICE GT 0 AND BEAM GT 0
DTR> PLOT LOGX_LOGY ALL LOA, BEAM
```

Plot for Example 2



ZK-00318-00

4.17 PLOT LOGX_Y Statement

Creates a scattergraph that uses logarithmic scaling for the horizontal (X) axis.

Format

```
                                                   ┌    ┌ file-spec ┐ ┐
PLOT LOGX_Y  [USING]  [ALL]  arg,  arg  [OF rse]   │ ON ┤           ├ │
                                                   └    └ *.prompt  ┘ ┘
```

Arguments

arg

The PLOT LOGX_Y statement takes two arguments. The arguments can be field names or other value expressions:

- The first argument provides the values for the horizontal (X) axis. This argument must contain numeric values greater than zero for all records in the collection or record stream.
- The second argument provides the values for the vertical (Y) axis.

Results

- PLOT LOGX_Y produces a scattergraph that uses logarithmic scaling for the horizontal (X) axis. Each pair of value expressions in the scattergraph is represented as a plus sign (+).
- With logarithmic scaling, each successive unit of length on the logarithmic scale covers a wider range of numbers. This helps to make trends more apparent.

Usage Note

You can use the following utility plots with the PLOT LOGX_Y statement:

- PLOT BIG
- PLOT CONNECT
- PLOT HARDCOPY
- PLOT LIMITS_X

- PLOT LIMITS_Y
- PLOT LR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT SHADE
- PLOT TITLE

See the section on each of these utility plot statements for more information.

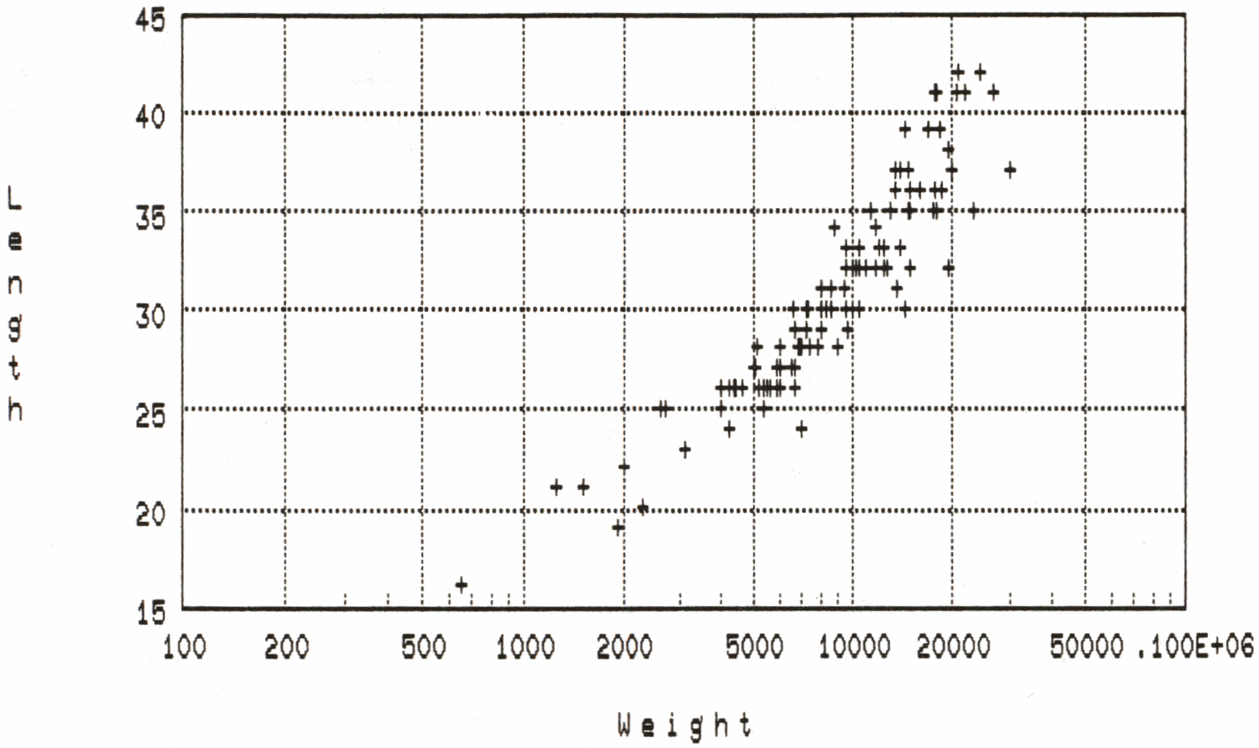
Examples

Example 1

Using the YACHTS domain, create a scattergraph using PLOT LOGX_Y that shows the relationship between weight and length for all yachts. The resulting scattergraph demonstrates the trend that as weight increases so does length:

```
DTR> PLOT LOGX_Y ALL DISP ("Weight"),
CON> LOA ("Length") OF YACHTS WITH DISP GT 0
```

Plot for Example 1



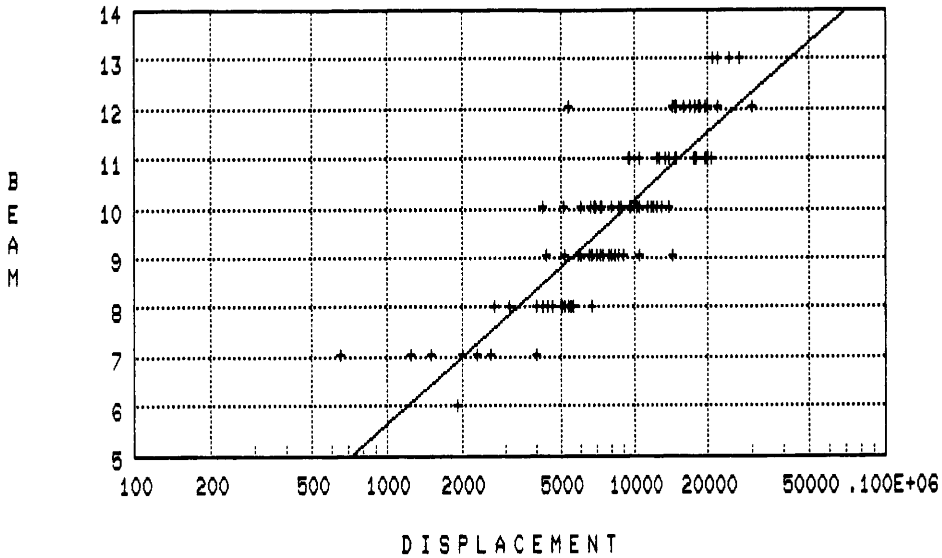
ZK-00319-00

Example 2

Create a collection from the YACHTS domain. Use the PLOT LOGX_Y statement to create a scattergraph showing the relationship between weight and beam. Add a linear regression line with the PLOT LR statement to emphasize the trend that heavier yachts have wider beams. Note that you can ensure that values for the plot arguments are greater than zero (0) by including relational operators in the RSE:

```
DTR> FIND YACHTS WITH DISP GT 0 AND BEAM GT 0
DTR> PLOT LOGX_Y ALL DISP, BEAM THEN
DTR> PLOT LR
```

Plot for Example 2



ZK-00320-00

4.18 PLOT LR Statement

Plots a linear regression line through the points on a scattergraph.

Format

```
PLOT LR
```

Argument

PLOT LR takes no argument.

Result

The PLOT LR statement plots a linear regression line on the scattergraph you have just created. The linear regression plot uses the least squares method to determine the regression line. The least squares method determines a straight line through a set of points so that the sum of the squares of the distances of the points from the line is a minimum. The resulting line is the “best fit” for a set of points.

Usage Note

You can use the PLOT LR statement with the following plots:

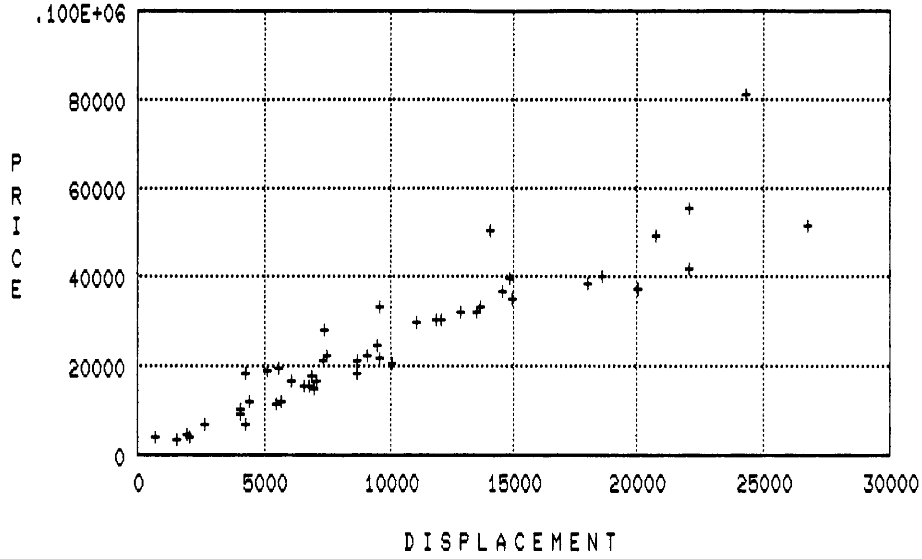
- PLOT DATE_LOGY
- PLOT DATE_Y
- PLOT LOGX_LOGY
- PLOT LOGX_Y
- PLOT X_LOGY
- PLOT X_Y

Example

1. Using the YACHTS domain, create a scattergraph that shows the relationship between weight and price for all yachts:

```
DTR> PLOT X_Y ALL DISP ("DISPLACEMENT"),
CON> PRICE OF YACHTS WITH PRICE NE 0
```

The following shows the plot before the linear regression line is added.

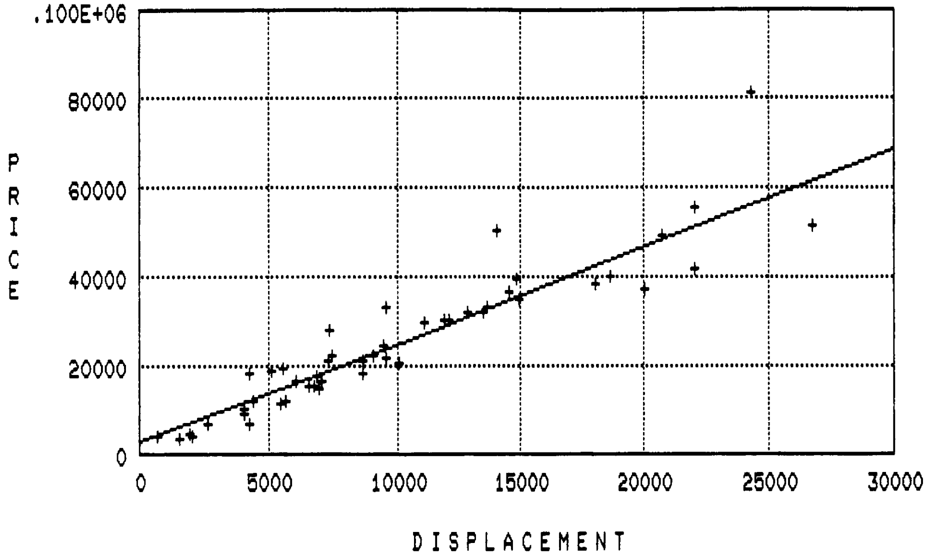


ZK-00321-00

2. Next, use a linear regression line to emphasize the trend that heavier yachts cost more than lighter ones:

```
DTR> PLOT LR
```

DATATRIEVE plots a regression line in the scattergraph.


.100E+06
80000
60000
40000
20000
0
PRICE
0
5000
10000
15000
20000
25000
30000
DISPLACEMENT

ZK-00322-00

4.19 PLOT MONITOR Statement

Displays the words RED, GREEN, and BLUE on your color monitor with their appropriate colors. It is used to verify that red, green, and blue cable connectors are correctly attached to the back of the color monitor.

Format

```
PLOT MONITOR
```

Argument

The PLOT MONITOR statement takes no argument.

Restriction

Use PLOT MONITOR only with a color terminal.

If you use PLOT MONITOR with a monochrome terminal that has light lettering and a dark background, the light lettering may change to the color of the dark background, essentially blacking out your screen.

Results

- The PLOT MONITOR statement displays the words RED in red, GREEN in green, and BLUE in blue. The words are listed vertically on the color monitor.
- The word SYNC appears on the screen to indicate that DATATRIEVE has reset the terminal to the terminal's default color settings after the words RED, GREEN, and BLUE have been displayed.

Usage Note

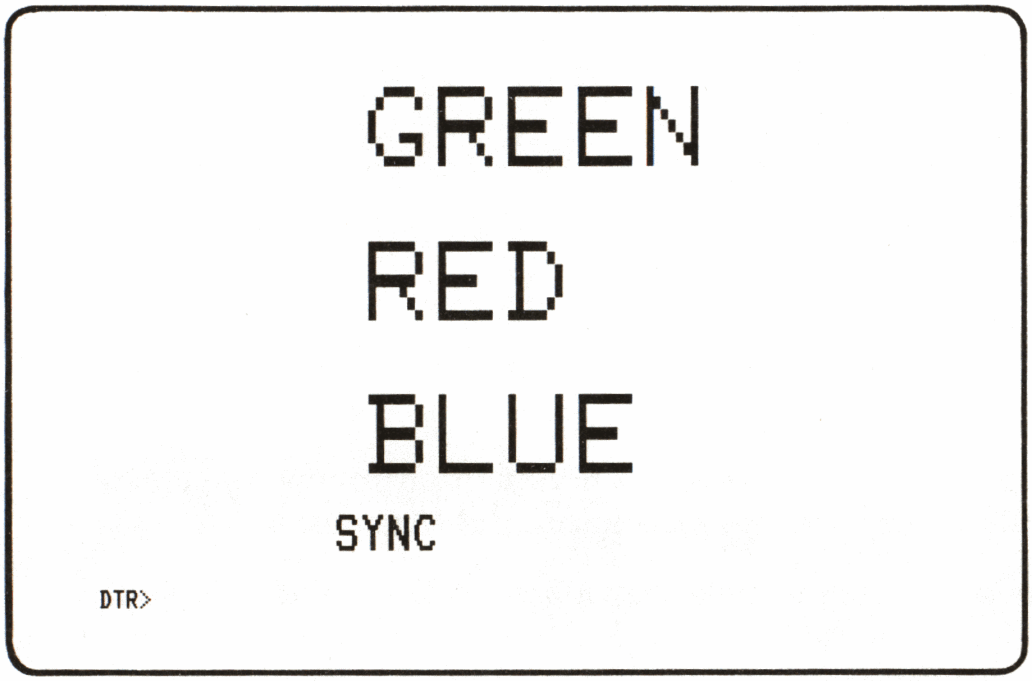
If the names of the colors do not correspond to the color in which they are displayed (for example, the word RED is displayed in the color green), you may not have attached the color cable connectors correctly. Make sure the cables from the video terminal match the proper connectors on the monitor. See the chapter about getting started with DATATRIEVE plots for more information.

Example

Type the following statement to verify that color cable connectors are appropriately attached and that other appropriate setup steps have been completed:

```
DTR> PLOT MONITOR
```

DATATRIEVE plots the following screen.



```
GREEN

RED

BLUE
SYNC
DTR>
```

ZK-00282-00

4.20 PLOT MULTI_BAR Statement

Creates a chart with bars grouped in clusters. Each bar in the cluster represents a field name or other value expression. Each cluster can contain as many as three bars.

Format

```
PLOT MULTI_BAR  [USING]  [ALL]  arg,  arg  [,  arg  [,  arg] ]

        [OF rse]  [ON  { file-spec }]
                  [    { *.prompt  }]
```

Arguments

arg

The PLOT MULTI_BAR statement takes from two to four arguments. Arguments can be field names or other value expressions:

- The first argument provides the common value that groups the clustered bars. It also labels the bars along the horizontal (X) axis.
- The remaining arguments provide the values for the vertical (Y) axis. These values determine the height of each bar. These arguments must evaluate to numbers.

Results

- PLOT MULTI_BAR produces a bar chart in which up to three related bars are clustered along the horizontal (X) axis. Unlike most bar charts, PLOT MULTI_BAR does not specify a limit on the number of records or clusters it will group on a single chart. It tries to plot all of the records or clusters onto a single graph.
- If there is room, PLOT MULTI_BAR prints a legend identifying what values each bar in the cluster represents. (If there is not enough space, you must use the PLOT LEGEND statement to display the legend separately.)

Usage Notes

- The difference between the PLOT MULTI_BAR and PLOT MULTI_BAR_GROUP statements is the first argument, which also labels the horizontal (X) axis:
  - With PLOT MULTI_BAR_GROUP, each bar or cluster of bars represents the totals from a number of records. The PLOT MULTI_BAR_GROUP statement groups the values of the subsequent arguments based on the first argument.
  - Unlike PLOT MULTI_BAR_GROUP, PLOT MULTI_BAR uses the first argument to produce a bar or cluster of bars for each unique record. In the following Example 1, each year corresponds to only one value for SERVICES, one value for EQUIPMENT_SALES, and one value for REVENUE.
- PLOT MULTI_BAR does not group the records or sort the bars. If you want a specific order, sort the record stream (for example, FIND domain-name SORTED BY field-name).
- You can use the following utility plots with the PLOT MULTI_BAR statement:
  - PLOT BIG
  - PLOT CROSS_HATCH
  - PLOT HARDCOPY
  - PLOT LEGEND
  - PLOT LIMITS_X
  - PLOT LIMITS_Y
  - PLOT PAUSE
  - PLOT REFERENCE_X
  - PLOT REFERENCE_Y
  - PLOT RE_PAINT
  - PLOT TITLE

  See the section on each of these utility plot statements for more information.

- The PLOT STACKED_BAR statement is a related plot that plots data similar to that used with PLOT MULTI_BAR in a stacked bar format.
- PLOT MULTI_BAR is restricted in the number of records it can plot. If the statement contains two arguments, then DATATRIEVE can plot approximately 450 records; for three arguments, it can plot approximately 250 records; and for four arguments, it can plot approximately 175 records. Before using the PLOT MULTI_BAR statement with a large number of records or clusters, you should consider the effect this will have on the chart readability.

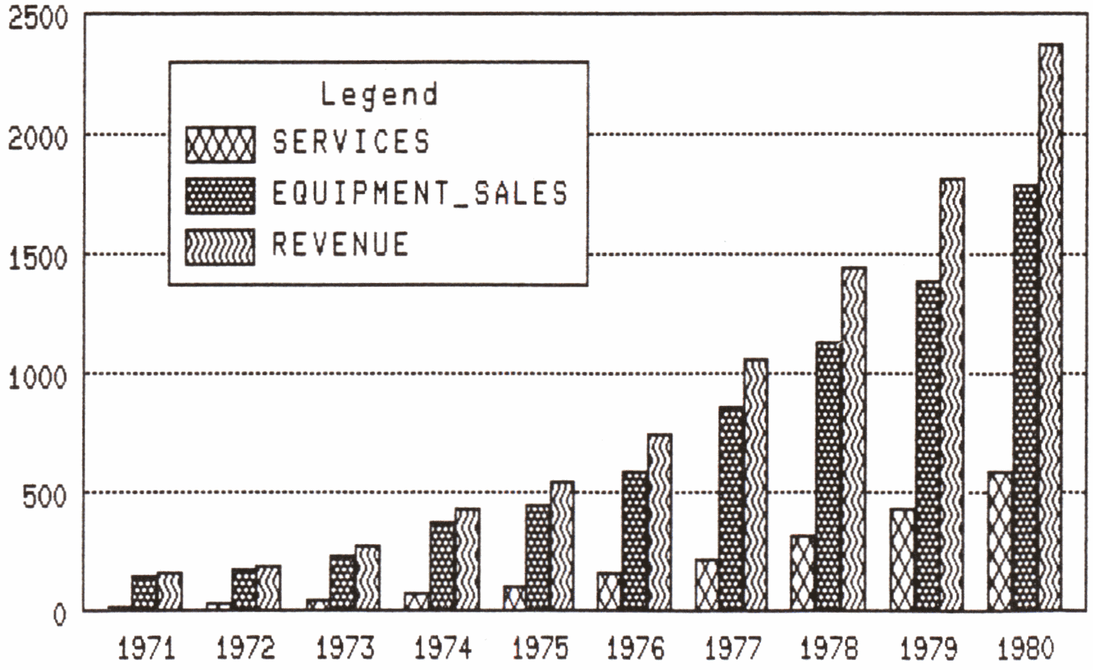
Examples

Example 1

Create a collection from the ANNUAL_REPORT domain. Then use PLOT MULTI_BAR to display a plot that shows the revenue from equipment sales and services and the total revenue for each year. Then use PLOT CROSS_HATCH to clearly differentiate bars in each cluster:

```
DTR> FIND ANNUAL_REPORT SORTED BY DATE
DTR> PLOT MULTI_BAR ALL FORMAT
CON> DATE USING Y(4),
CON> SERVICES, EQUIPMENT_SALES, REVENUE THEN
CON> PLOT CROSS_HATCH
```

Plot for Example 1



ZK-00323-00

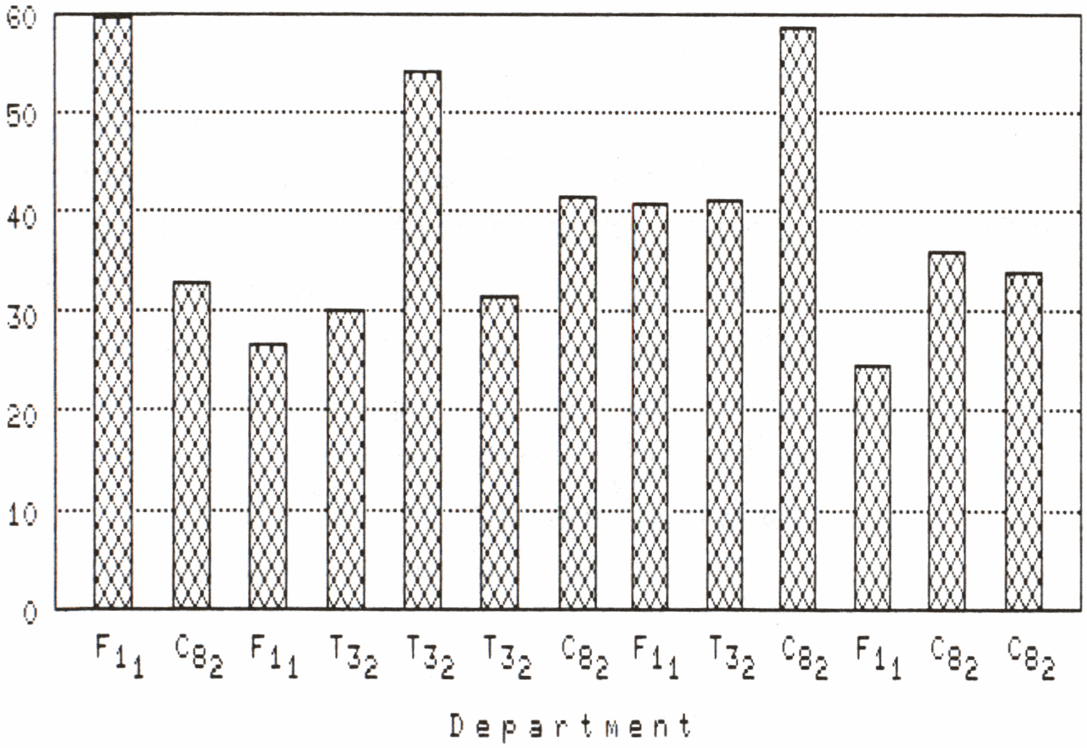
Example 2

Using the PERSONNEL domain, plot a multibar chart showing the value of each salary record and its corresponding department. The resulting bar chart shows the salary for each employee without using the employee's name. It allows you to see the ranges of employees' salaries.

