CHAPTER 1 Introduction

DATAPLOT is an interactive command driven language for

1. graphics (continuous and discrete);
2. fitting (linear and non-linear);
3. general data analysis;
4. mathematics.

DATAPLOT commands are high-level, English-syntax, and self-descriptive. Typical DATAPLOT commands are shown below.

```
PLOT Y X
PLOT EXP(-X**2) FOR X = -3 .1 3
FIT Y = A + B*EXP(-ALPHA*X)
BOX PLOT Y X
ANOVA Y X1 X2 X3
LET A = ROOTS SIN(X**2) + EXP(-X) FOR X = 0 TO 5
```

The three most important commands are PLOT, FIT, and LET. The “average” analyst commonly uses about 20 commands. The language as a whole consists of over 500 commands.

The reference manual is a two volume set consisting of:

1. DATAPLOT Reference Manual Volume I: Commands;
2. DATAPLOT Reference Manual Volume II: LET Subcommands and Library Functions.

This manual (Volume II) documents the subcommands available under the LET command and the library functions available in DATAPLOT. The above manuals are intended as references, not tutorials. A User’s Guide is being planned to provide a tutorial introduction to DATAPLOT.

Numbers, parameters, variables, matrices, and functions

DATAPLOT distinguishes between numbers, parameters, variables, matrices, and functions. Numbers are unnamed scalars (i.e., single values) while parameters are named scalers. For example, in the command

```
FIT Y = A + B*X + C*X**2
```

A, B, and C are parameters while 2 is a number. In general, parameters and numbers can be used interchangeably. For example,

```
BOX 20 20 50 50
```

can also be coded as

```
LET X1 = 20
LET Y1 = 20
LET X2 = 50
LET Y2 = 50
BOX X1 Y1 X2 Y2
```

The one exception to this rule is the FIT and PRE-FIT commands. The command lines

```
LET D = 2
FIT Y = A + B*X + C*X**D
```

are not treated the same as

```
FIT Y = A + B*X + C*X**2
```

In the first case, D is recognized as a parameter and a least squares estimate is computed for D (along with A, B, and C). However, in the second case, 2 is recognized as a number and least squares estimates are only computed for A, B, and C.

Arithmetic expressions cannot be used in place of parameters. For example,

```
BOX X1 Y1 X1+10 Y1+8
```

does not work and should be coded as

```
LET X2 = X1+10
LET Y2 = Y1+8
BOX X1 Y1 X2 Y2
```

Internally, DATAPLOT stores data in a row and column format. When DATAPLOT is built on a specific computer, the installer specifies limits on the maximum number of rows and the total number of data points (i.e., number of rows times number of columns). By default, these are set to 100,000 total points with a maximum of 10,000 rows. At sign-on, the user is given 10 columns with 10,000 rows each. The DIMENSION command, documented in the Support chapter of Volume I, allows the user to redimension the number rows and columns. However, the number of redimensioned rows cannot be set higher than the maximum number of rows.

A variable is a named one-dimensional array of numbers. In DATAPLOT, this corresponds to a single column of data.

A matrix is a two-dimensional array of numbers, divided into columns and rows, that can be accessed with a single name. DATAPLOT stores matrices as a collection of column variables. Only a few commands, documented in the “Matrix LET Subcommands” chapter of this volume, accept matrix names as input. However, the individual columns of the matrix can be accessed as variables on any command that accepts a variable name.

DATAPLOT supports named character strings and functions. Named character strings are most commonly used to label plots and identify file names while functions are used in various mathematical procedures such as non-linear fitting. Internally, DATAPLOT stores functions as named character strings. This means that named character strings and functions can be used interchangeably in DATAPLOT commands. The distinction in the documentation depends on the context in which they are used. DATAPLOT does not support arrays of character strings.

DATAPLOT does not require type specification for parameters, variables, matrices and functions. DATAPLOT determines the type from the command context the first time a name is used. However, once a type is determined, it cannot be changed without deleting the name first (see the DELETE command in the Support chapter of Volume I).

