CHAPTER 9

The COMPARE Procedure

Overview

The COMPARE procedure compares the contents of two SAS data sets, selected variables in different data sets, or variables within the same data set.

PROC COMPARE compares two data sets: the *base data set* and the *comparison data set*. The procedure determines matching variables and matching observations. *Matching variables* are variables with the same name or variables that you explicitly pair by using the VAR and WITH statements. Matching variables must be of the same type. *Matching observations* are observations that have the same values for all ID variables that you specify or, if you do not use the ID statement, that occur in the same position in the data sets. If you match observations by ID variables, both data sets must be sorted by all ID variables.

When you compare data sets using PROC COMPARE, you receive the following type of information:

- whether matching variables have different values
- whether one data set has more observations than the other
- what variables the two data sets have in common
- how many variables are in one data set but not in the other
- whether matching variables have different formats, labels, or types.
- a comparison of the values of matching observations.

Further, PROC COMPARE creates two kinds of output data sets that give detailed information about the differences between observations of variables it is comparing.

The following example compares the data sets PROCLIB.ONE and PROCLIB.TWO, which contain similar data about students:

```
data proclib.one(label='First Data Set');
   input student year $ state $ gr1 gr2;
   label year='Year of Birth';
   format gr1 4.1;
   datalines;
1000 1970 NC 85 87
1042 1971 MD 92 92
1095 1969 PA 78 72
1187 1970 MA 87 94
;

data proclib.two(label='Second Data Set');
   input student $ year $ state $ gr1
         gr2 major $;
   label state='Home State';
   format gr1 5.2;
   datalines;
1000 1970 NC 84 87 Math
1042 1971 MA 92 92 History
1095 1969 PA 79 73 Physics
1187 1970 MD 87 74 Dance
1204 1971 NC 82 96 French
;
```

PROC COMPARE produces lengthy output. You can use one or more options to determine the kinds of comparisons to make and the degree of detail in the report. For example, in the following PROC COMPARE step, the NOVALUES option suppresses the part of the output that shows the differences in the values of matching variables:

```
proc compare base=proclib.one
             compare=proclib.two novalues;
run;
```

Output 9.1 Comparison of Two Data Sets

```
                               The SAS System                              1

                             COMPARE Procedure
                   Comparison of PROCLIB.ONE with PROCLIB.TWO
                               (Method=EXACT)

                             Data Set Summary

Dataset               Created          Modified  NVar    NObs  Label

PROCLIB.ONE  13MAY98:15:01:42  13MAY98:15:01:42     5       4  First Data Set
PROCLIB.TWO  13MAY98:15:01:44  13MAY98:15:01:44     6       5  Second Data Set


                             Variables Summary

              Number of Variables in Common: 5.
              Number of Variables in PROCLIB.TWO but not in PROCLIB.ONE: 1.
              Number of Variables with Conflicting Types: 1.
              Number of Variables with Differing Attributes: 3.


              Listing of Common Variables with Conflicting Types

                     Variable  Dataset      Type  Length

                     student   PROCLIB.ONE  Num        8
                               PROCLIB.TWO  Char       8


            Listing of Common Variables with Differing Attributes

          Variable  Dataset      Type  Length  Format  Label

          year      PROCLIB.ONE  Char       8          Year of Birth
                    PROCLIB.TWO  Char       8
          state     PROCLIB.ONE  Char       8
                    PROCLIB.TWO  Char       8          Home State
```

```
                          The SAS System                          2

                        COMPARE Procedure
              Comparison of PROCLIB.ONE with PROCLIB.TWO
                          (Method=EXACT)

        Listing of Common Variables with Differing Attributes

      Variable  Dataset      Type  Length  Format  Label

      gr1       PROCLIB.ONE  Num        8  4.1
                PROCLIB.TWO  Num        8  5.2


                       Observation Summary

                  Observation      Base  Compare

                  First Obs           1        1
                  First Unequal       1        1
                  Last  Unequal       4        4
                  Last  Match         4        4
                  Last  Obs           .        5

   Number of Observations in Common: 4.
   Number of Observations in PROCLIB.TWO but not in PROCLIB.ONE: 1.
   Total Number of Observations Read from PROCLIB.ONE: 4.
   Total Number of Observations Read from PROCLIB.TWO: 5.

   Number of Observations with Some Compared Variables Unequal: 4.
   Number of Observations with All Compared Variables Equal: 0.
```

```
                          The SAS System                          3

                        COMPARE Procedure
              Comparison of PROCLIB.ONE with PROCLIB.TWO
                          (Method=EXACT)

                     Values Comparison Summary

   Number of Variables Compared with All Observations Equal: 1.
   Number of Variables Compared with Some Observations Unequal: 3.
   Total Number of Values which Compare Unequal: 6.
   Maximum Difference: 20.


                   Variables with Unequal Values

         Variable  Type  Len   Compare Label  Ndif   MaxDif

         state     CHAR    8   Home State        2
         gr1       NUM     8                     2    1.000
         gr2       NUM     8                     2   20.000
```

"Procedure Output" on page 242 shows the default output for these two data sets. Example 1 on page 251 shows the complete output for these two data sets.

Procedure Syntax

Restriction: You must use the VAR statement when you use the WITH statement.

Tip: Supports the Output Delivery System (see Chapter 2, “Fundamental Concepts for Using Base SAS Procedures”)

Reminder: You can use the LABEL, ATTRIB, FORMAT, and WHERE statements. See Chapter 3, "Statements with the Same Function in Multiple Procedures," for details. You can also use any global statements as well. See Chapter 2, "Fundamental Concepts for Using Base SAS Procedures," for a list.

PROC COMPARE <*option(s)*>;

BY <DESCENDING> *variable-1*
<...<DESCENDING> *variable-n*>
<NOTSORTED>;

ID <DESCENDING> *variable-1*
<...<DESCENDING> *variable-n*>
<NOTSORTED>;

VAR *variable(s)*;

WITH *variable(s)*;

To do this	Use this statement
Produce a separate comparison for each BY group	BY
Identify variables to use to match observations	ID
Restrict the comparison to values of specific variables	VAR
Compare variables of different names	WITH and VAR
Compare two variables in the same data set	WITH and VAR

PROC COMPARE Statement

Restriction: If you omit COMPARE=, you must use the WITH and VAR statements.

Restriction: PROC COMPARE reports errors differently if one or both of the compared data sets are not RADIX addressable. Version 6 compressed files are not RADIX addressable, while, beginning with Version 7, compressed files are RADIX addressable. (The integrity of the data is not compromised; the procedure simply numbers the observations differently.)

Reminder: You can use data set options with the BASE= and COMPARE= options.

PROC COMPARE <*option(s)*>;

To do this	Use this option
Specify the data sets to compare	
Specify the base data set	BASE=
Specify the comparison data set	COMPARE=
Control the output data set	
Create an output data set	OUT=
Write an observation for each observation in the BASE= and COMPARE= data sets	OUTALL
Write an observation for each observation in the BASE= data set	OUTBASE
Write an observation for each observation in the COMPARE= data set	OUTCOMP
Write an observation that contains the differences for each pair of matching observations	OUTDIF
Suppress the writing of observations when all values are equal	OUTNOEQUAL
Write an observation that contains the percent differences for each pair of matching observations	OUTPERCENT
Create an output data set that contains summary statistics	OUTSTATS=
Specify how the values are compared	
Specify the criterion for judging the equality of numeric values	CRITERION=
Specify the method for judging the equality of numeric values	METHOD=
Judge missing values equal to any value	NOMISSBASE and NOMISSCOMP
Control the details in the default report	
Include the values for all matching observations	ALLOBS
Print a table of summary statistics for all pairs of matching variables	ALLSTATS and STATS
Include in the report the values and differences for all matching variables	ALLVARS
Print only a short comparison summary	BRIEFSUMMARY
Change the report for numbers between 0 and 1	FUZZ=
Restrict the number of differences to print	MAXPRINT=
Suppress the print of creation and last-modified dates	NODATE
Suppress all printed output	NOPRINT
Suppress the summary reports	NOSUMMARY
Suppress the value comparison results.	NOVALUES
Produce a complete listing of values and differences	PRINTALL
Print the value differences by observation, not by variable	TRANSPOSE

To do this	Use this option
Control the listing of variables and observations	
List all variables and observations found in only one data set	LISTALL
List all variables and observations found only in the base data set	LISTBASE
List all observations found only in the base data set	LISTBASEOBS
List all variables found only in the base data set	LISTBASEVAR
List all variables and observations found only in the comparison data set	LISTCOMP
List all observations found only in the comparison data set	LISTCOMPOBS
List all variables found only in the comparison data set	LISTCOMPVAR
List variables whose values are judged equal	LISTEQUALVAR
List all observations found in only one data set	LISTOBS
List all variables found in only one data set	LISTVAR

Options

ALLOBS
includes in the report of value comparison results the values and, for numeric variables, the differences for all matching observations, even if they are judged equal.

Default: If you omit ALLOBS, PROC COMPARE prints values only for observations that are judged unequal.

Interaction: When used with the TRANSPOSE option, ALLOBS invokes the ALLVARS option and displays the values for all matching observations and variables.

ALLSTATS
prints a table of summary statistics for all pairs of matching variables.

See also: "Table of Summary Statistics" on page 245 for information on the statistics produced

ALLVARS
includes in the report of value comparison results the values and, for numeric variables, the differences for all pairs of matching variables, even if they are judged equal.

Default: If you omit ALLVARS, PROC COMPARE prints values only for variables that are judged unequal.

Interaction: When used with the TRANSPOSE option, ALLVARS displays unequal values in context with the values for other matching variables. If you omit the TRANSPOSE option, ALLVARS invokes the
ALLOBS option and displays the values for all matching observations and variables.

BASE=*SAS-data-set*
specifies the data set to use as the base data set.

