

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 61, 3rd Quarter 2020

This newsletter issue welcomes 2 new VIEWS Consultants: Jonas Bilenas and Norman Williams. It also contains an article from LeRoy Bessler about Donut Charts, an ODS article from Jonas Bilenas, a solution to including program documentation from Norman Williams, and I have added a review of the SAS Analytics Explorers communities. As usual I have publicised some SAS-related events for the next few months, and I have also added some more SAS formats, options and functions that you may or may not have come across before, but you might find interesting, and even useful.

If you would like to contribute an article, to re-visit and improve an existing article, or just discuss the possibility of doing so, please feel free to send an email to me at views-uk@hollandnumerics.org.uk.

Philip R Holland (Newsletter Editor)

Membership of VIEWS is **free** and you can register for notification of all future VIEWS News issues by registering at blog.hollandnumerics.org.uk, and then emailing views-uk@hollandnumerics.org.uk, or using the blog site menu, to request access to the VIEWS UK forum. Members will receive an email notification whenever a new issue of the free quarterly VIEWS News is published on the web site, please inform us when you change your email address, otherwise your registration may be cancelled.

Did You Know?

If you have a useful hint or tip, please send it to the Editor, and share it with the VIEWS membership.

The ODS Graphics Donut Chart: A Popular Pie Chart Alternative

Donut Charts are created with the new ODS Graphics SGPIE procedure, which also creates Pie Charts. PROC SGPIE, as of July 2020, is still in Pre-Production status in SAS Version 9.4 TS01M6. An anomaly described in the second example below will presumably be fixed for the Production release.

I used Google search to seek the answer to this question: “Who invented the donut chart?”. It yielded a lot of information about Pie Charts. Same result ensued when trying to find when it was invented. When I added “-pie” to the searches, it turned up nothing. Pie Chart critics who have never learned my recommended methods for creating communication-effective Pie Charts try to make the case that Donut Charts are inherently better. See *VIEWS News Issue 59, 1st Quarter 2020* for my recommended ways to create communication-effective pie charts.

The ONLY advantage of a Donut Chart is the option to imbed some text inside of the hole. I never found the answer to my questions. Was donut chart inventor somebody who prefers donuts over pie as a dessert? Being non-discriminatory, I like both of these desserts equally.

Before I proceed, I need to address a spelling matter. At <https://www.thesun.co.uk/fabulous/food/6426977/doughnut-donut-two-spellings-correct-american-english/> I found: “Doughnut is the original spelling of the word, and if often listed as the correct spelling with donut as a variant spelling. According to the Oxford English Dictionary, one of the first times the word appears in writing is in Washington Irving's History of New York in 1809”. In keeping with my non-discriminatory attitude, in the first image you will see **both** spellings.

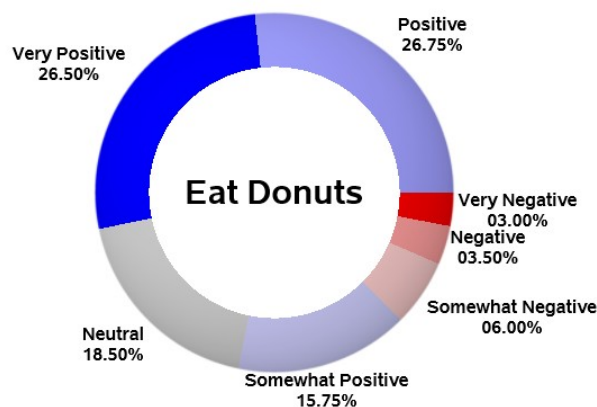
In Pie Charts, the portions are called “slices”, in conformity with how dessert pie sections are called, at least in my experience. In Donut Charts, my pre-emptive contribution to the yet to be published history of the Donut Chart is to suggest that the donut sections be called “bites”. Note that “bite order” for Donut Charts is controlled by an option that is inappropriately named “SLICEORDER”.

This article presents two examples of Donut Chart:

- categories (aka bites) ordered by size;
- categories/bites ordered by relative significance.

Any Donut Chart could have the bites ordered by alphabetic sequence of category/description, which is the default. That really has no real communication benefit. Alphabetic order for categories only has value when the number of categories is very large, and you want to expedite the finding process for any specific category (e.g. all of the two-hundred-plus countries in the United Nations). A huge number of categories is not well suited to either a donut chart or a pie chart, so alphabetic order seems always irrelevant, but something has to be the default option, and alphabetic order is harmless.

Survey Opinions About Doughnuts - Ordered By Percent



Data Source: <https://aytm.com/blog/doughnuts-survey/>

If you are body-weight-control-oriented or what is sometimes, at least in the USA, called “calorie conscious”, I hope

that you can forgive my irrelevant exhortation to “Eat Donuts”. It was, in my opinion, the most appropriate text string that would fit at a very readable font size in the space available. Just like donuts, big fonts are better than small.

