

Jane Eslinger Bio

Jane Eslinger is an independent programmer based in North Carolina with over 20 years of experience with SAS.

Jane has authored two books: The SAS® Programmer's PROC REPORT Handbook: Basic to Advanced Reporting Techniques and The SAS® Programmer's PROC REPORT Handbook: ODS Companion. Her SAS certifications include Advanced Programmer for SAS®9, SAS® Certified Data Scientist, and SAS® Certified Advanced Visual Business Analyst.

Highlighting the Differences: PROC COMPARE in Excel

Jane Eslinger

Eslinger Enterprises, LLC

Reasons for Using an OUT= Data Set

1. Allows the programmer to concentrate on just the value differences.
2. It might be easier to parse than the printed report when there are a large number of variables or differences.
3. Data sets are in a programmer's wheelhouse. 😊
4. The default printed report is not pretty. (But wait! See Jane's other PROC COMPARE paper.)

PROC COMPARE Statement Options

- **OUTBASE** - Write an observation for each observation in the BASE= data set
- **OUTCOMP** - Write an observation for each observation in the COMPARE= data set
- **OUTNOEQUAL** - Suppress the writing of observations when all values are equal
- **OUTDIF** - Write an observation that contains the differences for each pair of matching observations

PROC COMPARE Code

```
proc compare base=Base comp=Compare  
    out=__rslt0 criterion=&criterion noprint  
    outbase outcomp outnoequal outdif;  
run;
```

Upgrade in progress: BY statement

OUT= Data Set

TYPE	_OBS_	Latitude	Longitude	Name	Area	Type	Farenheit	Celsius
BASE	14	53.892	-166.93	Makushin Volcano Fumaroles	Unalaska	Hotspring	310	154
COMPARE	14	53.892	-166.93	Makushin Volcano Fumaroles	Unalaska	Hotspring	310	120
DIF	14	E	E				E	-34
BASE	88	58.37	-137.09	Hot Springs Near Icy Point	Mt. Fairweathe	Hotspring	153	67
COMPARE	88	58.37	-137.09	Hot Springs Near Icy Point	Mt. Fairweathe	Hotspring	153	67
DIF	88	E	E	XX.....			E	E
BASE	1400	48.77	-121.813	Sherman Crater Fumaroles	Concrete	Hotspring	266	130
COMPARE	1400	48.77	-121.813	Sherman Crater Fumaroles	Concrete	Hotspring	266	120
DIF	1400	E	E				E	-10

Character differences are designated by an X.

Numeric differences are the value of Base – Compare.

OUT= Data Set

TYPE	_OBS_	Latitude	Longitude ^	Name	Area	Type	Fahrenheit	Celsius
BASE	14	53.892	-166.93	Makushin Volcano Fumaroles	Unalaska	Hotspring	310	154
COMPARE	14	53.892	-166.93	Makushin Volcano Fumaroles	Unalaska	Hotspring	310	120
DIF	14	E	E				E	-34
BASE	88	58.37	-137.09	Hot Springs Near Icy Point	Mt. Fairweathe	Hotspring	153	67
COMPARE	88	58.37	-137.09	Hot Springs Near Icy Point	Mt. Fairweathe	Hotspring	153	67
DIF	88	E	E	XX.....			E	E
BASE	1400	48.77	-121.813	Sherman Crater Fumaroles	Concrete	Hotspring	266	130
COMPARE	1400	48.77	-121.813	Sherman Crater Fumaroles	Concrete	Hotspring	266	120
DIF	1400	E	E				E	-10

Character matches are designated by an period (.).

Numeric matches are the value .E.

Excel Results Goal

[illegible]

Excel Report Goal

1. The order of the columns matches the variable order in the OUT= data set.
2. The values of 'Production' and 'Validation' are supplied in a macro parameter. They correspond to the BASE= data set and the COMP= data set.
3. The value from each data set is highlighted when there is a difference. Foreground and background colors match those in Excel's default Neutral cell style.
4. Blank rows are inserted between each observation (comparison pair).