Alias: DATA=

Default: the most recently created SAS data set

Tip: You can use the WHERE= data set option with the BASE= option to limit the observations that are available for comparison.

BRIEFSUMMARY
produces a short comparison summary and suppresses the four default summary reports (data set summary report, variables summary report, observation summary report, and values comparison summary report).

Alias: BRIEF

Tip: By default, a listing of value differences accompanies the summary reports. To suppress this listing, use the NOVALUES option.

Featured in: Example 4 on page 258

COMPARE=*SAS-data-set*
specifies the data set to use as the comparison data set.

Aliases: COMP=, C=

Default: If you omit COMPARE=, the comparison data set is the same as the base data set, and PROC COMPARE compares variables within the data set.

Restriction: If you omit COMPARE=, you must use the WITH statement.

Tip: You can use the WHERE= data set option with COMPARE= to limit the observations that are available for comparison.

CRITERION=γ
specifies the criterion for judging the equality of numeric values. Normally, the value of γ (gamma) is positive, in which case the number itself becomes the equality criterion. If you use a negative value for γ, PROC COMPARE uses an equality criterion proportional to the precision of the computer on which the SAS System is running.

Default: 0.00001

See also: "The Equality Criterion" on page 238 for more information

ERROR
displays an error message in the SAS log when differences are found.

Interaction: This option overrides the WARNING option.

FUZZ=*number*
alters the values comparison results for numbers less than *number*. PROC COMPARE prints

- 0 for any variable value that is less than *number*
- a blank for difference or percent difference if it is less than *number*
- 0 for any summary statistic that is less than *number*.

Default 0

Range: 0 - 1

Tip: A report that contains many trivial differences is easier to read in this form.

LISTALL
lists all variables and observations that are found in only one data set.

Alias LIST

Interaction: using LISTALL is equivalent to using the following four options: LISTBASEOBS, LISTCOMPOBS, LISTBASEVAR, and LISTCOMPVAR.

LISTBASE
lists all observations and variables that are found in the base data set but not in the comparison data set.

Interaction: Using LISTBASE is equivalent to using the LISTBASEOBS and LISTBASEVAR options.

LISTBASEOBS
lists all observations that are found in the base data set but not in the comparison data set.

LISTBASEVAR
lists all variables that are found in the base data set but not in the comparison data set.

LISTCOMP
lists all observations and variables that are found in the comparison data set but not in the base data set.

Interaction: Using LISTCOMP is equivalent to using the LISTCOMPOBS and LISTCOMPVAR options.

LISTCOMPOBS
lists all observations that are found in the comparison data set but not in the base data set.

LISTCOMPVAR
lists all variables that are found in the comparison data set but not in the base data set.

LISTEQUALVAR
prints a list of variables whose values are judged equal at all observations in addition to the default list of variables whose values are judged unequal.

LISTOBS
lists all observations that are found in only one data set.

Interaction: Using LISTOBS is equivalent to using the LISTBASEOBS and LISTCOMPOBS options.

LISTVAR
lists all variables that are found in only one data set.

Interaction: Using LISTVAR is equivalent to using both the LISTBASEVAR and LISTCOMPVAR options.

MAXPRINT=*total* | (*per-variable, total*)
specifies the maximum number of differences to print, where

total
is the maximum total number of differences to print. The default value is 500 unless you use the ALLOBS option (or both the ALLVAR and TRANSPOSE options), in which case the default is 32000.

per-variable
is the maximum number of differences to print for each variable within a BY group. The default value is 50 unless you use the ALLOBS option (or both the ALLVAR and TRANSPOSE options), in which case the default is 1000.

The MAXPRINT= option prevents the output from becoming extremely large when data sets differ greatly.

METHOD=ABSOLUTE | EXACT | PERCENT | RELATIVE<(δ)>
specifies the method for judging the equality of numeric values. The constant δ (delta) is a number between 0 and 1 that specifies a value to add to the denominator when calculating the equality measure. By default, δ is 0.

Unless you use the CRITERION= option, the default method is EXACT. If you use CRITERION=, the default method is RELATIVE(ϕ), where ϕ (phi) is a small number that depends on the numerical precision of the computer on which you are running the SAS System and on the value of CRITERION=.

See also: "The Equality Criterion" on page 238

NODATE

suppresses the display in the data set summary report of the creation dates and the last modified dates of the base and comparison data sets.

NOMISSBASE

judges a missing value in the base data set equal to any value. (By default, a missing value is equal only to a missing value of the same kind, that is .=., .^=.A, .A=.A, .A^=.B, and so on.)

You can use this option to determine the changes that would be made to the observations in the comparison data set if it were used as the master data set and the base data set were used as the transaction data set in a DATA step UPDATE statement. For information on the UPDATE statement, see the chapter on SAS language statements in *SAS Language Reference: Dictionary*.

NOMISSCOMP

judges a missing value in the comparison data set equal to any value. (By default, a missing value is equal only to a missing value of the same kind, that is .=., .^=.A, .A=.A, .A^=.B, and so on.)

You can use this option to determine the changes that would be made to the observations in the base data set if it were used as the master data set and the comparison data set were used as the transaction data set in a DATA step UPDATE statement. For information on the UPDATE statement, see the chapter on SAS language statements in *SAS Language Reference: Dictionary*.

NOMISSING

judges missing values in both the base and comparison data sets equal to any value. By default, a missing value is only equal to a missing value of the same kind, that is .=., .^=.A, .A=.A, .A^=.B, and so on.

Alias: NOMISS

Interaction: Using NOMISSING is equivalent to using both NOMISSBASE and NOMISSCOMP.

NOPRINT

suppresses all printed output.

Tip: You may want to use this option when you are creating one or more output data sets.

Featured in: Example 6 on page 262

NOSUMMARY

suppresses the data set, variable, observation, and values comparison summary reports.

Tips: NOSUMMARY produces no output if there are no differences in the matching values.

Featured in: Example 2 on page 255

NOTE

displays notes in the SAS log describing the results of the comparison, whether or not differences were found.

NOVALUES

suppresses the report of the value comparison results.

Featured in: "Overview" on page 221

OUT=*SAS-data-set*
names the output data set. If *SAS-data-set* does not exist, PROC COMPARE creates it. *SAS-data-set* contains the differences between matching variables.

See also: “Output Data Set (OUT=)” on page 248

Featured in: Example 6 on page 262

OUTALL
writes an observation to the output data set for each observation in the base data set and for each observation in the comparison data set. The option also writes observations to the output data set containing the differences and percent differences between the values in matching observations.

Tip: Using OUTALL is equivalent to using the following four options: OUTBASE, OUTCOMP, OUTDIF, and OUTPERCENT.

See also: “Output Data Set (OUT=)” on page 248

OUTBASE
writes an observation to the output data set for each observation in the base data set, creating observations in which _TYPE_=BASE.

See also: “Output Data Set (OUT=)” on page 248

Featured in: Example 6 on page 262

OUTCOMP
writes an observation to the output data set for each observation in the comparison data set, creating observations in which _TYPE_=COMP.

See also: “Output Data Set (OUT=)” on page 248

Featured in: Example 6 on page 262

OUTDIF
writes an observation to the output data set for each pair of matching observations. The values in the observation include values for the differences between the values in the pair of observations. The value of _TYPE_ in each observation is DIF.

Default: The OUTDIF option is the default unless you specify the OUTBASE, OUTCOMP, or OUTPERCENT option. If you use any of these options, you must explicitly specify the OUTDIF option to create _TYPE_=DIF observations in the output data set.

See also: “Output Data Set (OUT=)” on page 248

Featured in: Example 6 on page 262

OUTNOEQUAL
suppresses the writing of an observation to the output data set when all values in the observation are judged equal. In addition, in observations containing values for some variables judged equal and others judged unequal, the OUTNOEQUAL option uses the special missing value ".E" to represent differences and percent differences for variables judged equal.

See also: “Output Data Set (OUT=)” on page 248

Featured in: Example 6 on page 262

OUTPERCENT
writes an observation to the output data set for each pair of matching observations. The values in the observation include values for the percent differences between the values in the pair of observations. The value of _TYPE_ in each observation is PERCENT.

See also: “Output Data Set (OUT=)” on page 248

OUTSTATS=*SAS-data-set*
writes summary statistics for all pairs of matching variables to the specified *SAS-data-set*.

Tip: If you want to print a table of statistics in the procedure output, use the STATS, ALLSTATS, or PRINTALL option.

See also: "Output Statistics Data Set (OUTSTATS=)" on page 249 and "Table of Summary Statistics" on page 245.

Featured in: Example 7 on page 265

PRINTALL
invokes the following options: ALLVARS, ALLOBS, ALLSTATS, LISTALL, and WARNING.

Featured in: Example 1 on page 251

STATS
prints a table of summary statistics for all pairs of matching numeric variables that are judged unequal.

See also: "Table of Summary Statistics" on page 245 for information on the statistics produced.

TRANSPOSE
prints the reports of value differences by observation instead of by variable.

Interaction: If you also use the NOVALUES option, the TRANSPOSE option lists only the *names* of the variables whose values compare as unequal for each observation, not the values and differences.

See also: "Comparison Results for Observations (Using the TRANSPOSE Option)" on page 247.

WARNING
displays a warning message in the SAS log when differences are found.

Interaction: The ERROR option overrides the WARNING option.

BY Statement

Produces a separate comparison for each BY group.

Main discussion: "BY" on page 68

BY <DESCENDING> *variable-1*
<...<DESCENDING> *variable-n*>
<NOTSORTED>;

Required Arguments

variable
specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. If you do not use the NOTSORTED option in the BY statement, the observations in the data set must be sorted by all the variables that you specify. Variables in a BY statement are called *BY variables*.

Options

DESCENDING
specifies that the observations are sorted in descending order by the variable that immediately follows the word DESCENDING in the BY statement.

