

PROC COMPARE

This procedure is used to give the full description to observation wise.
By default it gives the details of unrelated values.

Syntax:-

```
PROC COMPARE BASE=DATASET COMPARE=DATASET <OPTIONS>;  
BY VARIABLE<S>;  
VAR VARIABLES<S>;  
WITH VARIABLES<S>;  
RUN;
```

Examples:-

```
DATA ONE (LABEL='FIRST DATA SET');  
INPUT STUDENT YEAR STATE $ GRADE1 GRADE2;  
LABEL YEAR='YEAR OF BIRTH';  
FORMAT GRADE1 4.1;  
CARDS;  
1000 1970 NC 85 87  
1042 1971 MD 92 92  
1095 1969 PA 78 72  
1187 1970 MA 87 94  
;  
RUN;  
  
DATA TWO (LABEL='SECOND DATA SET');  
INPUT STUDENT $ YEAR STATE $ GRADE1 GRADE2 MAJOR $;  
LABEL STATE='HOME STATE';  
FORMAT GRADE1 5.2;  
CARDS;  
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  
;  
RUN;
```

Options:-

BASE - Specify the base data set

COMPARE - Specify the comparison data set

```
PROC COMPARE BASE=ONE COMPARE=TWO;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

OUT=DATASET - Create an output data set

```
PROC COMPARE BASE=ONE COMPARE=TWO OUT=THREE;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

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_.

OUTALL - Write an observation for each value in the BASE= and COMPARE= data sets

```
PROC COMPARE BASE=ONE COMPARE=TWO OUT=THREE OUTALL;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

OUTBASE - Write an observation for each observation in the BASE= data set

```
PROC COMPARE BASE=ONE COMPARE=TWO OUT=THREE OUTBASE;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

OUTCOMP - Write an observation for each observation in the COMPARE= data set

```
PROC COMPARE BASE=ONE COMPARE=TWO OUT=THREE OUTCOMP;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

OUTDIFF - Write an observation that contains the differences for each pair of matching observations

```
PROC COMPARE BASE=ONE COMPARE=TWO OUT=THREE OUTDIFF;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

OUTNOEQUAL - Suppress the writing of observations when all values are equal

```
PROC COMPARE BASE=ONE COMPARE=TWO OUT=THREE OUTNOEQUAL;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

OUTPERCENT - Write an observation that contains the percent differences for each pair of matching observations

```
PROC COMPARE BASE=ONE COMPARE=TWO OUT=THREE OUTPERCENT;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

OUTSTATS=DATASET - Create an output data set that contains summary statistics.

```
PROC COMPARE BASE=ONE COMPARE=TWO OUTSTATS=FOUR;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

When we use outstats=dataset proc compare creating following variables

VAR

Is a character variable that contains the name of the variable from the base data set for which the statistic in the observation was calculated.

WITH

Is a character variable that contains 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 that contains 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 that contains 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 dataset.

COMP

Is a numeric variable that contains 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 that contains 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 that contains 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 dataset.

PRINT ALL – Gives differences, percentage differences and Statistics such as min, max, mean, STD, n etc.

```
PROC COMPARE BASE=ONE COMPARE=TWO PRINTALL;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

ALLOBS – Include the values for all matching observations

```
PROC COMPARE BASE=ONE COMPARE=TWO ALLOBS;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

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

```
PROC COMPARE BASE=ONE COMPARE=TWO STATS;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

ALLSTATS - prints a table of summary statistics for all pairs of matching variables those are judged equal and unequal both.

```
PROC COMPARE BASE=ONE COMPARE=TWO ALLSTATS;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

ALLVARS - Include in the report the values and differences for all matching variables

```
PROC COMPARE BASE=ONE COMPARE=TWO ALLVARS;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

BRIEFSUMMARY – Gives summary briefly in report

```
PROC COMPARE BASE=ONE COMPARE=TWO BRIEFSUMMARY;  
TITLE 'COMPARING TWO DATA SETS: BRIEF REPORT';  
RUN;
```

NODATE – Suppress the print of creation and last-modified dates

```
PROC COMPARE BASE=ONE COMPARE=TWO NODATE;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

NOPRINT – Suppress the print of output.

