SASCRUNCH.COM

HOME

SAS® CERTIFICATION TRAINING

ONLINE COURSES

ARTICLES

SERVICES

MORE...

SASCRUNCH.COM

Proc Compare in SAS

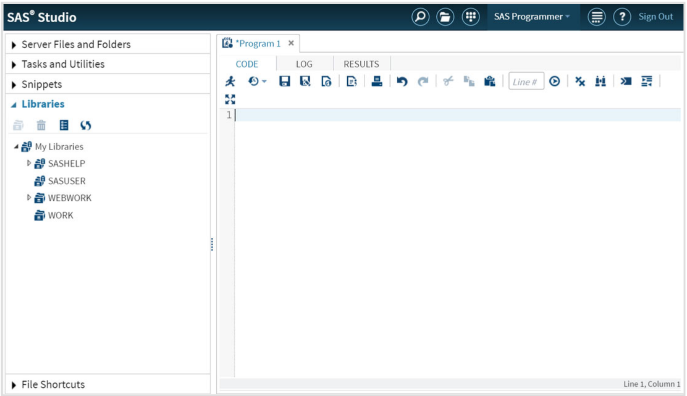
Are you looking for an efficient way to compare two datasets without having to merge them together? Would you like a simple way to summarize differences between two variables? Are you looking for a method to compare the metadata and structure of a dataset without comparing individual values? If so, then PROC COMPARE can be a very useful tool for you.

In this article, we will discuss the many different ways you can compare datasets and variables using PROC COMPARE. A variety of examples will be presented to highlight the different options available with PROC COMPARE that allow you to compare, contrast and report on the differences between datasets and the

6. Comparing specific values in an output dataset

Software

Before we continue, make sure you have SAS Studio or SAS 9.4 installed. Don't have the software? Download **SAS Studio** now. It's free!



datasets.

For more in-depth comparison examples, you will need to run the following code to create two new datasets, CUSTOMERS1 and CUSTOMERS2. These datasets contain a list of customers, their unique customer IDs, first names, last names and middle initials.

```
data customers1;
 infile datalines delimiter = ',';
 input customerid $ firstname $ lastname
$ middleinitial $;
 datalines;
 001, Bob, Smith, A
 002, Alex, Jones, B
 003, Tom, Cook, C
 004, Wendy, Jefferies, D
 005, Gail, Phillips, E
 006, Samantha, Williams, L
 007, Bill, Taylor, K
 datalines;
 001, Bobby, Smith
 002, Alex, jones
 004, Wendy, Jeffery
 005, Gail, Phillips
 006, Samantha, Williams
 007, George, Taylor
 012, Diane, Miller
 009, Pat, White
 010, Donna, Summer
 ;
run;
```

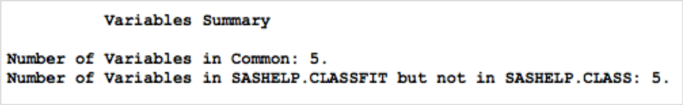
[Don't have the software yet? Download **SAS Studio** here for free.]

COMPARE is to simply specify one dataset as your “base” dataset using the BASE argument and then specify the dataset you would like to compare it against using the COMPARE argument:

```
proc compare
 base = sashelp.class
 compare = sashelp.classfit;
run;
```

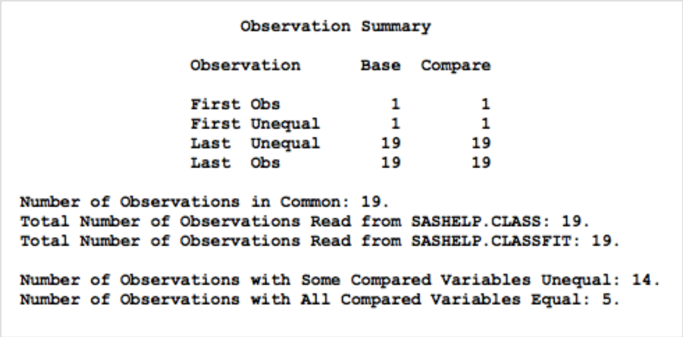
After running the above code, you will notice the Results are divided into 5 summaries. Let’s walk through each section in more detail

First, you will see the dataset summary which compares the structure or “metadata” of a dataset including the names, created dates, dataset but not the other. In this case, there are 5 variables in SASHELP.CLASSFIT which are not in SASHELP.CLASS and no variables that are in CLASS but not CLASSFIT:



```
         Variables Summary

Number of Variables in Common: 5.
Number of Variables in SASHELP.CLASSFIT but not in SASHELP.CLASS: 5.
```

The third section is the Observation Summary, which summarizes how many observations are in each dataset and how many have equal or unequal values in some of the variables:

