4. The DISPLAY Step for Real Variables

4.1 Introduction

Once a VPLX file has been created, the DISPLAY step can use it to display estimates, their standard errors, covariances, correlations of the sampling errors, and t-tests. Chapter 2 illustrates DISPLAY steps operating directly on the output of a CREATE step and on the output of a TRANSFORM step. In general, the DISPLAY step can accept any VPLX file as input, even if the file was created in an earlier run.

VPLX can write results to an optional output file, if desired. If the file exists, it will be overwritten. The output file will be in character format, suitable for use by another statistical package or spreadsheet. Section 4.6 describes default output for LIST requests, which is the most common application of this feature. Section 4.7 describes the default output for other requests.

This chapter introduces the DISPLAY step by describing the available features for the VPLX files with real variables created according to the rules of Chapter 3. As additional chapters introduce new variable types, real with missing, categorical, crossed categorical, crossed real, and class, the corresponding features in the DISPLAY step will also be presented.

Summary of this chapter:

- Section 4.2 describes the file specifications in the DISPLAY statement.
- Section 4.3 provides an overview of the remaining available statements in the DISPLAY step and describes the requests LIST, COV, CORR, T-TEST.
- Section 4.4 describes the available functions for real variables.
- Section 4.5 discusses the available options for real variables.
- Section 4.6 specifies the default format for output to the OUT = file from LIST.
- Section 4.7 specifies the default formats for the output files from COV, CORR, and T-TEST.
- Section 4.8 provides an example VPLX run to illustrate many of these features.

4.2 The DISPLAY Statement

4.2.1 General Form. Similar to the CREATE step, the DISPLAY step is initiated by a statement beginning with DISPLAY,

```
DISPLAY   [IN =  fname1]  [OUT =  fname2]
```

The input to the step is a VPLX file; the output is a character file to which results may be written in addition to or substituting for the writing of results to the print file.

Specification of the input and output files is optional. If no input file is explicitly named, then VPLX will select the previously named VPLX file according to the rules of Section 4.2.2. If no file is named and there is no default file according to rules of 4.2.2, then a fatal error occurs. If no output file is specified, then results of the step will only be written to the print file.

4.2.2 General Rules on Default Input VPLX files. As soon as a VPLX run explicitly names one or more VPLX files in a step, then VPLX maintains and continues to update information on a default input VPLX file according to the following rules:

- If an input VPLX file is not named, VPLX will use the last created or referenced VPLX file in the current run.

- If a step, such as the TRANSFORM step, has both an input and output VPLX file, then the default file for purposes of the next step is the output of the previous step, regardless of the order of the IN= and OUT= specification in the previous step.

To clarify the second rule,

```
transform   out = fileb.vpl  in = filea.vpl
...
display
```

The display will use the output of the TRANSFORM step, `fileb.vpl` , as its input.

Note: There are no rules for default outputs from any VPLX step. Thus, VPLX will not overwrite any file unless explicitly told to do so.

Note: As a matter of programming practice, it is often better to name files explicitly. On the other hand, use of defaults has its own advantages in preventing programming error. If it is possible to prepare a VPLX program in such a way that VPLX files are only explicitly named as

output files and are used as implicit inputs, errors from accidentally misspelling file names can be reduced.

4.3 General Form of DISPLAY Statements

Except for the statements OPTIONS, COMMENT, SCRATCH1 (*1*) and other global statements of Chapter 13, all other statements in the DISPLAY step are of the form:

request specification

where *request* is one of:

LIST - for estimates and standard errors for a list of variables.

COV (or COVARIANCES) - for a variance/covariance matrix for a set of variables.

CORR (or CORRELATIONS) - for a correlation matrix, derived from the covariance matrix, for a set of variables.

T-TEST - for standardized differences, that is, differences divided by estimated standard errors of the differences, for a set of variables. (Section 4.8 illustrates this feature.)

In general, the *specification* is made up of one or more of the following elements, separated by "/":

1) **Standard functions of VPLX tallies**, such as means or totals derived from the file,

2) **Class specifications**, to control combinations of class variables, and

3) **Options**, to control aspects of the calculation and display.

Section 4.4 presents the available functions for real variables. The treatment of class specifications is delayed until Chapter 7. Section 4.5 will describe options in the DISPLAY step, both within the specification of a request and in separate OPTIONS statements.