```
DTR> PLOT MULTI_BAR ALL DEPT ("Department"),
CON> SALARY/1000 ("Salary in 1000s") OF
CON> PERSONNEL WITH DEPT = "C82", "F11", "T32"
```

Compare this example to the similar example in the PLOT MULTI_BAR_GROUP section. PLOT MULTI_BAR_GROUP plots the same data but groups and totals the salaries for each department.

Plot for Example 2



ZK-00324-00

4.21 PLOT MULTI_BAR_GROUP Statement

Creates a chart with bars grouped in clusters. Each cluster can contain up to three bars.

PLOT MULTI_BAR_GROUP groups the records according to the argument you specify for the horizontal (X) axis. It plots the sum of each of the subsequent arguments along the vertical (Y) axis.

Format

```
PLOT MULTI_BAR_GROUP  [USING]  [ALL]  arg,  arg  [,  arg  [,  arg] ]

                          ┌     ┌ file-spec ┐ ┐
          [OF rse]        │ ON  │           │ │
                          └     └ *.prompt  ┘ ┘
```

Arguments

arg

The PLOT MULTI_BAR_GROUP statement takes up to four arguments. These arguments can be field names or other value expressions:

- The first argument labels the horizontal (X) axis. This argument is used to group records that share the same value for this argument.
- Each of the next three arguments is totaled and plotted as a bar. The total values are represented along the vertical (Y) axis.

  These arguments must evaluate to numbers.

Results

- PLOT MULTI_BAR_GROUP groups the records according to the argument you specify for the horizontal (X) axis. It plots the sum of each of the subsequent arguments along the Y axis.
- PLOT MULTI_BAR_GROUP does not sort the bars. If you want a specific order, sort the record stream (for example, FIND domain-name SORTED by field-name).
- If there is room, DATATRIEVE prints a legend explaining which bars represent which values. If there is not enough room, type PLOT LEGEND to see the legend.

Usage Notes

- The difference between the PLOT MULTI_BAR_GROUP and PLOT MULTI_BAR statements is the first argument, which also labels the horizontal (X) axis:
  - With PLOT MULTI_BAR_GROUP, each bar or cluster of bars represents the totals from a number of records. The PLOT MULTI_BAR_GROUP statement groups the values of the subsequent arguments based on the first argument.
  - Unlike PLOT MULTI_BAR_GROUP, PLOT MULTI_BAR uses the first argument to produce a bar or cluster of bars for each unique record.
- Most bar charts plot a maximum of six bars per chart. PLOT MULTI_BAR_GROUP, however, plots all the bars in one chart. Before using the PLOT MULTI_BAR_GROUP statement with a large number of records, you should consider the effect this will have on the readability of the chart.

  PLOT MULTI_BAR_GROUP is restricted in the number of records it can plot. If the statement contains two arguments, DATATRIEVE can plot approximately 450 records. For three arguments, it can plot approximately 250 records; for four arguments, it can plot approximately 175 records. Again, plots generated with PLOT MULTI_BAR_GROUP with a large number of records would be very difficult to read.
- You can use the following utility plots with the PLOT MULTI_BAR_GROUP statement:
  - PLOT BIG
  - PLOT CROSS_HATCH
  - PLOT HARDCOPY
  - PLOT LEGEND
  - PLOT LIMITS_X
  - PLOT LIMITS_Y
  - PLOT PAUSE
  - PLOT REFERENCE_X

- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT TITLE

See the section on each of these utility plot statements for more information.

Examples

Example 1

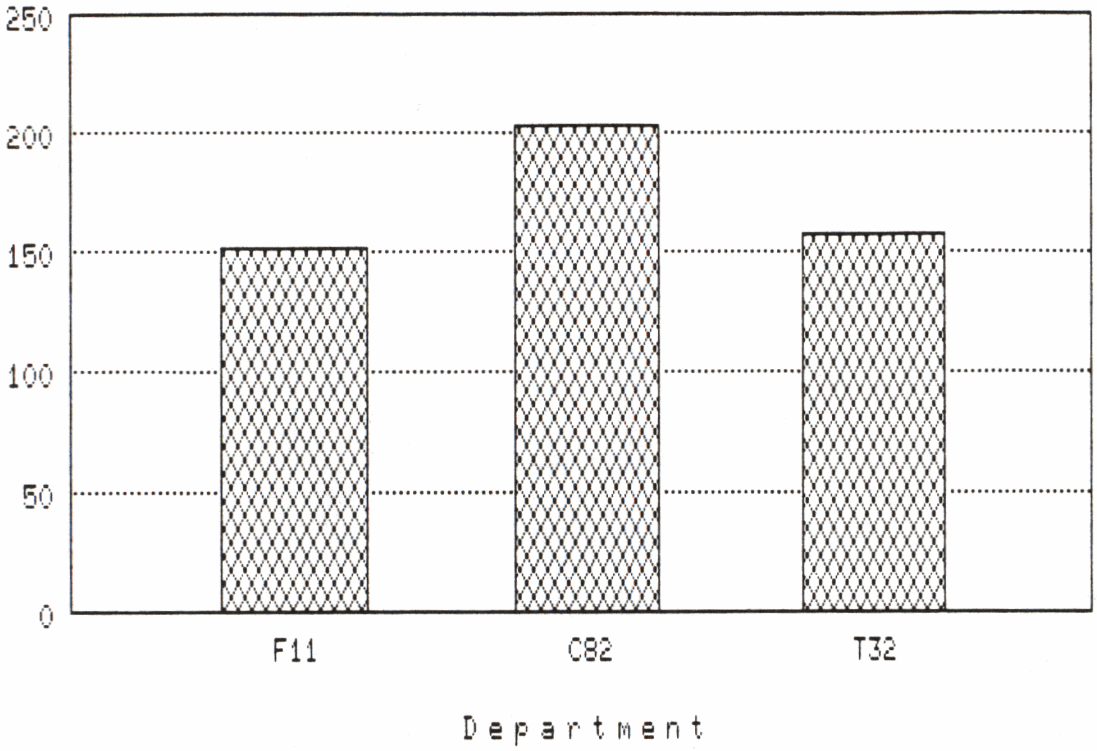
Use the PERSONNEL domain and PLOT MULTI_BAR_GROUP to plot the total salaries of employees who report to the departments F11, C82, and T32.

```
DTR> FIND PERSONNEL WITH DEPT = "F11", "C82", "T32"
DTR> PLOT MULTI_BAR_GROUP ALL DEPT ("Department"),
CON> SALARY / 1000 ("Salaries / 1000") THEN
CON> PLOT CROSS_HATCH
```

Compare this example to the similar example in the PLOT MULTI_BAR section:

- PLOT MULTI_BAR_GROUP plots the same data but groups and totals the salaries for each department.
- PLOT MULTI_BAR produces a bar showing the value of each salary record and its corresponding department. The bars show each employee's salary without using the employee's name.

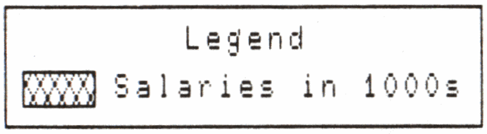
Plot for Example 1



ZK-00325-00

In this example, there is not enough room for the legend. Enter the PLOT LEGEND statement to plot the legend.

```
DTR> PLOT LEGEND
```



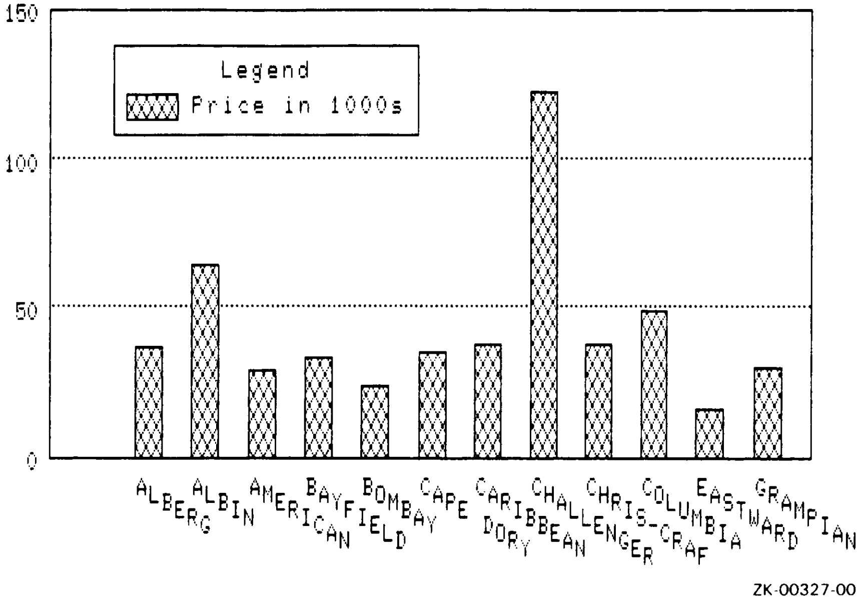
ZK-00326-00

Example 2

Use the YACHTS domain and establish a collection of the first 20 yachts. Then use the PLOT MULTI_BAR_GROUP statement to plot the total prices of those yachts according to their builder. Divide the total prices by 1000 to avoid scientific notation. The keyword ALL is required because the plot statement refers to the collection created by the first statement:

```
DTR> FIND FIRST 20 YACHTS WITH PRICE NE 0
DTR> PLOT MULTI_BAR_GROUP ALL BUILDER,
CON> PRICE/1000 ("Price in 1000s") THEN
CON> PLOT CROSS_HATCH
```

Plot for Example 2



4.22 PLOT MULTI_LINE Statement

Creates a line graph with up to three uniquely marked lines.

Format

```
PLOT MULTI_LINE  [USING]  [ALL]  arg,  arg  [,  arg  [,  arg] ]

        [OF rse]  [ ON  { file-spec  } ]
                  [     { *.prompt   } ]
```

Arguments

arg

The PLOT MULTI_LINE statement takes two to four arguments. The arguments can be field names or other value expressions and must be numeric:

- The first argument is used to label the horizontal (X) axis.
- The values for each of the next three arguments determine location of the points on the horizontal (X) and vertical (Y) axis. These points are connected, determining the lines of the graph.

Results

- PLOT MULTI_LINE produces a line graph that plots up to three uniquely marked lines.
- If there is room, DATATRIEVE prints a legend identifying what each line represents. If DATATRIEVE does not print the legend, you can produce it by subsequently typing PLOT LEGEND.

Usage Notes

- PLOT MULTI_LINE does not sort the records. You can specify a particular order by sorting the record stream (for example, FIND domain-name SORTED by field-name).
- The PLOT MULTI_SHADE statement produces a shaded multiline graph.

- You can use the following utility plots with the PLOT MULTI_LINE statement:

  – PLOT BIG

  – PLOT HARDCOPY

  – PLOT LEGEND

  – PLOT LIMITS_X

  – PLOT LIMITS_Y

  – PLOT PAUSE

  – PLOT REFERENCE_X

  – PLOT REFERENCE_Y

  – PLOT RE_PAINT

  – PLOT TITLE

  See the section on each of these utility plot statements for more information.

Examples

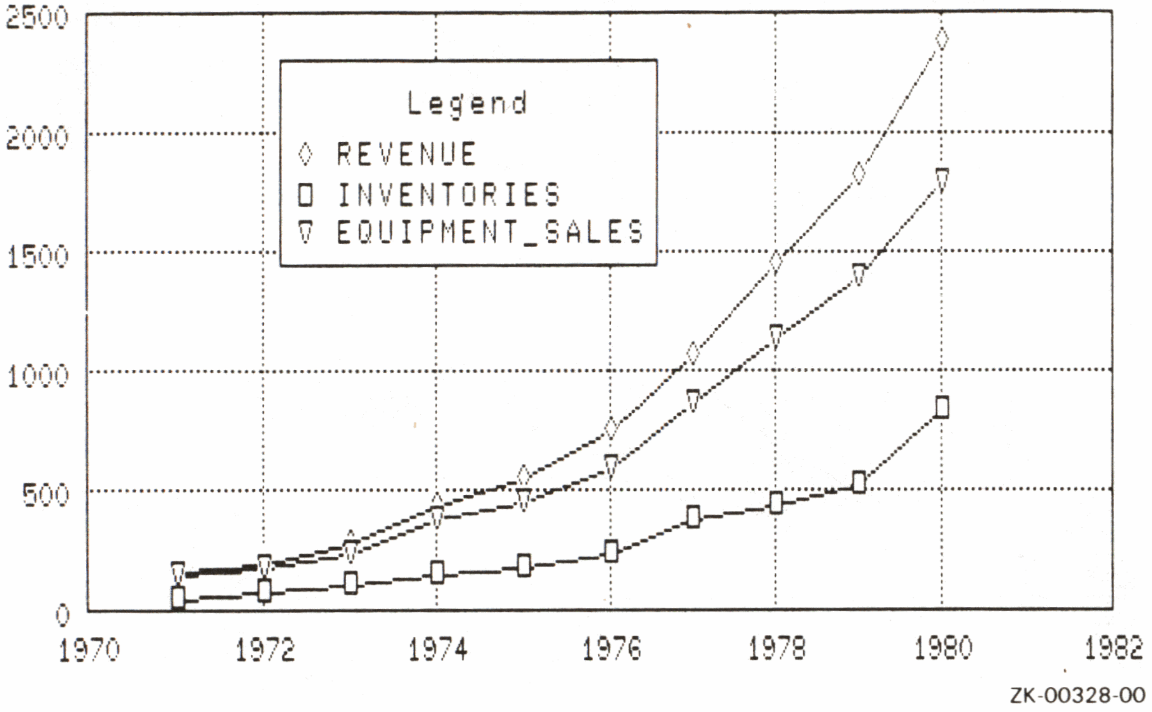
Example 1

Create a collection using the ANNUAL_REPORT domain sorted by date. Specify DATE as the first argument to label the horizontal (X) axis. Specify REVENUE, INVENTORIES, and EQUIPMENT_SALES as the last three arguments. These will each be plotted as uniquely marked lines. The keyword ALL is required because the plot statement refers to the collection created by the first statement.

The resulting graph shows the increases in revenue, inventories, and equipment sales between 1971 and 1980. The lines of the graph emphasize the trends:

```
DTR> FIND ANNUAL_REPORT SORTED BY DATE
DTR> PLOT MULTI_LINE ALL FORMAT
CON> DATE USING Y(4),
CON> REVENUE, INVENTORIES, EQUIPMENT_SALES
```

Plot for Example 1



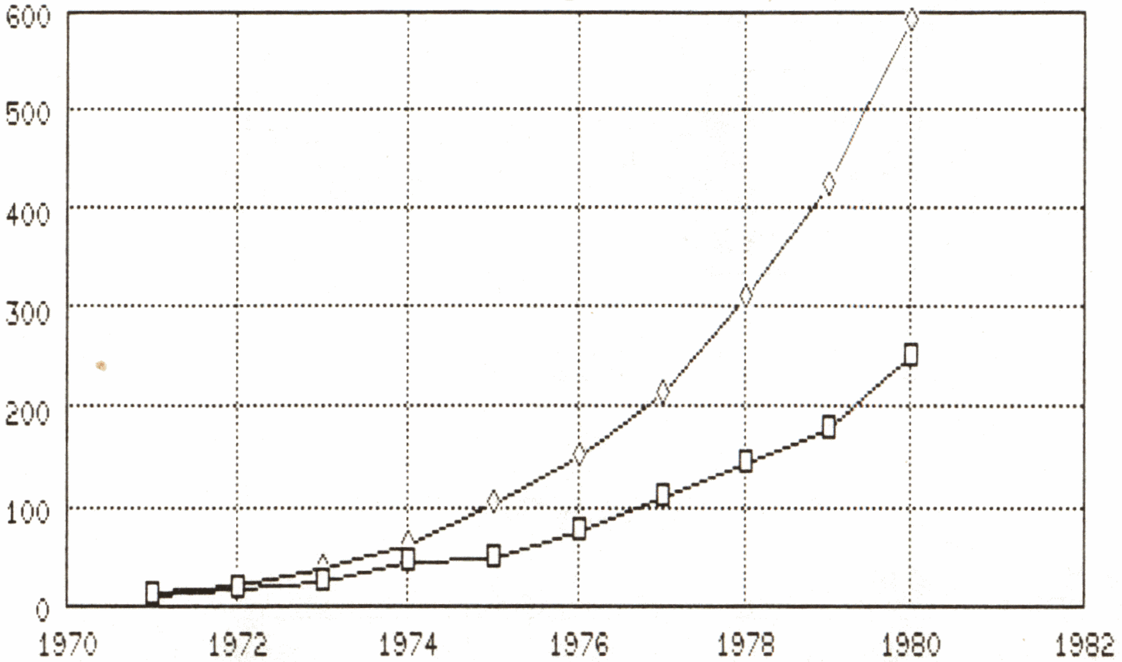
Example 2

Use ANNUAL_REPORT to plot a 2-line graph without a legend. Specify DATE as the first argument to label the horizontal (X) axis. Specify SERVICES and NET_INCOME as the last two arguments. They will be plotted as uniquely marked lines. The keyword ALL is required because the plot statement refers to the collection created in the first line.

The resulting plot shows how NET_INCOME and SERVICES increase over time:

```
DTR> FIND ANNUAL_REPORT SORTED BY DATE
DTR> PLOT MULTI_LINE ALL FORMAT
CON> DATE USING Y(4),
CON> SERVICES, NET_INCOME
```

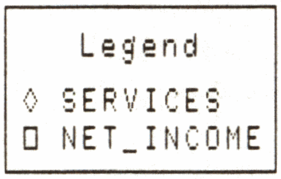
Plot for Example 2



ZK-00329-00

Use the PLOT LEGEND statement to produce the legend for Example 2:

```
DTR> PLOT LEGEND
```



ZK-00330-00

4.23 PLOT MULTI_LR Statement

Plots a line graph with up to three linear regression lines.

Format

```
PLOT MULTI_LR  [USING]  [ALL]  arg,  arg  [,  arg  [,  arg] ]

        [OF rse]  ┌    ┌ file-spec ┐ ┐
                  │ ON ┤           ├ │
                  └    └ *.prompt  ┘ ┘
```

Arguments

arg

The PLOT MULTI_LR statement takes two to four arguments. These arguments can be field names or other value expressions and must be numeric:

- The first argument is used to label the horizontal (X) axis.
- The values for each of the next three arguments determine location of the points on the horizontal (X) and vertical (Y) axis.

Results

- PLOT MULTI_LR plots points based on the arguments you provided. Those points are uniquely marked. PLOT MULTI_LR then uses these points and draws linear regression lines using the least squares method. (The least squares method determines a straight line through a set of points so that the sum of the squares of the distances of the points from the line is a minimum. The resulting line is the "best fit" for a set of points.)
- If there is room, DATATRIEVE prints a legend identifying what each line represents. If DATATRIEVE does not print the legend, you can produce it by subsequently typing PLOT LEGEND.

Usage Notes

- MULTI_LR does not sort the records. You can specify a particular order by sorting the record stream (for example, FIND domain-name SORTED by field-name).

- You can use the following utility plots with the PLOT MULTI_LR statement:
  - PLOT BIG
  - PLOT HARDCOPY
  - PLOT LEGEND
  - PLOT LIMITS_X
  - PLOT LIMITS_Y
  - PLOT PAUSE
  - PLOT REFERENCE_X
  - PLOT REFERENCE_Y
  - PLOT RE_PAINT
  - PLOT TITLE

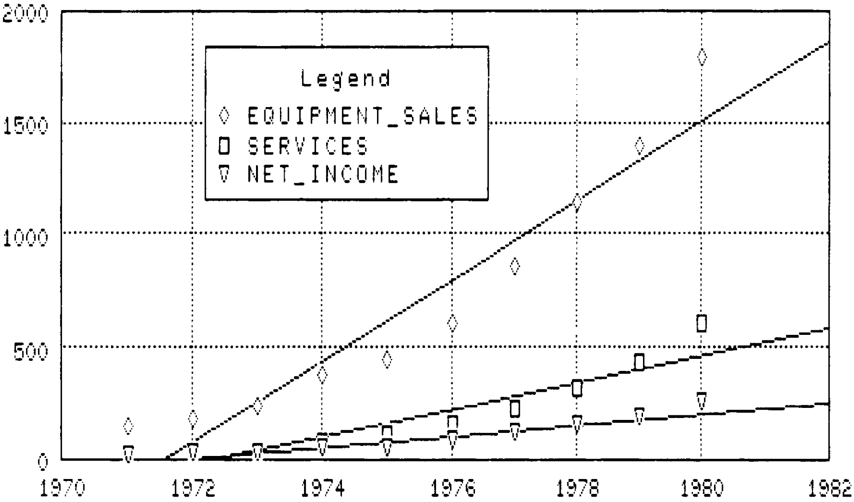
  See the section on each of these utility plot statements for more information.

Example

Using the ANNUAL_REPORT domain, plot a 3-line linear regression graph with a legend.

The resulting plot shows the increase in equipment sales, services, and net income between 1971 and 1980. The linear regression lines help to emphasize trends.