Overview of the LET command

The LET command is the single most powerful command in DATAPLOT. The most important capability of the LET command is carrying out function evaluations and variable transformations. Such evaluations and transformations are general and any Fortran-like expression can be used. For example,

```
LET Y = SIN(X)*COS(X)
LET Y = 3*MIN(Y1,Y2)-A
```

In addition to its use in function evaluation and data transformations, the LET command can be used by the analyst to carry out a broad spectrum of statistical, mathematical, and probabilistic operations. These operations are specified by the use of subcommands under the LET command. These subcommands fall into four general categories:

1. computing statistics;
2. performing mathematical operations;
3. operating on matrices;
4. generating random numbers.

There is a separate chapter to document the LET subcommands in each of these categories. Examples of LET subcommands include

```
LET A = MAXIMUM Y1
LET A = SUM Y1
LET YR YC = FFT Z
```

Unlike the general LET command, the LET subcommands do not accept expressions. That is, the command

```
LET A = MAXIMUM (Y1-3)/8
```

is not legal. It must be entered as the two commands

```
LET TEMP = (Y1-3)/8
LET A = MAXIMUM TEMP
```

Overview of library functions

DATAPLOT has an extensive library of built-in functions. These functions are typically used in the LET, LET FUNCTION, FIT, PRE-FIT, PLOT and 3D-PLOT commands. The available functions can be divided into three general categories.

1. mathematical functions;
2. trigonometric functions;
3. probability functions.

A separate chapter exists for each of these categories of library functions.

Distinction between LET subcommands and library functions

Library functions are distinguished from LET subcommands in the following ways.

1. Library functions enclose the input value in parenthesis while LET subcommands use spaces.
2. Library functions can accept and return either parameters (i.e., single values) or variables (i.e., arrays of values), while LET subcommands are specific in whether they accept a parameter or a variable as input and whether they return a parameter or a variable as the result. For example, the MAX function does a pairwise comparison of two numbers and can be entered as follows

   ```
   LET A = MAX(Y1,Y2)
   ```

   In this case, Y1 and Y2 can be either parameters or variables. If they are both variables, they should be the same length. In fact, one can be a parameter while the other is a variable. The result A is a parameter if both Y1 and Y2 are parameters. Otherwise, it is a variable. On the other hand, the LET subcommand MAXIMUM, which computes the maximum of a variable, and is entered as

   ```
   LET A = MAXIMUM Y1
   ```

   requires that Y1 be a variable and it returns A as a parameter.

3. Library functions can accept expressions, while LET subcommands do not. For example, the following statement using the library functions ABS and INT is acceptable.

```
LET Y2 = ABS(Y1-INT(Y1))
```

In order to use a LET subcommand, such as SUM, in conjunction with an expression, such as Y**2 + 8, enter the following commands.

```
LET YTEMP = Y**2 + 8
LET A = SUM YTEMP
```

4. Functions can appear in commands other than LET. For example,

```
PLOT SIN(X) FOR X = -3.14 0.1 3.14
```

can be used to plot the sine function.

(blank page)

CHAPTER 1 Introduction

DATAPLOT is an interactive command driven language for

1. graphics (continuous and discrete);
2. fitting (linear and non-linear);
3. general data analysis;
4. mathematics.

DATAPLOT commands are high-level, English-syntax, and self-descriptive. Typical DATAPLOT commands are shown below.

```
PLOT Y X
PLOT EXP(-X**2) FOR X = -3 .1 3
FIT Y = A + B*EXP(-ALPHA*X)
BOX PLOT Y X
ANOVA Y X1 X2 X3
LET A = ROOTS SIN(X**2) + EXP(-X) FOR X = 0 TO 5
```

The three most important commands are PLOT, FIT, and LET. The “average” analyst commonly uses about 20 commands. The language as a whole consists of over 500 commands.

The reference manual is a two volume set consisting of:

1. DATAPLOT Reference Manual Volume I: Commands;
2. DATAPLOT Reference Manual Volume II: LET Subcommands and Library Functions.

This manual (Volume II) documents the subcommands available under the LET command and the library functions available in DATAPLOT. The above manuals are intended as references, not tutorials. A User’s Guide is being planned to provide a tutorial introduction to DATAPLOT.