As an example, Exhibit 2.1 of Chapter 2 includes:

```
list     rooms  persons  total(rooms persons)
cov      total(rooms  persons)
```

which are two instances of the general syntax. The first request includes the means of `rooms` and `persons`, plus their totals. TOTAL is one of the available standard functions; another function, MEAN, is the implied default function for real variables. The second request also uses the TOTAL function. Neither request makes use of the CLASS or OPTIONS feature.

As a slightly more elaborate example,

```
list   option ndecimal =2 /  rooms  persons /
       option ndecimal =0 / total(rooms persons)
```

might be used to request 2 decimal places of accuracy for the means of rooms and persons and rounding to integers for the estimated totals of rooms and persons.

4.4 Standard Functions of Real Variables in the DISPLAY STEP

Standard functions are available in both the DISPLAY and TRANSFORM steps of VPLX to calculate simple statistics from the data. The two principal forms are:

func(*vlist*)

or

vlist

where *func* is the name of a standard function, such as TOTAL, MEAN, or N, and *vlist* is a list of variables (*2*). The second form implies that default functions, depending on the type of each variable, are to be used; for example, MEAN is the default function for a real variable, unless TOTALS has been specified in a previous OPTIONS statement.

There is also a third form,

N(*block number*)

where *block number* is a number of a block in the VPLX file, e.g., `N(1)`. Chapter 8 discusses blocking. In a file created without blocking, as in the examples in Chapter 3, `N(1)` is the weighted (or unweighted, in the case of an unweighted CREATE step) count of cases. `N(2)`, `N(3)`, *etc.*, do not exist in these cases. In general, if the VPLX file from the CREATE step is

weighted, then all counts referred to are weighted. For simplicity of exposition in this chapter, *weighted counts* will denote "weighted counts, or unweighted counts in the case of unweighted analysis."

The three basic elements, explicit functions of variables, implied functions, and `n` of a block, may be concatenated into lists:

```
cov      rooms  persons  total(rooms persons) n(1)
```

Standard functions may not be nested, however; that is, forms such as `mean(total(x))` or `total(N(1))` cannot be used (*3*).

For real variables in the DISPLAY step, the available functions and their usage are:

MEAN or **MEANS** - This function is the default for real variables. For real variables, the denominator is the weighted count of cases (in the corresponding block). When no explicit blocking is used in creating the VPLX file, the denominator is the number of included observations (*4*).

TOTAL or **TOTALS** - If this function is applied to a real variable, then it shows the weighted sum of the variable.

N - This function provides the weighted count of cases for a variable. When applied to a real variable, it gives the weighted sample size, or more specifically, the weighted count for the corresponding block.

As noted above, N may also be applied to a block number, such as N(1), to obtain the weighted number of cases for the block. Thus, it provides the same value as n(*realvar*), where *realvar* is any real variable included in the block.

4.5 Options

Options may be specified in two ways. An OPTIONS statement

```
OPTION[S]    specification
```

sets options globally. The contents of the ***`specification`*** override previous default and global choices. For example,

```
options  ndecimal=2
```

sets the number of displayed decimals to 2, overriding the default of 4 and any previous global specification for NDECIMAL, while not affecting other global options or defaults. For example,

```
options  ndecimal=6
list     rooms persons
options  ndecimal=2
list     rooms persons
```

may be used to produce outputs rounded to different levels of precision (*5*).

The options for real variables are:

CENTER (or CENTERED) - The default computation of the variance, uncentered, is of the form:

$$Var_g(X_0) = \sum_{r=1}^{n} b_r (X_r - X_0)^2 .$$

Alternatively, VPLX will implement the "centered" calculation

$$Var_g(X_0) = \sum_{r=1}^{n} b_r (X_r - X_0 - c_X)^2 .$$

where

$$c_X = \frac{\sum_{r=1}^{n} b_r (X_r - X_0)}{\sum_{r=1}^{n} b_r} .$$

This centered estimate of the variance is less, although usually not much less, than the uncentered version, but the difference may be large if the random group method is used or if the estimates have been jackknifed. (Chapter 12 includes further discussion of this point.)

DECIMAL (alternate spelling of **NDECIMAL**)

LINESIZE - LINESIZE=80 (default), LINESIZE=120, and LINESIZE=132 are the three allowed choices. These affect how many columns are used by the COV, CORR, and T-TEST displays. Most VPLX printing is designed for 80 columns, but the triangular displays for COV, CORR, and T-TESTS favor using 120 or 132, if the printer supports these widths.

MEAN - (or MEANS) This option reinstates the mean (and percent) as the default. In other words, this option cancels the effect of a TOTAL option (below).