NOTSORTED
specifies that observations are not necessarily sorted in alphabetic or numeric order. The observations are grouped in another way, for example, chronological order.
The requirement for ordering observations according to the values of BY variables is suspended for BY-group processing when you use the NOTSORTED option. The procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations with the same values for the BY variables are not contiguous, the procedure treats each contiguous set as a separate BY group.

BY Processing with PROC COMPARE

To use a BY statement with PROC COMPARE, you must sort both the base and comparison data sets by the BY variables. The nature of the comparison depends on whether all BY variables are in the comparison data set and, if they are, whether their attributes match those of the BY variables in the base data set. The following table shows how PROC COMPARE behaves under different circumstances:

Condition	Behavior of PROC COMPARE
All BY variables are in the comparison data set and all attributes match exactly	Compares corresponding BY groups
None of the BY variables are in the comparison data set	Compares each BY group in the base data set with the entire comparison data set
Some BY variables are not in the comparison data set	Writes an error message to the SAS log and terminates
Some BY variables have different types in the two data sets	Writes an error message to the SAS log and terminates

ID Statement

Lists variables to use to match observations.

See also: "A Comparison with an ID Variable" on page 237

Featured in: Example 5 on page 259

ID <DESCENDING> *variable-1*
<...<DESCENDING> *variable-n*>
<NOTSORTED>;

Required Arguments

variable
specifies the variable that the procedure uses to match observations. You can specify more than one variable, but the data set must be sorted by the variable or variables you specify. These variables are *ID variables*. ID variables also identify observations on the printed reports and in the output data set.

Options

DESCENDING
specifies that the data set is sorted in descending order by the variable that immediately follows the word DESCENDING in the ID statement.

If you use the DESCENDING option, you must sort the data sets. The SAS System does not use an index to process an ID statement with the DESCENDING option. Further, the use of DESCENDING for ID variables must correspond to the use of the DESCENDING option in the BY statement in the PROC SORT step that was used to sort the data sets.

NOTSORTED
specifies that observations are not necessarily sorted in alphabetic or numeric order. The data are grouped in another way, for example, chronological order.

See also: “Comparing Unsorted Data” on page 234

Requirements for ID Variables

- ID variables must be in the BASE= data set or PROC COMPARE stops processing.
- If an ID variable is not in the COMPARE= data set, PROC COMPARE prints a warning to the SAS log and does not use that variable to match observations in the comparison data set (but does write it to the OUT= data set).
- ID variables must be of the same type in both data sets.
- You should sort both data sets by the common ID variables (within the BY variables, if any) unless you specify the NOTSORTED option.

Comparing Unsorted Data

If you do not want to sort the data set by the ID variables, you can use the NOTSORTED option. When you specify the NOTSORTED option, or if the ID statement is omitted, PROC COMPARE matches the observations one-to-one. That is, PROC COMPARE matches the first observation in the base data set with the first observation in the comparison data set, the second with the second, and so on. If you use NOTSORTED, and the ID values of corresponding observations are not the same, PROC COMPARE prints an error message and stops processing.

If the data sets are not sorted by the common ID variables and you do not specify the NOTSORTED option, PROC COMPARE prints a warning message and continues to process the data sets as if you had specified NOTSORTED.

Avoiding Duplicate ID Values

The observations in each data set should be uniquely labeled by the values of the ID variables. If PROC COMPARE finds two successive observations with the same ID values in a data set, it

- prints the warning **`Duplicate Observations`** for the first occurrence for that data set

- prints the total number of duplicate observations found in the data set in the observation summary report
- uses the first observation with the duplicate value for the comparison.

When the data sets are not sorted, PROC COMPARE detects only those duplicate observations that occur in succession.

VAR Statement

Restricts the comparison of the values of variables to those named in the VAR statement.

Featured in: Example 2 on page 255, Example 3 on page 256, and Example 4 on page 258

VAR *variable(s)*;

Required Arguments

variable(s)
one or more variables that appear in the BASE= and COMPARE= data sets or only in the BASE= data set.

Details

- If you do not use the VAR statement, PROC COMPARE compares the values of all matching variables except those appearing in BY and ID statements.
- If a variable in the VAR statement does not exist in the COMPARE= data set, PROC COMPARE writes a warning to the SAS log and ignores the variable.
- If a variable in the VAR statement does not exist in the BASE= data set, PROC COMPARE stops processing and gives an error message.
- The VAR statement restricts only the comparison of values of matching variables. PROC COMPARE still reports on the total number of matching variables and compares their attributes. However, it produces neither error nor warning messages about these variables.

WITH Statement

Compares variables in the base data set with variables that have different names in the comparison data set, and compares different variables that are in the same data set.

Restriction: You must use the VAR statement when you use the WITH statement.

Featured in: Example 2 on page 255, Example 3 on page 256, and Example 4 on page 258

WITH *variable(s)*;

Required Arguments

variable(s)
one or more variables to compare with variables in the VAR statement.

Comparing Selected Variables

If you want to compare variables in the base data set with variables with different names in the comparison data set, specify the names of the variables in the base data set in the VAR statement and the names of the matching variables in the WITH statement. The first variable that you list in the WITH statement corresponds to the first variable that you list in the VAR statement, the second with the second, and so on. If the WITH statement list is shorter than the VAR statement list, PROC COMPARE assumes that the extra variables in the VAR statement have the same names in the comparison data set as they do in the base data set. If the WITH statement list is longer than the VAR statement list, PROC COMPARE ignores the extra variables.

A variable name can appear any number of times in the VAR statement or the WITH statement. By selecting VAR and WITH statement lists, you can compare the variables in any permutation.

If you omit the COMPARE= option in the PROC COMPARE statement, you must use the WITH statement. In this case, PROC COMPARE compares the values of variables with different names in the BASE= data set.

Concepts

PROC COMPARE first compares the following:

- □ data set attributes (set by the data set options TYPE= and LABEL=).
- □ variables. PROC COMPARE checks each variable in one data set to determine whether it matches a variable in the other data set.
- □ attributes (type, length, labels, formats, and informats) of matching variables.
- □ observations. PROC COMPARE checks each observation in one data set to determine whether it matches an observation in the other data set. PROC COMPARE either matches observations by their position in the data sets or by the values of the ID variable.

After making these comparisons, PROC COMPARE compares the values in the parts of the data sets that match. PROC COMPARE either compares the data by the position of observations or by the values of an ID variable.

A Comparison by Position of Observations

Figure 9.1 on page 237 shows two data sets. The data inside the shaded boxes show the part of the data sets that the procedure compares. Assume that variables with the same names have the same type.

Figure 9.1 Comparison by the Positions of Observations

Data Set ONE

IDNUM	NAME	GENDER	GPA
2998	Bagwell	f	3.722
9866	Metcalf	m	3.342
2118	Gray	f	3.177
3847	Baglione	f	4.000
2342	Hall	m	3.574

Data Set TWO

IDNUM	NAME	GENDER	GPA	YEAR
2998	Bagwell	f	3.722	2
9866	Metcalf	m	3.342	2
2118	Gray	f	3.177	3
3847	Baglione	f	4.000	4
2342	Hall	m	3.574	4
7565	Gold	f	3.609	2
1755	Syme	f	3.883	3

When you use PROC COMPARE to compare data set TWO with data set ONE, the procedure compares the first observation in data set ONE with the first observation in data set TWO, and it compares the second observation in the first data set with the second observation in the second data set, and so on. In each observation that it compares, the procedure compares the values of the IDNUM, NAME, GENDER, and GPA.

The procedure does not report on the values of the last two observations or the variable YEAR in data set TWO because there is nothing to compare them with in data set ONE.

A Comparison with an ID Variable

In a simple comparison, PROC COMPARE uses the observation number to determine which observations to compare. When you use an ID variable, PROC COMPARE uses the values of the ID variable to determine which observations to compare. ID variables should have unique values and must have the same type.

For the two data sets shown in Figure 9.2 on page 238, assume that IDNUM is an ID variable and that IDNUM has the same type in both data sets. The procedure compares the observations that have the same value for IDNUM. The data inside the shaded boxes show the part of the data sets that the procedure compares.

Figure 9.2 Comparison by the Value of the ID Variable

Data Set ONE

IDNUM	NAME	GENDER	GPA
2998	Bagwell	f	3.722
9866	Metcalf	m	3.342
2118	Gray	f	3.177
3847	Baglione	f	4.000
2342	Hall	m	3.574

Data Set TWO

IDNUM	NAME	GENDER	GPA	YEAR
2998	Bagwell	f	3.722	2
9866	Metcalf	m	3.342	2
2118	Gray	f	3.177	3
3847	Baglione	f	4.000	4
2342	Hall	m	3.574	4
7565	Gold	f	3.609	2
1755	Syme	f	3.883	3

The data sets contain three matching variables: NAME, GENDER, and GPA. They also contain five matching observations – the observations with values of **2998**, **9866**, **2118**, **3847**, and **2342** for IDNUM.

Data Set TWO contains two observations (IDNUM= **7565** and IDNUM= **1755**) for which data set ONE contains no matching observations. Similarly, no variable in data set ONE matches the variable YEAR in data set TWO.

See Example 5 on page 259 for an example that uses an ID variable.

The Equality Criterion

The COMPARE procedure judges numeric values unequal if the magnitude of their difference, as measured according to the METHOD= option, is greater than the value of the CRITERION= option. PROC COMPARE provides four methods for applying CRITERION=:

- The EXACT method tests for exact equality.
- The ABSOLUTE method compares the absolute difference to the value specified by CRITERION=.
- The RELATIVE method compares the absolute relative difference to the value specified by CRITERION=.
- The PERCENT method compares the absolute percent difference to the value specified by CRITERION=.