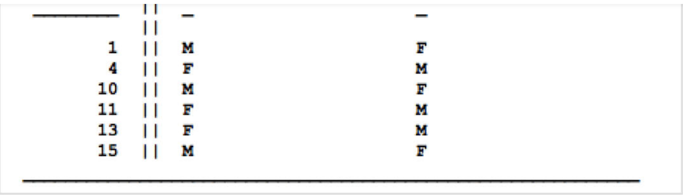


```
                        Observation Summary

                   Observation      Base  Compare

                   First Obs           1        1
                   First Unequal       1        1
                   Last  Unequal      19       19
                   Last  Obs          19       19

 Number of Observations in Common: 19.
 Total Number of Observations Read from SASHELP.CLASS: 19.
 Total Number of Observations Read from SASHELP.CLASSFIT: 19.

 Number of Observations with Some Compared Variables Unequal: 14.
 Number of Observations with All Compared Variables Equal: 5.
```

The fourth section is the Values Comparison summary, which lets you know how many

```
Weight    NUM    8    14    62.000
```

Finally, the details of the value comparisons are shown, which displays those observations, one variable at a time, with unequal values between the two datasets. In the partial output below, you can see that all 19 observations have different values for NAME between the two datasets but SEX is only unequal in 6 of the 19 observations:



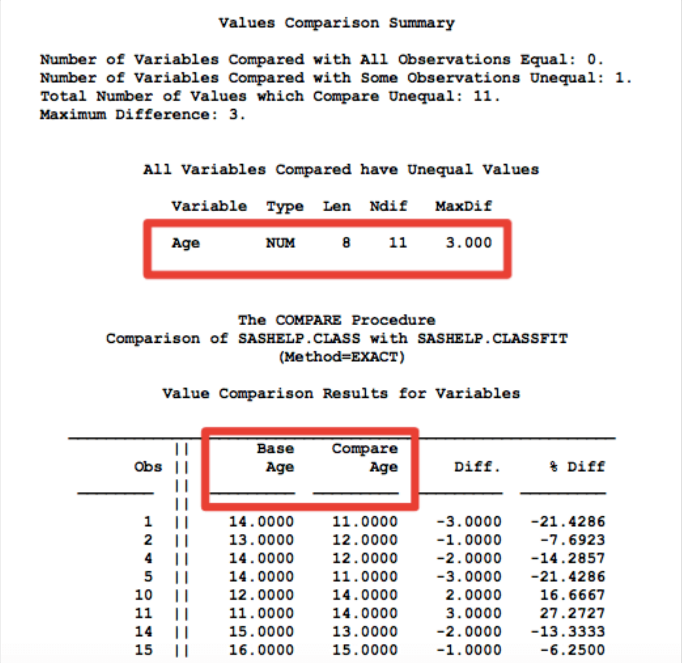
```
  ________  ||  _                  _
            ||
        1   ||  M                  F
        4   ||  F                  M
       10   ||  M                  F
       11   ||  F                  M
       13   ||  F                  M
       15   ||  M                  F
 ______________________________________________________
```

Limiting the Comparison to Certain variables

With large datasets, the Value Comparison Results (the final summary) can become quite lengthy when you are comparing many variables. Fortunately, you can limit the comparison to a list of variables that you specify.

For example, if you would only like the details of

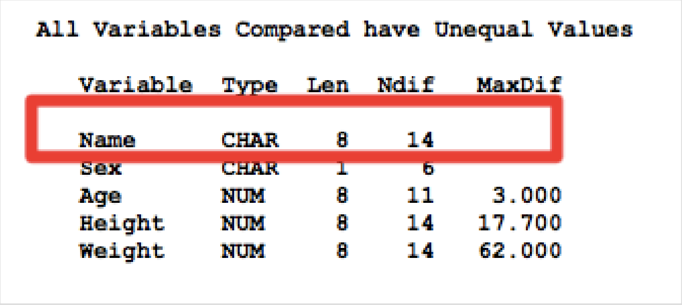
Although the initial summaries about the metadata are the same as before, you will notice the details in the Value Comparison Summary are now limited to only the age variable:



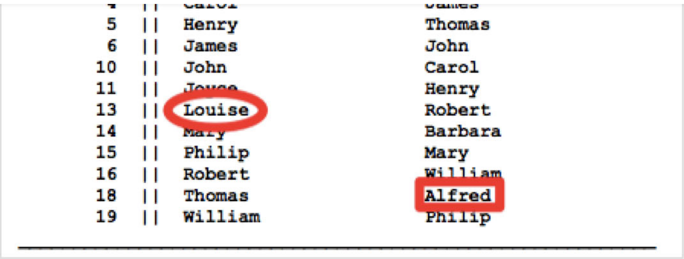
```
                    Values Comparison Summary

Number of Variables Compared with All Observations Equal: 0.
Number of Variables Compared with Some Observations Unequal: 1.
Total Number of Values which Compare Unequal: 11.
Maximum Difference: 3.


             All Variables Compared have Unequal Values

                Variable  Type  Len  Ndif   MaxDif

                Age        NUM    8    11    3.000


                      The COMPARE Procedure
         Comparison of SASHELP.CLASS with SASHELP.CLASSFIT
                         (Method=EXACT)

              Value Comparison Results for Variables

     __________________________________________________________
                  ||       Base    Compare
           Obs    ||        Age        Age      Diff.     % Diff
      ________    ||  _________  _________  _________  _________
                  ||
             1    ||    14.0000    11.0000    -3.0000   -21.4286
             2    ||    13.0000    12.0000    -1.0000    -7.6923
             4    ||    14.0000    12.0000    -2.0000   -14.2857
             5    ||    14.0000    11.0000    -3.0000   -21.4286
            10    ||    12.0000    14.0000     2.0000    16.6667
            11    ||    11.0000    14.0000     3.0000    27.2727
            14    ||    15.0000    13.0000    -2.0000   -13.3333
            15    ||    16.0000    15.0000    -1.0000    -6.2500
```

can merge the files together and conduct a more in-depth record level comparison. By running a PROC COMPARE with ID variable, you can compare those records with a matching ID without having to explicitly merge your data together or create a new dataset.

