Proc Compare to Validate Datasets
Angelina Cecilia Casas M.S., PPD Development, Research Triangle Park, NC

ABSTRACT
Comparison of two datasets is a technique used to know that two datasets are equal or if they have discrepancies. This method can be used to validate that a dataset has been created correctly or that the changes in a dataset are only those that are expected. I.e. observations were added or removed or corrections were done.

INTRODUCTION
Proc Compare is a procedure that allows two datasets to be compared for properties, number of observations, number of variables, and properties of the datasets.
For a variable, you can get output about differences in:
  - Type, length, formats, informats and label(s).
For a dataset, you can find differences in:
  - date of creation, last modification of the datasets was modified,
  - number of variables and observations of the datasets. You can also see the labels of the datasets, but differences are not reported for that.
For observations, you can compare the value of the record for each variable. Also, you can decide how different the values of the observations can be.

THE DATASETS THAT WILL BE COMPARED.

```
proc print data=demog noobs;

Unique   WEIGHTKG   dob
20120    101.6      08/19/1949
20130    94.3       06/02/1934
20110    85.7       09/17/1949
20202    99.3       10/17/1931
```

```
proc contents data=demog;

Data Set Name: WORK.DEMOG Observations: 4
Member Type: DATA Variables: 3

--Alphabetic List of Variables and Attributes--

Variable Type Len Pos Format Label
-----------------------------------------------
UNIQUE Char 200 32 Unique Record
WEIGHTKG Num 8 8 5.1 Weight in kg
dob Num 8 24 MMDDYY10. Date of Birth
```

```
proc sql;
  create table Compare as select * from demog;
quit;
```

AN EXAMPLE OF A CLEAN COMPARE OUTPUT

```
proc compare base=here.demog compare=compare;
run;
```

```
Variables Summary
Number of Variables in Common: 3.

Observation Summary
Observation  Base  Compare
First Obs   1      1
Last Obs    4      4

Number of Observations in Common: 4.
Total Number of Observations Read from WORK.DEMOG: 4.
Total Number of Observations Read from WORK.COMpare: 4.

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

NOTE: No unequal values were found. All values compared are exactly equal.
```

The records in DEMOG and the records in compare have the same order and an ID statement is not needed. However if the order of the two datasets is not the same, records might be compared incorrectly.

WHEN THE DATASETS HAVE A DIFFERENT ORDER;

```
proc sort data=compare;
  by DOB;
run;
```

```
proc compare base=demog (keep=unique weightkg) compare=compare (keep=unique weightkg);
run;
```

As in other SAS procedures and data step, it is possible to select observations and variables in the datasets that are going to be used.

```
Dataset       Created       Modified   NVar   Nobs
WORK.DEMOG    20JAN03:13:17 20JAN03:13:17 2      4
WORK.COMpare   20JAN03:13:17 20JAN03:13:17 2      4

Vars Summary
# of Vars in Common: 2.

Observation Summary
Observation  Base  Compare

# of Obs in Common: 4.
Total # of Obs Read from WORK.DEMOG: 4.
Total # of Obs Read from WORK.COMpare: 4.

# of Obs with Some Compared Vars Unequal: 3.
# of Obs with All Compared Vars Equal: 1.

Values Comparison Summary
# of Vars Compared with All Obs Equal: 0.
# of Vars Compared with Some Obs Unequal: 2.
Total # of Values which Compare Unequal: 6.
Maximum Difference: 15.9.
```

All Vars Compared have Unequal Values

```
Variable Type Len Label   Ndif   MaxDif
Unique  CHAR 200 Unique Record 3
WEIGHTKG NUM 8 Weight in kg 3 15.900
```

```
```

The COMPARE Procedure
Comparison of WORK.DEMOG with WORK.COMpare
(Method=EXACT)
```

```
Data Set Summary
Dataset      Created       Modified   NVar   Nobs
WORK.DEMOG   14JAN03:16:03 14JAN03:16:06 3      4
WORK.COMpare 14JAN03:16:06 14JAN03:16:06 3      4
```

```
The values of the variable that was to the right of the `BASE` or `DATA` statement is in the `BASE` column. The values of dataset listed to right of the `COMPARE` dataset is in the `COMPARE` column.