This Donut Chart starts at Three O’Clock (0 degrees) and the bites are sequenced counterclockwise. I would prefer to start at Twelve O’Clock (90 degrees), and have the bites sequenced clockwise. Unfortunately, in the latter case, the labels for Very Negative and Negative collide, unless readability is sacrificed by resorting to a small enough font.

The essential part of the code to create this given below is rather simple, but extras like SAS input data set creation, TITLE & FOOTNOTE statements, and helpful comments make it longer.

Pay particular attention to the remarks about the HoleLabel, which is the distinguishing feature and possible advantage of the Donut Chart over a Pie Chart. Of course, one could put the same HoleLabel text into a subtitle on a Pie Chart, but the HoleLabel is a currently popular data visualization gimmick.

Here is the code to create the chart above:

```
DATA ForChart;
  INFILE CARDS;
  INPUT
    @1 Percent 5.2
    @7 Opinion $17.
    @25 RankByAttitude 1.;
  CARDS;
03.00 Very Negative      7
03.50 Negative           6
06.00 Somewhat Negative 5
18.50 Neutral            4
15.75 Somewhat Positive 3
26.75 Positive           2
26.50 Very Positive      1
;
RUN;

PROC FORMAT;
  PICTURE PctFmt LOW-HIGH='009.99%';
RUN;

ODS RESULTS OFF;
ODS _ALL_ CLOSE;
ODS LISTING STYLE=LISTING
  GPATH="C:\VIEWS_Aug2020\Results";

ODS GRAPHICS / RESET=ALL NOSCALE
  WIDTH=6.5IN HEIGHT=4.875IN
  IMAGENAME='Donut_Chart_Order_By_Percent';

TITLE1 FONT='Arial/Bold' HEIGHT=14pt
  'Survey Opinions About Doughnuts - Ordered By
Percent';
TITLE2 COLOR=White FONT='ARIAL/BOLD' HEIGHT=14pt
  'INVISIBLE TEXT: For desired white space.';
FOOTNOTE1 COLOR=White FONT='ARIAL/BOLD'
  HEIGHT=14pt
  'INVISIBLE TEXT: For desired white space.';
FOOTNOTE2 FONT='Arial' HEIGHT=14pt
  'Data Source: https://aytm.com/blog/doughnuts-
survey/';

PROC SGPIE DATA=ForChart;
  STYLEATTRS DATACOLORS=(CX9999FF CX0000FF
    CXCCCCCXCCCCFF CXFFCCCC CXFF9999
    CXFF0000
  );
```

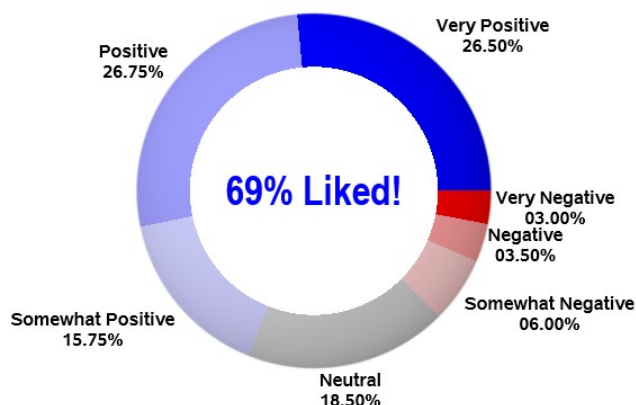
```
DONUT Opinion /
  RESPONSE=Percent SLICEORDER=RESPDESC
  OTHERPercent=0 STARTANGLE=0 STARTPOS=EDGE
  DIRECTION=COUNTERCLOCKWISE
  /* HOLEVALUEATTRS=() */
  /* HOLEVALUE=2020 displays as 2K */
  HOLELABEL='Eat Donuts'
  /* HOLELABEL=
    'National Doughnut Day, 7 June 2019'
    would be too big.
    There is no text wrap capability, and
    no SIZE= option to control font size. */
  HOLELABELATTRS=
    (FAMILY='Arial Narrow' WEIGHT=Bold)
  /* The only ATTRS supported are
    FAMILY, COLOR, STYLE, WEIGHT
    No control for SIZE. SIZE is adjusted by
    software to fit, if possible.
    Too long text is truncated or omitted.
    E.g. Enjoy Donuts would be truncated. */
  DATALABELDISPLAY=(CATEGORY RESPONSE)
  DATALABELLOC=OUTSIDE
  DATALABELATTRS=
    (FAMILY=ARIAL SIZE=12pt WEIGHT= Bold);
  FORMAT Percent PctFmt.;
RUN;

ODS LISTING CLOSE;
```

NOTE: If the categories in this example were not already Percents, and the DataLabels were specified with DATALABEL=(CATEGORY RESPONSE PERCENT), the Percents in the DataLabels would be rounded to tenths, with no way to override that format.