Using the same CLASS and CLASSFIT datasets from the SASHELP library, you may have noticed both datasets contain a NAME variable. When you review the results from the initial PROC COMPARE, notice the summary indicates that none of the values for NAME match:



```
 All Variables Compared have Unequal Values

    Variable  Type  Len  Ndif   MaxDif

    Name      CHAR    8    14
    Sex       CHAR    1     6
    Age       NUM     8    11    3.000
    Height    NUM     8    14   17.700
    Weight    NUM     8    14   62.000
```



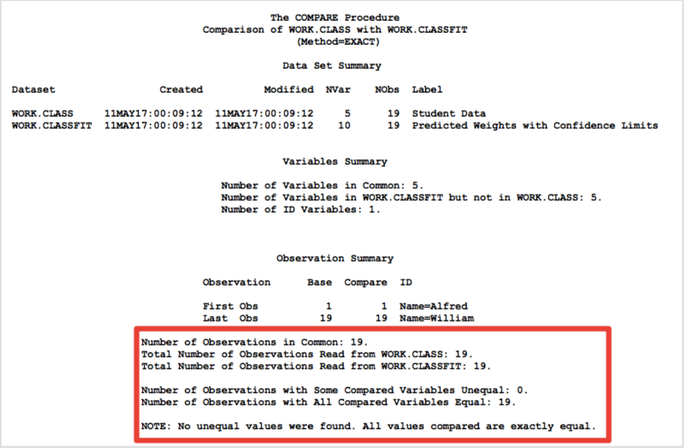
```
    4  ||  Carol           James
    5  ||  Henry           Thomas
    6  ||  James           John
   10  ||  John            Carol
   11  ||  Joyce           Henry
   13  ||  Louise          Robert
   14  ||  Mary            Barbara
   15  ||  Philip          Mary
   16  ||  Robert          William
   18  ||  Thomas          Alfred
   19  ||  William         Philip
```

When you run a standard PROC COMPARE with no options, SAS will just compare observation 1 from the BASE dataset with observation 1 from the COMPARE dataset followed by comparing observation 2 from the BASE dataset compared with observation 2 from the COMPARE dataset and so on. Thus, if the observations are not sorted and do not line up exactly, it will appear as though nothing matches between your two datasets.

To make the compare more useful, you can include the ID option. To demonstrate this with an example, we can compare the SASHELP.CLASS dataset to create 2 new datasets, WORK.CLASS and WORK.CLASSFIT. After sorting, simply add the ID option and specify the NAME variable as the ID variable in your call to PROC COMPARE:

```
proc sort data=sashelp.class out=class;
 by name;
run;

proc sort data=sashelp.classfit
out=classfit;
 by name;
run;

proc compare base=class
compare=classfit;
 id name;
run;
```

are no unequal values in any variable now that we have matched the datasets on NAME:



```
                              The COMPARE Procedure
                     Comparison of WORK.CLASS with WORK.CLASSFIT
                                 (Method=EXACT)

                               Data Set Summary

Dataset                 Created          Modified  NVar    NObs  Label

WORK.CLASS     11MAY17:00:09:12  11MAY17:00:09:12     5      19  Student Data
WORK.CLASSFIT  11MAY17:00:09:12  11MAY17:00:09:12    10      19  Predicted Weights with Confidence Limits


                               Variables Summary

                     Number of Variables in Common: 5.
                     Number of Variables in WORK.CLASSFIT but not in WORK.CLASS: 5.
                     Number of ID Variables: 1.


                              Observation Summary

                  Observation      Base  Compare  ID

                  First Obs           1        1  Name=Alfred
                  Last  Obs          19       19  Name=William

              Number of Observations in Common: 19.
              Total Number of Observations Read from WORK.CLASS: 19.
              Total Number of Observations Read from WORK.CLASSFIT: 19.

              Number of Observations with Some Compared Variables Unequal: 0.
              Number of Observations with All Compared Variables Equal: 19.

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

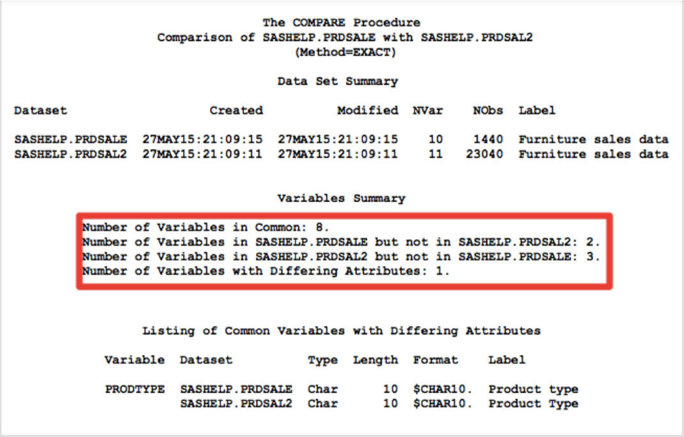
Comparing Metadata (Dataset Attributes)

In many scenarios, comparing the values between two datasets may not be useful if you widely between the two datasets but you would like to know how the structure of the datasets compare, you can add a few different options to the PROC COMPARE.

First, you can add the NOVALUES option, which suppresses any value comparisons in the PROC COMPARE output:

```
proc compare
 base = sashelp.prdsale
 compare = sashelp.prdsal2
 novalues;
run;
```

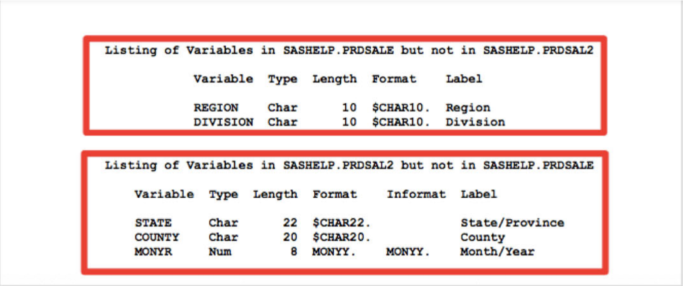
The Results output will be similar to before, listing the Data Set Summary, Variables Summary, Common variables with differing attributes, Observation Summary, Values



```
                        The COMPARE Procedure
            Comparison of SASHELP.PRDSALE with SASHELP.PRDSAL2
                            (Method=EXACT)

                          Data Set Summary

Dataset                  Created          Modified  NVar    NObs  Label

SASHELP.PRDSALE  27MAY15:21:09:15  27MAY15:21:09:15    10    1440  Furniture sales data
SASHELP.PRDSAL2  27MAY15:21:09:11  27MAY15:21:09:11    11   23040  Furniture sales data


                          Variables Summary

          Number of Variables in Common: 8.
          Number of Variables in SASHELP.PRDSALE but not in SASHELP.PRDSAL2: 2.
          Number of Variables in SASHELP.PRDSAL2 but not in SASHELP.PRDSALE: 3.
          Number of Variables with Differing Attributes: 1.


                Listing of Common Variables with Differing Attributes

             Variable  Dataset          Type  Length  Format     Label

             PRODTYPE  SASHELP.PRDSALE  Char      10  $CHAR10.   Product type
                       SASHELP.PRDSAL2  Char      10  $CHAR10.   Product Type
```

To expand the comparison and see which variables are in one dataset but not the other, simply add the listvar option:

```
proc compare
 base = sashelp.prdsale
 compare = sashelp.prdsal2
 novalues
 listvar;
run;
```



```
 Listing of Variables in SASHELP.PRDSALE but not in SASHELP.PRDSAL2

            Variable  Type  Length  Format    Label

            REGION    Char      10  $CHAR10.  Region
            DIVISION  Char      10  $CHAR10.  Division
```