```
DTR> PLOT MULTI_LR ALL FORMAT DATE USING Y(4),
CON> EQUIPMENT_SALES, SERVICES, NET_INCOME OF
CON> ANNUAL_REPORT
```


Legend
◊ EQUIPMENT_SALES
□ SERVICES
▽ NET_INCOME
2000
1500
1000
500
0
1970
1972
1974
1976
1978
1980
1982

ZK-00331-00

4.24 PLOT MULTI_SHADE Statement

Creates a line graph with one to three lines. Areas below the lines are shaded in up to three unique shading patterns.

Format

```
PLOT MULTI_SHADE  [USING]  [ALL]  arg,  arg  [,  arg  [,  arg] ]

        [OF rse]  [ON { file-spec  } ]
                  [   { *.prompt   } ]
```

Arguments

arg

The PLOT MULTI_SHADE statement takes two to four arguments. The arguments can be field names or other value expressions and must be numeric:

- The first argument is used to label the horizontal (X) axis.
- The values for each of the next three arguments determine location of the points on the horizontal (X) and vertical (Y) axis. These points determine the lines of the graph, which in turn determine shading patterns.

Results

- The PLOT MULTI_SHADE statement creates a shaded multiple line graph with up to three unique shades.
- Rather than actually plotting points and lines like PLOT MULTI_LINE, PLOT MULTI_SHADE fills in the area below where the line would appear. PLOT MULTI_SHADE performs the following sequence:

  1. Produces a unique shaded area below the line of the values for the second argument
  2. Overwrites the first shaded area with a uniquely shaded area for the third argument
  3. Overwrites the first and second shaded areas with a uniquely shaded area for the fourth argument

- If there is room, DATATRIEVE prints a legend explaining which shading patterns represent which values. If DATATRIEVE does not print the legend, you can produce it by subsequently typing PLOT LEGEND.

Usage Notes

- DATATRIEVE plots the shaded area for the last three arguments in the order you enter them.

  To avoid completely overwriting the smaller shaded areas, you should first determine the size of the values in the fields. (Use the PRINT statement to print the fields and look at the data.) You should then enter the arguments in decreasing order of size:

  - Enter the argument representing the largest values first
  - Enter the argument representing the second largest values next
  - Enter the argument representing the smallest values last

  PLOT MULTI_SHADE then plots the arguments with the larger values first. It subsequently plots the arguments with smaller values over the larger values. You can then see *all* the shaded areas, not just the largest.

- You can use the following utility plots with the PLOT MULTI_SHADE statement:

  - PLOT BIG
  - PLOT CROSS_HATCH
  - PLOT HARDCOPY
  - PLOT LEGEND
  - PLOT LIMITS_X
  - PLOT LIMITS_Y
  - PLOT PAUSE
  - PLOT REFERENCE_X
  - PLOT REFERENCE_Y
  - PLOT RE_PAINT
  - PLOT TITLE

  See the section on each of these utility plot statements for more information.

- PLOT MULTI_SHADE does not sort the records. If you want a specific order, sort the record stream (for example, FIND domain-name SORTED by field-name).

Examples

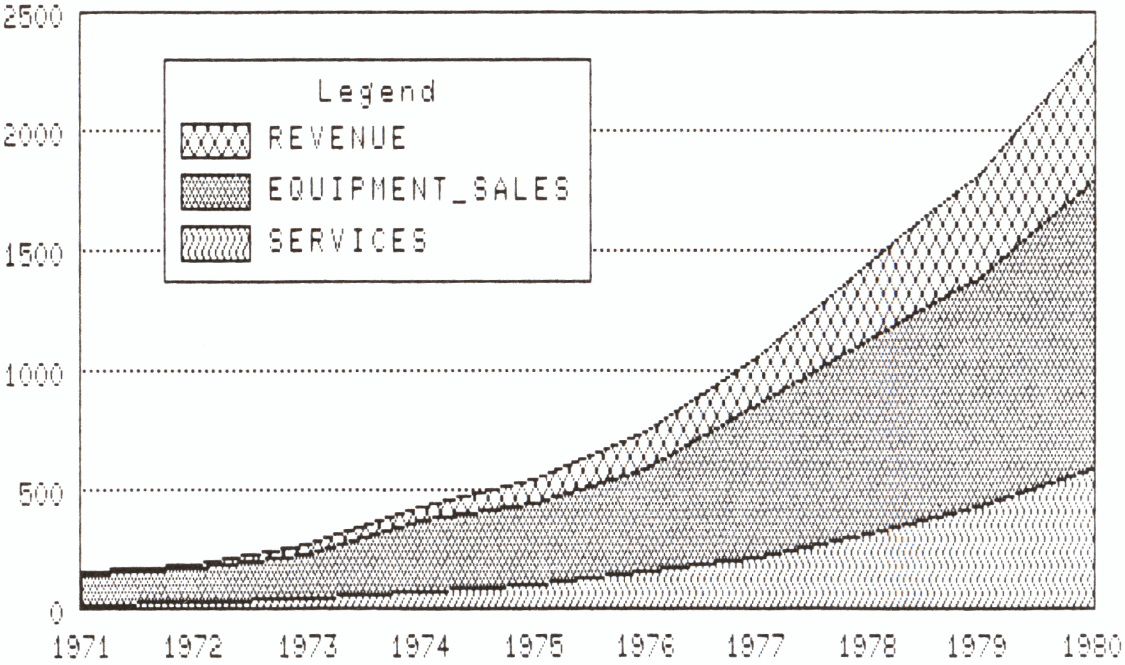
Example 1

Using the ANNUAL_REPORT domain, specify DATE as the first argument, labeling the horizontal (X) axis. Enter the remaining arguments in decreasing order.

PLOT MULTI_SHADE uses the values for the last three arguments to shade the area below each line. It produces the shaded areas in the order you entered the arguments. Because you entered the arguments in decreasing order, all shaded areas appear in the plot:

```
DTR> PLOT MULTI_SHADE ALL FORMAT DATE USING Y(4),
CON> REVENUE, EQUIPMENT_SALES, SERVICES OF
CON> ANNUAL_REPORT SORTED BY DATE THEN
CON> PLOT CROSS_HATCH
```

Plot for Example 1



ZK-00332-00

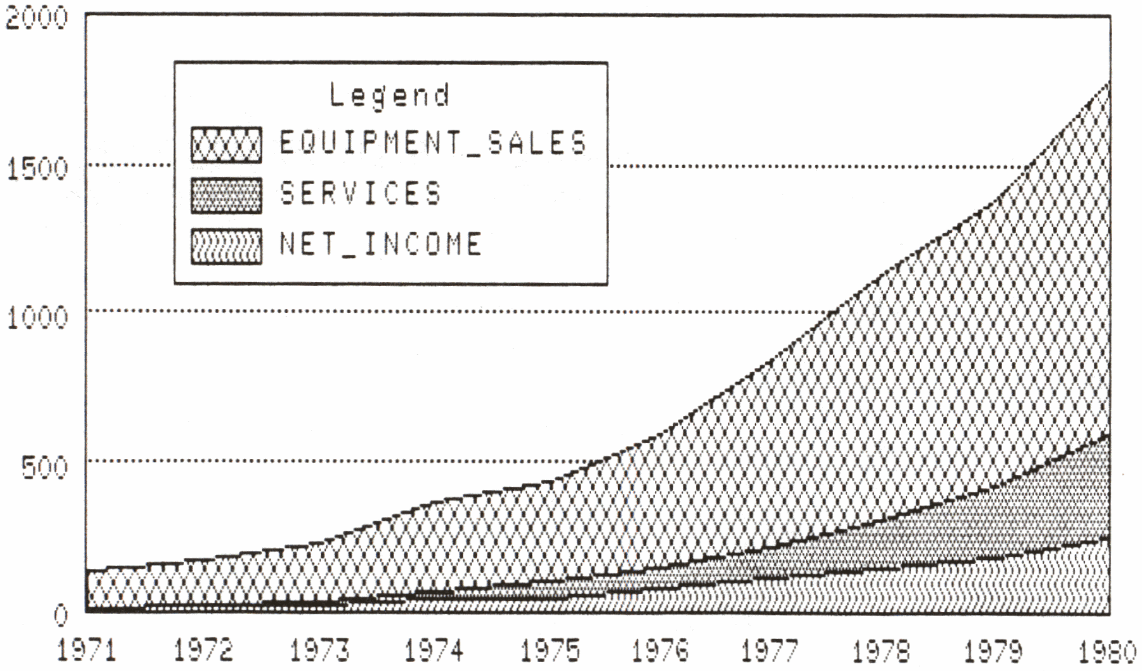
Example 2

Specify DATE as the first argument, labeling the horizontal (X) axis. Enter the remaining arguments in decreasing order. The keyword ALL is required because it refers to the collection created by the first statement.

PLOT MULTI_SHADE uses the values for each argument to shade the area below each line. It produces the shaded areas in the order you entered the arguments. Because you entered the arguments in decreasing order of size, all shaded areas appear in the plot.

```
DTR> FIND ANNUAL_REPORT SORTED BY DATE
DTR> PLOT MULTI_SHADE ALL FORMAT
CON> DATE USING Y(4),
CON> EQUIPMENT_SALES, SERVICES, NET_INCOME THEN
CON> PLOT CROSS_HATCH
```

Plot for Example 2



ZK-00333-00

4.25 PLOT NEXT_BAR Statement

Plots the next six bars of a bar chart or histogram.

Format

```
PLOT NEXT_BAR
```

Argument

The PLOT NEXT_BAR statement takes no argument.

Results

- A bar chart can contain up to six bars. The notice (More...) appears in the lower right corner of a chart if additional bars remain to be plotted.
- The PLOT NEXT_BAR statement plots the next six bars.

Usage Notes

- To see all the remaining bars, continue entering the PLOT NEXT_BAR statement after each chart is plotted. The bars have all been charted when the notice (More...) no longer appears.
- Use the PLOT NEXT_BAR statement with the following statements:
  - PLOT BAR
  - PLOT BAR_AVERAGE
  - PLOT HISTO
  - PLOT RAW_BAR
- Use the PLOT SORT_BAR statement to sort and replot bars in descending order. Use the PLOT BAR_ASCENDING statement to sort and replot bars in ascending order.
- If you use the PLOT NEXT_BAR statement with the PLOT TITLE statement and either the PLOT SORT_BAR or PLOT BAR_ASCENDING statements, enter the statements in this sequence:

  1. PLOT BAR

2. PLOT TITLE
3. PLOT SORT_BAR (or PLOT BAR_ASCENDING)
4. PLOT NEXT_BAR

If you enter the PLOT TITLE statement after PLOT SORT_BAR (or PLOT BAR_ASCENDING), the sorting will not appear in the subsequent plots produced by PLOT NEXT_BAR.

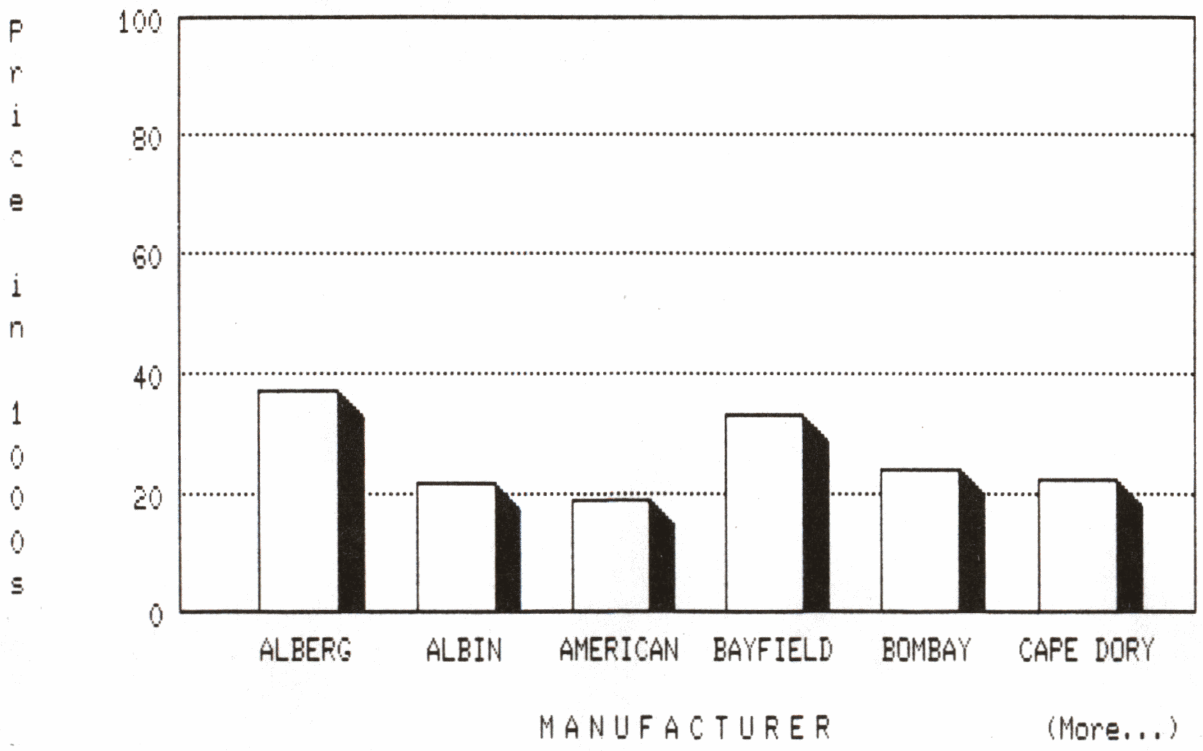
Example

The following steps show how to use the PLOT NEXT_BAR statement:

1. Create a bar chart or histogram using a PLOT BAR or PLOT HISTO statement.

   Dividing the PRICE argument by 1000 eliminates the default scientific notation that DATATRIEVE uses to display large numbers:

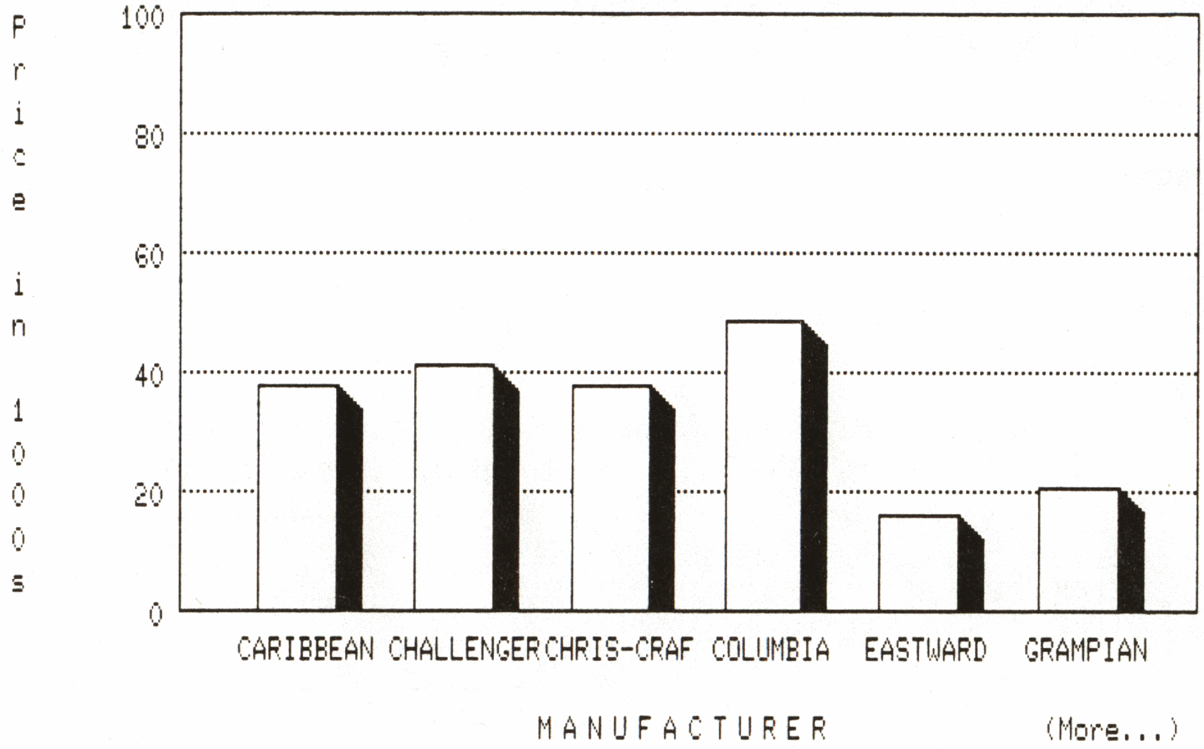
```
DTR> PLOT BAR_AVERAGE ALL BUILDER,
CON> PRICE/1000 ("Price in 1000s") OF
CON> YACHTS WITH PRICE GT 10000
```



ZK-00334-00

2. Enter the PLOT NEXT_BAR statement to display the next six bars of the plot. Because there are 50 bars in the entire plot, you would have to enter PLOT NEXT_BAR eight times to see all the bars:

```
DTR> PLOT NEXT_BAR
```



ZK-00335-00

4.26 PLOT PAUSE Statement

Delays processing the next plot statement in a DATATRIEVE session for several seconds.

Format

```
PLOT PAUSE
```

Argument

The PLOT PAUSE statement takes no argument.

Result

The PLOT PAUSE statement delays processing the next statement for several seconds during your DATATRIEVE plots session.

Usage Note

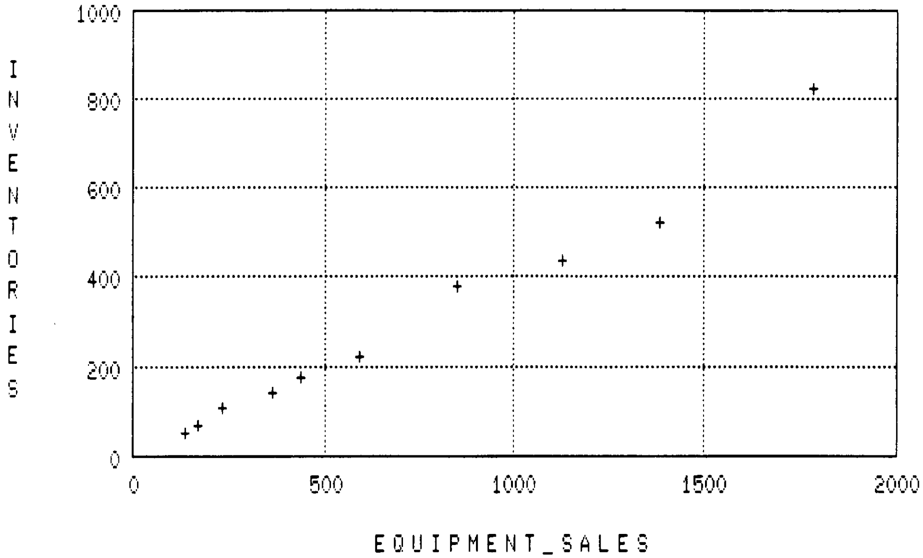
Even though this plot appears to be very simple, you must still use a DIGITAL supported ReGIS terminal with the PLOT PAUSE statement.

Example

The following example shows how you can use PLOT PAUSE to view one plot before DATATRIEVE processes the next plot statement:

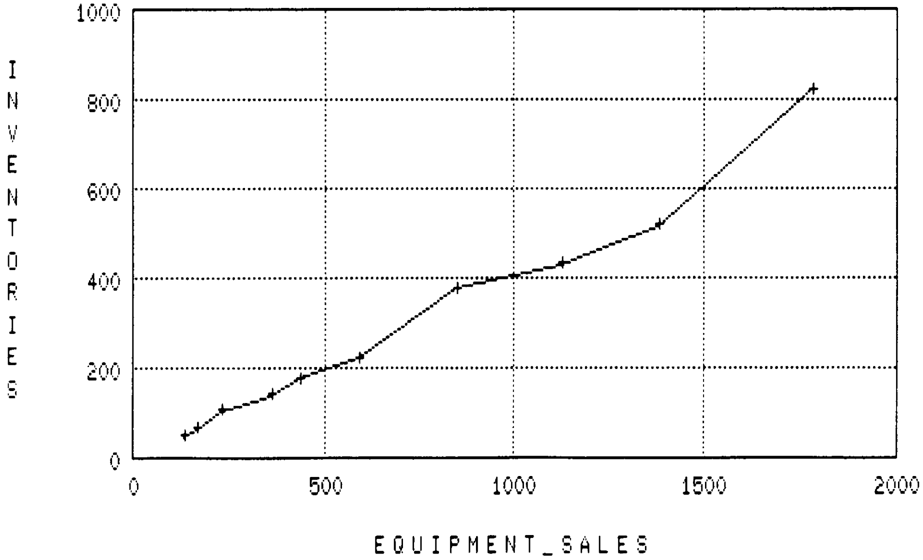
1. DATATRIEVE executes the PLOT X_Y statement. It then produces the scattergraph plot:

```
DTR> PLOT X_Y ALL EQUIPMENT_SALES,
CON> INVENTORIES OF ANNUAL_REPORT THEN
CON> PLOT PAUSE THEN
CON> PLOT CONNECT
```



ZK-00336-00

2. Because you entered the PLOT PAUSE statement, DATATRIEVE now pauses several seconds. It then executes the next plot statement, PLOT CONNECT, producing the following plot.



ZK-00337-00

4.27 PLOT PIE Statement

Plots a pie chart with the slices of a pie representing the number of occurrences of each unique field value. The slices represent percentages proportional to the whole pie (or 100 percent).

Format

```
PLOT PIE  [USING]  [ALL]  arg  [OF rse] [ON { file-spec } ]
                                        [   { *.prompt  } ]
```

Argument

arg

The PLOT PIE statement takes one argument, which can be a field name or other value expression.

Results

- The PLOT PIE statement groups records according to the argument you provide. It counts the number of records for each unique field value or value expression in each group. It then calculates what percentage of the whole each group represents.
- PLOT PIE plots these percentages as slices of a pie chart. Each slice is proportional to the sum of the slices, or the whole pie.
- PLOT PIE prints the percentage and a label for groups with a large enough slice to accommodate these labels. Groups with a small percentage (and thus a small slice) might not have a label or percentage displayed.

Usage Notes

Both PLOT PIE and PLOT VALUE_PIE create pie shaped charts:

- PLOT PIE groups and counts the number of occurrences for each unique field value or value expression. It then calculates what percentage of the whole each group represents. It plots these percentages as slices of a pie chart with each slice proportional to the sum of the slices.

- In contrast, PLOT VALUE_PIE groups records, calculates the total value of the field name or other value expression (value, not number or count), and then plots this total value as a percentage of the whole pie.
- You can specify a title other than the default title by using the PLOT TITLE statement.

  You can also customize the title by including a label string following the field name or other value expression:

  ```
  DTR> PLOT PIE DEPT ("Department") OF PERSONNEL
  ```

- You can use the following utility plots with the PLOT PIE statement:
  - PLOT BIG
  - PLOT HARDCOPY
  - PLOT PAUSE
  - PLOT RE_PAINT
  - PLOT TITLE

  See the section on each of these utility plot statements for more information.

Examples

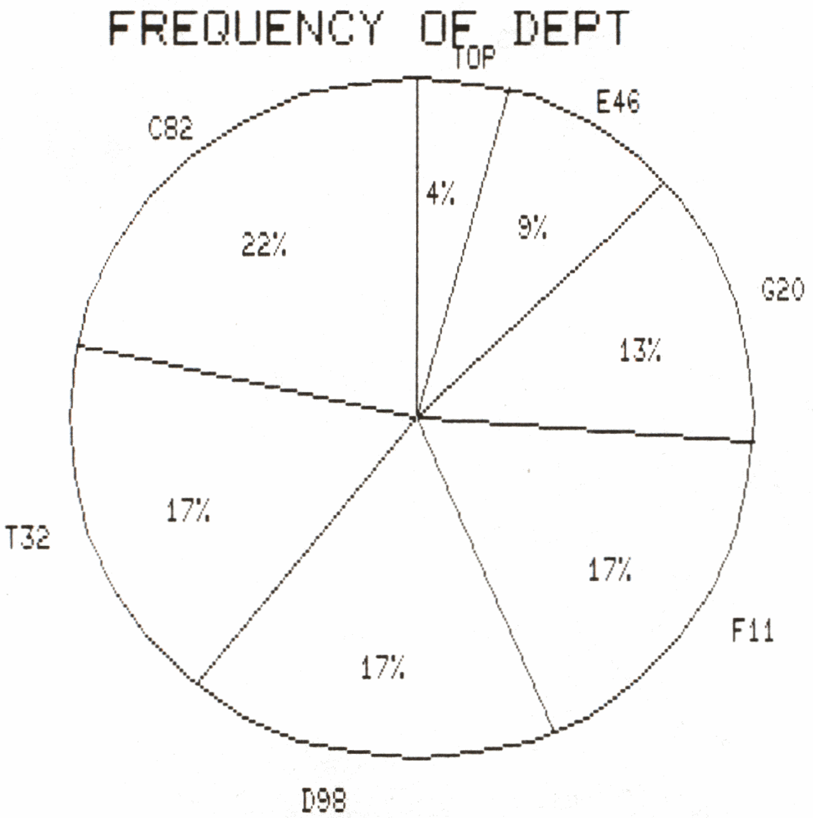
Example 1

Use the PLOT PIE statement to plot a pie chart using the PERSONNEL domain.

The total count of records for each department is calculated. These totals are plotted as percentages proportional to the total of all departments:

```
DTR> PLOT PIE DEPT OF PERSONNEL
```

Plot for Example 1



ZK-00338-00

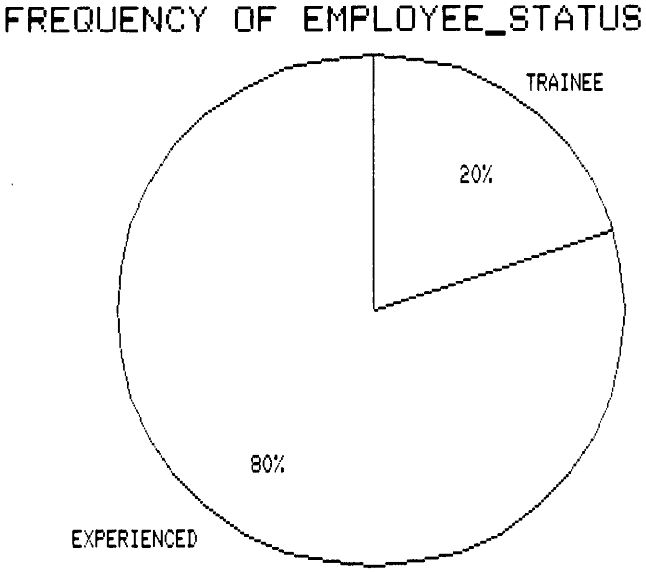
Example 2

Again using the PERSONNEL domain, use the PLOT PIE statement to plot the employees in department C82 according to status.

DATATRIEVE calculates the total count of employees for each status (TRAINEE and EXPERIENCED). These totals are plotted as percentages relative to the total of all employees in department C82.

```
DTR> PLOT PIE ALL STATUS OF
CON> PERSONNEL WITH DEPT = C82
```

Plot for Example 2



ZK-00339-00

4.28 PLOT RAW_BAR Statement

Creates a simple bar chart from value expressions you supply.

Format

```
                                               ┌    ┌ file-spec ┐ ┐
PLOT RAW_BAR   arg   [ [,  arg]...]   [OF rse] │ ON ┤           ├ │
                                               └    └ *.prompt  ┘ ┘
```

Arguments

arg

The PLOT RAW_BAR statement takes from 1 to 24 arguments:

- You can provide data values for arguments, as shown in the following examples.
- PLOT RAW_BAR does not use field values from a stream of records. You can select a record and use field names as value expressions, or you can use variables, numbers, and arithmetic expressions. If you include value expressions that are not field names, you should provide a label string with each one.

Results

The PLOT RAW_BAR statement evaluates each argument and plots its value as the height of a bar.

Usage Note

You can use the following utility plots with the PLOT RAW_BAR statement:

- PLOT BAR_ASCENDING
- PLOT BIG
- PLOT HARDCOPY
- PLOT LIMITS_X
- PLOT LIMITS_Y
- PLOT PAUSE

- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT SORT_BAR
- PLOT TITLE

See the section on each of these utility plot statements for more information.

Examples

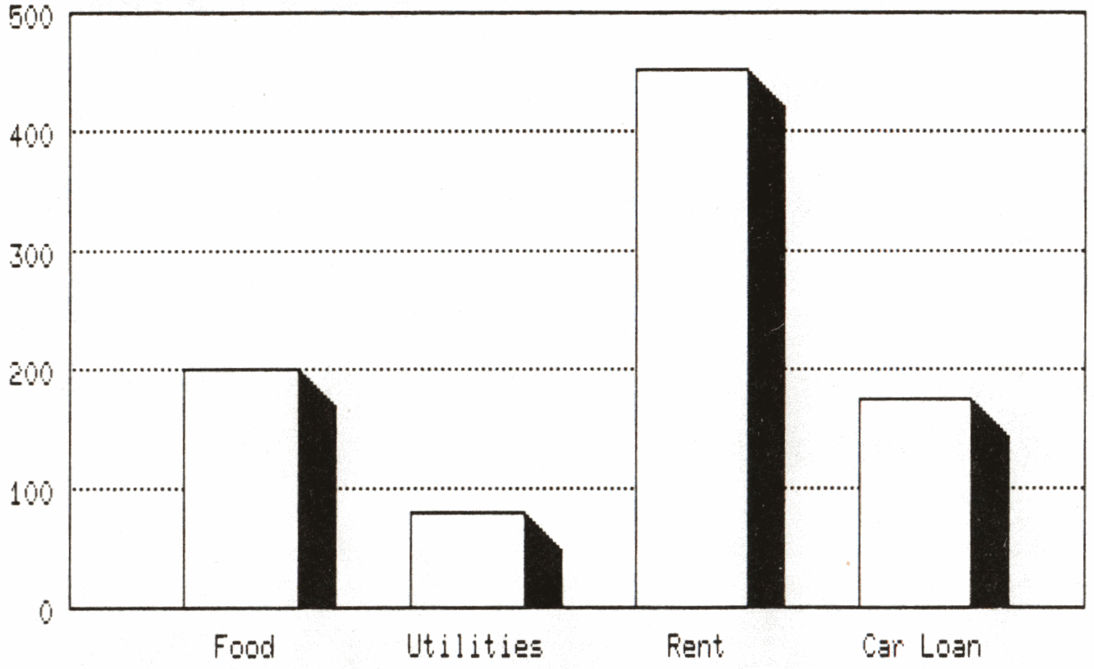
Example 1

Plot a bar chart by supplying specific data values for arguments.

The resulting bar chart plots the values you supplied:

```
DTR> PLOT RAW_BAR 200 ("Food"), 80 ("Utilities"),
CON> 450 ("Rent"), 175 ("Car Loan")
```

Plot for Example 1



ZK-00340-00

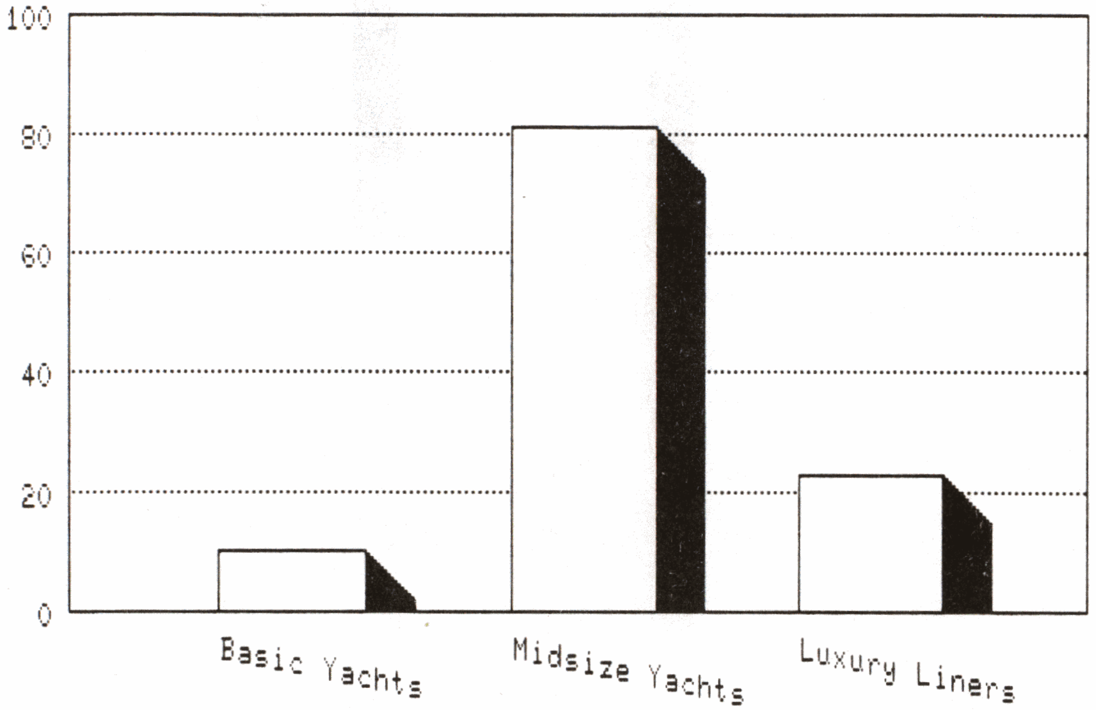
Example 2

Using the YACHTS domain and PLOT RAW_BAR, count and plot the number of yachts according to their length (LOA).

The resulting bar chart shows the count of yachts according to length. It labels each bar with the labels you supplied:

```
DTR> PLOT RAW_BAR COUNT OF YACHTS WITH LOA LT 25 ("Basic Yachts"),
CON> COUNT OF YACHTS WITH LOA BT 25 AND 35 ("Midsize Yachts"),
CON> COUNT OF YACHTS WITH LOA GT 35 ("Luxury Liners")
```

Plot for Example 2



ZK-00341-00

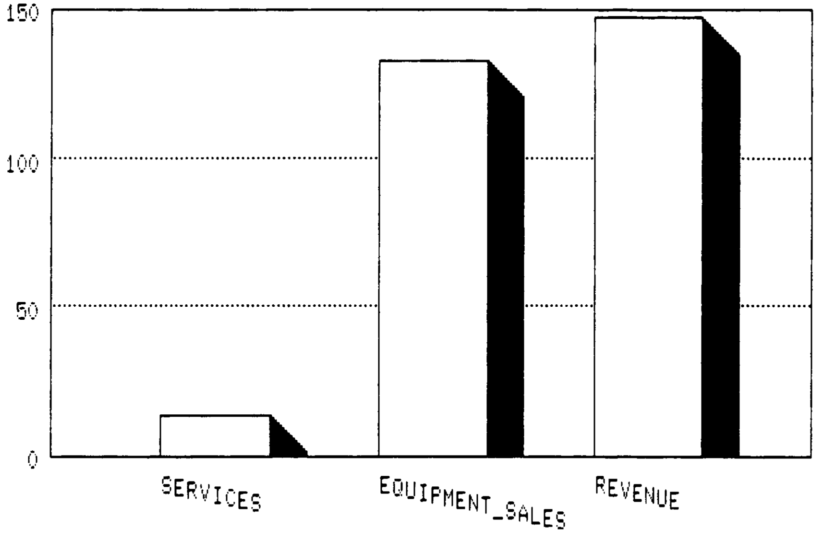
Example 3

The following example shows how you can select a record with the PLOT RAW_BAR statement. Using ANNUAL_REPORT, select the first record in the collection and plot the services, equipment sales, and revenue for that record.

The resulting bar chart shows the values for each of those arguments.

```
DTR> FIND ANNUAL_REPORT
DTR> SELECT 1
DTR> PLOT RAW_BAR ALL SERVICES,
CON> EQUIPMENT_SALES, REVENUE
```

PLOT RAW_BAR



ZK-00342-00

4.29 PLOT RAW_PIE Statement

Creates a simple pie chart that plots data you supply.

Format

```
                                                 ┌    ┌ file-spec ┐ ┐
PLOT RAW_PIE  arg  [ [,  arg]...]  [OF rse]      │ON  ┤           ├ │
                                                 └    └ *.prompt  ┘ ┘
```

Arguments

arg

The PLOT RAW_PIE statement takes from 1 to 24 arguments:

- You can provide data values for arguments, as shown in the examples.
- PLOT RAW_PIE does not use field names from a stream of records. You can select a record and use field names as value expressions, or you can use variables, numbers, and arithmetic expressions. If you include value expressions that are not field names, you should provide a label string with each one.

Results

- The PLOT RAW_PIE statement evaluates each argument, then calculates what percentage of the whole each argument represents.
- The PLOT RAW_PIE statement next plots these percentages as slices of a circular, pie-shaped chart. Each slice is proportional to the whole pie.
- PLOT RAW_PIE prints the percentage and a label for arguments with a large enough slice to accommodate these labels. Arguments with a small percentage (and thus a small slice) might not have a label or percentage displayed.

Usage Note

You can use the following utility plots with the PLOT RAW_PIE statement:

- PLOT BIG
- PLOT HARDCOPY

- PLOT PAUSE
- PLOT RE_PAINT
- PLOT TITLE

See the section on each of these utility plot statements for more information.

Examples

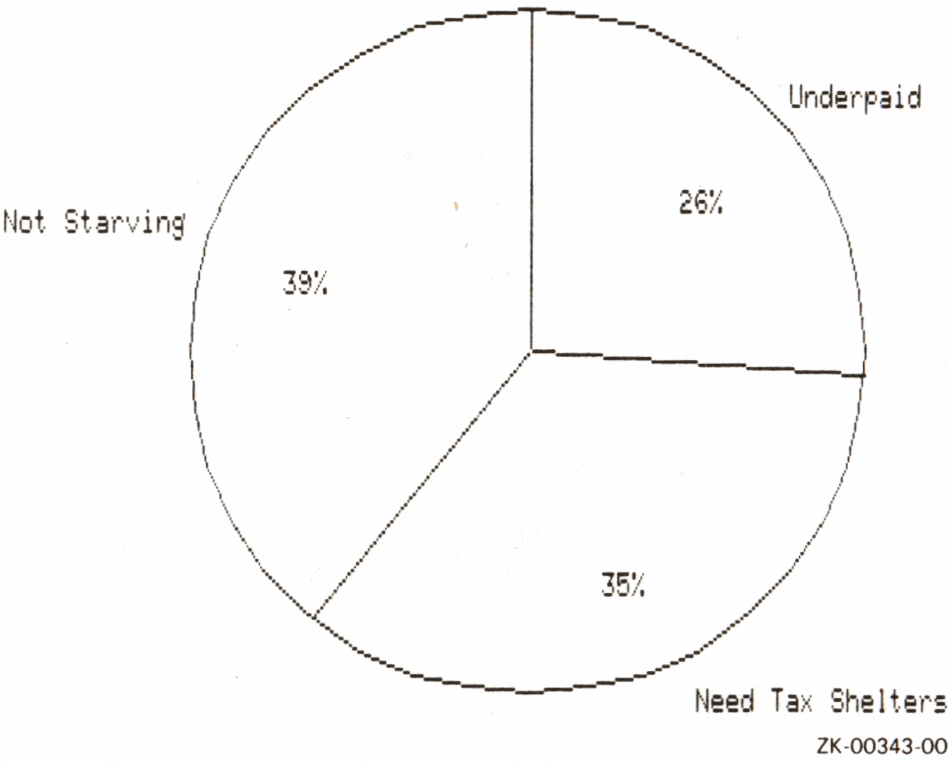
Example 1

Using the PERSONNEL domain and PLOT RAW_PIE, count and plot the number of employees according to the salary groupings you specify.

The resulting pie chart shows the count of employees according to the salary groupings you specified. The slices are labeled with the label strings you supplied:

```
DTR> PLOT RAW_PIE COUNT OF PERSONNEL WITH
CON> SALARY LT 30000("Underpaid"),
CON> COUNT OF PERSONNEL WITH
CON> SALARY BT 30000 45000 ("Not Starving"),
CON> COUNT OF PERSONNEL WITH
CON> SALARY GT 45000 ("Need Tax Shelters")
```

Plot for Example 1



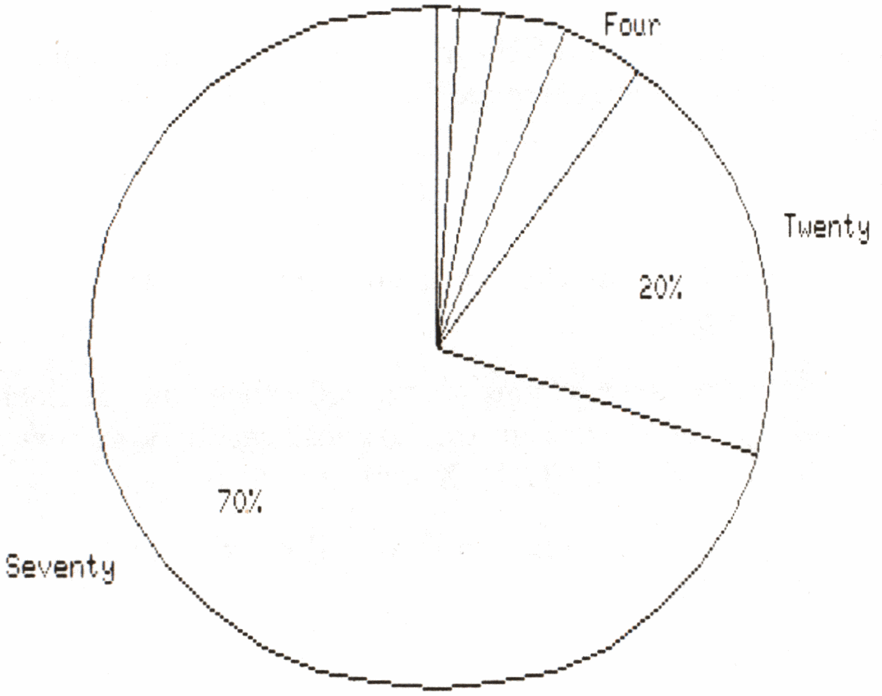
Example 2

Plot a pie chart by supplying specific data values and labels for arguments.

The resulting pie chart plots the values you supplied. Note that the percentages and labels are printed only for the slices large enough to accommodate these labels:

```
DTR> PLOT RAW_PIE 70 ("Seventy"), 20 ("Twenty"),
CON> 4 ("Four"), 3 ("Three"), 2 ("Two"), 1 ("One")
```

Plot for Example 2



ZK-00344-00

4.30 PLOT REFERENCE_X Statement

Plots a solid vertical reference line on bar charts, line graphs, and scattergraphs to mark a point on the horizontal (X) axis that has special meaning.

Format

```
PLOT REFERENCE_X  n
```

Argument

n

Is a value expression identifying the value of the horizontal (X) axis where you want DATATRIEVE to draw a solid reference line. The value of n must be within the range of the X values on the plot.

Results

- PLOT REFERENCE_X adds a solid line at the point you specify on the X axis of the most recently generated plot.
- DATATRIEVE adds to the existing plot when you plot a reference line; it does not replot it. DATATRIEVE adds a reference line to the most recently generated plot for each use of the PLOT REFERENCE_X statement.
- DATATRIEVE returns an error message for entries that fall outside the range of values for the horizontal (X) axis.

Usage Notes

- Be sure that the values on the horizontal (X) axis are numeric.
- Look at the X labels on the existing plot to see what units are used for X values. Use the same units for the n value in the PLOT REFERENCE statement.
- Use PLOT REFERENCE_X repeatedly to add multiple reference lines along the horizontal (X) axis of an existing plot.

Restrictions

- You cannot draw reference lines for date values or string values.
- You cannot specify a scientific notation value for the X coordinate of a reference line.

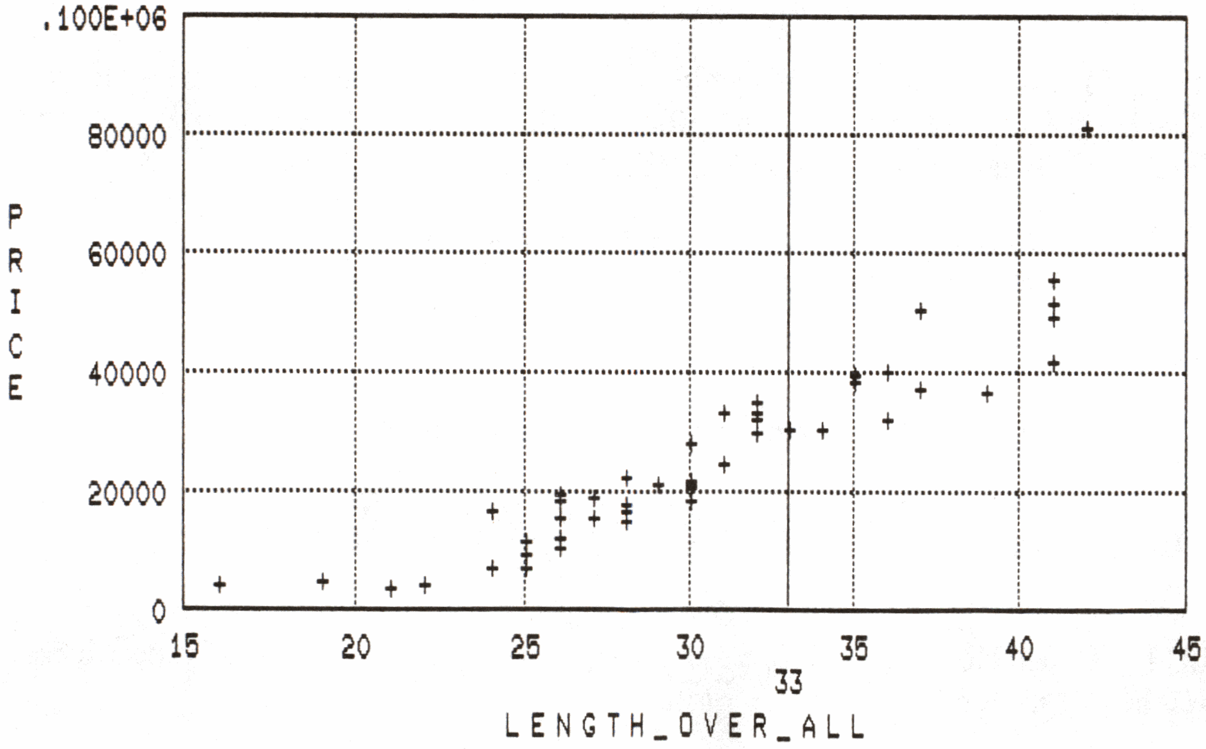
- DATATRIEVE uses scientific notation for labels with integer values greater than 100,000. In addition, DATATRIEVE rounds some real number values from the vertical (Y) axis or uses scientific notation values rounded to three digits.

  You can avoid this scientific notation by dividing the argument for the Y axis by 1000.

Example

Create a collection from the YACHTS domain. Use PLOT X_Y to create a scattergraph showing the trend that as LOA increases, so does price. Then select a value from the range of values along the horizontal (X) axis at which you want to draw a solid reference line. Use that value as the argument of the PLOT REFERENCE_X statement. The keyword ALL is required because the plot statement refers to the collection created by the first statement.