Numbers, parameters, variables, matrices, and functions

DATAPLOT distinguishes between numbers, parameters, variables, matrices, and functions. Numbers are unnamed scalars (i.e., single values) while parameters are named scalers. For example, in the command

```
FIT Y = A + B*X + C*X**2
```

A, B, and C are parameters while 2 is a number. In general, parameters and numbers can be used interchangeably. For example,

```
BOX 20 20 50 50
```

can also be coded as

```
LET X1 = 20
LET Y1 = 20
LET X2 = 50
LET Y2 = 50
BOX X1 Y1 X2 Y2
```

The one exception to this rule is the FIT and PRE-FIT commands. The command lines

```
LET D = 2
FIT Y = A + B*X + C*X**D
```

are not treated the same as

```
FIT Y = A + B*X + C*X**2
```

In the first case, D is recognized as a parameter and a least squares estimate is computed for D (along with A, B, and C). However, in the second case, 2 is recognized as a number and least squares estimates are only computed for A, B, and C.

Arithmetic expressions cannot be used in place of parameters. For example,

```
BOX X1 Y1 X1+10 Y1+8
```

does not work and should be coded as

```
LET X2 = X1+10
LET Y2 = Y1+8
BOX X1 Y1 X2 Y2
```

Internally, DATAPLOT stores data in a row and column format. When DATAPLOT is built on a specific computer, the installer specifies limits on the maximum number of rows and the total number of data points (i.e., number of rows times number of columns). By default, these are set to 100,000 total points with a maximum of 10,000 rows. At sign-on, the user is given 10 columns with 10,000 rows each. The DIMENSION command, documented in the Support chapter of Volume I, allows the user to redimension the number rows and columns. However, the number of redimensioned rows cannot be set higher than the maximum number of rows.

A variable is a named one-dimensional array of numbers. In DATAPLOT, this corresponds to a single column of data.

A matrix is a two-dimensional array of numbers, divided into columns and rows, that can be accessed with a single name. DATAPLOT stores matrices as a collection of column variables. Only a few commands, documented in the “Matrix LET Subcommands” chapter of this volume, accept matrix names as input. However, the individual columns of the matrix can be accessed as variables on any command that accepts a variable name.

DATAPLOT supports named character strings and functions. Named character strings are most commonly used to label plots and identify file names while functions are used in various mathematical procedures such as non-linear fitting. Internally, DATAPLOT stores functions as named character strings. This means that named character strings and functions can be used interchangeably in DATAPLOT commands. The distinction in the documentation depends on the context in which they are used. DATAPLOT does not support arrays of character strings.

DATAPLOT does not require type specification for parameters, variables, matrices and functions. DATAPLOT determines the type from the command context the first time a name is used. However, once a type is determined, it cannot be changed without deleting the name first (see the DELETE command in the Support chapter of Volume I).

Overview of the LET command

The LET command is the single most powerful command in DATAPLOT. The most important capability of the LET command is carrying out function evaluations and variable transformations. Such evaluations and transformations are general and any Fortran-like expression can be used. For example,

```
LET Y = SIN(X)*COS(X)
LET Y = 3*MIN(Y1,Y2)-A
```

In addition to its use in function evaluation and data transformations, the LET command can be used by the analyst to carry out a broad spectrum of statistical, mathematical, and probabilistic operations. These operations are specified by the use of subcommands under the LET command. These subcommands fall into four general categories:

1. computing statistics;
2. performing mathematical operations;
3. operating on matrices;
4. generating random numbers.

There is a separate chapter to document the LET subcommands in each of these categories. Examples of LET subcommands include

```
LET A = MAXIMUM Y1
LET A = SUM Y1
LET YR YC = FFT Z
```

Unlike the general LET command, the LET subcommands do not accept expressions. That is, the command

```
LET A = MAXIMUM (Y1-3)/8
```

is not legal. It must be entered as the two commands

```
LET TEMP = (Y1-3)/8
LET A = MAXIMUM TEMP
```

Overview of library functions

DATAPLOT has an extensive library of built-in functions. These functions are typically used in the LET, LET FUNCTION, FIT, PRE-FIT, PLOT and 3D-PLOT commands. The available functions can be divided into three general categories.

1. mathematical functions;
2. trigonometric functions;
3. probability functions.

A separate chapter exists for each of these categories of library functions.

Distinction between LET subcommands and library functions

Library functions are distinguished from LET subcommands in the following ways.

