

Colour me useful

Roberto de Gois, Veramed Ltd, London, United Kingdom

ABSTRACT

PROC FORMAT is a procedure that allows the creation of user defined formats and/or informats that can be associated to variables using the format statement. In other words, this procedure grants the user creation of values that can be applied to variables, in order to modify how data is read and stored in SAS® or how data is printed.

Programmers might not be aware of the usefulness of PROC FORMAT when used in a different way. The procedure has the capability of producing a format that can change the background, foreground and border colours of printed variables when used in a reporting procedure. This method can enable more possibilities, including the creation of a traffic light review system within SAS®.

INTRODUCTION

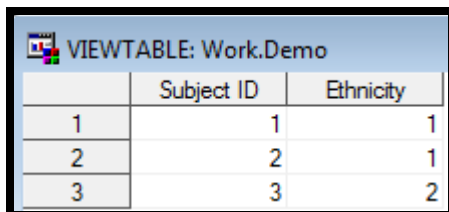
The purpose of this paper is not to introduce the reader to PROC FORMAT, but to demonstrate a slightly different approach when using the FORMAT procedure in SAS®.

FOUNDATION

First, we will start by creating a dataset called DEMO and input some dummy data using the CARDS statement. Note that the dummy data could also be done with the DATALINES statement.

```
data demo;
  input subjid ethnic;
  label subjid = "Subject ID"
        ethnic = "Ethnicity";
  cards;
  1 1
  2 1
  3 2
  ;
run;
```

The above code generates the DEMO dataset with 2 variables and 3 observations. The data are in a numerical format. When opening the dataset this information will not make much sense as we do not know what the Ethnicity values 1 or 2 means.



	Subject ID	Ethnicity
1	1	1
2	2	1
3	3	2

THE USUAL PROC FORMAT

By using the FORMAT procedure we can create a user defined format that can be subsequently associated to the ETHNIC variable allowing a more comprehensive interpretation of the data.

```
proc format;
  value ethnic
    1 = "Hispanic or latino"
    2 = "Not hispanic or latino"
  ;
run;
```

PhUSE 2016

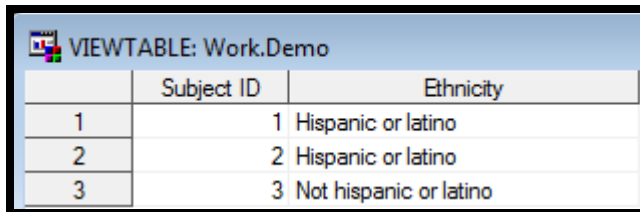
The PROC FORMAT defines formats or informats for variables whilst the VALUE statement creates a value that specifies character strings to be used to print variable values. A format/informat name can be up to 32 characters long (including the '\$' for character formats), cannot end in a number and cannot be the same name as a pre-existing format.

USING THE FORMAT STATEMENT

Since a format has already been created we can associate this with the ETHNIC variable in a data step.

```
data demo;
  input subjid ethnic;
  label subjid = "Subject ID"
        ethnic = "Ethnicity";
  format ethnic ethnic.;
  cards;
1 1
2 1
3 2
;
run;
```

The format ethnic. is now applied to the variable ETHNIC in the data step above and the output values of ETHNIC are now self-explanatory when the code is submitted.



	Subject ID	Ethnicity
1	1	Hispanic or latino
2	2	Hispanic or latino
3	3	Not hispanic or latino

COLOURING WITH PROC FORMAT

Instead of creating a format to specify character strings in an output, it is also possible to create a format that can define the background, foreground or border colour of a variable.

Let's have a look at the output below:

Obs	subjid	stdate	endate	days
1	1	2013-05-01	2014-05-01	365
2	2	2013-05-02	2014-05-01	364
3	3	2013-05-13	2014-03-01	292
4	4	2013-05-10	2014-05-01	356
5	5	2013-05-01	2013-05-01	0
6	3	2013-05-01	2014-05-01	365
7	7	2013-05-01	2014-05-02	366
8	8	2013-05-01	2014-06-01	396
9	9	2013-05-01	2014-05-01	365
10	10	2013-05-01	2014-05-01	365

At a quick glance how can we tell:

- Which observations have a value of 365 in the DAYS variable?
- Which observations have a +/- 1 deviation from 365 in the DAYS variable?
- Which observations have a value less than 363 or greater than 367 in the DAYS variable?

PhUSE 2016

As we can see, it is difficult to answer posed questions from the output in the current state. PROC FORMAT can help us to highlight the output without subsetting the data.

CREATING THE COLOURING FORMAT

A FORMAT procedure is called in order to define a format for a variable but this time we will specify, in the VALUE statement, a value that will be used to modify the background colour of a variable in the output window for each selection criterion.

```
proc format;
  value check
    364,366 = "yellow"
    365     = "green"
    other   = "red"
  ;
run;
```

We have used actual colour names as the character strings in the code above, but hexadecimal codes can also be used instead as such:

```
proc format;
  value check
    364,366 = "CXFFFF00"
    365     = "CX00FF00"
    other   = "CXFF0000"
  ;
run;
```

APPLYING THE FORMAT

We will apply the newly created format whilst using a simple reporting procedure. The dataset TEST is output using PROC PRINT. The variables displayed in the output window are defined in the VAR statement.

```
proc print data=test;
  var subjid stdate endate;
  var days / style = [background = check.];
run;
```

To modify the background or any other characteristic of specific variables these are defined in a separate VAR statement along with the STYLE option. In the STYLE option the element or attribute to be modified, in this case 'background', is stated within square brackets followed by the format name check.. This will enable the conditional background colouring of the variable DAYS.