```
 Listing of Variables in SASHELP.PRDSAL2 but not in SASHELP.PRDSALE

    Variable  Type  Length  Format    Informat  Label

    STATE     Char      22  $CHAR22.            State/Province
    COUNTY    Char      20  $CHAR20.            County
    MONYR     Num        8  MONYY.    MONYY.    Month/Year
```

Are you totally new to SAS?

Comparing Datasets by Observations

In addition to determining what are the differences in variables between two datasets, it can also be helpful to understand what are the differences in observations between two datasets. To see observation differences, there are a variety of other LIST options that can be used.

Some of the most common and useful LIST options that you can use include:

1. LISTBASEOBS – list observations found in the BASE dataset but not the COMPARE dataset
2. LISTCOMPOBS – list observations found in the COMPARE dataset but not the BASE dataset

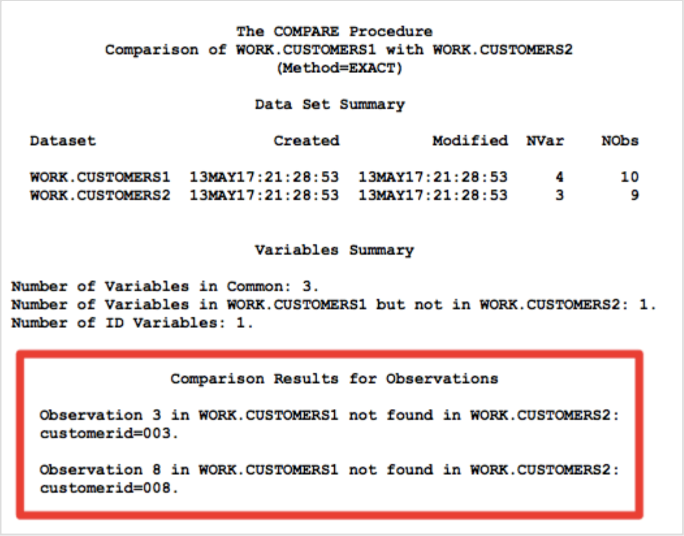
beginning of this article, let's look at a couple examples.

First, say for example you would like to know which observations are in WORK.CUSTOMERS1 but not WORK.CUSTOMERS2. Here, we will select the CUSTOMERS1 dataset as the BASE dataset and CUSTOMERS2 as the COMPARE dataset. Since we would like to know which observations are in CUSTOMERS1 and not CUSTOMERS2, and CUSTOMERS1 is our BASE dataset, we will add the LISTBASEOBS option.

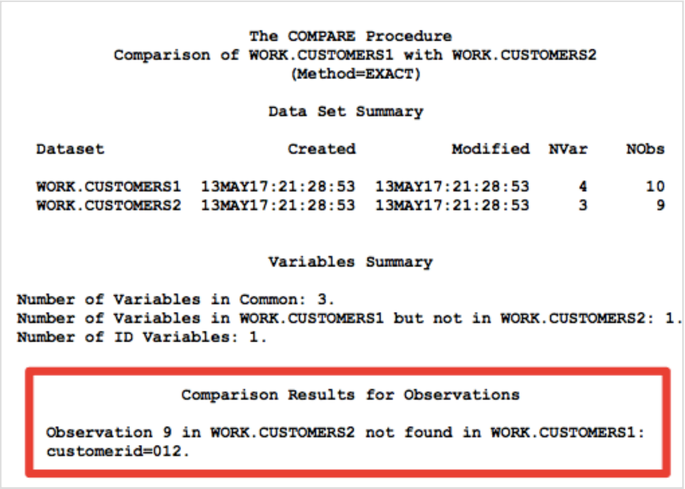
Recall that since we are comparing observations, we will use the CUSTOMERID variable as our ID variable in PROC COMPARE to ensure the correct records are being compared. Both input datasets are also sorted by ID prior to running the comparison.

```
 compare=customers2 listbaseobs;
 id customerid;
run;
```

As you can see from the output, there is now a new section that lists out the CUSTOMERID's which are found in the BASE dataset (CUSTOMERS1) but not the COMPARE dataset (CUSTOMERS2):



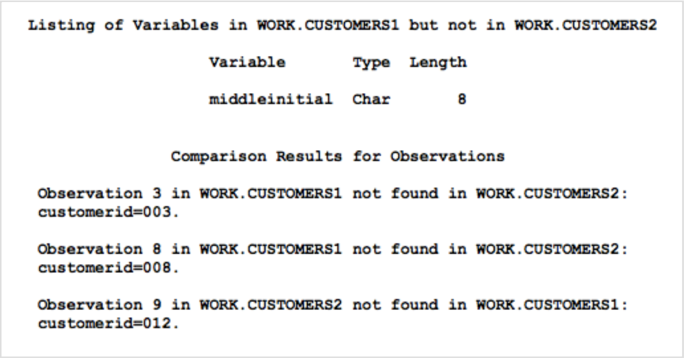
```
                      The COMPARE Procedure
         Comparison of WORK.CUSTOMERS1 with WORK.CUSTOMERS2
                         (Method=EXACT)

                        Data Set Summary

  Dataset                Created          Modified  NVar    NObs

  WORK.CUSTOMERS1  13MAY17:21:28:53  13MAY17:21:28:53     4      10
  WORK.CUSTOMERS2  13MAY17:21:28:53  13MAY17:21:28:53     3       9


                        Variables Summary

Number of Variables in Common: 3.
Number of Variables in WORK.CUSTOMERS1 but not in WORK.CUSTOMERS2: 1.
Number of ID Variables: 1.

                Comparison Results for Observations

  Observation 3 in WORK.CUSTOMERS1 not found in WORK.CUSTOMERS2:
  customerid=003.

  Observation 8 in WORK.CUSTOMERS1 not found in WORK.CUSTOMERS2:
  customerid=008.
  base = customers1
  compare = customers2 listcompareobs;
  id customerid;
run;
```



```
                     The COMPARE Procedure
        Comparison of WORK.CUSTOMERS1 with WORK.CUSTOMERS2
                        (Method=EXACT)

                       Data Set Summary

  Dataset                 Created          Modified  NVar    NObs

  WORK.CUSTOMERS1  13MAY17:21:28:53  13MAY17:21:28:53     4      10
  WORK.CUSTOMERS2  13MAY17:21:28:53  13MAY17:21:28:53     3       9


                       Variables Summary

Number of Variables in Common: 3.
Number of Variables in WORK.CUSTOMERS1 but not in WORK.CUSTOMERS2: 1.
Number of ID Variables: 1.