```
PROC COMPARE BASE=ONE COMPARE=TWO NODATE;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

NOSUMMARY – Suppress the summary reports

```
PROC COMPARE BASE=ONE COMPARE=TWO NOSUMMARY;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

NOVALUES – Suppress the value comparison results.

```
PROC COMPARE BASE=ONE COMPARE=TWO NOVALUES;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

TRANPOSE – Print the value differences by observation, not by variable.

```
PROC COMPARE BASE=ONE COMPARE=TWO TRANPOSE;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

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

```
PROC COMPARE BASE=ONE COMPARE=TWO ERROR;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

WARNING - Displays warning message in the SAS log when differences are found.

```
PROC COMPARE BASE=ONE COMPARE=TWO WARNING;  
TITLE 'COMPARING TWO DATA SETS';  
RUN;
```

BY STATEMENT – Compare datasets on each by group wise.

Default compares one to one observations.

```
PROC SORT DATA=ONE;  
BY STATE;  
RUN;  
PROC SORT DATA=TWO;  
BY STATE;  
RUN;  
PROC COMPARE BASE=ONE COMPARE=TWO;  
BY STATE;  
TITLE 'COMPARING TWO DATA SETS: BASED ON BY VARIABLE';  
RUN;  
PROC SORT DATA=ONE;  
BY STATE;  
RUN;  
PROC COMPARE BASE=ONE;  
BY STATE;  
VAR GRADE1;  
WITH GRADE2;  
TITLE 'COMPARING TWO VARIABLES WITH IN DATA SETS';  
RUN;
```

VAR STATEMENT

Comparing selected variables from both datasets

Select variables only which involves in comparing. Otherwise all variables involve.

```
PROC COMPARE BASE=ONE COMPARE=TWO NOSUMMARY;  
VAR GRADE1;  
TITLE 'COMPARING VARIABLES FOR SELECTED VARIABLES';  
RUN;
```

WITH STATEMENT

Comparing between different variables from both datasets

Select variable which compares with Var statement variable.

```
PROC COMPARE BASE=ONE COMPARE=TWO NOSUMMARY ALLSTATS NOVALUES;
VAR GRADE1;
WITH GRADE2;
TITLE 'COMPARING VARIABLES WITH DIFFERENT VARIABLES';
RUN;
```

Comparing between different variables from base dataset

```
PROC COMPARE BASE=ONE NOSUMMARY ALLSTATS NOVALUES;
VAR GRADE1;
WITH GRADE2;
TITLE 'COMPARING DIFFERENT VARIABLES WITHIN A DATA SET';
RUN;
```

Creating a dataset with statistical values











```
PROC COMPARE BASE=ONE COMPARE=TWO OUTSTATS=DATASET NOPRINT;
RUN;
```

When we can use Proc compare

It is invaluable for the many situations in which one needs to compare two data sets. For example,

- To identify differences between variables and observations count.
- To identify Conflicting of data types and differing attributes.
- whether matching variables have different values
- what variables the two data sets have in common
- To evaluate newly collected data in comparison to an existing file.
- To test whether data set updates or edits have occurred as expected.
- To examine whether two algorithms for computing certain variables produce comparable results.
- To prepare for a merge/joint of two large data sets with many variables, so that one knows what variables may need to be renamed.
- To joining or merging two data sets, followed by identifying mis-matches and analyzing similarities and differences among variables on matching observations.

An important document which helps to understand the Proc compare

 PROC COMPARE – Worth Another Look.	 Proc Compare as validation tool.pdf	 Demystifying PROC COMPARE A Program	 Don't Get Blindsided by PROC COMPARE.†	 Using PROC COMPARE to identify
 PROC COMPARE – Worth Another Look.	 Validating clinical trail data reporting with S	 20 character limit for proc compare.htm	 Don't Get Blindsided by PROC COMPARE.†	 Q.pdf