NDECIMAL - NDECIMAL = *nplaces*, where *nplaces* is the number of places after the decimal to be used in displaying estimates and standard errors to the print file after a LIST request. The option has no effect on the display of other requests or on the format of the output file. The option only affects the rounding of the results; all calculations of the estimates and variances are in full precision until rounding in the display. The default is NDECIMAL=4.

NOPRINT - This option turns off the display of results to the print file, which is useful on occasion if the sole interest is in the separate output file.

NOWRITE - This option turns off writing to the output file, if one has been specified. By combining PRINT, NOPRINT, WRITE, and NOWRITE options, it is possible to direct the results from requests to the print file, the output file, or both.

OUTFORMAT - OUTFORMAT = *choice*, where *choice* is 1, 2, 3, or 4. The default is OUTFORMAT=1. This option selects the format for the output file. This chapter describes only the default. Chapter XX describes the other choices (*6*).

PAGESIZE - PAGESIZE = *nlines*, where *nlines* is the number of lines (30 or more) per page. The default is PAGESIZE=57. This option affects page skips within the DISPLAY output (*7*).

PRINT - (default) This requests that the results be included on the print file, opposite of the NOPRINT option.

TOTAL - (or TOTALS) This option sets the default to TOTAL for variables in requests, instead of means and percents.

UNCENTERED - (or UNCENTER) (default) This option requests the default calculation of variance, opposite of the CENTERED option.

WRITE - (default if an output file has been selected) This option requests that results be written to the output file.

As an example, the statement

```
OPTIONS   linesize = 120, pagesize=60  noprint
```

sets global choices for the width of the COV, CORR, and T-TESTS printing, and for the number of lines per page. Results will not be shown on the print file. (The comma in the example has no effect on VPLX's interpretation. Commas may be added in this way to add clarity, if desired.)

Options may be included within a request specification. By experience, however, the best single strategy is to use this feature only for NDECIMAL options (*8*). Options specified within a request take precedence over any corresponding default or global value for the request, but they do not change global values and consequently have no effect across different requests (*9*). As an example of specifying NDECIMAL within a request,

```
list   options  ndecimal=6 / rooms persons /
       options  ndecimal=2 / rooms persons
```

Note the placement of " / " between each options specification and each list of variables.

4.6 Default Output for LIST

Under the default option, that is, OUTFORMAT=1, the output to the OUT= file for list request is written as a series of records, each containing an estimate and its corresponding estimated standard error. Each record is of the form:

Positions

1-2	"01".
3-6	Number of the request within the DISPLAY step.
7-12	An ascending sequence number within the request, which corresponds to the order in which estimates are displayed by LIST.
13-34	The estimate, in F22.8 format.
35-56	The estimated standard error, in F22.8 format

The example in Section 4.8 includes an output file in this form.

If any estimates are missing, results of -98765.43210900 will be shown.

4.7 Default Output for COV, CORR, and T-TEST

4.7.1 Output from COV. Generally COV (or COVARIANCE) produces a square *n* by *n* matrix, where *n* is the number of values specified. To indicate the relationship between COV and LIST: if the COV request were converted to a LIST request, then LIST would output *n* estimates with corresponding standard errors. The squares of the standard error estimates from LIST are the diagonal elements of the covariance matrix from COV.

Under the default option, that is, OUTFORMAT=1, the output to the OUT= file for COV request takes the strategy of writing out each of the *n* rows of the matrix separately. Each row is written out with as many records as required. Each record holds up to 3 elements of the row.

Each record is of the form:

Positions

1-2	"03".
3-6	Number of the request within the DISPLAY step.
7-10	An ascending sequence number within the request, which numbers the row of the covariance matrix.
11-14	The total number of elements in the row (*10*).
15-36	A covariance estimate, in D22.14 format (*11*).
37-58	A covariance estimate, in D22.14 format.
59-80	A covariance estimate, in D22.14 format.

The example in Section 4.8 writes out a 4 by 4 covariance matrix. Each row requires 2 records; the first supplies the first three elements and the second the fourth element, in positions 15-36.

Note that the implicit limit is a 9999 by 9999 covariance matrix.

4.7.2 Output from CORR. CORR (or CORRELATION) also produces a square *n* by *n* matrix, where *n* is the number of values specified. Under the default option, that is, OUTFORMAT=1, the output to the OUT= file for CORR similarly writes each row of information separately with as many records as required. Each record holds up to 6 elements of the row.