For a numeric variable compared, let x be its value in the base data set and let y be its value in the comparison data set. If both x and y are nonmissing, the values are judged unequal according to the value of METHOD= and the value of CRITERION= (γ) as follows:

- If METHOD=EXACT, the values are unequal if y does not equal x.
- If METHOD=ABSOLUTE, the values are unequal if

$$\mathrm{ABS}\,(y-x) > \gamma$$

- If METHOD=RELATIVE, the values are unequal if

$$\mathrm{ABS}\,(y-x)\,/\,((\mathrm{ABS}\,(x)+\mathrm{ABS}\,(y))\,/2+\delta) > \gamma$$

  The values are equal if *x*=*y*=0.

- If METHOD=PERCENT, the values are unequal if

$$100\,(\mathrm{ABS}\,(y-x)\,/\mathrm{ABS}\,(x)) > \gamma \;\text{ for }\; x \neq 0$$

  or

$$y \neq 0 \;\text{ for }\; \mathrm{x} = 0\;\;.$$

If *x* or *y* is missing, then the comparison depends on the NOMISSING option. If NOMISSING is in effect, a missing value will always compare equal to anything. Otherwise, a missing value is judged equal only to a missing value of the same type, (that is, .=., .^=.A, .A=.A, .A^=.B, and so on).

If the value specified for CRITERION= is negative, the actual criterion used is made equal to the absolute value of γ times a very small number ϵ (epsilon) that depends on the numerical precision of the computer. This number ϵ is defined as the smallest positive floating-point value such that, using machine arithmetic, $1-\epsilon<1<1+\epsilon$. Round-off or truncation error in floating-point computations is typically a few orders of magnitude larger than ϵ. This means that CRITERION=–1000 often provides a reasonable test of the equality of computed results at the machine level of precision.

The value δ added to the denominator in the RELATIVE method is specified in parentheses after the method name: METHOD=RELATIVE(δ). If not specified in METHOD=, δ defaults to 0. The value of δ can be used to control the behavior of the error measure when both *x* and *y* are very close to 0. If δ is not given and *x* and *y* are very close to 0, any error produces a large relative error (in the limit, 2).

Specifying a value for δ avoids this extreme sensitivity of the RELATIVE method for small values. If you specify METHOD=RELATIVE(δ) CRITERION=γ when both *x* and *y* are much smaller than δ in absolute value, the comparison is as if you had specified METHOD=ABSOLUTE CRITERION=$\delta\gamma$. However, when either *x* or *y* is much larger than δ in absolute value, the comparison is like METHOD=RELATIVE CRITERION=γ. For moderate values of *x* and *y*, METHOD=RELATIVE(δ) CRITERION=γ is, in effect, a compromise between METHOD=ABSOLUTE CRITERION=$\delta\ \gamma$ and METHOD=RELATIVE CRITERION=γ.

For character variables, if one value has a greater length than the other, the shorter value is padded with blanks for the comparison. Nonblank character values are judged equal only if they agree at each character. If NOMISSING is in effect, blank character values compare equal to anything.

Definition of Difference and Percent Difference

In the reports of value comparisons and in the OUT= data set, PROC COMPARE displays difference and percent difference values for the numbers compared. These quantities are defined using the value from the base data set as the reference value. For a numeric variable compared, let x be its value in the base data set and let y be its value in the comparison data set. If x and y are both nonmissing, the difference and percent difference are defined as follows:

Difference = $y - x$

Percent Difference = $(y - x)/x * 100 \text{ for } x \neq 0$

Percent Difference = missing for $x = 0$.

Formatted Values

PROC COMPARE compares unformatted values. If you have two matching variables that are formatted differently, PROC COMPARE lists the formats of the variables.

Results

PROC COMPARE reports the results of its comparisons in the following ways:

- the SAS log
- return codes stored in the automatic macro SYSINFO
- procedure output
- output data sets.

SAS Log

When you use the WARNING, PRINTALL, or ERROR option, PROC COMPARE writes a description of the differences to the SAS log.

Macro Return Codes (SYSINFO)

PROC COMPARE stores a return code in the automatic macro variable SYSINFO. The value of the return code provides information about the result of the comparison. By checking the value of SYSINFO after PROC COMPARE has run and before any other step begins, SAS macros can use the results of a PROC COMPARE step to determine what action to take or what parts of a SAS program to execute.

Table 9.1 on page 240 is a key for interpreting the SYSINFO return code from PROC COMPARE. For each of the conditions listed, the associated value is added to the return code if the condition is true. Thus, the SYSINFO return code is the sum of the codes listed in Table 9.1 on page 240 for the applicable conditions:

Table 9.1 Macro Return Codes

Bit	Condition	Code	Hex	Description
1	DSLABEL	1	0001X	Data set labels differ
2	DSTYPE	2	0002X	Data set types differ
3	INFORMAT	4	0004X	Variable has different informat
4	FORMAT	8	0008X	Variable has different format

Bit	Condition	Code	Hex	Description
5	LENGTH	16	0010X	Variable has different length
6	LABEL	32	0020X	Variable has different label
7	BASEOBS	64	0040X	Base data set has observation not in comparison
8	COMPOBS	128	0080X	Comparison data set has observation not in base
9	BASEBY	256	0100X	Base data set has BY group not in comparison
10	COMPBY	512	0200X	Comparison data set has BY group not in base
11	BASEVAR	1024	0400X	Base data set has variable not in comparison
12	COMPVAR	2048	0800X	Comparison data set has variable not in base
13	VALUE	4096	1000X	A value comparison was unequal
14	TYPE	8192	2000X	Conflicting variable types
15	BYVAR	16384	4000X	BY variables do not match
16	ERROR	32768	8000X	Fatal error: comparison not done

These codes are ordered and scaled to allow a simple check of the degree to which the data sets differ. For example, if you want to check that two data sets contain the same variables, observations, and values, but you do not care about differences in labels, formats, and so forth, use the following statements:

```
proc compare base=SAS-data-set
             compare=SAS-data-set;
run;

%if &sysinfo >= 64 %then
   %do;
      handle error;
   %end;
```

You can examine individual bits in the SYSINFO value by using DATA step bit-testing features to check for specific conditions. For example, to check for the presence of observations in the base data set that are not in the comparison data set, use the following statements:

```
proc compare base=SAS-data-set
             compare=SAS-data-set;
run;

%let rc=&sysinfo;
data _null_;
   if &rc='1......'b then
      put 'Observations in Base but not
           in Comparison Data Set';
run;
```

PROC COMPARE must run before you check SYSINFO and you must obtain the SYSINFO value before another SAS step starts because every SAS step resets SYSINFO.

Procedure Output

The following sections show and describe the default output of the two data sets shown in “Overview” on page 221. Because PROC COMPARE produces lengthy output, the output is presented in seven pieces.

Data Set Summary

This report lists the attributes of the data sets being compared. These attributes include the following:

- the data set names
- the data set types, if any
- the data set labels, if any
- the dates created and last modified
- the number of variables in each data set
- the number of observations in each data set.

Output 9.2 on page 242 shows the Data Set Summary.

Output 9.2 Partial Output

```
                           COMPARE Procedure
                 Comparison of PROCLIB.ONE with PROCLIB.TWO
                              (Method=EXACT)

                            Data Set Summary

Dataset              Created          Modified  NVar    NObs  Label


PROCLIB.ONE  11SEP97:15:11:07  11SEP97:15:11:09     5       4  First Data Set
PROCLIB.TWO  11SEP97:15:11:10  11SEP97:15:11:10     6       5  Second Data Set
```

Variables Summary

This report compares the variables in the two data sets. The first part of the report lists the following:

- the number of variables the data sets have in common
- the number of variables in the base data set that are not in the comparison data set and vice versa
- the number of variables in both data sets that have different types
- the number of variables that differ on other attributes (length, label, format, or informat)
- the number of BY, ID, VAR, and WITH variables specified for the comparison.

The second part of the report lists matching variables with different attributes and shows how the attributes differ. (The COMPARE procedure omits variable labels if the line size is too small for them.)

Output 9.3 on page 243 shows the Variables Summary.

Output 9.3 Partial Output

```
                          Variables Summary

                Number of Variables in Common: 5.
                Number of Variables in PROCLIB.TWO but not in PROCLIB.ONE: 1.
                Number of Variables with Conflicting Types: 1.
                Number of Variables with Differing Attributes: 3.


                Listing of Common Variables with Conflicting Types

                       Variable  Dataset      Type  Length

                       student   PROCLIB.ONE  Num        8
                                 PROCLIB.TWO  Char       8


              Listing of Common Variables with Differing Attributes

            Variable  Dataset      Type  Length  Format  Label

            year      PROCLIB.ONE  Char       8          Year of Birth
                      PROCLIB.TWO  Char       8
            state     PROCLIB.ONE  Char       8
                      PROCLIB.TWO  Char       8          Home State
            gr1       PROCLIB.ONE  Num        8  4.1
                      PROCLIB.TWO  Num        8  5.2
```

Observation Summary

This report provides information about observations in the base and comparison data sets. First of all, the report identifies the first and last observation in each data set, the first and last matching observations, and the first and last differing observations. Then, the report lists the following:

- the number of observations that the data sets have in common
- the number of observations in the base data set that are not in the comparison data set and vice versa
- the total number of observations in each data set
- the number of matching observations for which PROC COMPARE judged some variables unequal
- the number of matching observations for which PROC COMPARE judged all variables equal.

Output 9.4 on page 243 shows the Observation Summary.

Output 9.4 Partial Output

```
                       Observation Summary

                    Observation      Base  Compare

                    First Obs           1        1
                    First Unequal       1        1
                    Last  Unequal       4        4
                    Last  Match         4        4
                    Last  Obs           .        5

     Number of Observations in Common: 4.
     Number of Observations in PROCLIB.TWO but not in PROCLIB.ONE: 1.
     Total Number of Observations Read from PROCLIB.ONE: 4.
     Total Number of Observations Read from PROCLIB.TWO: 5.

     Number of Observations with Some Compared Variables Unequal: 4.
     Number of Observations with All Compared Variables Equal: 0.
```

Values Comparison Summary

This report first lists the following:

- the number of variables compared with all observations equal
- the number of variables compared with some observations unequal
- the number of variables with differences involving missing values, if any
- the total number of values judged unequal
- the maximum difference measure between unequal values for all pairs of matching variables (for differences not involving missing values).