```
DTR> FIND YACHTS WITH PRICE NE 0
DTR> PLOT X_Y ALL LOA, PRICE THEN
CON> PLOT REFERENCE_X 33
```



ZK-00345-00

4.31 PLOT REFERENCE_Y Statement

Plots a solid horizontal reference line on bar charts, line graphs, and scattergraphs to mark a point on the vertical (Y) axis that has special meaning.

Format

```
PLOT REFERENCE_Y  n
```

Argument

n

Is a value expression identifying the value of the vertical (Y) axis where you want DATATRIEVE to draw a solid reference line. The value of n must be within the range of values for the vertical (Y) axis.

Results

- PLOT REFERENCE_Y adds a solid line at the point you specify on the vertical (Y) axis of the most recently generated plot.
- DATATRIEVE adds to the existing plot when you plot a reference line; it does not replot it. DATATRIEVE adds a reference line to the most recently generated plot for each use of the PLOT REFERENCE_Y statement.
- DATATRIEVE returns an error message for entries that fall outside the range of values for the vertical (Y) axis.

Usage Notes

- Be sure that values on the vertical (Y) axis are numeric.
- Look at the Y labels on the existing plot to see what units are used for Y values. Use the same units for the n value in the PLOT REFERENCE_Y statement.
- Use PLOT REFERENCE_Y repeatedly to add multiple reference lines along the vertical (Y) axis of an existing plot.

Restrictions

- You cannot draw reference lines for date values or string values.
- You cannot specify a scientific notation value for the Y coordinate of a reference line.
- DATATRIEVE uses scientific notation for labels with integer values greater than 100,000. In addition, DATATRIEVE rounds some real number values from the vertical (Y) axis or uses scientific notation values rounded to three digits.

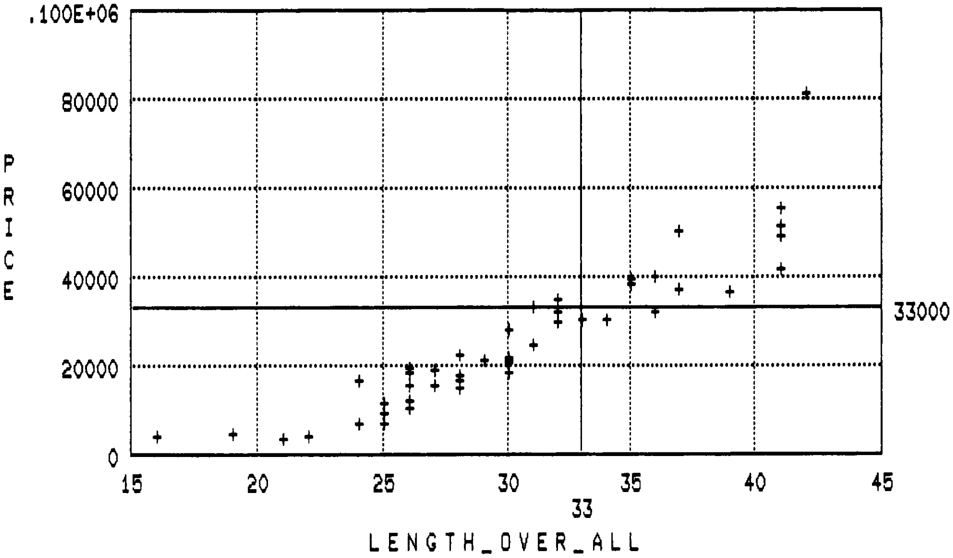
  You can avoid this scientific notation by dividing the argument for the Y axis by 1000.

Example

Create a collection from the YACHTS domain. Use PLOT X_Y to create a scattergraph showing the trend that as LOA increases, so does price. Then select a value from the range of values along the horizontal (X) axis at which you want to draw a solid reference line. Use that value as the argument of the PLOT REFERENCE_X statement. Next select a value from the vertical (Y) axis and use it with the PLOT REFERENCE_Y statement to plot a reference line along the vertical (Y) axis.

```
DTR> FIND YACHTS WITH PRICE NE 0
DTR> PLOT X_Y ALL LOA, PRICE THEN
CON> PLOT REFERENCE_X 33 THEN
CON> PLOT REFERENCE_Y 33000
```

This example produces a base scattergraph, plots a reference line on top of the base graph for the horizontal (X) axis, then plots a reference line on top of that for the vertical (Y) axis.


.100E+06
80000
60000
40000
20000
0
PRICE
33000
15
20
25
30
35
40
45
33
LENGTH_OVER_ALL

ZK-00346-00

4.32 PLOT RE_PAINT Statement

Restores the most recent plot to your screen.

Format

```
PLOT RE_PAINT
```

Argument

The PLOT RE_PAINT statement takes no argument.

Results

After you generate a plot, it can become distorted for a number of reasons, including:

- DATATRIEVE scrolls up the existing plot with each new command or statement you issue other than a plot statement.
- On some terminals, if the plot is large enough (such as a pie chart), typing a subsequent command can overwrite the plot displayed on your screen.

In each of the previous situations, the PLOT RE_PAINT statement restores the most recent plot to your screen.

Usage Notes

- DATATRIEVE erases the existing plot from your terminal as soon as you issue a new plot statement.
- Note that in the following sequence, DATATRIEVE restores the subsequent, most recent plot, not the first plot:

  1. You enter a plot statement
  2. You enter a subsequent plot statement
  3. You enter a PLOT RE_PAINT statement

Example

Enter the PLOT RE_PAINT statement to restore the most recent plot:

```
DTR> PLOT RE_PAINT
```

The most recent plot displays on your terminal.

4.33 PLOT SHADE Statement

Shades the area beneath the points in a scattergraph.

Format

```
PLOT SHADE
```

Argument

The PLOT SHADE statement takes no argument.

Result

After producing a scattergraph with one of the statements in the following list, enter the PLOT SHADE statement. PLOT SHADE shades the area beneath the points in the scattergraph, emphasizing trends.

Usage Notes

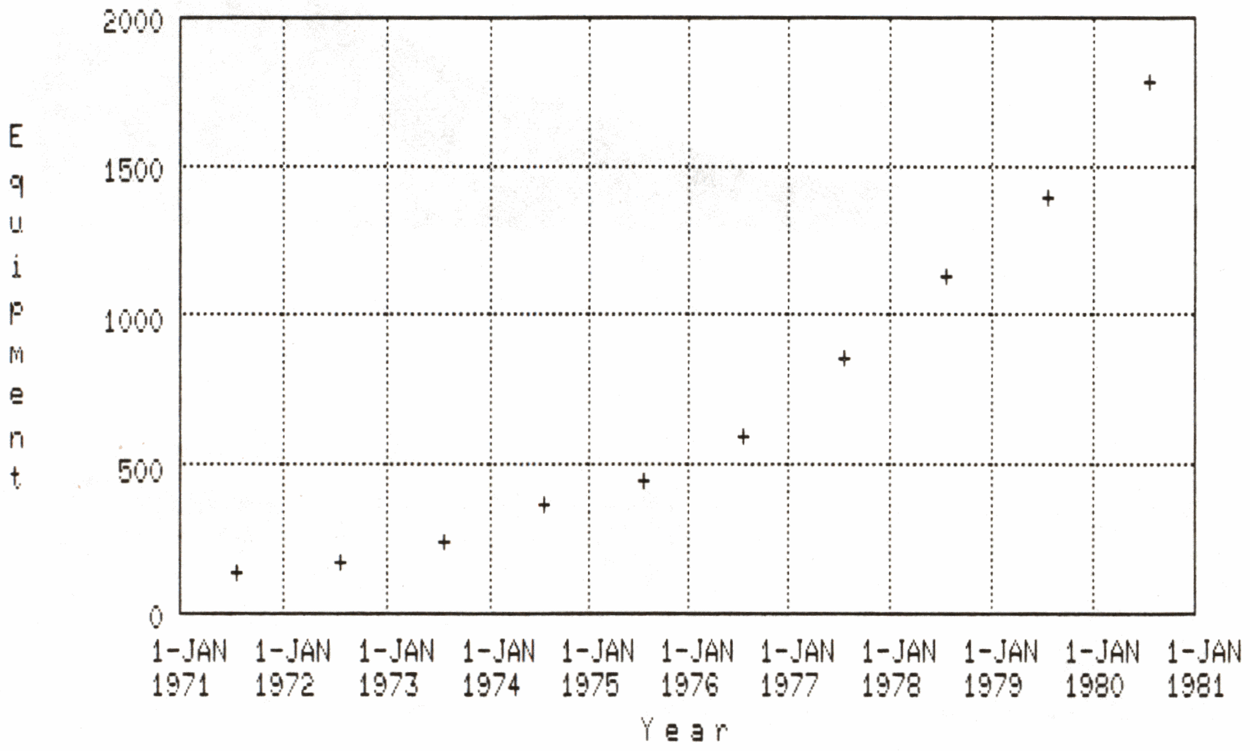
- Use the PLOT SHADE statement with any of the following plot statements:
  - PLOT DATE_LOGY
  - PLOT DATE_Y
  - PLOT LOGX_LOGY
  - PLOT LOGX_Y
  - PLOT X_LOGY
  - PLOT X_Y
- You can also use the PLOT CONNECT statement to emphasize trends in your data. It connects the points in a scattergraph.

Example

The following example shows a typical application of the PLOT SHADE statement:

1. Enter one of the plot statements from the preceding list to generate a scattergraph. This example uses the PLOT DATE_Y statement.

```
DTR> PLOT DATE_Y ALL DATE ("Year"),
CON> EQUIPMENT_SALES ("Equipment") OF ANNUAL_REPORT
```

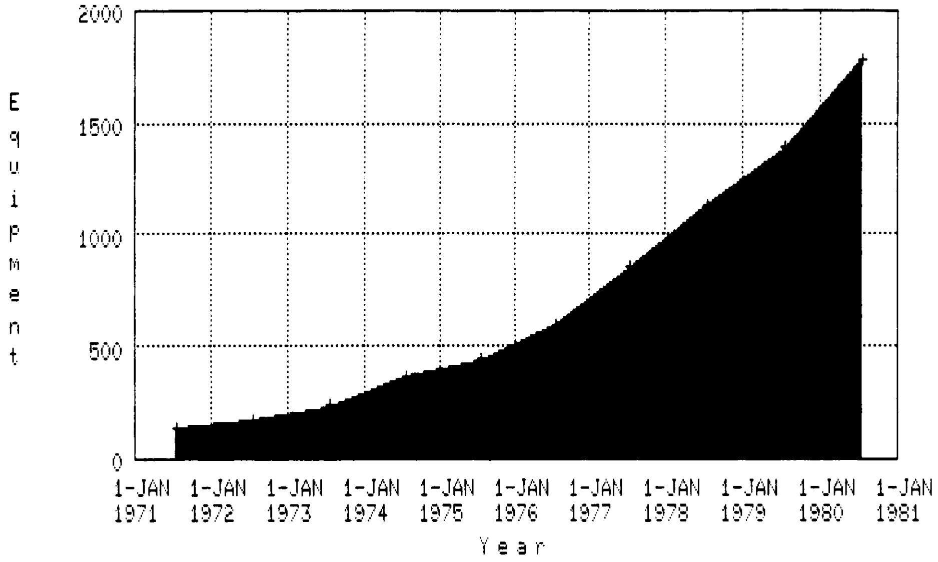


ZK-00349-00

2. Enter the PLOT SHADE statement. The area below the points is now shaded for emphasis.

```
DTR> PLOT SHADE
```

PLOT SHADE



ZK-00350-00

4.34 PLOT SORT_BAR Statement

Sorts the bars of a histogram or simple bar chart by descending height.

Format

```
PLOT SORT_BAR
```

Argument

The PLOT SORT_BAR statement takes no argument.

Result

After producing a histogram or bar chart with one of the statements in the following list, enter the PLOT SORT_BAR statement. PLOT SORT_BAR sorts the bars of the chart by descending height.

Usage Notes

- Use the PLOT SORT_BAR statement with any of the following statements:
  - PLOT BAR
  - PLOT BAR_AVERAGE
  - PLOT HISTO
  - PLOT RAW_BAR
- If you use the PLOT TITLE statement with the PLOT NEXT_BAR and the PLOT SORT_BAR statements, enter the statements in this sequence:

  1. PLOT BAR
  2. PLOT TITLE
  3. PLOT SORT_BAR
  4. PLOT NEXT_BAR

  If you enter the PLOT TITLE statement after PLOT SORT_BAR, the sorting will not appear in the subsequent plots produced by PLOT NEXT_BAR.

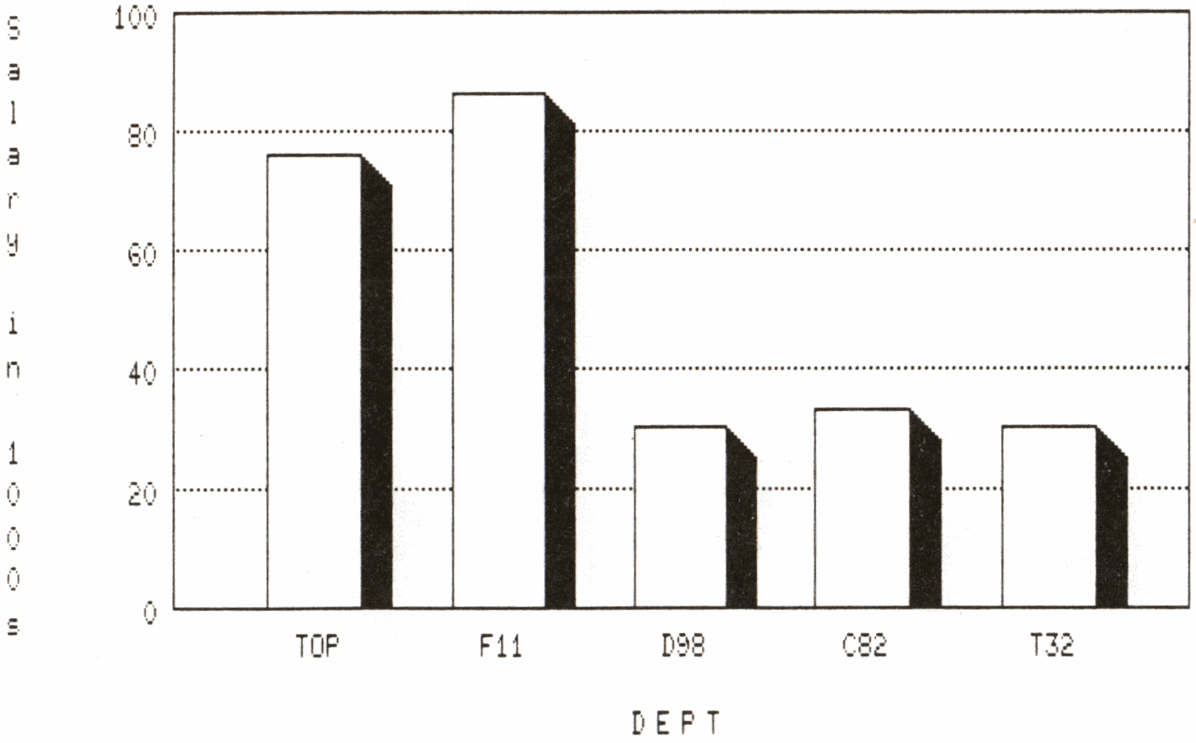
- Use the PLOT NEXT_BAR statement to see the next six bars of the chart if your chart has more than six bars.
- Use the PLOT BAR_ASCENDING statement to sort and replot bars in ascending order.

Example

The following example shows a typical application of the PLOT SORT_BAR statement:

1. Use the PERSONNEL domain to plot a simple bar chart showing the total salaries for the first six departments. The keyword ALL is required because the plot statement refers to the collection created by the first statement. Dividing the SALARY argument by 1000 eliminates default scientific notation that DATATRIEVE uses to display large numbers:

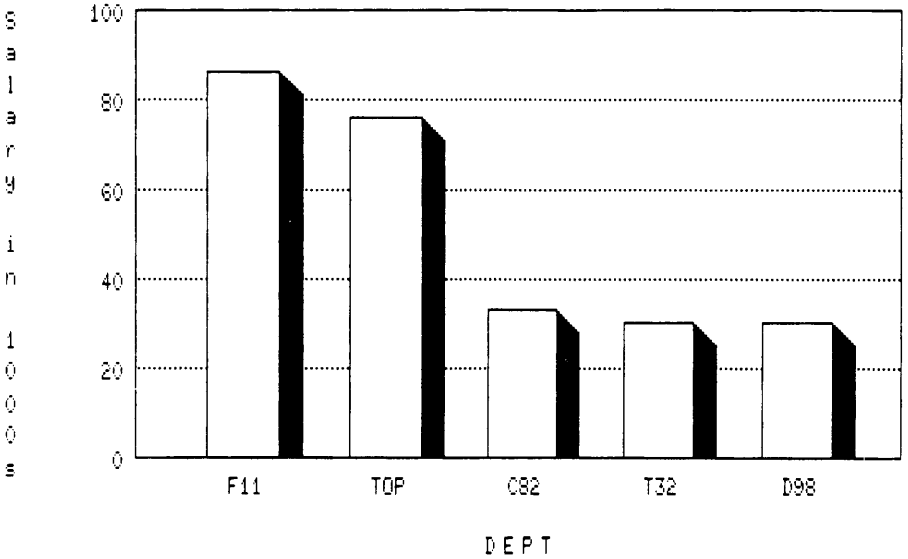
```
DTR> FIND FIRST 6 PERSONNEL
DTR> PLOT BAR ALL DEPT, SALARY/1000 ("Salary in 1000s")
```



ZK-00351-00

2. Enter the PLOT SORT_BAR statement to sort the bars of the chart by descending height:

```
DTR> PLOT SORT_BAR
```



ZK-00352-00

4.35 PLOT STACKED_BAR Statement

Creates a bar chart with one to three shaded values plotted on top of each other in stacked bars.

Format

```
PLOT STACKED_BAR  [USING]  [ALL]  arg,  arg  [,  arg  [,  arg] ]

        [OF rse]  [ON { file-spec } ]
                  [   { *.prompt  } ]
```

Arguments

arg

The PLOT STACKED_BAR statement takes two to four arguments, which can be field names or other value expressions:

- The first argument labels the bars on the horizontal (X) axis.
- Each of the next three arguments must evaluate to numbers. The values of each argument are plotted in a unique shade in the stacked bars and determine the height on the vertical (Y) axis.

Results

- The PLOT STACKED_BAR statement creates a bar chart with the shaded values stacked on top of each other.
- If there is room, DATATRIEVE prints a legend explaining which bars represent which values. If DATATRIEVE does not print the legend, you can produce it by subsequently typing PLOT LEGEND.

Usage Notes

- Most bar charts plot a maximum of six bars per chart. However, PLOT STACKED_BAR plots all the bars in one chart. Before using the PLOT STACKED_BAR statement with a large number of records, you should consider the effect this will have on the chart's readability.

PLOT STACKED_BAR is restricted in the number of records it can plot. If the statement contains two arguments, DATATRIEVE can plot up to approximately 480 records. For three arguments, it can plot approximately 260 records; for four arguments, it can plot approximately 180 records. Again, plots generated with PLOT STACKED_BAR with a large number of records would be very difficult to read.

- PLOT STACKED_BAR is similar to PLOT MULTI_BAR. Both statements plot three values. PLOT MULTI_BAR plots the values in clusters, side-by-side, while PLOT STACKED_BAR plots the values on top of each other.

- You can use the following utility plots with the PLOT STACKED_BAR statement:

  - PLOT BIG
  - PLOT CROSS_HATCH
  - PLOT HARDCOPY
  - PLOT LEGEND
  - PLOT LIMITS_X
  - PLOT LIMITS_Y
  - PLOT PAUSE
  - PLOT REFERENCE_X
  - PLOT REFERENCE_Y
  - PLOT RE_PAINT
  - PLOT TITLE

  See the section on each of these utility plot statements for more information.

- The PLOT STACKED_BAR statement does not group the records or sort the bars. You can specify a particular order by sorting the record stream (for example, FIND domain-name SORTED by field-name).

Examples

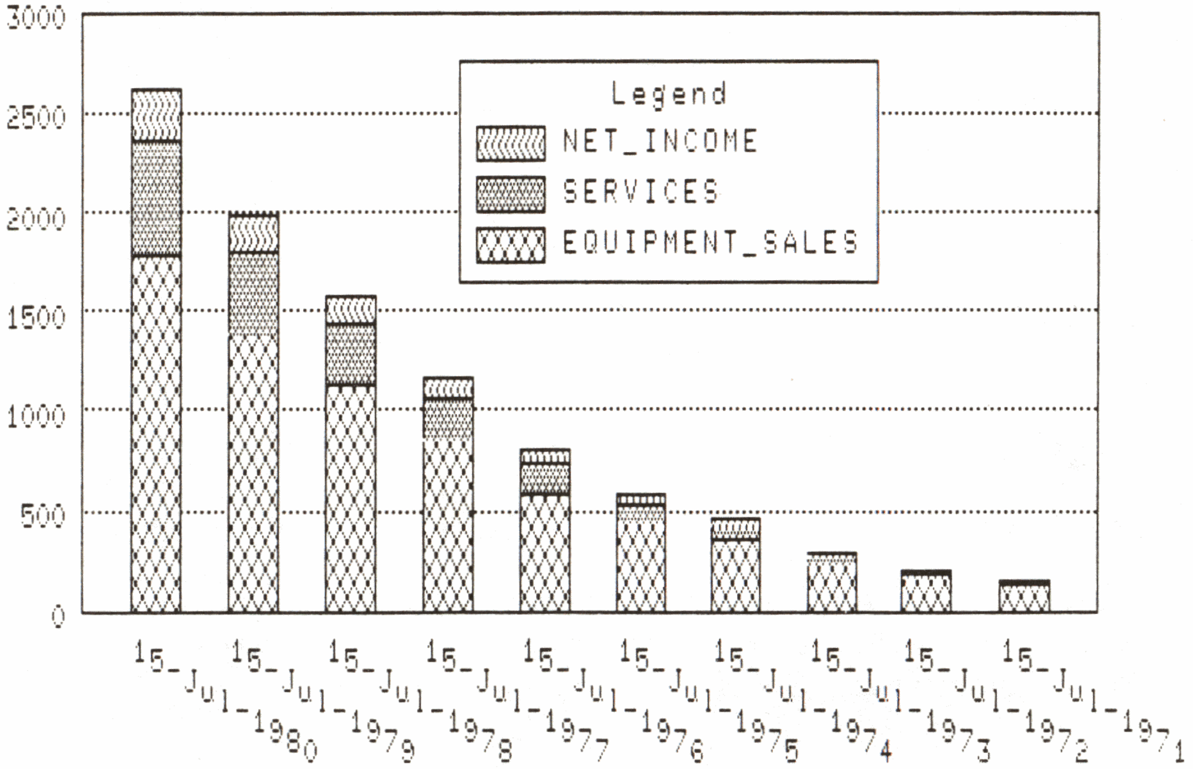
Example 1

Using the domain ANNUAL_REPORT, enter DATE as the first argument to label the horizontal (X) axis. Enter the arguments EQUIPMENT_SALES, SERVICE, and NET_INCOME to be plotted as stacked bars. The last three arguments determine the height of the bars on the vertical (Y) axis.

The resulting plot shows the annual figures for equipment sales, services, and net income. Each figure is plotted in a unique crosshatched pattern, with the figures for each year stacked on top of each other. The plot includes a legend identifying each crosshatch pattern.