Each record is of the form:

Positions	
1-2	"05".
3-6	Number of the request within the DISPLAY step.
7-10	An ascending sequence number within the request, which numbers the row of the correlation matrix.
11-14	The total number of elements in the row.
15-25	A correlation estimate, in F11.8 format.
26-36	A correlation estimate, in F11.8 format.
37-47	A correlation estimate, in F11.8 format.
48-58	A correlation estimate, in F11.8 format.
59-69	A correlation estimate, in F11.8 format.
70-80	A correlation estimate, in F11.8 format.

4.7.3 Output from T-TEST. Like the other 2 procedures, the output from T-TEST is also a square symmetric matrix. If $i < j$, elements (i, j) and (j, i) have signs consistent with the difference of the jth estimate minus the ith. Diagonal elements (i, i) are set to 0.

The output from T-TEST differs from CORR in two ways:

- Positions 1-2 have "07" instead of "05."
- The format is F11.6 instead of F11.8.

If a test result exceeds 999.999, it will be truncated to 999.999. Similarly, if a test result is less than -999.999, it will be truncated to -999.999. This convention insures that the test result will conform to the required format.

4.8 An Example

The following example is based on an elaboration of example 4 of Section 2.3. As in Chapter 2, annotations, #1, #2, etc. have been added at the right margin.

```
comment  EXAM14

create  in = example1.dat  out = example1.vpl

input    rooms persons cluster

      3 variables are specified

format   (3f2.0)

divide   rooms by persons into proom_indiv                                    #1

    (Simple) jackknife replication assumed

    Size of block   1  =              4

    Total size of tally matrix =      4

    Unnamed scratch file opened on unit 13

    Unnamed scratch file opened on unit 14

    End of primary input file after obs #       6

transform  in = example1.vpl out=exampl1a.vpl

user2

old  rooms persons

derived   proom

    (assigned to block   2)

labels  rooms 'Number of rooms' persons 'Persons'
        proom  'Rooms per person'
REPLICATE  0, V1=   36.00, V2=   24.00 RATIO=  1.5000
REPLICATE  1, V1=   37.20, V2=   20.40 RATIO=  1.8235
REPLICATE  2, V1=   36.00, V2=   19.20 RATIO=  1.8750
REPLICATE  3, V1=   37.20, V2=   26.40 RATIO=  1.4091
REPLICATE  4, V1=   38.40, V2=   27.60 RATIO=  1.3913
REPLICATE  5, V1=   33.60, V2=   24.00 RATIO=  1.4000
REPLICATE  6, V1=   33.60, V2=   26.40 RATIO=  1.2727


display  out  = exam14.dat                                                    #2

list     options  ndecimal = 6 / mean ( rooms persons ) proom_indiv proom

options  centered
```

4.12

```
list     options  ndecimal = 6 / mean ( rooms persons ) proom_indiv proom


                                                 Estimate      Standard error

Number of rooms         :  MEAN                  6.000000             .683130

Persons                 :  MEAN                  4.000000            1.183216

proom_indiv             :  MEAN                  2.327381             .600624

Rooms per person        :  VALUE                 1.500000             .522038

                                                 Estimate      Standard error

Number of rooms         :  MEAN                  6.000000             .683130

Persons                 :  MEAN                  4.000000            1.183216

proom_indiv             :  MEAN                  2.327381             .600624

Rooms per person        :  VALUE                 1.500000             .518104

display  out = exam14.cov                                                      #3

cov      mean (rooms persons ) proom_indiv proom

                     Covariances of the Sample Estimates

                                      Estimate              1              2      #4

    Number of rooms         :  MEAN
   1                                    6.0000   .46666667D+00

    Persons                 :  MEAN
   2                                    4.0000   .33333333D-01   .14000000D+01

    proom_indiv             :  MEAN
   3                                    2.3274   .26190476D-01  -.66190476D+00

    Rooms per person        :  VALUE
   4                                    1.5000   .11162908D+00  -.57217314D+00

                                           3              4                     #5

    proom_indiv             :  MEAN
   3                           .36074972D+00

    Rooms per person        :  VALUE
   4                           .27638388D+00   .27252365D+00


display  out = exam14.cor                                                      #6
```

```
corr     mean (rooms persons ) proom_indiv proom

                    Correlations of the Sample Estimates

                                        Estimate       1       2       3       4

    Number of rooms         :  MEAN
   1                                     6.0000   1.000

    Persons                 :  MEAN
   2                                     4.0000    .041   1.000

    proom_indiv             :  MEAN
   3                                     2.3274    .064   -.931   1.000

    Rooms per person        :  VALUE
   4                                     1.5000    .313   -.926    .881   1.000


display  out = exam14.tst                                                        #7

t-test   mean (rooms persons ) proom_indiv proom

                           T-Tests for Differences

                                        Estimate       1       2       3       4


    Number of rooms         :  MEAN
   1                                     6.0000    .000

    Persons                 :  MEAN
   2                                     4.0000  -1.491    .000

    proom_indiv             :  MEAN
   3                                     2.3274  -4.172   -.952    .000

    Rooms per person        :  VALUE
   4                                     1.5000  -6.265  -1.490  -2.916    .000
```