It is necessary that the two datasets that are going to be compared have the same order UNLESS the order of the datasets is something that is being tested.

What happens when the order of the observations in each of the datasets is different?

```sql
proc sql;
  update compare set unique='34343' where unique='20202';
quit;
proc sort data=compare;
  by unique;
run;
```

```sql
proc compare base=demog(keep = unique weightkg)
  compare=compare(keep = unique weightkg);
run;
```

Even though, there is only one discrepancy (UNIQUE) in the datasets, several discrepancies are reported. It is necessary to specify which observations should be compared.

The option that tells proc compare, which variables should be together, is `ID`. After it, you should list the number of variables that define which observation in each of the datasets should be compared. This is similar to the way datasets are merged using a `by` statement.

```sql
proc compare base=demog compare=compare;
  id unique;
run;
```

**LIMITING THE PRINTED OUTPUT**

Sometimes it is not necessary to know all the observations that have an error, once it is known that an error exists, it might not be needed to know all the discrepancies.

It might be desirable to limit the size of the output; for that, the option `MAXPRINT` is very useful.

First it is necessary to use the original dataset and modify the variable `WEIGHTKG` to have something to report.

For example, let’s change the value of one of the variables in the `ID` statement from ’34343’ to ’20120’.

```sql
proc sql;
  update compare set unique='20120' where unique='34343';
  update compare set weightkg=int(weightkg);
quit;
```

```sql
proc sort data=compare;
  by unique;
run;
```

```sql
proc compare data=demog
  compare=compare
  maxprint=(4,2);
  id unique;
run;
```

At the most, 4 differences will be printed, 2 for every variable that
has discrepancies.

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Len</th>
<th>Label</th>
<th>Ndif</th>
<th>MaxDif</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEIGHTKG</td>
<td>NUM</td>
<td>Weight in kg</td>
<td>4</td>
<td>0.700</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unique</th>
<th>Base</th>
<th>Compare</th>
<th>Diff.</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>20110</td>
<td>85.7</td>
<td>85.0</td>
<td>-0.700</td>
<td>-0.8168</td>
</tr>
<tr>
<td>20120</td>
<td>101.6</td>
<td>101.0</td>
<td>-0.600</td>
<td>-0.5906</td>
</tr>
</tbody>
</table>

NOTE: The MAXPRINT=2 printing limit has been reached for the variable WEIGHTKG.
No more values will be printed for this comparison.

PRINTING RELATED INFORMATION TOGETHER.

Sometimes it is desired to see the information with discrepancies grouped together by the variables in the ID statement.

```
proc compare data=demog
  compare=compare
  transpose
  id unique;
run;
```

```
proc compare data=demog
  compare=compare
  criterion=.01
  var age;
  with nage;
run;
```

COMPARING VARIABLES WITH DIFFERENT NAMES.

Sometimes, it is necessary to compare datasets when it is known that the variable names are different yet they should have the same values.

```
proc sql;
  alter table compare
    add nage num label "Numeric age";
  alter table demog
    add age num label "New Numeric age";
  update demog set
    age=int((date()-dob)/365.25);
  update compare set
    nage=int((date()-dob)/365.25);
quit;
```

```
proc compare data=demog
  compare=compare
  outnoequal
  out=toprint;
run;
```

Sometimes, there are differences in the datasets that are not important for the purpose of the comparison. For this scenario, it is possible to give a value for which only observations that are bigger than this number will be marked as a difference.

```
proc sql;
  update compare set nage=nage+0.001;
quit;
```

```
proc compare data=demog
  compare=compare
  criterion=.01
  var age;
  with nage;
run;
```

```
proc sql;
  update compare set nage=nage+0.001;
  quit;
```

Sometimes, it is desirable to create an output dataset that contains only the equalities or discrepancies. Here, there is an example to create a dataset that has the differences.

```
proc compare data=demog
  compare=compare
  outnoequal
  out=toprint;
run;
```

The COMPARE Procedure
Comparison of WORK.DEMOG with WORK.COMPARE
(Method=RELATIVE(2.21E-12), Criterion=0.01)

Variables Summary
Number of Variables in Common: 3.
Number of Variables in WORK.DEMOG but not in WORK.COMPARE: 1.
Number of Variables in WORK.COMPARE but not in WORK.DEMOG: 1.
Number of VAR Statement Variables: 1.
Number of WITH Statement Variables: 1.