In addition, for the variables for which some matching observations have unequal values, the report lists

- the name of the variable
- other variable attributes
- the number of times PROC COMPARE judged the variable unequal
- the maximum difference measure found between values (for differences not involving missing values)
- the number of differences caused by comparison with missing values, if any.

Output 9.5 on page 244 shows the Values Comparison Summary.

Output 9.5 Partial Output

```
                    Values Comparison Summary

     Number of Variables Compared with All Observations Equal: 1.
     Number of Variables Compared with Some Observations Unequal: 3.
     Total Number of Values which Compare Unequal: 6.
     Maximum Difference: 20.

                  Variables with Unequal Values

           Variable  Type  Len   Compare Label  Ndif   MaxDif

           state     CHAR    8   Home State        2
           gr1       NUM     8                     2    1.000
           gr2       NUM     8                     2   20.000
```

Value Comparison Results

This report consists of a table for each pair of matching variables judged unequal at one or more observations. When comparing character values, PROC COMPARE displays only the first 20 characters. When you use the TRANSPOSE option, it displays only the first 12 characters. Each table shows

- □ the number of the observation or, if you use the ID statement, the values of the ID variables
- □ the value of the variable in the base data set
- □ the value of the variable in the comparison data set
- □ the difference between these two values (numeric variables only)
- □ the percent difference between these two values (numeric variables only).

Output 9.6 on page 245 shows the Value Comparison Results for Variables.

Output 9.6 Partial Output

```
                  Value Comparison Results for Variables

            ___________________________________________________________
                      ||  Home State
                      ||  Base Value          Compare Value
                  Obs ||  state                state
             ________ ||  ________             ________
                      ||
                    2 ||  MD                   MA
                    4 ||  MA                   MD
            ___________________________________________________________


            ___________________________________________________________
                      ||       Base    Compare
                  Obs ||        gr1        gr1      Diff.     % Diff
             ________ ||  _________  _________  _________  _________
                      ||
                    1 ||       85.0      84.00    -1.0000    -1.1765
                    3 ||       78.0      79.00     1.0000     1.2821
            ___________________________________________________________


            ___________________________________________________________
                      ||       Base    Compare
                  Obs ||        gr2        gr2      Diff.     % Diff
             ________ ||  _________  _________  _________  _________
                      ||
                    3 ||    72.0000    73.0000     1.0000     1.3889
                    4 ||    94.0000    74.0000   -20.0000   -21.2766
            ___________________________________________________________
```

You can suppress the value comparison results with the NOVALUES option. If you use both the NOVALUES and TRANSPOSE options, PROC COMPARE lists for each observation the names of the variables with values judged unequal but does not display the values and differences.

Table of Summary Statistics

If you use the STATS, ALLSTATS, or PRINTALL options, the Value Comparison Results for Variables section contains summary statistics for the numeric variables being compared. The STATS option generates these statistics for only the numeric variables whose values are judged unequal. The ALLSTATS and PRINTALL options generate these statistics for all numeric variables, even if all values are judged equal.

Note: In all cases PROC COMPARE calculates the summary statistics based on all matching observations that do not contain missing values, not just on those containing unequal values. △

Output 9.7 on page 246 shows the following summary statistics for base data set values, comparison data set values, differences, and percent differences:

N
: the number of nonmissing values

MEAN
: the mean, or average, of the values

STD
: the standard deviation

MAX
: the maximum value

MIN
: the minimum value

STDERR
: the standard error of the mean

T
: the T ratio (MEAN/STDERR)

PROB> | T |
: the probability of a greater absolute T value if the true population mean is 0.

NDIF
: the number of matching observations judged unequal, and the percent of the matching observations that were judged unequal.

DIFMEANS
: the difference between the mean of the base values and the mean of the comparison values. This line contains three numbers. The first is the mean expressed as a percentage of the base values mean. The second is the mean expressed as a percentage of the comparison values mean. The third is the difference in the two means (the comparison mean minus the base mean).

R
: the correlation of the base and comparison values for matching observations that are nonmissing in both data sets.

RSQ
: the square of the correlation of the base and comparison values for matching observations that are nonmissing in both data sets.

Output 9.7 on page 246 is from the ALLSTATS option using the two data sets shown in “Overview”:

Output 9.7 Partial Output

```
                Value Comparison Results for Variables

           __________________________________________________________
                     ||       Base    Compare
                 Obs ||        gr1        gr1      Diff.     % Diff
            ________ ||  _________  _________  _________  _________
                     ||
                   1 ||       85.0      84.00    -1.0000    -1.1765
                   3 ||       78.0      79.00     1.0000     1.2821
            ________ ||  _________  _________  _________  _________
                     ||
             N       ||          4          4          4          4
             Mean    ||    85.5000    85.5000          0     0.0264
             Std     ||     5.8023     5.4467     0.8165     1.0042
             Max     ||    92.0000    92.0000     1.0000     1.2821
             Min     ||    78.0000    79.0000    -1.0000    -1.1765
             StdErr  ||     2.9011     2.7234     0.4082     0.5021
             t       ||    29.4711    31.3951     0.0000     0.0526
            Prob>|t| ||     <.0001     <.0001     1.0000     0.9614
                     ||
             Ndif    ||          2    50.000%
            DifMeans ||     0.000%     0.000%          0
             r, rsq  ||      0.991      0.983
           __________________________________________________________


           __________________________________________________________
                     ||       Base    Compare
                 Obs ||        gr2        gr2      Diff.     % Diff
            ________ ||  _________  _________  _________  _________
                     ||
                   3 ||    72.0000    73.0000     1.0000     1.3889
                   4 ||    94.0000    74.0000   -20.0000   -21.2766
            ________ ||  _________  _________  _________  _________
                     ||
             N       ||          4          4          4          4
             Mean    ||    86.2500    81.5000    -4.7500    -4.9719
             Std     ||     9.9457     9.4692    10.1776    10.8895
             Max     ||    94.0000    92.0000     1.0000     1.3889
             Min     ||    72.0000    73.0000   -20.0000   -21.2766
             StdErr  ||     4.9728     4.7346     5.0888     5.4447
             t       ||    17.3442    17.2136    -0.9334    -0.9132
            Prob>|t| ||     0.0004     0.0004     0.4195     0.4285
                     ||
             Ndif    ||          2    50.000%
            DifMeans ||    -5.507%    -5.828%    -4.7500
             r, rsq  ||      0.451      0.204
           __________________________________________________________
```

Note: If you use a wide line size with PRINTALL, PROC COMPARE prints the value comparison result for character variables next to the result for numeric variables. In that case, PROC COMPARE calculates only NDIF for the character variables. △

Comparison Results for Observations (Using the TRANSPOSE Option)

The TRANSPOSE option prints the comparison results by observation instead of by variable. The comparison results precede the observation summary report. By default, the source of the values for each row of the table is indicated by the following label:

```
_OBS_1=number-1  _OBS_2=number-2
```

where *number-1* is the number of the observation in the base data set for which the value of the variable is shown, and *number-2* is the number of the observation in the comparison data set.

Output 9.8 on page 248 shows the differences in PROCLIB.ONE and PROCLIB.TWO by observation instead of by variable.

Output 9.8 Partial Output

```
                    Comparison Results for Observations

        _OBS_1=1 _OBS_2=1:
        Variable   Base Value       Compare         Diff.        % Diff
             gr1         85.0         84.00     -1.000000     -1.176471

        _OBS_1=2 _OBS_2=2:
        Variable   Base Value       Compare
           state           MD            MA

        _OBS_1=3 _OBS_2=3:
        Variable   Base Value       Compare         Diff.        % Diff
             gr1         78.0         79.00      1.000000      1.282051
             gr2    72.000000     73.000000      1.000000      1.388889

        _OBS_1=4 _OBS_2=4:
        Variable   Base Value       Compare         Diff.        % Diff
             gr2    94.000000     74.000000    -20.000000    -21.276596
           state           MA            MD
```

If you use an ID statement, the identifying label has the following form:

ID-1=ID-value-1 ... ID-n=ID-value-n

where *ID* is the name of an ID variable and *ID-value* is the value of the ID variable.

Note: When you use the TRANSPOSE option, PROC COMPARE prints only the first 12 characters of the value. △

Output Data Set (OUT=)

By default, the OUT= data set contains an observation for each pair of matching observations. The OUT= data set contains the following variables from the data sets you are comparing:

- all variables named in the BY statement
- all variables named in the ID statement
- all matching variables or, if you use the VAR statement, all variables listed in the VAR statement.

In addition, the data set contains two variables created by PROC COMPARE to identify the source of the values for the matching variables: _TYPE_ and _OBS_.

TYPE
is a character variable of length 8. Its value indicates the source of the values for the matching (or VAR) variables in that observation. (For ID and BY variables, which are not compared, the values are the values from the original data sets.) _TYPE_ has the label **`Type of Observation`**. The four possible values of this variable are as follows:

BASE
The values in this observation are from an observation in the base data set. PROC COMPARE writes this type of observation to the OUT= data set when you specify the OUTBASE option.

COMPARE
The values in this observation are from an observation in the comparison data set. PROC COMPARE writes this type of observation to the OUT= data set when you specify the OUTCOMP option.

DIF
The values in this observation are the differences between the values in the base and comparison data sets. For character variables, PROC COMPARE uses a period (.) to represent equal characters and an X to represent unequal characters. PROC COMPARE writes this type of observation to the OUT= data set by default. However, if you request any other type of observation with the OUTBASE, OUTCOMP, or OUTPERCENT option, you must specify the OUTDIF option to generate observations of this type in the OUT= data set.