Exhibit 4.1 An example illustrating several features of the DISPLAY step, including the use of options, the printing from LIST, COV, CORR, and T-TEST, and the concurrent writing of results to output files.

The DIVIDE statement at #1 creates `proom_indiv` , the ratio of `rooms` to `persons` at the individual level. The subsequent TRANSFORM step adds `proom` to the file. The first DISPLAY step, at #2, uses the output from the TRANSFORM step as its default input file and illustrates the difference between the UNCENTERED and CENTERED computations of the standard errors. Note that the only difference between the two methods occurs for `proom`, since the computation of a ratio in the TRANSFORM step makes this variable a nonlinear estimator.

For illustration, the example includes the function MEAN in DISPLAY step requests, even though each request would have the same meaning without it.

DISPLAY wrote to `exam14.dat` :

```
01   1     1           6.00000000            .68313005
01   1     2           4.00000000           1.18321596
01   1     3           2.32738095            .60062444
01   1     4           1.50000000            .52203798
01   2     1           6.00000000            .68313005
01   2     2           4.00000000           1.18321596
01   2     3           2.32738095            .60062444
01   2     4           1.50000000            .51810363
```

Exhibit 4.2 The contents of exam14.dat, illustrating output from a LIST specification.

Note that OPTIONS NDECIMAL = does not extend to this file. Instead, all estimates and standard errors are displayed in a FORTRAN F22.8 format, as Section 4.6 notes.

Asterisks after estimated standard errors. An asterisk on the right of the standard errors from LIST warns that the value is undefined for one or more replicates, although not for the full sample. (No instances occur in the example.) Consequently, such estimated standard errors must be interpreted with caution. This warning appears only on the print file and is not included on the output file. When a value is defined for the complete sample but undefined for one or more replicates, VPLX will only include in the estimated variance contributions from the defined replicates, in effect, adding nothing to the variance estimate from replicates that are undefined.

Missing values in DISPLAY. As a general rule, if a characteristic is undefined for the complete sample, the LIST display will show its value as (M) and standard error as -. Elsewhere, such as COV, a VPLX missing value indicator, -98765.432109, will appear. (No instances occur in the example.)

The second DISPLAY step, at #3, illustrates the display of covariances. Notice that each estimate is displayed in the first colum of numbers. The heading at #4 also identifies this first column as "Estimate." The heading then uses the numbers 1 and 2 to reference the first and second variables, number of rooms, and persons, listed in the stub. At #5 a new heading appears, indicating that the columns refer to variables 3 and 4 in the list. The remaining covariances with variables 3 and 4 are shown. Thus, VPLX displays the covariances on the print file in triangular form, but will slice the triangle by groups of columns. If LINESIZE is set to 120 or 132, VPLX will use the additional space to display more columns at a time.

DISPLAY wrote to `exam14.cov` :

```
03   1   1   4   .46666666666667D+00   .33333333333333D-01   .26190476190476D-01
03   1   1   4   .11162907850887D+00
03   1   2   4   .33333333333333D-01   .14000000000000D+01  -.66190476190476D+00
03   1   2   4  -.57217313802992D+00
```

```
03   1   3   4   .26190476190476D-01  -.66190476190476D+00   .36074971655329D+00
03   1   3   4   .27638387782649D+00
03   1   4   4   .11162907850887D+00  -.57217313802992D+00   .27638387782649D+00
03   1   4   4   .27252365037519D+00
```

Exhibit 4.3 The contents of exam14.cov, illustrating output from COV.

Because the covariance matrix is 4 by 4, each row requires 2 records.

A display of a correlation matrix is specified at #6. The strategy for displaying the output is similar to a covariance matrix. Again, a first column of estimates appears, followed by columns of the triangular correlation matrix.