                Comparison Results for Observations

   Observation 9 in WORK.CUSTOMERS2 not found in WORK.CUSTOMERS1:
   customerid=012.
```

To avoid having to run PROC COMPARE many times, you can take advantage of the LISTALL option, which will essentially combine the output of LISTVARS, LISTBASEOBS and LISTCOMPOBS into a single output.

and a list of any observation found in one dataset but not the other (equivalent to what LISTBASEOBS, LISTCOMPOBS or also LISTOBS produces):



```
 Listing of Variables in WORK.CUSTOMERS1 but not in WORK.CUSTOMERS2

                   Variable       Type  Length

                   middleinitial  Char       8


               Comparison Results for Observations

  Observation 3 in WORK.CUSTOMERS1 not found in WORK.CUSTOMERS2:
  customerid=003.

  Observation 8 in WORK.CUSTOMERS1 not found in WORK.CUSTOMERS2:
  customerid=008.

  Observation 9 in WORK.CUSTOMERS2 not found in WORK.CUSTOMERS1:
  customerid=012.
```

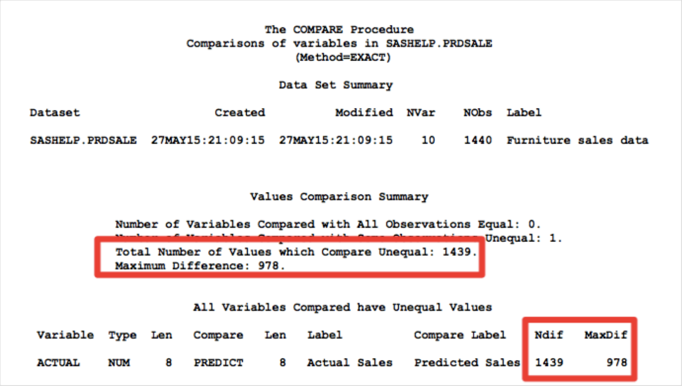
Comparing Values within a Dataset

Not only can PROC COMPARE be used for

examine how many times the predicted value was different from the actual value, the largest difference between values as well as the specific differences and percentage differences between the two you can use a single call to PROC COMPARE.

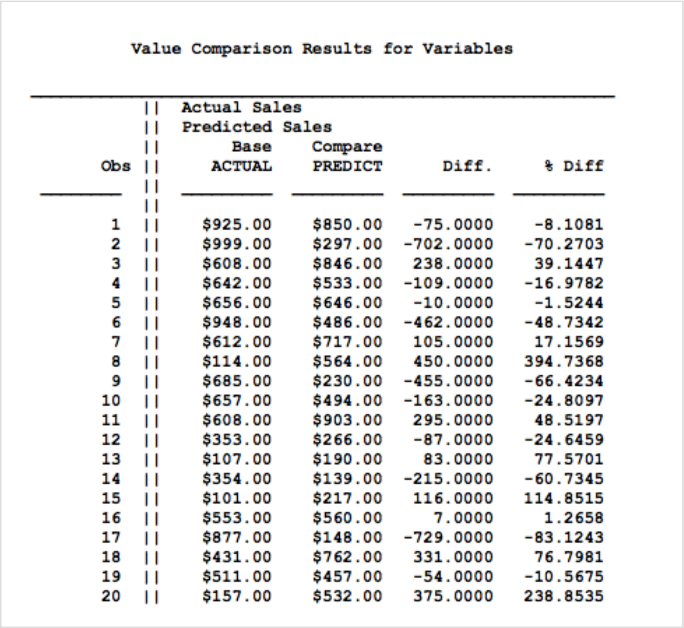
The syntax for within dataset comparisons is quite simple. As before, you specify a BASE dataset (PRDSALE in this example) but this time you do not specify a COMPARE dataset. Next, you use the VAR statement to list any variables you want to compare. After specifying the variables you want to compare, the WITH statement is used to specify the corresponding variables you want to compare the first set of variables with. So, to compare ACTUAL with PREDICT in the PRDSALE data, you would use the following syntax:

equal to the actual sale value for any observation. Also, the maximum difference was found to be 978 ($978.00) between Actual and Predicted Sales:



```
                          The COMPARE Procedure
                 Comparisons of variables in SASHELP.PRDSALE
                             (Method=EXACT)

                            Data Set Summary

Dataset                    Created          Modified  NVar    NObs  Label

SASHELP.PRDSALE  27MAY15:21:09:15  27MAY15:21:09:15    10    1440  Furniture sales data


                        Values Comparison Summary

           Number of Variables Compared with All Observations Equal: 0.
           Number of Variables Compared with Some Observations Unequal: 1.
           Total Number of Values which Compare Unequal: 1439.
           Maximum Difference: 978.