```
DTR> PLOT STACKED_BAR ALL DATE,
CON> EQUIPMENT_SALES, SERVICES, NET_INCOME OF
CON> ANNUAL_REPORT SORTED BY DATE THEN
CON> PLOT CROSS_HATCH
```

Plot for Example 1



ZK-00353-00

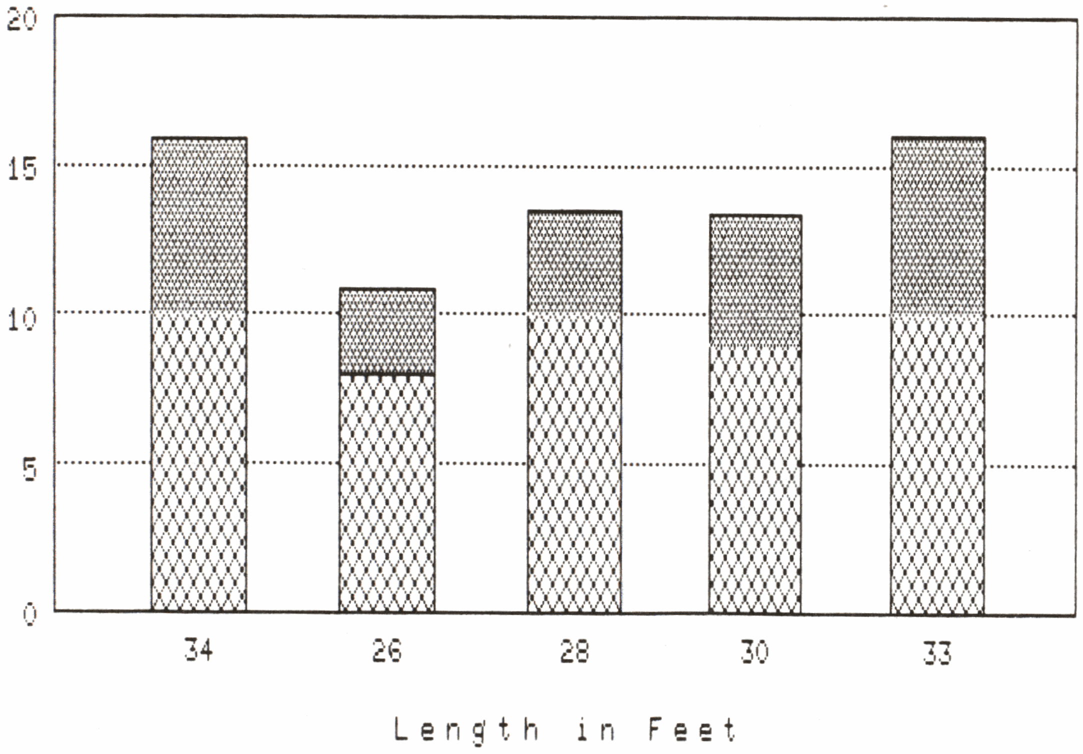
Example 2

Establish a collection from the domain YACHTS. Enter LOA as the first argument to label the horizontal (X) axis. Enter the arguments BEAM and DISP to be plotted as stacked bars. The values of these last two arguments determine the height of the bars on the vertical (Y) axis. The keyword ALL is required because the plot statement refers to the collection created by the first statement.

In the resulting plot, the scale on the vertical (X) axis applies to both arguments you supplied. For example, the first bar shows the figures for a 34-foot long Grampian. The lower crosshatch pattern for the bar indicates the yacht is 10 feet wide. The upper crosshatch pattern indicates the yacht weighs 6 tons:

```
DTR> FIND YACHTS WITH BUILDER EQ "GRAMPIAN"
DTR> PLOT STACKED_BAR ALL LOA ("Length in Feet"),
CON> BEAM ("Width in Feet"),
CON> DISP / 2000 ("Weight in Tons") THEN
CON> PLOT CROSS_HATCH
```

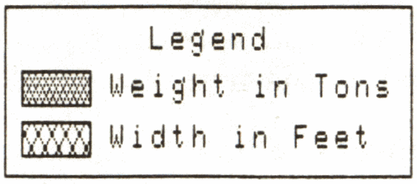
Plot for Example 2



ZK-00354-00

The plot did not leave enough room for the legend. Enter the PLOT LEGEND statement to generate the legend:

```
DTR> PLOT LEGEND
```



```
      Legend
▨ Weight in Tons
▨ Width in Feet
```

ZK-00355-00

4.36 PLOT TITLE Statement

Adds a title to the most recently generated plot.

Format

```
PLOT TITLE “text”  [/“text”]  [/“text”]
```

Argument

“text”

The PLOT TITLE statement requires a “text” string, which is a character string enclosed in matching quotation marks. The text for a plot title can be up to three lines long, with a slash (/) separating the lines of text. A single title line cannot exceed 49 characters including blanks. Each segment of a multiple line title must be enclosed in quotation marks. For example:

```
PLOT TITLE "text line 1"/"text line 2"/"text line 3"
```

Results

- PLOT TITLE replots the previous plot and shrinks the Y, or vertical, dimension to make room for a title at the top of the plot. Additional lines in the title mean more reduction of the vertical dimension.

- PLOT TITLE centers each line of the plot title.

Usage Note

- PLOT TITLE can be used with plot statements for all bar charts, line graphs, scattergraphs, and pie charts.

- Use PLOT TITLE before any other utility plot statements. PLOT TITLE replots the base plot and adds a title. If you have used a utility plot to add any utility features (for example, a linear regression line or hatching) to a base plot, you lose those additions when DATATRIEVE repaints the base plot to add the title. You can use the PLOT TITLE statement before or after PLOT LIMITS_X and PLOT LIMITS_Y, however.

- If you use the PLOT TITLE statement with the PLOT NEXT_BAR statement and either the PLOT SORT_BAR or PLOT BAR_ASCENDING statements, enter the statements in this sequence:

  1. PLOT BAR
  2. PLOT TITLE
  3. PLOT SORT_BAR (or PLOT BAR_ASCENDING)
  4. PLOT NEXT_BAR

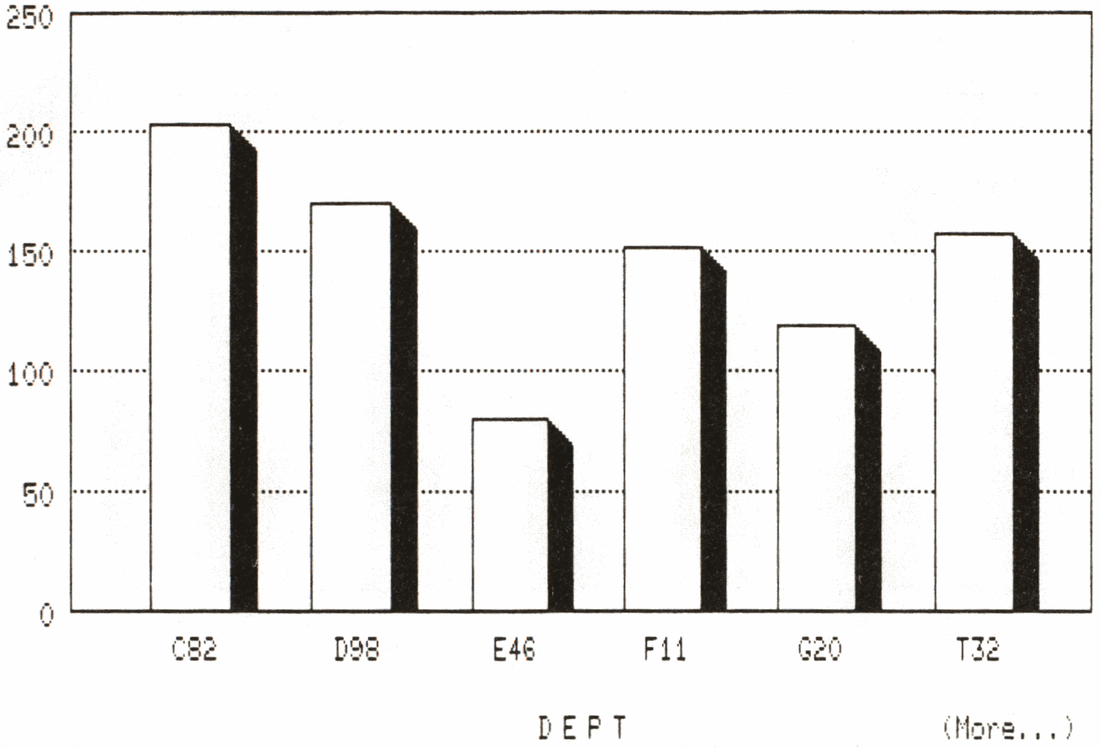
  If you enter the PLOT TITLE statement after PLOT SORT_BAR or PLOT BAR_ASCENDING, the sorting will not appear in the subsequent plots produced by PLOT NEXT_BAR.

Example

1. Using the PERSONNEL domain and the PLOT BAR statement, create a base plot showing the total salary for each department. Dividing the SALARY argument by 1000 eliminates the default scientific notation that DATATRIEVE uses to display large numbers.

   The resulting plot provides the base plot but needs a title to explain the units of measure used on the vertical (Y) axis.

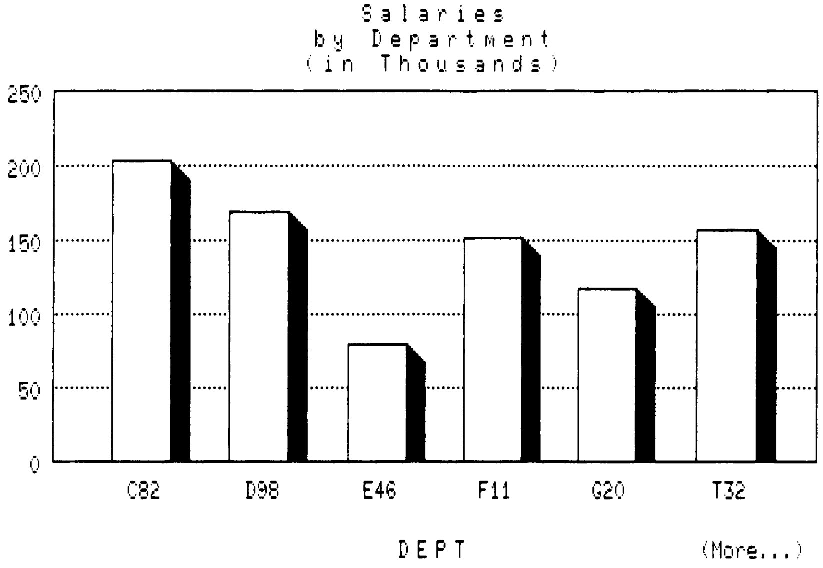
```
DTR> PLOT BAR ALL DEPT, SALARY/1000 OF
CON> PERSONNEL SORTED BY DEPT
```



ZK-00356-00

2. To add a title that also helps label the units of measure, use the PLOT TITLE statement:

```
DTR> PLOT TITLE "Salaries"/"by Department"/"(in Thousands)"
```


Salaries
by Department
(in Thousands)
250
200
150
100
50
0
C82
D98
E46
F11
G20
T32
DEPT
(More...)

ZK-00357-00

4.37 PLOT VALUE_PIE Statement

Plots a pie chart with the total values of an argument represented as slices of a pie. Each slice represents a percentage proportional to the whole pie.

Format

```
                                                     [    { file-spec }  ]
PLOT VALUE_PIE  [USING]  [ALL]  arg,  arg  [OF rse]  [ ON {           }  ]
                                                     [    { *.prompt  }  ]
```

Arguments

arg

The PLOT VALUE_PIE statement takes two arguments, which can be field names or other value expressions:

- The first argument groups the records.
- The second argument specifies which values will be totaled.

Results

- After grouping records according to the first argument, PLOT VALUE_PIE totals the values specified by the second argument. Each total value is plotted as a percentage (or slice) of the pie proportional to the total of the other slices.
- PLOT VALUE_PIE produces a default title for the entire plot showing "argument-2 by argument-1."
- PLOT VALUE_PIE prints the percentage and a label for groups with a large enough slice to accommodate these labels. Groups with a small percentage (and thus a small slice) might not have a label or percentage displayed.

Usage Notes

- Both PLOT PIE and PLOT VALUE_PIE create pie shaped charts:
  - PLOT PIE groups and counts the number of occurrences for each unique field value or value expression. It then calculates what percentage of the whole each group represents. It plots these percentages as slices of a pie chart with each slice proportional to the sum of the slices.

- In contrast, PLOT VALUE_PIE groups records, calculates the total value of the field name or other value expression (value, not number or count), and then plots this total value as a percentage of the whole pie.

- You can specify a title other than the default title by using the PLOT TITLE statement.

  You can also customize the title by including a label string following the field name or other value expression:

```
DTR> PLOT VALUE_PIE DEPT ("Department"),
CON> SALARY ("Salary Distribution") OF PERSONNEL
```

- You can use the following utility plots with the PLOT VALUE_PIE statement:

  - PLOT BIG
  - PLOT HARDCOPY
  - PLOT PAUSE
  - PLOT RE_PAINT
  - PLOT TITLE

  See the section on each of these utility plot statements for more information.

Examples

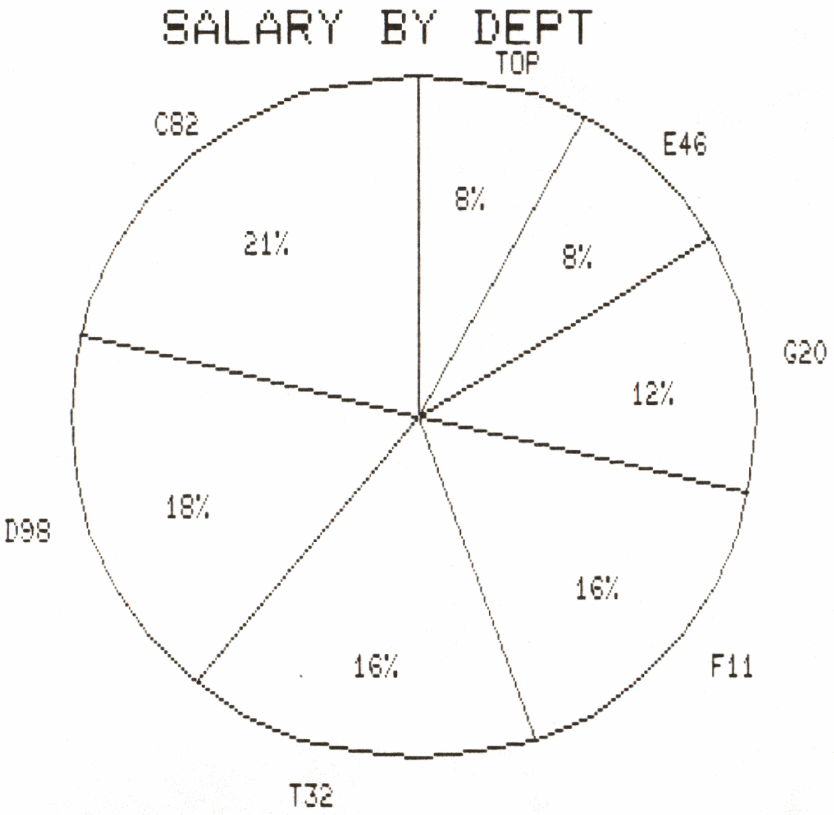
Example 1

Using the domain ANNUAL_REPORT, plot what percentage the total salaries of each department represent proportional to the total salaries of the whole company. Specify DEPT as the first argument and SALARY as the second argument.

PLOT VALUE_PIE plots the total salaries of each department as separate pie slices. Each pie slice represents a percentage proportional to the total salaries of all departments.

```
DTR> PLOT VALUE_PIE ALL DEPT, SALARY OF PERSONNEL
```

Plot for Example 1



ZK-00358-00

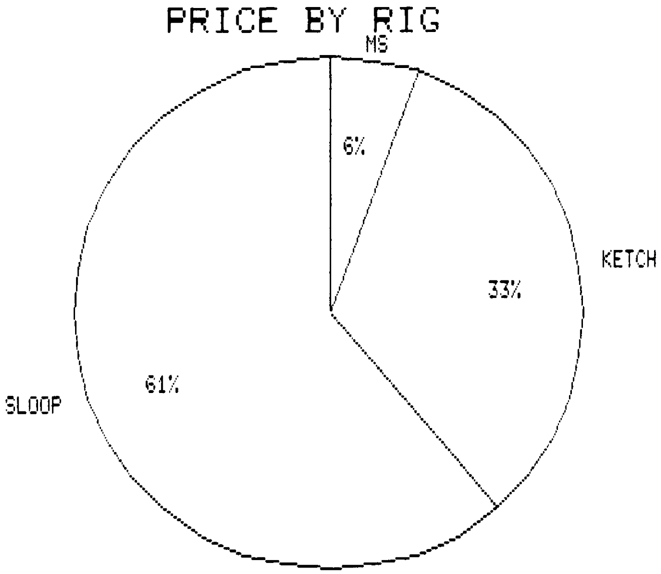
Example 2

Using the domain YACHTS, plot the percentage the total prices each type of yacht (RIG) represent compared to the total prices of all the yachts. Specify RIG as the first argument and PRICE as the second argument.

PLOT VALUE_PIE plots the total prices of each type of yacht as a separate pie slice. Each pie slice represents a percentage proportional to the total price of all yachts.