PERCENT
The values in this observation are the percent differences between the values in the base and comparison data sets. For character variables the values in observations of type PERCENT are the same as the values in observations of type DIF.

OBS
is a numeric variable containing a number further identifying the source of the OUT= observations.

For observations with _TYPE_ equal to BASE, _OBS_ is the number of the observation in the base data set from which the values of the VAR variables were copied. Similarly, for observations with _TYPE_ equal to COMPARE, _OBS_ is the number of the observation in the comparison data set from which the values of the VAR variables were copied.

For observations with _TYPE_ equal to DIF or PERCENT, _OBS_ is a sequence number that counts the matching observations in the BY group.

OBS has the label `Observation Number`.

The COMPARE procedure takes variable names and attributes for the OUT= data set from the base data set except for the lengths of ID and VAR variables, for which it uses the longer length regardless of which data set that length is from. This behavior has two important repercussions:

- If you use the VAR and WITH statements, the names of the variables in the OUT= data set come from the VAR statement. Thus, observations with _TYPE_ equal to `BASE` contain the values of the VAR variables, while observations with _TYPE_ equal to `COMPARE` contain the values of the WITH variables.
- If you include a variable more than once in the VAR statement in order to compare it with more than one variable, PROC COMPARE can include only the first comparison in the OUT= data set because each variable must have a unique name. Other comparisons produce warning messages.

For an example of the OUT= option, see Example 6 on page 262.

Output Statistics Data Set (OUTSTATS=)

When you use the OUTSTATS= option, PROC COMPARE calculates the same summary statistics as the ALLSTATS option for each pair of numeric variables compared (see “Table of Summary Statistics” on page 245). The OUTSTATS= data set contains an observation for each summary statistic for each pair of variables. The data set also contains the BY variables used in the comparison and several variables created by PROC COMPARE:

VAR
: is a character variable containing the name of the variable from the base data set for which the statistic in the observation was calculated.

WITH
: is a character variable containing the name of the variable from the comparison data set for which the statistic in the observation was calculated. The _WITH_ variable is not included in the OUTSTATS= data set unless you use the WITH statement.

TYPE
: is a character variable containing the name of the statistic contained in the observation. Values of the _TYPE_ variable are **N**, **MEAN**, **STD**, **MIN**, **MAX**, **STDERR**, **T**, **PROBT**, **NDIF**, **DIFMEANS**, and **R**, **RSQ**.

BASE
: is a numeric variable containing the value of the statistic calculated from the values of the variable named by _VAR_ in the observations in the base data set with matching observations in the comparison data set.

COMP
: is a numeric variable containing the value of the statistic calculated from the values of the variable named by the _VAR_ variable (or by the _WITH_ variable if you use the WITH statement) in the observations in the comparison data set with matching observations in the base data set.

DIF
: is a numeric variable containing the value of the statistic calculated from the differences of the values of the variable named by the _VAR_ variable in the base data set and the matching variable (named by the _VAR_ or _WITH_ variable) in the comparison data set.

PCTDIF
: is a numeric variable containing the value of the statistic calculated from the percent differences of the values of the variable named by the _VAR_ variable in the base data set and the matching variable (named by the _VAR_ or _WITH_ variable) in the comparison data set.

Note: For both types of output data sets, PROC COMPARE assigns one of the following data set labels:

```
Comparison of base-SAS-data-set
with comparison-SAS-data-set

Comparison of variables in base-SAS-data-set
```

△

Labels are limited to 40 characters.

See Example 7 on page 265 for an example of an OUTSTATS= data set.

Examples

Example 1: Producing a Complete Report of the Differences

Procedure features:
PROC COMPARE statement options
BASE=
PRINTALL
COMPARE=

Data sets:
PROCLIB.ONE, PROCLIB.TWO on page 222

This example shows the most complete report that PROC COMPARE produces as procedure output.

Program

```
libname proclib 'SAS-data-library';
```

```
options nodate pageno=1 linesize=80 pagesize=40;
```

BASE= and COMPARE= specify the data sets to compare. PRINTALL prints a full report of the differences.

```
proc compare base=proclib.one compare=proclib.two printall;
   title 'Comparing Two Data Sets: Full Report';
run;
```

Output

A ✓ in the output marks information that is in the full report but not in the default report. The additional information includes a listing of variables found in one data set but not the other, a listing of observations found in one data set but not the other, a listing of variables with all equal values, and summary statistics. For an explanation of the statistics, see “Table of Summary Statistics” on page 245.

```
                     Comparing Two Data Sets: Full Report                     1

                              COMPARE Procedure
                   Comparison of PROCLIB.ONE with PROCLIB.TWO
                                (Method=EXACT)

                              Data Set Summary

Dataset               Created          Modified  NVar    NObs  Label

PROCLIB.ONE  11SEP97:16:19:59  11SEP97:16:20:01     5       4  First Data Set
PROCLIB.TWO  11SEP97:16:20:01  11SEP97:16:20:01     6       5  Second Data Set


                              Variables Summary

              Number of Variables in Common: 5.
              Number of Variables in PROCLIB.TWO but not in PROCLIB.ONE: 1.
              Number of Variables with Conflicting Types: 1.
              Number of Variables with Differing Attributes: 3.


          Listing of Variables in PROCLIB.TWO but not in PROCLIB.ONE

                            Variable  Type  Length

✓                            major     Char       8


              Listing of Common Variables with Conflicting Types

                     Variable  Dataset      Type  Length

                     student   PROCLIB.ONE  Num        8
                               PROCLIB.TWO  Char       8
```

```
                    Comparing Two Data Sets: Full Report                    2

                              COMPARE Procedure
                   Comparison of PROCLIB.ONE with PROCLIB.TWO
                                (Method=EXACT)

              Listing of Common Variables with Differing Attributes

            Variable  Dataset      Type  Length  Format  Label

            year      PROCLIB.ONE  Char       8          Year of Birth
                      PROCLIB.TWO  Char       8
            state     PROCLIB.ONE  Char       8
                      PROCLIB.TWO  Char       8          Home State
            gr1       PROCLIB.ONE  Num        8  4.1
                      PROCLIB.TWO  Num        8  5.2


                       Comparison Results for Observations

✓         Observation 5 in PROCLIB.TWO not found in PROCLIB.ONE.


                               Observation Summary

                        Observation      Base  Compare

                        First Obs           1        1
                        First Unequal       1        1
                        Last  Unequal       4        4
                        Last  Match         4        4
                        Last  Obs           .        5

         Number of Observations in Common: 4.
         Number of Observations in PROCLIB.TWO but not in PROCLIB.ONE: 1.
         Total Number of Observations Read from PROCLIB.ONE: 4.
         Total Number of Observations Read from PROCLIB.TWO: 5.

         Number of Observations with Some Compared Variables Unequal: 4.
         Number of Observations with All Compared Variables Equal: 0.
```

```
                    Comparing Two Data Sets: Full Report                    3

                              COMPARE Procedure
                   Comparison of PROCLIB.ONE with PROCLIB.TWO
                                (Method=EXACT)

                           Values Comparison Summary

         Number of Variables Compared with All Observations Equal: 1.
         Number of Variables Compared with Some Observations Unequal: 3.
         Total Number of Values which Compare Unequal: 6.
         Maximum Difference: 20.


                         Variables with All Equal Values

✓                      Variable  Type  Len   Label

                     year      CHAR    8   Year of Birth

                          Variables with Unequal Values

               Variable  Type  Len   Compare Label  Ndif   MaxDif

               state     CHAR    8   Home State        2
               gr1       NUM     8                     2    1.000
               gr2       NUM     8                     2   20.000
```

```
                 Comparing Two Data Sets: Full Report                   4

                          COMPARE Procedure
                Comparison of PROCLIB.ONE with PROCLIB.TWO
                            (Method=EXACT)

                Value Comparison Results for Variables

         __________________________________________________________
                    ||  Year of Birth
                    ||  Base Value           Compare Value
               Obs  ||  year                  year
          ________  ||  ________              ________
                    ||
                 1  ||  1970                  1970
                 2  ||  1971                  1971
                 3  ||  1969                  1969
                 4  ||  1970                  1970
         __________________________________________________________


         __________________________________________________________
                    ||  Home State
                    ||  Base Value           Compare Value
               Obs  ||  state                 state
          ________  ||  ________              ________
                    ||
                 1  ||  NC                    NC
                 2  ||  MD                    MA
                 3  ||  PA                    PA
                 4  ||  MA                    MD
         __________________________________________________________
```

```
                 Comparing Two Data Sets: Full Report                   5

                          COMPARE Procedure
                Comparison of PROCLIB.ONE with PROCLIB.TWO
                            (Method=EXACT)

                Value Comparison Results for Variables

         __________________________________________________________
                    ||       Base    Compare
               Obs  ||        gr1        gr1      Diff.     % Diff
          ________  ||  _________  _________  _________  _________
                    ||
                 1  ||       85.0      84.00    -1.0000    -1.1765
                 2  ||       92.0      92.00          0          0
                 3  ||       78.0      79.00     1.0000     1.2821
                 4  ||       87.0      87.00          0          0
          ________  ||  _________  _________  _________  _________
          ✓          ||
             N      ||          4          4          4          4
            Mean    ||    85.5000    85.5000          0     0.0264
            Std     ||     5.8023     5.4467     0.8165     1.0042
            Max     ||    92.0000    92.0000     1.0000     1.2821
            Min     ||    78.0000    79.0000    -1.0000    -1.1765
           StdErr   ||     2.9011     2.7234     0.4082     0.5021
             t      ||    29.4711    31.3951     0.0000     0.0526
          Prob>|t|  ||     <.0001     <.0001     1.0000     0.9614
                    ||
           Ndif     ||          2    50.000%
          DifMeans  ||     0.000%     0.000%          0
           r, rsq   ||      0.991      0.983
         __________________________________________________________
```

```
                    Comparing Two Data Sets: Full Report                     6

                              COMPARE Procedure
                 Comparison of PROCLIB.ONE with PROCLIB.TWO
                                (Method=EXACT)

                    Value Comparison Results for Variables

            __________________________________________________________
                       ||       Base     Compare
                   Obs ||        gr2         gr2       Diff.      % Diff
             ________  ||  _________   _________   _________   _________
                       ||
                     1 ||    87.0000     87.0000           0           0
                     2 ||    92.0000     92.0000           0           0
                     3 ||    72.0000     73.0000      1.0000      1.3889
                     4 ||    94.0000     74.0000    -20.0000    -21.2766
             ________  ||  _________   _________   _________   _________
            ✓           ||
                N      ||          4           4           4           4
               Mean    ||    86.2500     81.5000     -4.7500     -4.9719
               Std     ||     9.9457      9.4692     10.1776     10.8895
               Max     ||    94.0000     92.0000      1.0000      1.3889
               Min     ||    72.0000     73.0000    -20.0000    -21.2766
              StdErr   ||     4.9728      4.7346      5.0888      5.4447
                t      ||    17.3442     17.2136     -0.9334     -0.9132
             Prob>|t|  ||     0.0004      0.0004      0.4195      0.4285
                       ||
              Ndif     ||          2     50.000%
             DifMeans  ||    -5.507%     -5.828%     -4.7500
              r, rsq   ||      0.451       0.204
            __________________________________________________________
```