Obs	subjid	stdate	endate	days
1	1	2013-05-01	2014-05-01	365
2	2	2013-05-02	2014-05-01	364
3	3	2013-05-13	2014-03-01	292
4	4	2013-05-10	2014-05-01	356
5	5	2013-05-01	2013-05-01	0
6	3	2013-05-01	2014-05-01	365
7	7	2013-05-01	2014-05-02	366
8	8	2013-05-01	2014-06-01	396
9	9	2013-05-01	2014-05-01	365
10	10	2013-05-01	2014-05-01	365

We now have an output that quickly lets us find the answers to our initial questions.

PhUSE 2016

Note that we can change other attributes of the output variables. The colours of the foreground or border colour can also be altered.

In order to modify the foreground we'll use the same syntax as described above and just change the attribute name in the STYLE option.

```
proc print data=test;
  var subjid stdate endate;
  var days / style = [foreground = check.];
run;
```

Obs	subjid	stdate	endate	days
1	1	2013-05-01	2014-05-01	365
2	2	2013-05-02	2014-05-01	364
3	3	2013-05-13	2014-03-01	292
4	4	2013-05-10	2014-05-01	356
5	5	2013-05-01	2013-05-01	0
6	3	2013-05-01	2014-05-01	365
7	7	2013-05-01	2014-05-02	366
8	8	2013-05-01	2014-06-01	396
9	9	2013-05-01	2014-05-01	365
10	10	2013-05-01	2014-05-01	365

Next, we will use the bordercolor attribute on the DAYS variable. It will still emphasize the conditional colouring.

```
proc print data=test;
  var subjid stdate endate;
  var days / style = [bordercolor = check.];
run;
```

Obs	subjid	stdate	endate	days
1	1	2013-05-01	2014-05-01	365
2	2	2013-05-02	2014-05-01	364
3	3	2013-05-13	2014-03-01	292
4	4	2013-05-10	2014-05-01	356
5	5	2013-05-01	2013-05-01	0
6	3	2013-05-01	2014-05-01	365
7	7	2013-05-01	2014-05-02	366
8	8	2013-05-01	2014-06-01	396
9	9	2013-05-01	2014-05-01	365
10	10	2013-05-01	2014-05-01	365

PhUSE 2016

BEYOND PROC PRINT

The previous example showed us how the format containing the colour character strings can be applied using PROC PRINT. The same can be used in PROC REPORT and PROC TABULATE. Let us see how.

When using PROC REPORT we use the DEFINE statement to describe how to display and use an item. It is within the DEFINE statement where we can call the STYLE option and modify the required attribute. We will follow this procedure in the example below.

```
proc report data = test nowd;
  columns subjid stdate endate days;

  define subjid / display;
  define stdate / display;
  define endate / display;
  define days / style = [background = check.];
run;
```

subjid	stdate	endate	days
1	2013-05-01	2014-05-01	365
2	2013-05-02	2014-05-01	364
3	2013-05-13	2014-03-01	292
4	2013-05-10	2014-05-01	356
5	2013-05-01	2013-05-01	0
6	2013-05-01	2014-05-01	365
7	2013-05-01	2014-05-02	366
8	2013-05-01	2014-06-01	396
9	2013-05-01	2014-05-01	365
10	2013-05-01	2014-05-01	365

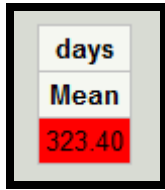
When using PROC TABULATE the process is slightly different as this procedure displays descriptive statistics in a tabular format. In this example we will obtain the average value of the DAYS variable and see whether this is equal to 365 and highlight it green. Otherwise we will colour it red.

First, let us create a new format called avg_check with the aforementioned requirements.

```
proc format;
  value avg_check
    other = "red"
    365 = "green"
  ;
run;
```

Next, we will use PROC TABULATE to obtain the average value of DAYS and highlight the background attribute using the STYLE option and the newly created format.

```
proc tabulate data=test;
  class subjid stdate endate;
  var days;
  table days * mean * [style = [background = avg_check.]];
run;
```



days
Mean
323.40

Note that unlike the previous procedures (PRINT and REPORT), the STYLE option is within square brackets. As the variable is below 365, it is displayed on a red background as specified above.

CONCLUSION

PROC FORMAT allows the user to create their own formats and/or informats that can be linked to variables using the format statement. However, this procedure can also be used to change variable styles (in an output) such as the variable BACKGROUND, FOREGROUND and/or BORDERCOLOR.

By applying this method the programmer can facilitate a much more efficient, quicker and easier review of the data focusing the reader's attention on key results. One can also enable a traffic light review system by highlighting good results in green, neutral results in yellow and poor results in red, if desired.

The FORMAT containing the preferred colours to be used in the output can be applied during a PROC PRINT, PROC REPORT or PROC TABULATE.

REFERENCES

<http://support.sas.com/documentation/cdl/en/proc/65145/HTML/default/viewer.htm#n1c16dxnndwzfzyn14o1kb8a4312m.htm>
<http://support.sas.com/documentation/cdl/en/proc/65145/HTML/default/viewer.htm#p1798jn94kxfo2n1g0dnuv4ccufu.htm>

CONTACT INFORMATION

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

Roberto de Gois
Veramed Ltd
5th Floor Regal House
70 London Road
Twickenham / TW1 3QS
Work Phone: +44 (0)20 3696 7240
Fax:
Email: roberto.degois@veramed.co.uk
Web: www.veramed.co.uk

Brand and product names are trademarks of their respective companies.