DISPLAY wrote to `exam14.cor` :

```
05   1   1   4 1.00000000  .04123930  .06383179  .31301983
05   1   2   4  .04123930 1.00000000 -.93138339 -.92632066
05   1   3   4  .06383179 -.93138339 1.00000000  .88147015
05   1   4   4  .31301983 -.92632066  .88147015 1.00000000
```

Exhibit 4.4 The contents of exam14.cor, illustrating output from CORR.

Finally, the display step beginning at #7 illustrates the t-test feature. Note that the column of estimates may be used as a reminder of the direction of the test. For example, the test of the difference between the average of `proom_indiv` and the ratio `proom` gives a negative test statistic. Comparison of the values quickly indicates that `proom` is significantly less than `proom_indiv` , according to the test.

DISPLAY wrote to `exam14.tst` :

```
07   1   1   4    .000000  -1.490712  -4.171720  -6.264930
07   1   2   4  -1.490712    .000000   -.952359  -1.489555
07   1   3   4  -4.171720   -.952359    .000000  -2.916033
07   1   4   4  -6.264930  -1.489555  -2.916033    .000000
```

Exhibit 4.5 The contents of exam14.tst, illustrating output from T-TEST.

This example used different display steps to clarify the different types of output. It is sometimes advantageous to write different results to different files in this manner. Requests may be output to a single file, however. For example, all four display steps could have been combined. The information in positions 1 and 2 serves to identify the type of output from DISPLAY.

NOTES

1. The DISPLAY step uses one scratch file to hold a summary of all the requests made in the step. This usage is almost invisible to the user, since the required size is small and, if a scratch file has not been implicitly assigned, VPLX will almost always, in practice, be able to assign a default scratch file of sufficient size. Section 10.4 describes the SCRATCH1 statement. In the rare event that some problem occurs with the scratch file during the DISPLAY step, then a SCRATCH1 statement may be used to assign a scratch file explicitly. The SCRATCH1 statement should come either before the DISPLAY step or just after, so that it precedes any requests.
   As DISPLAY encodes and records all requests on the scratch file, it computes for each request the required amount of storage. If the storage requirements for all requests are less than the total available storage, VPLX will read the input file only once to satisfy all of the requests. If the total requirements exceed the available storage, VPLX will group the requests, without changing their order, into groups, each with as many requests as can be satisfied simultaneously. If a single request requires more storage than VPLX has available, however, DISPLAY is unable to complete the task and terminates. This may occur, for example, if a large covariance matrix is requested.
2. Chapter 7 will clarify that the list of variables may not contain class variables.
3. In fact, it is not clear what meaning any concatenation of functions should be given that is not already covered by the available functions.
4. In other words, the number of observations after exclusions from SELECT statements, if any.
5. Of course, a more economical statement is

```
LIST  OPTIONS NDECIMAL = 6 / ROOMS PERSONS /
    OPTIONS NDECIMAL = 2 / ROOMS PERSONS
```

   which appears at the end of this section.
6. Unlike the other options statements, OUTFORMAT may be specified only once during a DISPLAY step.
7. VPLX attempts some aesthetic choices in preparing the print file output from the DISPLAY step. It will try to decide when, if necessary, to skip to the top of the next page. For example, VPLX attempts to avoid printing a variable label at the end of one page and all of the resulting values on the next. These decisions are made on the basis of the number of lines per page for the target printer.
8. Except for NDECIMAL (and DECIMAL), any other option appearing within a request takes effect for the entire request, regardless of its placement. For example, CENTERED determines the method of variance calculation for the entire requestion; it is not possible, by using both CENTERED and UNCENTERED as an option within a request, to specify both calculations within the same request. CENTERED and UNCENTERED may be used for separate requests within the same display step, however.
   Except for NDECIMAL, the clearest strategy is to place all options in OPTIONS statements to avoid misinterpretation. For example, it is not possible to print selectively results within a request by using PRINT and NOPRINT options together within the request specification; VPLX will simply will apply the last PRINT or NOPRINT it finds in the specification to the entire request. Thus, placing all options, except NDECIMAL, in OPTIONS statements makes the meaning clearer than including them in requests.
9. As noted at (*5*), OUTFORMAT may be only specified once.
10. This number is helpful to a general SAS or FORTRAN program written to read in a covariance matrix from the DISPLAY step, since it provides the dimension of the matrix.
11. This format was discussed in Section 3.3.5.