Example 2: Comparing Variables in Different Data Sets

Procedure features:

PROC COMPARE statement option

NOSUMMARY

VAR statement

WITH statement

Data sets:

PROCLIB.ONE, PROCLIB.TWO on page 222.

This example compares a variable from the base data set with a variable in the comparison data set. All summary reports are suppressed.

Program

```
libname proclib 'SAS-data-library';
```

```
options nodate pageno=1 linesize=80 pagesize=40;
```

BASE= specifies the base data set and COMPARE= specifies the comparison data set. NOSUMMARY suppresses all summary reports.

```
proc compare base=proclib.one compare=proclib.two nosummary;
```

The VAR and WITH statements specify the variables to compare. This example compares GR1 from the base data set with GR2 from the comparison data set.

```
   var gr1;
   with gr2;
   title 'Comparison of Variables in Different Data Sets';
run;
```

Output

```
                  Comparison of Variables in Different Data Sets              1

                                 COMPARE Procedure
                    Comparison of PROCLIB.ONE with PROCLIB.TWO
                                  (Method=EXACT)

NOTE: Data set PROCLIB.TWO contains 1 observations not in PROCLIB.ONE.
NOTE: Values of the following 1 variables compare unequal: gr1^=gr2


                      Value Comparison Results for Variables

             __________________________________________________________
                        ||       Base    Compare
                    Obs ||        gr1        gr2       Diff.     % Diff
              ________  ||  _________  _________  _________  _________
                        ||
                      1 ||       85.0    87.0000     2.0000     2.3529
                      3 ||       78.0    73.0000    -5.0000    -6.4103
                      4 ||       87.0    74.0000   -13.0000   -14.9425
             __________________________________________________________
```

Example 3: Comparing a Variable Multiple Times

Procedure features:

VAR statement

WITH statement

Data sets:

PROCLIB.ONE, PROCLIB.TWO on page 222.

This example compares one variable from the base data set with two variables in the comparison data set.

Program

```
libname proclib 'SAS-data-library';


options nodate pageno=1 linesize=80 pagesize=40;
```

BASE= specifies the base data set and COMPARE= specifies the comparison data set. NOSUMMARY suppresses all summary reports.

```
proc compare base=proclib.one compare=proclib.two nosummary;
```

The VAR and WITH statements specify the variables to compare.This example compares GR1 from the base data set with GR1 and GR2 from the comparison data set.

```
   var gr1 gr1;
   with gr1 gr2;
   title 'Comparison of One Variable with Two Variables';
run;
```

Output

The Value Comparison Results section shows the result of the comparison.

```
                  Comparison of One Variable with Two Variables                1

                                COMPARE Procedure
                    Comparison of PROCLIB.ONE with PROCLIB.TWO
                                  (Method=EXACT)

NOTE: Data set PROCLIB.TWO contains 1 observations not in PROCLIB.ONE.
NOTE: Values of the following 2 variables compare unequal: gr1^=gr1 gr1^=gr2


                      Value Comparison Results for Variables

            __________________________________________________________
                       ||       Base    Compare
                   Obs ||        gr1        gr1       Diff.     % Diff
             ________  ||  _________  _________  _________  _________
                       ||
                     1 ||       85.0      84.00     -1.0000    -1.1765
                     3 ||       78.0      79.00      1.0000     1.2821
            __________________________________________________________


            __________________________________________________________
                       ||       Base    Compare
                   Obs ||        gr1        gr2       Diff.     % Diff
             ________  ||  _________  _________  _________  _________
                       ||
                     1 ||       85.0    87.0000      2.0000     2.3529
                     3 ||       78.0    73.0000     -5.0000    -6.4103
                     4 ||       87.0    74.0000    -13.0000   -14.9425
            __________________________________________________________
```

Example 4: Comparing Variables That Are in the Same Data Set

Procedure features:
- PROC COMPARE statement options
  - ALLSTATS
  - BRIEFSUMMARY
- VAR statement
- WITH statement

Data set:
PROCLIB.ONE on page 222.

This example shows that PROC COMPARE can compare two variables that are in the same data set.

Program

```
libname proclib 'SAS-data-library';
```

```
options nodate pageno=1 linesize=80 pagesize=40;
```

ALLSTATS prints summary statistics. BRIEFSUMMARY suppresses all the summary reports.

```
proc compare base=proclib.one allstats briefsummary;
```

The VAR and WITH statements specify the variables in the base data set to compare. This example compares GR1 with GR2. Because there is no comparison data set, the variables GR1 and GR2 must be in the base data set.

```
   var gr1;
   with gr2;
   title 'Comparison of Variables in the Same Data Set';
run;
```

Output

```
                 Comparison of Variables in the Same Data Set                1

                              COMPARE Procedure
                   Comparisons of variables in PROCLIB.ONE
                               (Method=EXACT)

NOTE: Values of the following 1 variables compare unequal: gr1^=gr2


                    Value Comparison Results for Variables

            __________________________________________________________
                      ||       Base    Compare
                  Obs ||        gr1        gr2      Diff.     % Diff
                      ||
             ________ ||  _________  _________  _________  _________
                      ||
                    1 ||       85.0    87.0000     2.0000     2.3529
                    3 ||       78.0    72.0000    -6.0000    -7.6923
                    4 ||       87.0    94.0000     7.0000     8.0460
             ________ ||  _________  _________  _________  _________
                      ||
                N     ||          4          4          4          4
               Mean   ||    85.5000    86.2500     0.7500     0.6767
               Std    ||     5.8023     9.9457     5.3774     6.5221
               Max    ||    92.0000    94.0000     7.0000     8.0460
               Min    ||    78.0000    72.0000    -6.0000    -7.6923
              StdErr  ||     2.9011     4.9728     2.6887     3.2611
                t     ||    29.4711    17.3442     0.2789     0.2075
             Prob>|t| ||     <.0001     0.0004     0.7984     0.8489
                      ||
              Ndif    ||          3    75.000%
             DifMeans ||     0.877%     0.870%     0.7500
              r, rsq  ||      0.898      0.807
            __________________________________________________________
```

Example 5: Comparing Observations with an ID Variable

Procedure features:
ID statement

In this example, PROC COMPARE compares only the observations that have matching values for the ID variable.

Program

```
libname proclib 'SAS-data-library';
```

```
options nodate pageno=1 linesize=80 pagesize=40;
```

PROCLIB.EMP95 and PROCLIB.EMP96 contain employee data. IDNUM works well as an ID variable because it has unique values. A DATA step on page 1507 creates PROCLIB.EMP95. A DATA step on page 1508 creates PROCLIB.EMP96.

```
data proclib.emp95;
   input #1 idnum $4. @6 name $15.
         #2 address $42.
         #3 salary 6.;
   datalines;
2388 James Schmidt
100 Apt. C Blount St. SW Raleigh NC 27693
92100
2457 Fred Williams
99 West Lane  Garner NC 27509
33190
... more data lines...
3888 Kim Siu
5662 Magnolia Blvd Southeast Cary NC 27513
77558
;

data proclib.emp96;
   input #1 idnum $4. @6 name $15.
         #2 address $42.
         #3 salary 6.;
   datalines;
2388 James Schmidt
100 Apt. C Blount St. SW Raleigh NC 27693
92100
2457 Fred Williams
99 West Lane  Garner NC 27509
33190
...more data lines...
6544 Roger Monday
3004 Crepe Myrtle Court Raleigh NC 27604
47007
;
```

Both data sets must be sorted by the variable that will be used as the ID variable in the PROC COMPARE step. OUT= specifies the location of the sorted data.

```
proc sort data=proclib.emp95 out=emp95_byidnum;

 by idnum;
run;

proc sort data=proclib.emp96 out=emp96_byidnum;
   by idnum;
run;
```

The ID statement makes IDNUM the ID variable.

```
proc compare base=emp95_byidnum compare=emp96_byidnum;
   id idnum;
   title 'Comparing Observations that Have Matching IDNUMs';
run;
```