                  All Variables Compared have Unequal Values

 Variable  Type  Len  Compare   Len  Label          Compare Label    Ndif    MaxDif

 ACTUAL    NUM     8  PREDICT     8  Actual Sales   Predicted Sales  1439       978
```

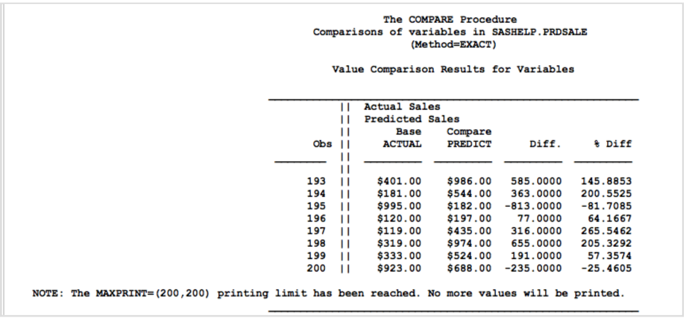
Exa- ining the output further, you can see the observation level comparison where the difference and percentage difference for the first 50 observations are shown, as you can see in the partial output below:



```
                Value Comparison Results for Variables

 __________________________________________________________
           ||  Actual Sales
           ||  Predicted Sales
           ||       Base    Compare
       Obs ||     ACTUAL    PREDICT      Diff.     % Diff
           ||
  ________ ||  _________  _________  _________  _________
           ||
         1 ||    $925.00    $850.00   -75.0000    -8.1081
         2 ||    $999.00    $297.00  -702.0000   -70.2703
         3 ||    $608.00    $846.00   238.0000    39.1447
         4 ||    $642.00    $533.00  -109.0000   -16.9782
         5 ||    $656.00    $646.00   -10.0000    -1.5244
         6 ||    $948.00    $486.00  -462.0000   -48.7342
         7 ||    $612.00    $717.00   105.0000    17.1569
         8 ||    $114.00    $564.00   450.0000   394.7368
         9 ||    $685.00    $230.00  -455.0000   -66.4234
        10 ||    $657.00    $494.00  -163.0000   -24.8097
        11 ||    $608.00    $903.00   295.0000    48.5197
        12 ||    $353.00    $266.00   -87.0000   -24.6459
        13 ||    $107.00    $190.00    83.0000    77.5701
        14 ||    $354.00    $139.00  -215.0000   -60.7345
        15 ||    $101.00    $217.00   116.0000   114.8515
        16 ||    $553.00    $560.00     7.0000     1.2658
        17 ||    $877.00    $148.00  -729.0000   -83.1243
        18 ||    $431.00    $762.00   331.0000    76.7981
        19 ||    $511.00    $457.00   -54.0000   -10.5675
        20 ||    $157.00    $532.00   375.0000   238.8535
```

Note that by default, the observation level comparisons are limited to 50 records, which is true both for within dataset comparisons and comparisons between two datasets. To expand the comparison to 200 records for example, simply use the MAXPRINT option with PROC COMPARE, as shown here:

dataset comparisons. As you can see the output shown partially below, the comparison results are now printed for up to 200 observations:



```
                              The COMPARE Procedure
                    Comparisons of variables in SASHELP.PRDSALE
                                 (Method=EXACT)

                      Value Comparison Results for Variables

                  __________________________________________________________
                          ||  Actual Sales
                          ||  Predicted Sales
                          ||       Base    Compare
                     Obs  ||     ACTUAL    PREDICT       Diff.     % Diff
                   ________  ||  _________  _________  _________  _________
                          ||
                     193  ||    $401.00    $986.00    585.0000   145.8853
                     194  ||    $181.00    $544.00    363.0000   200.5525
                     195  ||    $995.00    $182.00   -813.0000   -81.7085
                     196  ||    $120.00    $197.00     77.0000    64.1667
                     197  ||    $119.00    $435.00    316.0000   265.5462
                     198  ||    $319.00    $974.00    655.0000   205.3292
                     199  ||    $333.00    $524.00    191.0000    57.3574
                     200  ||    $923.00    $688.00   -235.0000   -25.4605

NOTE: The MAXPRINT=(200,200) printing limit has been reached. No more values will be printed.
                  __________________________________________________________
```

Comparing Specific Values in an Output Dataset

PROC COMPARE is also a useful tool for conducting more in-depth comparisons between two datasets. If there are differences in values between the two datasets you are comparing and you would like to drill down to find out exactly how the values of two matching variables are different, PROC COMPARE has a

- OUTBASE – includes the values from the BASE dataset
- OUTDIF – includes indicators to show what position the differences are found
- OUTNOEQUAL – restrict output observations to only those observations with differences

Using the CUSTOMERS1 and CUSTOMERS2 datasets created at the beginning of this article, the following example illustrates how to create both a convenient Output dataset and report on differences in values.

As before, we will be using the ID option with PROC COMPARE so we need to first ensure that the CUSTOMERS1 and CUSTOMERS2 datasets are sorted prior to running the compare:

```
proc sort data = customers1;
 by customerid;
 value missing_num
    . = 'Missing'
    other = 'Present'
    ;
run;
```

To create an output dataset of differences, you simply need to add the **OUT** argument to the PROC COMPARE statement. Immediately following the **OUT** argument is the name of the output dataset that will be created. In the following example, the output dataset will be named CUSTOMER_COMPARE.

After the dataset name, the desired options OUTCOMPARE, OUTBASE, OUTDIFF and OUTNOEQUAL as described above are added.

Since we are only interested in the output dataset in this example, the NOPRINT option is

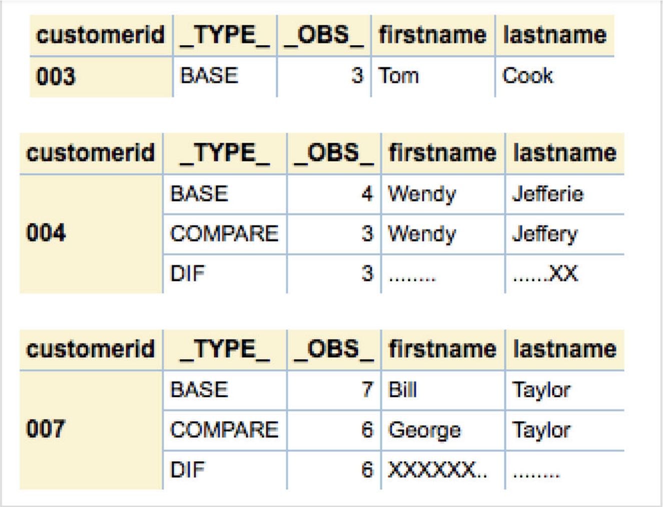
```
run;
```

As you can see in the output dataset shown below, we now have groups of records where there is one record for each BASE observation, one for each COMPARE observation, as well as a DIF record which highlights which position the differences occur between the two values. When there is no matching observation (based on CUSTOMERID) between the BASE and COMPARE datasets, there will be only the respective BASE or COMPARE observation that is found in one dataset but not the other.

While comparing differences within a dataset is certainly useful, it can often be easier to review differences within CUSTOMERIDs by further summarizing the results with PROC PRINT. Using the following PROC PRINT syntax, the output dataset CUSTOMER_COMPARE we created above with PROC COMPARE can be easily grouped by CUSTOMERID for further review:

```
proc print data = customer_compare;
 by customerid;
 id customerid;
run;
```

In the resulting output shown partially below, you can now easily compare the values from the



customerid	_TYPE_	_OBS_	firstname	lastname
003	BASE	3	Tom	Cook

customerid	_TYPE_	_OBS_	firstname	lastname
004	BASE	4	Wendy	Jefferie
	COMPARE	3	Wendy	Jeffery
	DIF	3	....	XX

customerid	_TYPE_	_OBS_	firstname	lastname
007	BASE	7	Bill	Taylor
	COMPARE	6	George	Taylor
	DIF	6	XXXXXX..	....

SASCRUNCH.COM

HOME

SAS® CERTIFICATION TRAINING

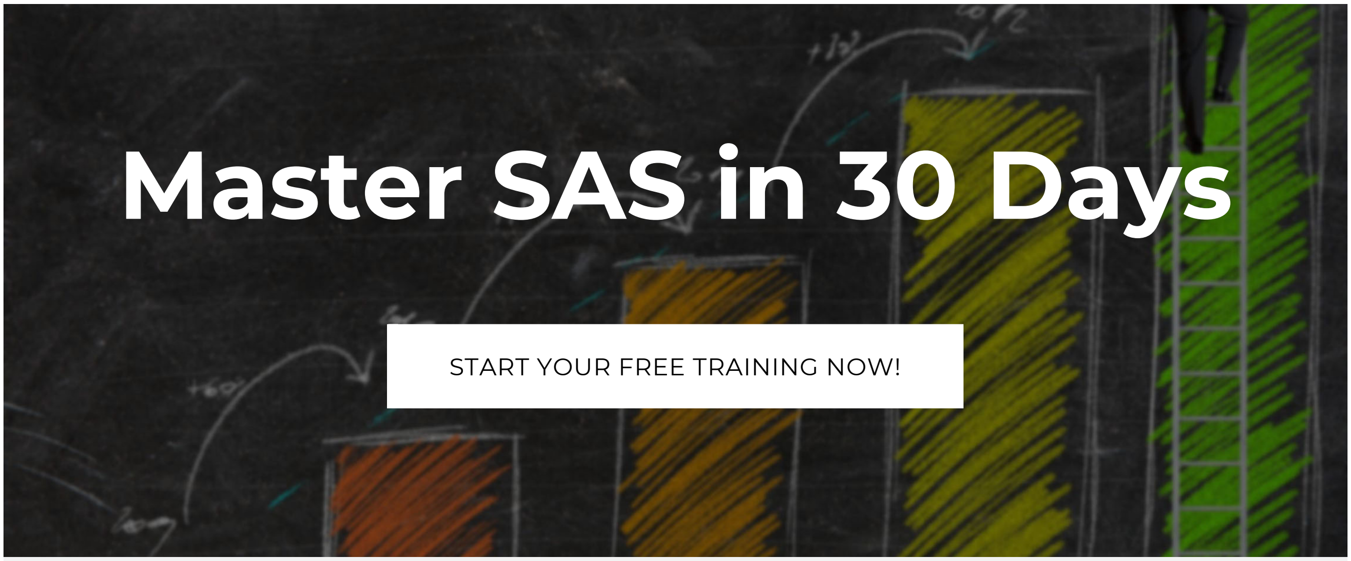
ONLINE COURSES

ARTICLES

SERVICES

MORE...

Name



SASCRUNCH.COM

HOME

SAS® CERTIFICATION TRAINING

ONLINE COURSES

ARTICLES

SERVICES

MORE...