Separate the Observations

```
data __rslt1 __rslt1_dif(drop=dataset);  
    set __rslt0;  
    length Dataset $10;  
  
    if _type_ = 'BASE' then dataset = "&baselbl";  
    else if _type_ = 'COMPARE' then dataset = "&complbl";  
  
    if _type_ = 'DIF' then output __rslt1_dif;  
    else output __rslt1;  
  
    drop _type_;  
run;
```

Variable Lists

```
proc contents data=__rslt0(drop=_obs_ _type_)  out=__rslt1_vars(keep=name type varnum)
    noprint;

proc sort data=__rslt1_vars; by varnum;  run;

proc sql noprint;
    select name into: varorder separated by ' ' from __rslt1_vars;
    select name into: charnames separated by ' ' from __rslt1_vars
        where type = 2;
    select '_'||name into: _charnames separated by ' ' from __rslt1_vars
        where type = 2;
    select name into: numnames separated by ' ' from __rslt1_vars  where type = 1;
    select '_'||name into: _numnames separated by ' ' from __rslt1_vars
        where type = 1;

quit;
```

Indicator Variables

```
data __rslt2_dif;  
    set __rslt1_dif;  
    array _char1{*} $ &charnames;  
    array _nums1{*} &numnames;  
    array _char2{*} $ &_charnames; *indicator;  
    array _nums2{*} &_numnames; *indicator;  
    do i=1 to dim(_char1);  
        if ^index(_char1{i},'X') then _char2{i} = ' '  
        else _char2{i} = 'X';  
    end;  
    do j=1 to dim(_nums1);  
        if _nums1{j} = .E then _nums2{j} = .;  
        else _nums2{j} = 9999;  
    end;  
    keep _;;  
  
run;
```

__rslt2_dif Data Set

OBS	_Name	_Area	_Ty... ^	_Latitude	_Longitude	_Fahrenheit	_Celsius
14				.	.	.	9999
88	X			.	.	.	.
1400				.	.	.	9999

```
data tab_dif;  
    merge __rslt1 __rslt2_dif;  
    by _obs_;  
run;
```

PROC REPORT Code

```
proc report data=tab_dif ;  
    column dataset _obs_ &varorder &_charnames  
        &_numnames dummy;  
    define _obs_ / order;
```

&varorder contains the variables in the desired order

&_charnames contains the character indicator variables

&_numnames contains the numeric the indicator variables

dummy is for computing purposes

PROC REPORT Code (continued)

```
%let ind1=1;  
%let _curr1 = %scan(&_charnames,&ind1,%str( ));  
%do %while (%length(&_curr1)>0);  
    define &_curr1 / noprint;  
        %let ind1 = %eval(&ind1 + 1);  
        %let _curr1 = %scan(&_charnames,&ind1,%str( ));  
%end;
```

PROC REPORT Code (continued)

```
%let ind2=1;
%let _curr2 = %scan(&_numnames,&ind2,%str( ));
%let _xcurr2 = %scan(&numnames,&ind2,%str( ));
%do %while (%length(&_curr2)>0);
    define &_curr2 / display noprint;
    define &_xcurr2 / display;
    %let ind2 = %eval(&ind2 + 1);
    %let _curr2 = %scan(&_numnames,&ind2,%str( ));
    %let _xcurr2 = %scan(&numnames,&ind2,%str( ));
%end;
```

PROC REPORT Code (continued)

```
define dummy / noprint  
compute after _obs_;  
    line ' ';  
endcomp;
```

PROC REPORT Code (continued)

```
compute dummy;
  array _char1{*} $ &charnames;
  array _nums1{*} &numnames;
  array _char2{*} $ &_charnames;
  array _nums2{*} &_numnames;
  do i=1 to dim(_char1);
    if ^missing(_char2{i}) then call define(vname(_char1{i}),
      'style','style={foreground=#9C6500 background=#FFEB9C}');
  end;
  do j=1 to dim(_nums1);
    if ^missing(_nums2{j}) then call define(vname(_nums1{j}),
      'style','style={foreground=#9C6500 background=#FFEB9C}');
  end;
endcomp;
```


Questions?