1. Library functions enclose the input value in parenthesis while LET subcommands use spaces.
2. Library functions can accept and return either parameters (i.e., single values) or variables (i.e., arrays of values), while LET subcommands are specific in whether they accept a parameter or a variable as input and whether they return a parameter or a variable as the result. For example, the MAX function does a pairwise comparison of two numbers and can be entered as follows

   ```
   LET A = MAX(Y1,Y2)
   ```

   In this case, Y1 and Y2 can be either parameters or variables. If they are both variables, they should be the same length. In fact, one can be a parameter while the other is a variable. The result A is a parameter if both Y1 and Y2 are parameters. Otherwise, it is a variable. On the other hand, the LET subcommand MAXIMUM, which computes the maximum of a variable, and is entered as

   ```
   LET A = MAXIMUM Y1
   ```

   requires that Y1 be a variable and it returns A as a parameter.

3. Library functions can accept expressions, while LET subcommands do not. For example, the following statement using the library functions ABS and INT is acceptable.

```
LET Y2 = ABS(Y1-INT(Y1))
```

In order to use a LET subcommand, such as SUM, in conjunction with an expression, such as Y**2 + 8, enter the following commands.

```
LET YTEMP = Y**2 + 8
LET A = SUM YTEMP
```

4. Functions can appear in commands other than LET. For example,

```
PLOT SIN(X) FOR X = -3.14 0.1 3.14
```

can be used to plot the sine function.

(blank page)

CHAPTER 1 Introduction

DATAPLOT is an interactive command driven language for

1. graphics (continuous and discrete);
2. fitting (linear and non-linear);
3. general data analysis;
4. mathematics.

DATAPLOT commands are high-level, English-syntax, and self-descriptive. Typical DATAPLOT commands are shown below.

```
PLOT Y X
PLOT EXP(-X**2) FOR X = -3 .1 3
FIT Y = A + B*EXP(-ALPHA*X)
BOX PLOT Y X
ANOVA Y X1 X2 X3
LET A = ROOTS SIN(X**2) + EXP(-X) FOR X = 0 TO 5
```

The three most important commands are PLOT, FIT, and LET. The “average” analyst commonly uses about 20 commands. The language as a whole consists of over 500 commands.

The reference manual is a two volume set consisting of:

1. DATAPLOT Reference Manual Volume I: Commands;
2. DATAPLOT Reference Manual Volume II: LET Subcommands and Library Functions.

This manual (Volume II) documents the subcommands available under the LET command and the library functions available in DATAPLOT. The above manuals are intended as references, not tutorials. A User’s Guide is being planned to provide a tutorial introduction to DATAPLOT.

Numbers, parameters, variables, matrices, and functions

DATAPLOT distinguishes between numbers, parameters, variables, matrices, and functions. Numbers are unnamed scalars (i.e., single values) while parameters are named scalers. For example, in the command

```
FIT Y = A + B*X + C*X**2
```

A, B, and C are parameters while 2 is a number. In general, parameters and numbers can be used interchangeably. For example,

```
BOX 20 20 50 50
```

can also be coded as

```
LET X1 = 20
LET Y1 = 20
LET X2 = 50
LET Y2 = 50
BOX X1 Y1 X2 Y2
```

The one exception to this rule is the FIT and PRE-FIT commands. The command lines

```
LET D = 2
FIT Y = A + B*X + C*X**D
```

are not treated the same as

```
FIT Y = A + B*X + C*X**2
```

In the first case, D is recognized as a parameter and a least squares estimate is computed for D (along with A, B, and C). However, in the second case, 2 is recognized as a number and least squares estimates are only computed for A, B, and C.

Arithmetic expressions cannot be used in place of parameters. For example,

```
BOX X1 Y1 X1+10 Y1+8
```

does not work and should be coded as

```
LET X2 = X1+10
LET Y2 = Y1+8
BOX X1 Y1 X2 Y2
```

Internally, DATAPLOT stores data in a row and column format. When DATAPLOT is built on a specific computer, the installer specifies limits on the maximum number of rows and the total number of data points (i.e., number of rows times number of columns). By default, these are set to 100,000 total points with a maximum of 10,000 rows. At sign-on, the user is given 10 columns with 10,000 rows each. The DIMENSION command, documented in the Support chapter of Volume I, allows the user to redimension the number rows and columns. However, the number of redimensioned rows cannot be set higher than the maximum number of rows.