```
DTR> PLOT VALUE_PIE ALL RIG, PRICE OF
CON> YACHTS WITH PRICE GT 0
```

Plot for Example 2



ZK-00359-00

4.38 PLOT WOMBAT Statement

Displays a picture of a wombat.

Format

```
PLOT WOMBAT
```

Argument

The PLOT WOMBAT statement takes no argument.

Usage Notes

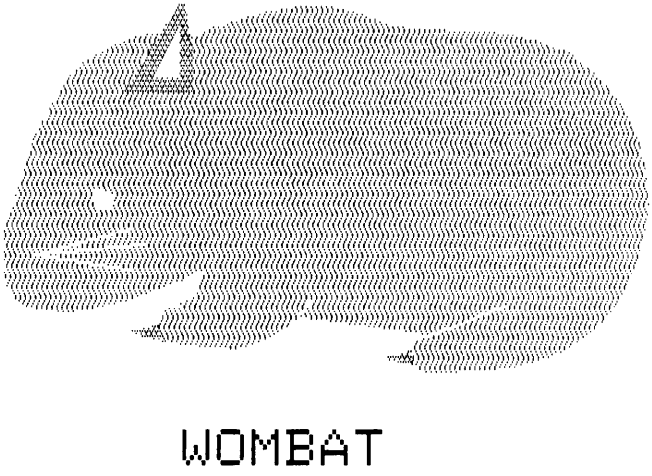
- Type HELP WOMBAT for more information about wombats.
- You can use the following utility plots with the PLOT WOMBAT statement:
  - PLOT BIG
  - PLOT CROSS_HATCH
  - PLOT HARDCOPY
  - PLOT RE_PAINT
  - PLOT TITLE

  See the section on each of these utility plot statements for more information.

Example

Type PLOT WOMBAT to display a wombat. Use PLOT CROSS_HATCH to enhance the display:

```
DTR> PLOT WOMBAT THEN
CON> PLOT CROSS_HATCH
```



ZK-00360-00

4.39 PLOT X_LOGY Statement

Creates a scattergraph that uses a logarithmic scale for the vertical (Y) axis, allowing you to accommodate a wide range of values.

Format

```
                                                     ┌    ┌ file-spec ┐ ┐
PLOT X_LOGY  [USING]  [ALL]  arg,  arg  [OF rse]    │ ON ┤           ├ │
                                                     └    └ *.prompt  ┘ ┘
```

Arguments

arg

The PLOT X_LOGY statement takes two arguments, which can be field names or other value expressions. The values of the arguments are plotted as points:

- The value of the first argument determines the locations of the points on the horizontal (X) axis.
- The value of the second argument determines the locations of the points along the vertical (Y) axis. Note the vertical (Y) axis uses a logarithmic scale.

  The second argument must be numeric and greater than 0 for each record.

Results

- The PLOT X_LOGY statement plots points according to their values on the horizontal (X) and vertical (Y) axes. These points are represented by the plus sign (+).
- Unlike a simple scattergraph (such as PLOT X_Y), PLOT X_LOGY uses a logarithmic scale for the vertical (Y) axis. With a logarithmic scale, each successive unit of length on the scaled axis covers a wider range of values. This allows you to plot a wider range of values in your graph.

PLOT X_LOGY

Usage Note

You can use the following utility plots with the PLOT X_LOGY statement:

- PLOT BIG
- PLOT CONNECT
- PLOT HARDCOPY
- PLOT LIMITS_X
- PLOT LIMITS_Y
- PLOT LR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT TITLE

See the section on each of these utility plot statements for more information.

Examples

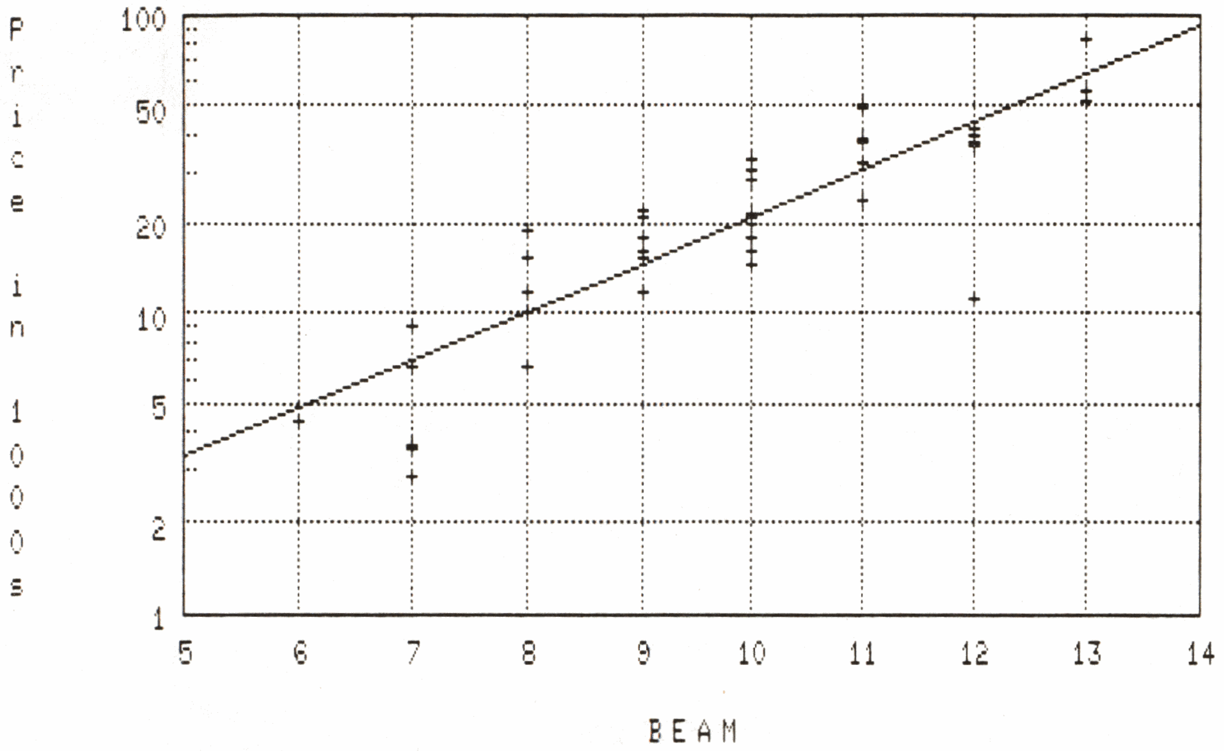
Example 1

Establish a collection using YACHTS. Use the PLOT X_LOGY statement to plot the beam sizes and prices. The keyword ALL is required because the plot statement refers to the collection created by the first statement.

The resulting plot shows how prices increase as beam size increases. It includes a linear regression line to emphasize the trend.

```
DTR> FIND YACHTS WITH PRICE GT 0 AND BEAM GT 0
DTR> PLOT X_LOGY ALL BEAM,
CON> PRICE/1000 ("Price in 1000s") THEN
CON> PLOT LR
```

Plot for Example 1



ZK-00361-00

Example 2

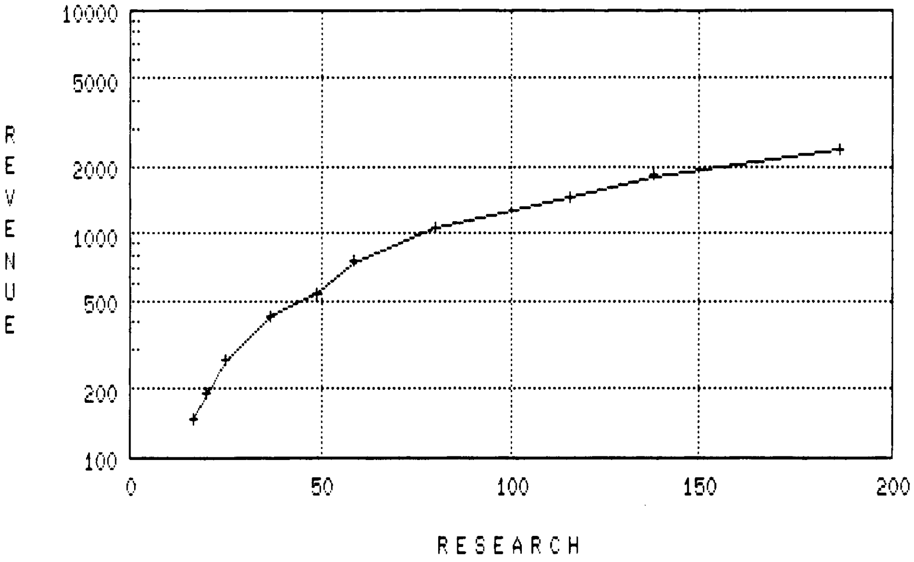
Use the PLOT X_LOGY statement to plot research funding and revenue from the ANNUAL_REPORT domain.

The resulting plot shows how revenue increases as research funding increases. The points are connected to emphasize the trend.

```
DTR> PLOT X_LOGY ALL RESEARCH, REVENUE OF
CON> ANNUAL_REPORT THEN
CON> PLOT CONNECT
```

PLOT X_LOGY

Plot for Example 2



ZK-00362-00

4.40 PLOT X_Y Statement

Creates a simple scattergraph with points plotted according to their values on the horizontal (X) and vertical (Y) axes.

Format

```
                                                 [    ( file-spec )]
PLOT X_Y  [USING]  [ALL]  arg,  arg  [OF rse]    [ON  {           }]
                                                 [    ( *.prompt  )]
```

Arguments

arg

The PLOT X_Y statement takes two arguments, which can be field names or other value expressions. The values of the arguments are plotted as a series of points:

- The value of the first argument determines the locations of the points on the horizontal (X) axis.
- The value of the second argument determines the locations of the points along the vertical (Y) axis.

  The second argument must be numeric and greater than 0 for each record.

Result

The PLOT X_Y statement plots points according to their values on the horizontal (X) and vertical (Y) axis. Each point is represented by a plus sign (+).

Usage Note

You can use the following utility plots with the PLOT X_Y statement:

- PLOT BIG
- PLOT CONNECT
- PLOT HARDCOPY
- PLOT LIMITS_X

- PLOT LIMITS_Y
- PLOT LR
- PLOT PAUSE
- PLOT REFERENCE_X
- PLOT REFERENCE_Y
- PLOT RE_PAINT
- PLOT TITLE

See the section on each of these utility plot statements for more information.

Examples

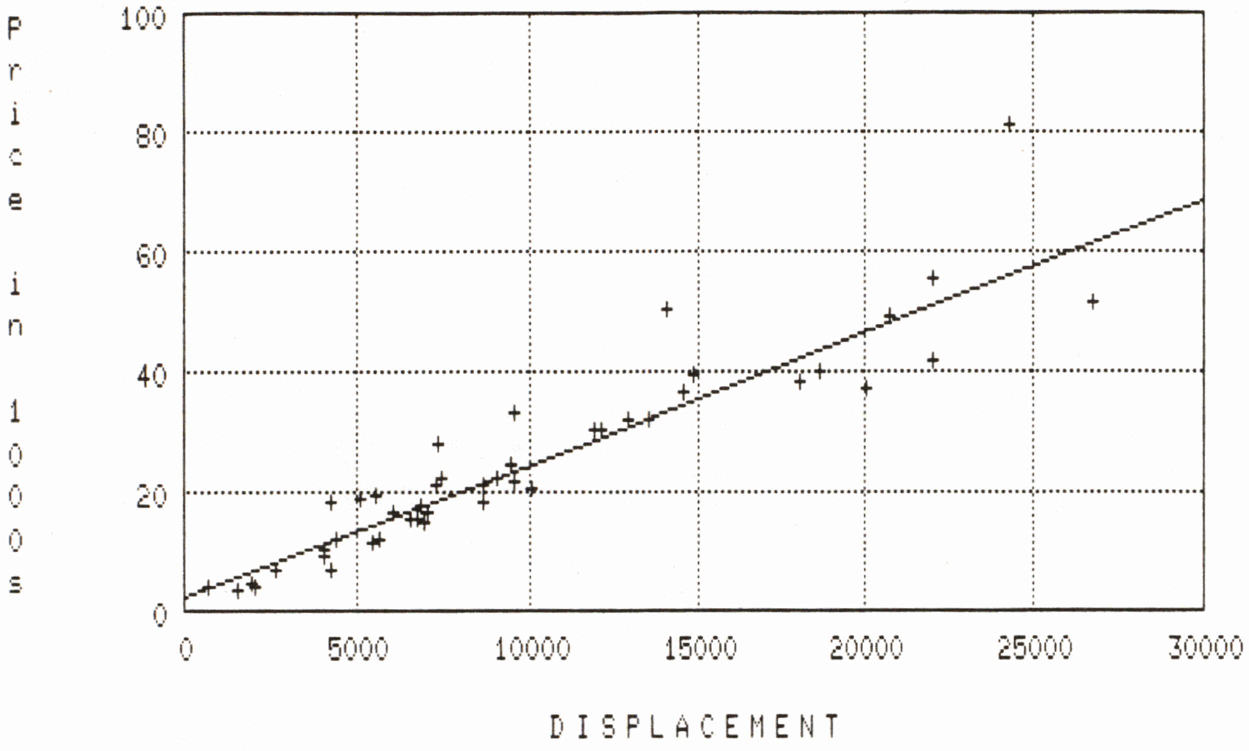
Example 1

Establish a collection using YACHTS. Use the PLOT X_Y statement to plot the displacement and prices. The keyword ALL is required because the plot statement refers to the collection created by the first statement. Dividing the PRICE argument by 1000 eliminates the default scientific notation that DATATRIEVE uses to display large numbers.

The resulting plot shows how the prices increase as displacement increases. It includes a linear regression line to emphasize the trend:

```
DTR> FIND YACHTS WITH PRICE GT 0 AND BEAM GT 0
DTR> PLOT X_Y ALL DISP,
CON> PRICE/1000 ("Price in 1000s") THEN
CON> PLOT LR
```

Plot for Example 1



ZK-00363-00

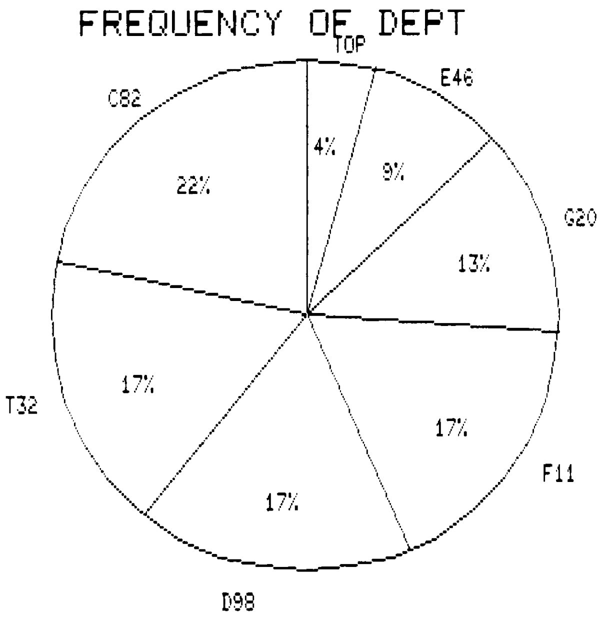
Example 2

Use the PLOT X_Y statement with the ANNUAL_REPORT domain to plot equipment sales and inventories.

The resulting plot shows how inventories change with the demand caused by increased equipment sales. The PLOT CONNECT statement emphasizes the trend by connecting the points.

```
DTR> FIND ANNUAL_REPORT WITH INVENTORIES GT 0
DTR> PLOT X_Y ALL EQUIPMENT_SALES, INVENTORIES THEN
DTR> PLOT CONNECT
```

Plot for Example 2



ZK-00338-00

Using DATATRIEVE Plots with Other Database Products 5

This chapter demonstrates the use of DATATRIEVE plots with other DIGITAL database products. These include:

- VAX DBMS
- VAX Rdb/VMS
- VAX Rdb/ELN
- VIDA

You might use DATATRIEVE with these database products because:

- You regularly use DATATRIEVE, but the data you need to do a particular job is in one of these other databases.
- You regularly use one of these products, but you need the DATATRIEVE Report Writer, graphics capabilities, or distributed access.

5.1 Using DATATRIEVE Plots with VAX DBMS

You can use DATATRIEVE graphics with any VAX DBMS database. To access and maintain a DBMS database, you may need to learn some extensions to the basic DATATRIEVE language. For more information about using a DBMS database with DATATRIEVE, see the *VAX DATATRIEVE User's Guide.*

The following section uses the sample DBMS PARTS database and the associated DBMS domains that are located in the CDD directory CDD$TOP.DTR$LIB.DEMO.DBMS. (See the sample database definitions and procedures in the *VAX DATATRIEVE User's Guide* for a list of all procedures and domains provided with the database.)

With DATATRIEVE graphics, you can plot data from the DBMS PARTS database to display graphic representations of information pertaining to the usage, ordering, and cost of various parts and supplies.

Set the CDD default to the directory that contains the database definitions used in this section:

```
DTR> SET DICTIONARY CDD$TOP.DTR$LIB.DEMO.DBMS
```

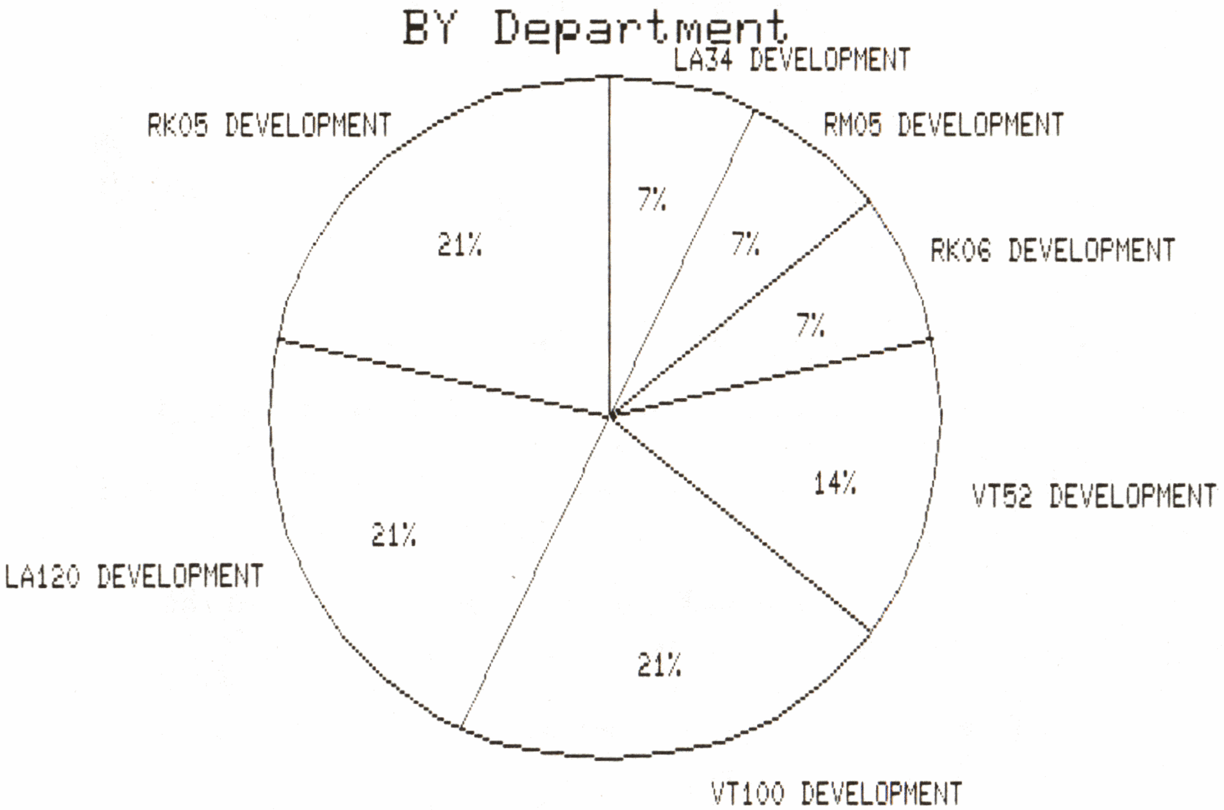
Once your database or any of its records have been properly defined, you have the option of readying the database directly or of readying any of the domains created from its records.

The following commands establish a collection of all departments involved in product development:

```
DTR> READY EMPLOYEES, DIVISIONS
DTR> FIND DIVISIONS WITH DIV_NAME CONT "DEVELOPMENT"
[7 records found]
DTR>
```

Next, entering the PLOT VALUE_PIE statement gives you a pie chart. Each section of the chart represents the percentage of employees associated with each product:

```
DTR> PLOT VALUE_PIE ALL DIV_NAME ("Department"),
CON> COUNT OF EMPLOYEES MEMBER CONSISTS_OF
```

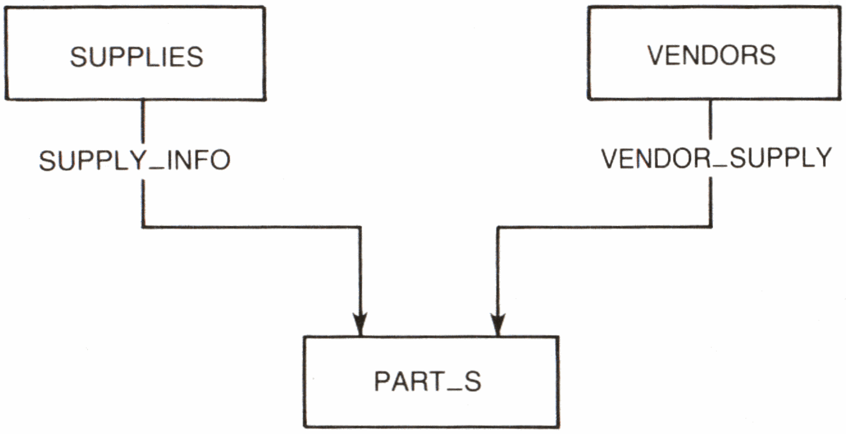


ZK-00364-00

In addition to common hierarchies such as EMPLOYEES and DIVISIONS, VAX DBMS also lets you model more complex relationships. For example, if a factory receives its parts from a variety of suppliers, it is likely that:

- A single vendor might supply many different parts
- A single part might be supplied by many different vendors

The following figure represents this VAX DBMS many-to-many relationship.



ZK-00365-00

The records SUPPLIES, VENDORS, and PART_S contain a large amount of data about the parts and the vendors who supply the parts. For example, the PART_S record contains a field that specifies the price for each part. A PRINT statement can tell you what the price is for each part, but a plot statement allows you to compare at a glance the relative price of each supplier. To plot the relative price of each supplier:

1. Ready all three DBMS domains. These READY statements also ready the set relationships between the records:

```
DTR> READY SUPPLIES, VENDORS, PART_S
DTR>
```

2. Flatten the DBMS hierarchy using the CROSS clause. Because the record PART_S links the SUPPLIES and VENDORS records, you must flatten two different hierarchies. You can use a FIND statement with two CROSS clauses:

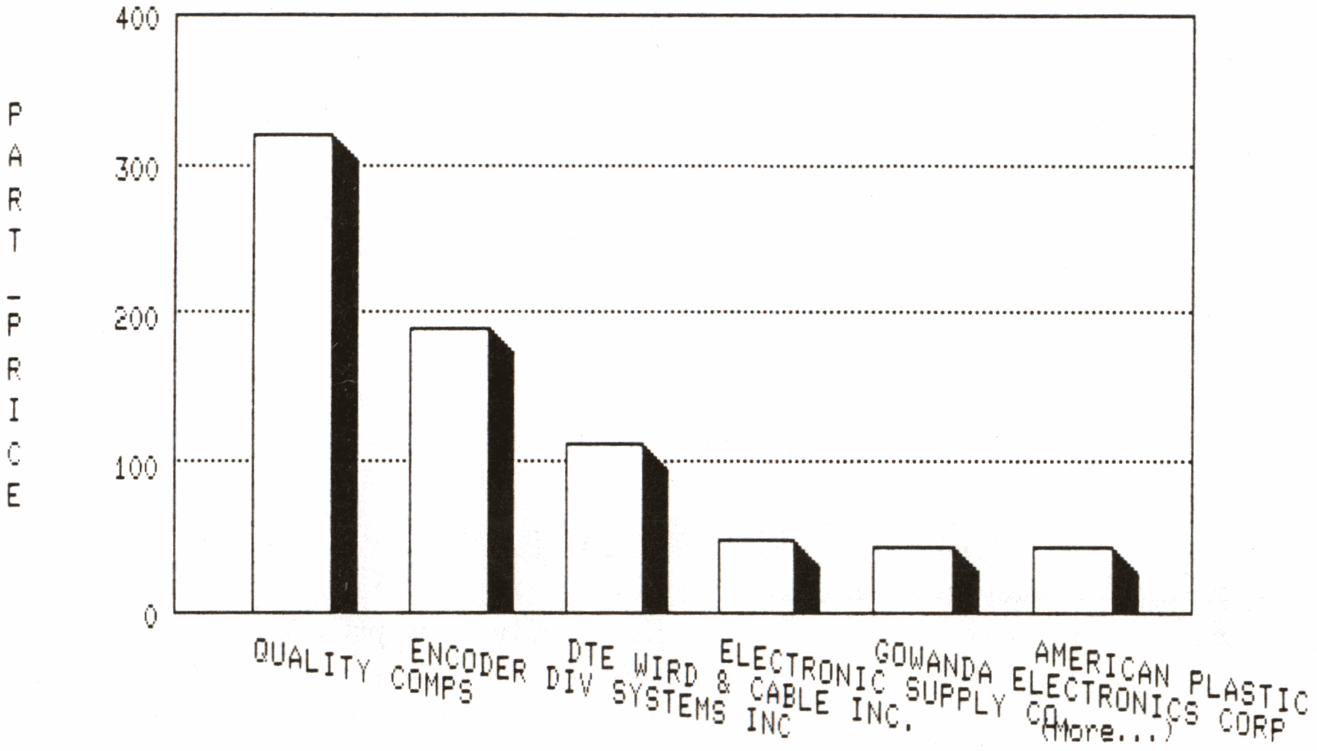
```
DTR> FIND SUPPLIES CROSS VENDORS OWNER
CON> VENDOR_SUPPLY CROSS PART_S OWNER PART_INFO
[65 records found]