Number of Observations in Common: 4.
Total Number of Observations Read from WORK.DEMOG: 4.
Total Number of Observations Read from WORK.COMPARE: 4.
Number of Observations with Some Compared Variables Unequal: 0.
Number of Observations with All Compared Variables Equal: 4.

Values Comparison Summary
Number of Variables Compared with All Observations Equal: 1.
Number of Variables Compared with Some Observations Unequal: 0.
Total Number of Values which Compare Unequal: 0.
Total Number of Values not EXACTLY Equal: 4.
Maximum Difference Criterion Value: 0.000018868.

CREATING AN OUTPUT DATASET
Sometimes, it is desirable to create an output dataset that contains only the equalities or discrepancies. Here, there is an example to create a dataset that has the differences.

The outnoequal option, specifies that only the observations with discrepancies will exist in the dataset toprint.

```
proc compare data=demog
  compare=compare
  outnoequal
  out=toprint;
run;
```
### Variables with Unequal Values

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Len</th>
<th>Label</th>
<th>Ndif</th>
<th>MaxDif</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEIGHTKG</td>
<td>NUM</td>
<td>8</td>
<td>4</td>
<td>0.700</td>
</tr>
</tbody>
</table>

### Value Comparison Results for Variables

<table>
<thead>
<tr>
<th>Unique</th>
<th>Base WEIGHTKG</th>
<th>Base Diff</th>
<th>Compare WEIGHTKG</th>
<th>Diff.</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>20110</td>
<td>85.7</td>
<td>0.7000</td>
<td>85</td>
<td>0.8168</td>
<td></td>
</tr>
<tr>
<td>20120</td>
<td>101.6</td>
<td>0.6000</td>
<td>101.0</td>
<td>0.5906</td>
<td></td>
</tr>
<tr>
<td>20130</td>
<td>94.3</td>
<td>0.3000</td>
<td>94.0</td>
<td>0.3181</td>
<td></td>
</tr>
<tr>
<td>20202</td>
<td>99.3</td>
<td>0.3000</td>
<td>99.0</td>
<td>0.3021</td>
<td></td>
</tr>
</tbody>
</table>

### OUTPUT OF PROC PRINT

<table>
<thead>
<tr>
<th>Obs</th>
<th><em>TYPE</em></th>
<th><em>OBS</em></th>
<th>Unique</th>
<th>WEIGHTKG</th>
<th>dob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DIF</td>
<td>1</td>
<td>20110</td>
<td>85.7</td>
<td>09/17/1949</td>
</tr>
<tr>
<td>2</td>
<td>DIF</td>
<td>2</td>
<td>20120</td>
<td>101.6</td>
<td>08/19/1949</td>
</tr>
<tr>
<td>3</td>
<td>DIF</td>
<td>3</td>
<td>20130</td>
<td>94.3</td>
<td>06/02/1934</td>
</tr>
<tr>
<td>4</td>
<td>DIF</td>
<td>4</td>
<td>20202</td>
<td>99.3</td>
<td>10/17/1931</td>
</tr>
</tbody>
</table>

### CONCLUSION

You are the person who decides what is important to validate, values of the observations and how different can they be. Is it important to have the same labels, formats and informats in a variable. Proc Compare is a tool that is flexible to allow you find the differences that are important to you.

### ACKNOWLEDGMENTS

I want to thank, Craig Mauldin, Andy Barnett, George Clark, Bonnie Duncan and Kim Sturgen for their comments and support.

### CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

A. Cecilia Casas  
PPD Development  
3900 Paramount Parkway  
Morrisville, NC 27560  
Phone: (919) 462-4199  
Fax: (919) 379-6151  
Email: Cecilia.Casas@rtp.ppdi.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.