A variable is a named one-dimensional array of numbers. In DATAPLOT, this corresponds to a single column of data.

A matrix is a two-dimensional array of numbers, divided into columns and rows, that can be accessed with a single name. DATAPLOT stores matrices as a collection of column variables. Only a few commands, documented in the “Matrix LET Subcommands” chapter of this volume, accept matrix names as input. However, the individual columns of the matrix can be accessed as variables on any command that accepts a variable name.

DATAPLOT supports named character strings and functions. Named character strings are most commonly used to label plots and identify file names while functions are used in various mathematical procedures such as non-linear fitting. Internally, DATAPLOT stores functions as named character strings. This means that named character strings and functions can be used interchangeably in DATAPLOT commands. The distinction in the documentation depends on the context in which they are used. DATAPLOT does not support arrays of character strings.

DATAPLOT does not require type specification for parameters, variables, matrices and functions. DATAPLOT determines the type from the command context the first time a name is used. However, once a type is determined, it cannot be changed without deleting the name first (see the DELETE command in the Support chapter of Volume I).

Overview of the LET command

The LET command is the single most powerful command in DATAPLOT. The most important capability of the LET command is carrying out function evaluations and variable transformations. Such evaluations and transformations are general and any Fortran-like expression can be used. For example,

```
LET Y = SIN(X)*COS(X)
LET Y = 3*MIN(Y1,Y2)-A
```

In addition to its use in function evaluation and data transformations, the LET command can be used by the analyst to carry out a broad spectrum of statistical, mathematical, and probabilistic operations. These operations are specified by the use of subcommands under the LET command. These subcommands fall into four general categories:

1. computing statistics;
2. performing mathematical operations;
3. operating on matrices;
4. generating random numbers.

There is a separate chapter to document the LET subcommands in each of these categories. Examples of LET subcommands include

```
LET A = MAXIMUM Y1
LET A = SUM Y1
LET YR YC = FFT Z
```

Unlike the general LET command, the LET subcommands do not accept expressions. That is, the command

```
LET A = MAXIMUM (Y1-3)/8
```

is not legal. It must be entered as the two commands

```
LET TEMP = (Y1-3)/8
LET A = MAXIMUM TEMP
```

Overview of library functions

DATAPLOT has an extensive library of built-in functions. These functions are typically used in the LET, LET FUNCTION, FIT, PRE-FIT, PLOT and 3D-PLOT commands. The available functions can be divided into three general categories.

1. mathematical functions;
2. trigonometric functions;
3. probability functions.

A separate chapter exists for each of these categories of library functions.

Distinction between LET subcommands and library functions

Library functions are distinguished from LET subcommands in the following ways.

1. Library functions enclose the input value in parenthesis while LET subcommands use spaces.
2. Library functions can accept and return either parameters (i.e., single values) or variables (i.e., arrays of values), while LET subcommands are specific in whether they accept a parameter or a variable as input and whether they return a parameter or a variable as the result. For example, the MAX function does a pairwise comparison of two numbers and can be entered as follows

   ```
   LET A = MAX(Y1,Y2)
   ```

   In this case, Y1 and Y2 can be either parameters or variables. If they are both variables, they should be the same length. In fact, one can be a parameter while the other is a variable. The result A is a parameter if both Y1 and Y2 are parameters. Otherwise, it is a variable. On the other hand, the LET subcommand MAXIMUM, which computes the maximum of a variable, and is entered as

   ```
   LET A = MAXIMUM Y1
   ```

   requires that Y1 be a variable and it returns A as a parameter.

3. Library functions can accept expressions, while LET subcommands do not. For example, the following statement using the library functions ABS and INT is acceptable.

```
LET Y2 = ABS(Y1-INT(Y1))
```

In order to use a LET subcommand, such as SUM, in conjunction with an expression, such as Y**2 + 8, enter the following commands.

```
LET YTEMP = Y**2 + 8
LET A = SUM YTEMP
```

4. Functions can appear in commands other than LET. For example,

```
PLOT SIN(X) FOR X = -3.14 0.1 3.14
```

can be used to plot the sine function.

(blank page)