Output

PROC COMPARE identifies specific observations by the value of IDNUM. In the **`Value Comparison Results for Variables`** section, PROC COMPARE prints the nonmatching addresses and nonmatching salaries. For salaries, PROC COMPARE computes the numerical difference and the percent difference. Because ADDRESS is a character variable, PROC COMPARE displays only the first 20 characters. For addresses where the observation has an IDNUM of **`0987`**, **`2776`**, or **`3888`**, the differences occur after the 20th character and the differences do not appear in the output. The plus sign in the output indicates that the full value is not shown. To see the entire value, create an output data set. See Example 6 on page 262.

```
                 Comparing Observations that Have Matching IDNUMs               1

                                COMPARE Procedure
             Comparison of WORK.EMP95_BYIDNUM with WORK.EMP96_BYIDNUM
                                  (Method=EXACT)

                                Data Set Summary

      Dataset                      Created          Modified  NVar    NObs

      WORK.EMP95_BYIDNUM  13MAY98:16:03:36  13MAY98:16:03:36     4      10
      WORK.EMP96_BYIDNUM  13MAY98:16:03:36  13MAY98:16:03:36     4      12


                                Variables Summary

                     Number of Variables in Common: 4.
                     Number of ID Variables: 1.


                               Observation Summary

                    Observation      Base  Compare  ID

                    First Obs           1        1  idnum=0987
                    First Unequal       1        1  idnum=0987
                    Last  Unequal      10       12  idnum=9857
                    Last  Obs          10       12  idnum=9857

 Number of Observations in Common: 10.
 Number of Observations in WORK.EMP96_BYIDNUM but not in WORK.EMP95_BYIDNUM: 2.
 Total Number of Observations Read from WORK.EMP95_BYIDNUM: 10.
 Total Number of Observations Read from WORK.EMP96_BYIDNUM: 12.

 Number of Observations with Some Compared Variables Unequal: 5.
 Number of Observations with All Compared Variables Equal: 5.
```

```
              Comparing Observations that Have Matching IDNUMs            2

                             COMPARE Procedure
          Comparison of WORK.EMP95_BYIDNUM with WORK.EMP96_BYIDNUM
                              (Method=EXACT)

                         Values Comparison Summary

      Number of Variables Compared with All Observations Equal: 1.
      Number of Variables Compared with Some Observations Unequal: 2.
      Total Number of Values which Compare Unequal: 8.
      Maximum Difference: 2400.


                        Variables with Unequal Values

                     Variable  Type  Len  Ndif   MaxDif

                     address   CHAR   42     4
                     salary    NUM     8     4     2400



                    Value Comparison Results for Variables

          __________________________________________________________
                  ||  Base Value           Compare Value
           idnum  ||  address               address
           _____  ||  ___________________+  ___________________+
                  ||
           0987   ||  2344 Persimmons Bran  2344 Persimmons Bran
           2776   ||  12988 Wellington Far  12988 Wellington Far
           3888   ||  5662 Magnolia Blvd S  5662 Magnolia Blvd S
           9857   ||  1000 Taft Ave. Morri  100 Taft Ave. Morris
          __________________________________________________________
```

```
              Comparing Observations that Have Matching IDNUMs            3

                             COMPARE Procedure
          Comparison of WORK.EMP95_BYIDNUM with WORK.EMP96_BYIDNUM
                              (Method=EXACT)

                    Value Comparison Results for Variables

          __________________________________________________________
                  ||       Base    Compare
           idnum  ||     salary     salary      Diff.     % Diff
           _____  ||  _________  _________  _________  _________
                  ||
           0987   ||      44010      45110       1100     2.4994
           3286   ||      87734      89834       2100     2.3936
           3888   ||      77558      79958       2400     3.0945
           9857   ||      38756      40456       1700     4.3864
          __________________________________________________________
```

Example 6: Comparing Values of Observations Using an Output Data Set (OUT=)

Procedure features:

PROC COMPARE statement options:

NOPRINT

OUT=
OUTBASE
OUTBASE
OUTCOMP
OUTDIF
OUTNOEQUAL

Other features: PRINT procedure

Data sets: PROCLIB.EMP95 and PROCLIB.EMP96 on page 260

This example creates and prints an output data set that shows the differences between matching observations.

In Example 5 on page 259, the output does not show the differences past the 20th character. The output data set in this example shows the full values. Further, it shows the observations that occur in only one of the data sets.

Program

```
libname proclib 'SAS-data-library';
```

```
options nodate pageno=1 linesize=120 pagesize=40;
```

Both data sets must be sorted by the variable that will be used as the ID variable in the PROC COMPARE step. OUT= specifies the location of the sorted data.

```
proc sort data=proclib.emp95 out=emp95_byidnum;

 by idnum;
run;

proc sort data=proclib.emp96 out=emp96_byidnum;
   by idnum;
run;
```

BASE= and COMPARE= specify the data sets to compare.

```
proc compare base=emp95_byidnum compare=emp96_byidnum
```

OUT= names and creates the output data set. NOPRINT suppresses the printing of the procedure output. OUTNOEQUAL includes only observations that are judged unequal. OUTBASE writes an observation to the output data set for each observation in the base data set. OUTCOMP writes an observation to the output data set for each observation in the comparison data set. OUTDIF writes an observation to the output data set that contains the differences between the two observations.

```
             out=result outnoequal outbase outcomp outdif
     noprint;
```

The ID statement specifies IDNUM as the ID variable.

```
   id idnum;
run;
```

PROC PRINT prints the output data set. Using the BY and ID statements with the same variable makes the output easy to read. See Chapter 28, “The PRINT Procedure,” on page 783 for more information on this technique.

```
proc print data=result noobs;
   by idnum;
   id idnum;
   title ’The Output Data Set RESULT’;
run;
```

Output

The differences for character variables are noted with an X or a period (.). An X shows that the characters do not match. A period shows that the characters do match. For numeric variables, an E means that there is no difference. Otherwise, the numeric difference is shown. By default, the output data set shows that two observations in the comparison data set have no matching observation in the base data set. You do not have to use an option to make those observations appear in the output data set.

```
                                     The Output Data Set RESULT                                        1

idnum    _TYPE_     _OBS_    name               address                                     salary

0987     BASE          1     Dolly Lunford      2344 Persimmons Branch  Apex NC 27505        44010
         COMPARE       1     Dolly Lunford      2344 Persimmons Branch Trail Apex NC 27505   45110
         DIF           1     ...............    .......................XXXXX.XXXXXXXXXXXXX    1100

2776     BASE          5     Robert Jones       12988 Wellington Farms Ave. Cary NC 27512    29025
         COMPARE       5     Robert Jones       12988 Wellington Farms Ave. Cary NC 27511    29025
         DIF           5     ...............    .......................................X.       E

3278     COMPARE       6     Mary Cravens       211 N. Cypress St. Cary NC 27512             35362

3286     BASE          6     Hoa Nguyen         2818 Long St. Cary NC 27513                  87734
         COMPARE       7     Hoa Nguyen         2818 Long St. Cary NC 27513                  89834
         DIF           6     ...............    ..........................................    2100

3888     BASE          7     Kim Siu            5662 Magnolia Blvd Southeast Cary NC 27513   77558
         COMPARE       8     Kim Siu            5662 Magnolia Blvd Southwest Cary NC 27513   79958
         DIF           7     ...............    ........................XX................    2400

6544     COMPARE       9     Roger Monday       3004 Crepe Myrtle Court Raleigh NC 27604     47007

9857     BASE         10     Kathy Krupski      1000 Taft Ave. Morrisville NC 27508          38756
         COMPARE      12     Kathy Krupski      100 Taft Ave. Morrisville NC 27508           40456
         DIF          10     ...............    ...XXXXXXXXXXXXXX.XXXXX.XXXXXXXXXXX.......    1700
```

Example 7: Creating an Output Data Set of Statistics (OUTSTATS=)

Procedure features:
PROC COMPARE statement options:
NOPRINT
OUTSTATS=

Data sets: PROCLIB.EMP95, PROCLIB.EMP96 on page 260

This example creates an output data set that contains summary statistics for the numeric variables that are compared.

Program

```
libname proclib 'SAS-data-library';
```

```
options nodate pageno=1 linesize=80 pagesize=40;
```

Both data sets must be sorted by the variable that will be used as the ID variable in the PROC COMPARE step. OUT= specifies the location of the sorted data.

```
proc sort data=proclib.emp95 out=emp95_byidnum;
   by idnum;
run;

proc sort data=proclib.emp96 out=emp96_byidnum;
   by idnum;
run;
```

BASE= and COMPARE= specify the data sets to compare. OUTSTATS= creates the output data set DIFFSTAT. NOPRINT suppresses the procedure output. The ID statement specifies IDNUM as the ID variable. PROC COMPARE uses the values of IDNUM to match observations.

```
proc compare base=emp95_byidnum compare=emp96_byidnum
             outstats=diffstat noprint;
   id idnum;
run;
```

PROC PRINT prints the output data set DIFFSTAT.

```
proc print data=diffstat noobs;
   title 'The DIFFSTAT Data Set';
run;
```

Output

The variables are described in “Output Statistics Data Set (OUTSTATS=)” on page 249.

```
                        The DIFFSTAT Data Set                              1

  _VAR_     _TYPE_        _BASE_       _COMP_       _DIF_     _PCTDIF_

  salary    N              10.00        10.00       10.00      10.0000
  salary    MEAN        52359.00     53089.00      730.00       1.2374
  salary    STD         24143.84     24631.01      996.72       1.6826
  salary    MAX         92100.00     92100.00     2400.00       4.3864
  salary    MIN         29025.00     29025.00        0.00       0.0000
  salary    STDERR       7634.95      7789.01      315.19       0.5321
  salary    T               6.86         6.82        2.32       2.3255
  salary    PROBT           0.00         0.00        0.05       0.0451
  salary    NDIF            4.00        40.00         .           .
  salary    DIFMEANS        1.39         1.38      730.00         .
  salary    R,RSQ           1.00         1.00         .           .
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Procedures Guide, Version 8,* Cary, NC: SAS Institute Inc., 1999. 1729 pp.

SAS® Procedures Guide, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1–58025–482–9

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999