DTR>
```

3. Plot the average part price by supplier:

```
DTR> PLOT BAR_AVERAGE ALL VEND_NAME, PART_PRICE THEN
CON> PLOT SORT_BAR
```

These plot statements produce two successive displays because of the THEN clause. The second of these displays, the PLOT SORT_BAR, shows the average price of parts for each supplier sorted in descending order by price.



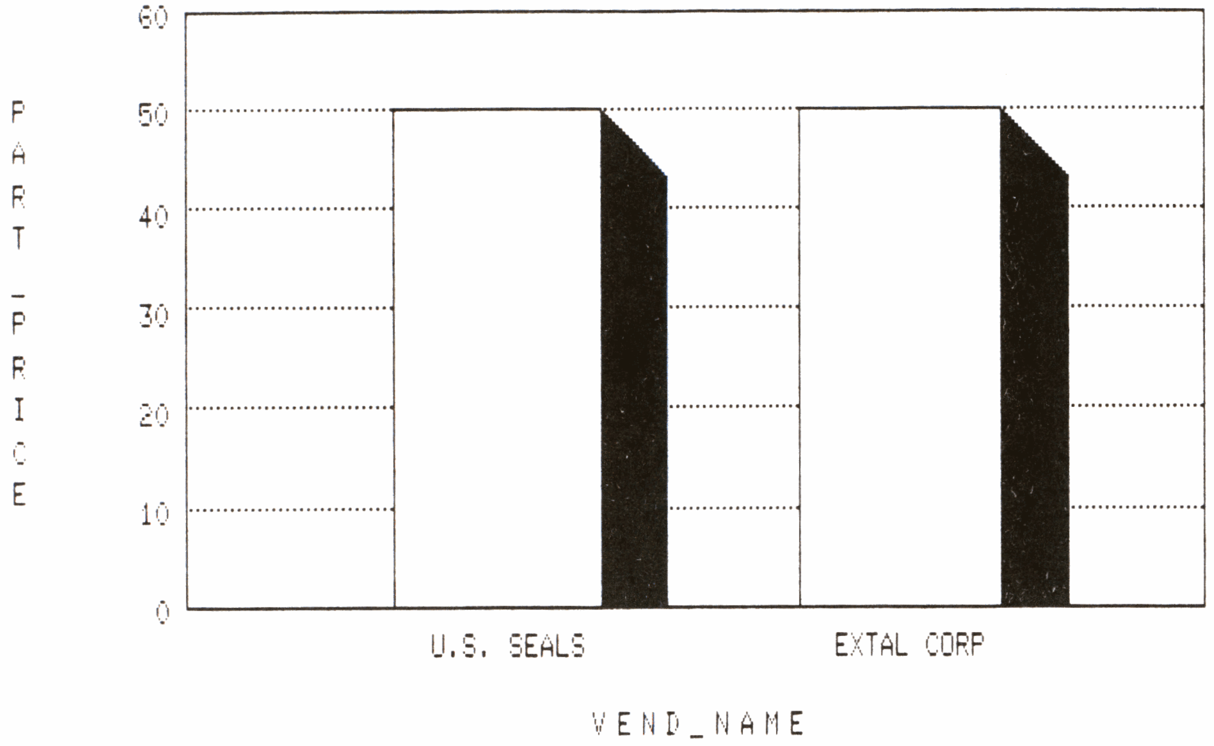
ZK-00366-00

4. Type PLOT NEXT_BAR at the DTR> prompt to see the remaining vendors.

Seeing the specific price each vendor charges for a specific part is also useful. The following example establishes a collection of vendors of keyboard casings, located in PART_S:

```
DTR> FIND KEYBOARDS IN CURRENT WITH PART_DESC CONT
CON> "KEYBOARD CASING"
[2 records found]
DTR> PLOT BAR ALL VEND_NAME, PART_PRICE
```

The resulting bar chart shows the prices two different vendors charge for keyboard casings.



ZK-00367-00

5.2 Using DATATRIEVE with Relational Databases

You can use DATATRIEVE graphics with the DIGITAL family of relational database management products. These include Rdb/VMS, Rdb/ELN, and the VAX-IBM Data Access (VIDA) facility.

To access and maintain a relational database, you need to learn some extensions to the DATATRIEVE language. For information about using the relational databases with VAX DATATRIEVE, see the *VAX DATATRIEVE User's Guide*.

The examples and references in this section refer to the sample Rdb/VMS PERSONNEL database installed with DATATRIEVE. (See the sample database definitions and procedures in the *VAX DATATRIEVE User's Guide* for a list of all procedures and domains provided with the database. See the chapter on using DATATRIEVE with relational databases in the *VAX DATATRIEVE User's Guide* for a figure that shows the relations and fields used in the database.) You can use the Rdb/VMS PERSONNEL sample database to plot data from personnel records to display graphic representations of a corporate personnel structure.

Set the CDD default to the directory that contains the database definitions used in this section:

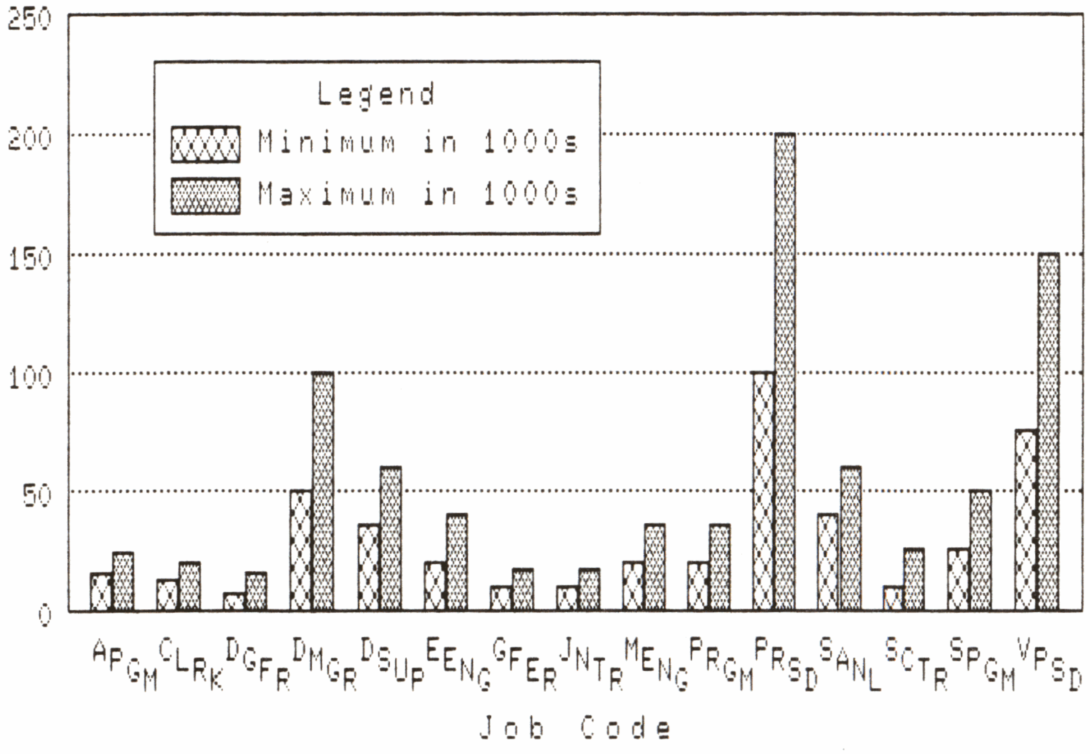
```
DTR> SET DICTIONARY CDD$TOP.DTR$LIB.DEMO.RDB
```

Once you have set the CDD default to the appropriate directory, you can begin using the Rdb/VMS PERSONNEL database and the various domains that have been created from the relations of the database.

To begin, you must ready the database. You can ready the database directly or, if the database or any of its relations have been defined as DATATRIEVE domains, you can ready the relevant domains.

In the following example, the information being plotted is contained in a single relation, JOBS, which has been defined as a DATATRIEVE domain. The example uses PLOT MULTI_BAR to create a chart showing the minimum and maximum salaries for each job code. Note that dividing by 1000 eliminates the default scientific notation labeling on the vertical (Y) axis.

```
DTR> READY JOBS
DTR> PLOT MULTI_BAR ALL JOB_CODE ("Job Code"),
DTR> MINIMUM_SALARY/1000 ("Minimum in 1000's"),
DTR> MAXIMUM_SALARY/1000 ("Maximum in 1000's") OF
DTR> JOBS SORTED BY JOB_CODE THEN
DTR> PLOT CROSS_HATCH
```



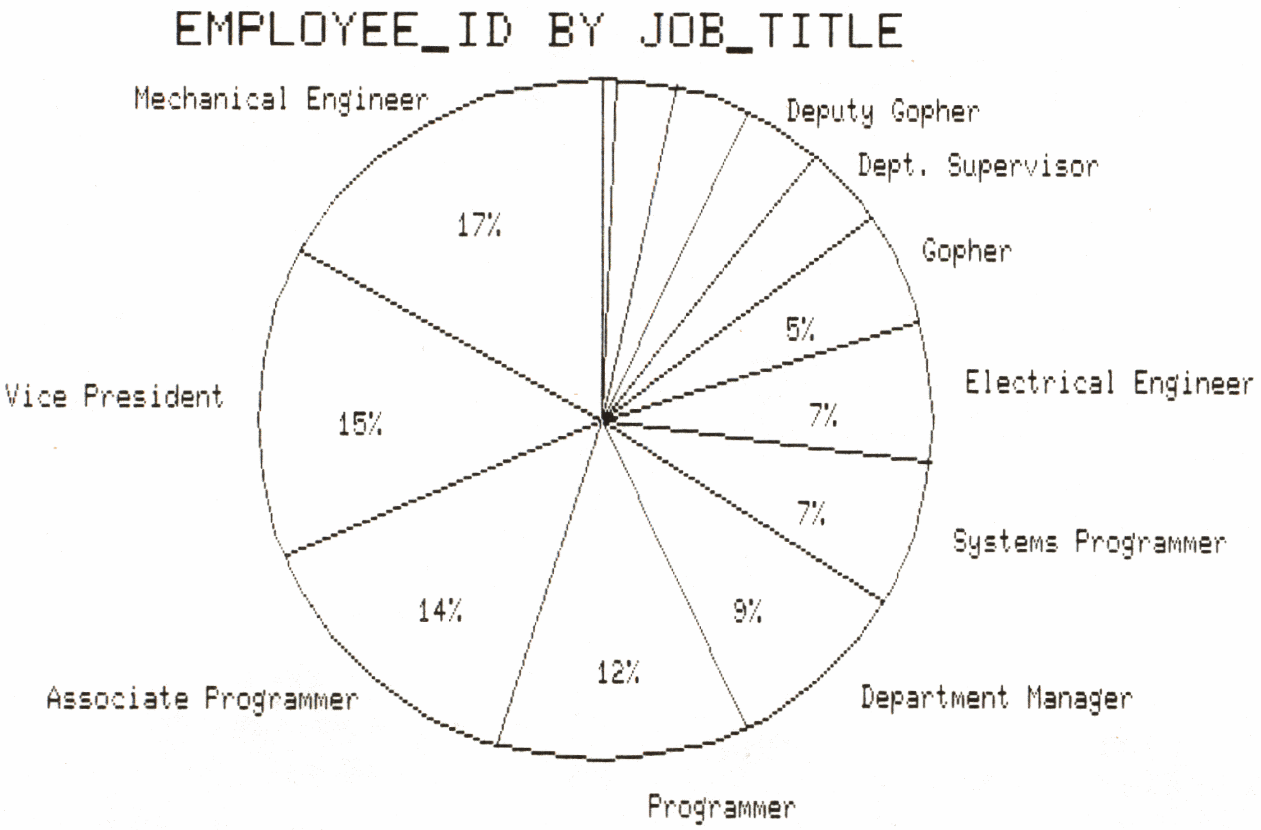
ZK-00368-00

You can also use DATATRIEVE plot statements to plot data from more than one relation or domain by using the DATATRIEVE CROSS clause. The CROSS clause of the record selection expression joins the relations or domains that share a common field name. You could also create a view domain of fields from two or more frequently used relations or domains.

The relations used in the following example are JOB_HISTORY and JOBS. Both are defined as domains. They share a common field name, JOB_CODE, through which you can link the two domains using the CROSS clause. Then, using the PLOT VALUE_PIE statement, you can create a pie chart showing the percentage of employees that work at a given type of job.

```
DTR> READY JOB_HISTORY, JOBS
DTR> FIND JOB_HISTORY CROSS JOBS OVER JOB_CODE
DTR> PLOT VALUE_PIE ALL JOB_TITLE, EMPLOYEE_ID
```

Note that when a percentage is below a certain minimum, DATATRIEVE does not display a label. This avoids overwriting when pie wedges are small.



ZK-00369-00

Record Definitions Used in This Manual A

This appendix lists the record definitions used in this manual. These definitions are stored in CDD$TOP.DTR$LIB.DEMO. The files associated with these records can be found in SYS$SYSROOT:[DTR].

A.1 ANNUAL_REC Record Definition

```
DEFINE RECORD ANNUAL_REC
01 DATA.
   03 DATE DATE PIC YYYY.
   03 EQUIPMENT_SALES REAL
              EDIT_STRING ZZZ9.9.
   03 SERVICES REAL
              EDIT_STRING ZZ9.9.
   03 REVENUE COMPUTED BY
              EQUIPMENT_SALES + SERVICES
              EDIT_STRING ZZZ9.9.
   03 NET_INCOME REAL
              EDIT_STRING ZZ9.9.
   03 NET_INCOME_PER_SHARE REAL
              EDIT_STRING ZZ9.9.
   03 RESEARCH REAL
              QUERY_NAME DEVELOPMENT
              EDIT_STRING ZZ9.9.
   03 INVENTORIES REAL
              EDIT_STRING ZZ9.9.
   03 EMPLOYEES REAL
               EDIT_STRING IS ZZ,ZZZ.
   03 FILLER PIC X(68).
```

A.2 PERSONNEL_REC Record Definition

```
DEFINE RECORD PERSONNEL_REC USING
01  PERSON.
    05 ID                   PIC IS 9(5).
    05 EMPLOYEE_STATUS      PIC IS X(11)
                            QUERY_NAME IS STATUS
                            QUERY_HEADER IS "STATUS"
                            VALID IF STATUS EQ "TRAINEE","EXPERIENCED".
    05 EMPLOYEE_NAME        QUERY_NAME IS NAME.
        10 FIRST_NAME           PIC IS X(10)
                                QUERY_NAME IS F_NAME.
        10 LAST_NAME            PIC IS X(10)
                                QUERY_NAME IS L_NAME.
    05 DEPT                 PIC IS XXX.
    05 START_DATE           USAGE IS DATE
                            DEFAULT VALUE IS "TODAY".
    05 SALARY               PIC IS 9(5)
                            EDIT_STRING IS $$$,$$$.
    05 SUP_ID               PIC IS 9(5)
                            MISSING VALUE IS 0.
```

A.3 YACHT Record Definition

```
DEFINE RECORD YACHT USING
01 BOAT.
  03 TYPE.
     06 MANUFACTURER PIC X(10)
        QUERY_NAME IS BUILDER.
     06 MODEL PIC X(10).
  03 SPECIFICATIONS
     QUERY_NAME SPECS.
     06 RIG PIC X(6)
        VALID IF RIG EQ "SLOOP","KETCH","MS","YAWL".
     06 LENGTH_OVER_ALL PIC XXX
        VALID IF LOA BETWEEN 15 AND 50
        QUERY_NAME IS LOA.
     06 DISPLACEMENT PIC 99999
        QUERY_HEADER IS "WEIGHT"
        EDIT_STRING IS ZZ,ZZ9
        QUERY_NAME IS DISP.
     06 BEAM PIC 99 MISSING VALUE IS 0.
     06 PRICE PIC 99999
        MISSING VALUE IS 0
        VALID IF PRICE)DISP*1.3 OR PRICE EQ 0
        EDIT_STRING IS $$$,$$$.
     06 PRICE_PER_POUND    COMPUTED BY (PRICE/DISP)
        EDIT_STRING IS $$$.
```

A.4 SALES_REC Record Definition

```
DEFINE RECORD SALES_REC USING
01 SALESREC.
      05 SALES_NAME   PIC IS X(20).
      05 START_DATE   USAGE DATE.
      05 MONTHS_EMP   COMPUTED BY ("TODAY" - START_DATE)/30
                      EDIT_STRING IS ZZ9.
      05 AMOUNT       PIC IS 9(5)V99
                      EDIT_STRING IS $$$,$$$.99.
      05 COMM_PCT      COMPUTED BY
                      CHOICE
                         (MONTHS_EMP LE 6 AND AMOUNT > 5000) THEN 10
                         (MONTHS_EMP LE 6) THEN 05
                         (AMOUNT > 10000) THEN 12
                         ELSE 07
                      END_CHOICE
                      EDIT_STRING IS Z9%.
      05 RATING       COMPUTED BY
                      CHOICE
                         (MONTHS_EMP LE 6 AND AMOUNT > 5000)
                         THEN "ABOVE QUOTA"
                         (AMOUNT > 10000) THEN "ABOVE QUOTA"
                         ELSE "BELOW QUOTA"
                      END_CHOICE.
```

A.5 PAYABLES_REC Record Definition

```
DEFINE RECORD PAYABLES_REC USING
     01 PAYABLE.
         05 ORDER_NUM PIC 9(7).
         05 TYPE.
             10 MANUFACTURER    PIC IS X(10)
                QUERY_NAME IS BUILDER
                QUERY_HEADER IS "VENDOR".
             10 MODEL           PIC IS X(10)
                QUERY_HEADER IS "ITEM_TYPE".
         05 WHSLE_PRICE   PIC 9(5)
                 EDIT_STRING IS $$$,$$$.
         05 ITEMS_RECEIVED      USAGE IS DATE
                MISSING VALUE IS 010101
                EDIT_STRING IS NN/DD/YY?"NO GOODS".
         05 INVOICE_DUE         USAGE IS DATE
                MISSING VALUE IS 010101
                EDIT_STRING IS NN/DD/YY?"NO INVCE".
         05 BILL_PAID           USAGE IS DATE
                MISSING VALUE IS 010101
                EDIT_STRING IS NN/DD/YY?"NOT PAID".
         05 AGE COMPUTED BY FN$FLOOR(("TODAY" - INVOICE_DUE)/30)
                EDIT_STRING IS Z9.
```

Index

In this index, a page number followed by a "t" indicates a table reference. A page number followed by an "f" indicates a figure reference. A page number followed by an "e" indicates an example reference.

How to Order Additional Documentation

If you live in:	Call:	or Write:
New Hampshire, Alaska	603-884-6660	Digital Equipment Corp. P.O. Box CS2008 Nashua, NH 03061-2698
Continental USA, Puerto Rico, Hawaii	1-800-258-1710	Same as above.
Canada (Ottawa-Hull)	613-234-7726	Digital Equipment Corp. 940 Belfast Road Ottawa, Ontario K1G 4C2 Attn: P&SG Business Manager or approved distributor
Canada (British Columbia)	1-800-267-6146	Same as above.
Canada (All other)	112-800-267-6146	Same as above.
All other areas	—	Digital Equipment Corp. Peripherals & Supplies Centers P&SG Business Manager c/o DIGITAL's local subsidiary

Note: Place prepaid orders from Puerto Rico with the local DIGITAL subsidiary (phone 809-754-7575).

Place internal orders with the Software Distribution Center, Digital Drive, Westminster, MA 01473-0471.

Reader's Comments

VAX DATATRIEVE
Guide to Using Graphics
AA-L631D-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

I would like to see more/less ____________________

____

What I like best about this manual is ____________________

____

What I like least about this manual is ____________________

____

I found the following errors in this manual:
Page Description

____ ____________________

____ ____________________

____ ____________________

Additional comments or suggestions to improve this manual:

____

____

____

I am using **Version** ______ of the software this manual describes.

Name/Title ____________________ Dept. ____________

Company ____________________ Date ____________

Mailing Address ____________________

____________________ Phone ____________

-- Do Not Tear - Fold Here and Tape

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here

Cut Along Dotted Line

Reader's Comments

VAX DATATRIEVE
Guide to Using Graphics
AA–L631D–TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

I would like to see more/less ______________________________

____

What I like best about this manual is ______________________________

____

What I like least about this manual is ______________________________

____

I found the following errors in this manual:
Page Description

______ ______________________________

______ ______________________________

______ ______________________________

Additional comments or suggestions to improve this manual:

____

____

____

I am using **Version** ______ of the software this manual describes.

Name/Title ______________________ Dept. ____________

Company ______________________ Date ____________

Mailing Address ______________________________

______________________ Phone ____________

-- Do Not Tear - Fold Here and Tape

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here

Cut Along Dotted Line

digital™

DIGITAL EQUIPMENT CORPORATION

Printed in U.S.A.

Reference Section