Given the clearly comparative nature of the categories for this data set, a different bite sequence seems natural. Here is the Order by Attitude version:

Survey Opinions About Doughnuts - Ordered By Attitude



Data Source: <https://aytm.com/blog/doughnuts-survey/>

Here is the code to create the chart above:

```
ODS RESULTS OFF;
ODS _ALL_ CLOSE;

ODS LISTING STYLE=LISTING
  GPATH="C:\VIEWS_Aug2020\Results";
```

```

DATA ToChart(DROP=RankByAttitude);
  LENGTH RankedOpinion $ 19;
  SET ForChart;
  RankedOpinion =
    PUT(RankByAttitude, 1.) || ':' || Opinion;
RUN;

DATA ToFormat;
  RETAIN FMTNAME 'OpinionFORMAT' TYPE 'C';
  SET ToChart(RENAME=(
    RankedOpinion=START
    Opinion=LABEL)
  );
RUN;

PROC FORMAT CNTLIN=ToFormat;
RUN;

PROC SORT DATA=ToChart(DROP=Opinion);
  BY RankedOpinion;
RUN;

ODS GRAPHICS / RESET=ALL NOSCALE
  WIDTH=6.5IN HEIGHT=4.875IN
  IMAGENAME='Donut_Chart_Order_By_Attitude';

TITLE1 FONT='Arial/Bold' HEIGHT=14pt
  'Survey Opinions About Doughnuts - Ordered By
  Attitude';
TITLE2 COLOR=White FONT='ARIAL/BOLD'
  HEIGHT=14pt
  'INVISIBLE TEXT: For desired white space.';
FOOTNOTE1 COLOR=White FONT='ARIAL/BOLD'
  HEIGHT=14pt
  'INVISIBLE TEXT: For desired white space.';
FOOTNOTE2 FONT='Arial' HEIGHT=14pt
  'Data Source: https://aytm.com/blog/doughnuts-
  survey/';

PROC SGPIE DATA=ToChart;
  STYLEATTRS DATACOLORS=(CX0000FF CX9999FF
    CXCCCCFF CXCCCCCX CXFFCCCC CXFF9999
    CXFF0000
  );
  DONUT RankedOpinion /
    RESPONSE=Percent SLICEORDER=UNFORMATTED
    OTHERPercent=0 STARTANGLE=0 STARTPOS=EDGE
    DIRECTION=COUNTERCLOCKWISE
    HOLELABEL=' 69% Liked!'
    HOLELABELATTRS=(FAMILY='Arial Narrow'
      COLOR=CX0000FF WEIGHT=Bold)
    DATALABELDISPLAY=(CATEGORY RESPONSE)
    DATALABELLOC=OUTSIDE
    DATALABELATTRS=
      (FAMILY=Arial SIZE=12pt WEIGHT=Bold);
  FORMAT RankedOpinion $OpinionFORMAT.;
  FORMAT Percent PctFmt.;
RUN;

ODS LISTING CLOSE;

```

A desirable, but not guaranteed by me, enhancement for the Production release of PROC SGPIE would be the option to create separator lines between the bites (and, in a Pie Chart, the slices). If using only intense colours, separators add no value, but with fainter colours I find that separators are comforting to the eye, at least to **my** eye.

LeRoy Bessler, Visual Data Insights™

Fancy Joining an Exclusive SAS Community?

I can't give an exact date, but SAS Analytics Explorers was created sometime before 16 October 2019, because the first SAS Analytics Explorers Programming Challenge in SAS Communities has that date:

"This programme is a VIP community designed to show our appreciation for your affinity and loyalty to SAS. We invite you to register to receive an e-mail invitation to join our exclusive network and link up with SAS members and other SAS users."

In fact there is not one community, but five geographically separate communities:

- Asia Pacific
- Canada
- Europe, Middle East & Africa (EMEA)
- Latin America
- US

In each community you can:

- *"JOIN THE NETWORK: Connect with peers, experts and industry leaders to gain new insights and share ideas."*
- *"EARN POINTS: Complete challenges of your choice - both fun and educational. Tap into fresh content and resources along the way."*
- *"HAVE FUN: Spark your competitive spirit. Indulge your curiosity. Earn rewards and watch your hard work pay off."*

I have been a member of the EMEA community since April 2020, so, if you live in that area and would like an invitation to join, please follow this link: <https://infl.tv/hKDr>. Otherwise please visit <https://communities.sas.com/t5/SAS-Analytics-Explorers/bg-p/programming-challenge>, follow the link to request an invitation, and select your own community there.

However, from personal experience, please do not expect an immediate response. I have found the EMEA community to act quickly in some situations, but not when referral invitations need to be processed.

Philip R Holland, Newsletter Editor

Ask The Consultant

This part of VIEWS News is where you can get your technical questions answered. Send your questions to the Editor.

Using ODS OUTPUT and ODS TRACE to capture SAS® output elements into MACRO Variables

Q: How do you transfer PROC output values into macro variables?

A:[SAS 9] The output from most SAS procedures are saved in ODS OUTPUT tables which can be captured and saved into SAS data sets and also generate MACRO variables. The example shown in this paper is "how to calculate a R-Square from a

regression model applied to a validation (or hold-out) data set?”. We then compare the R-Square results for both model development data and validation data, both captured using ODS TRACE and ODS OUTPUT. We then insert the macro variables using the INSET option in PROC SGPLOT.

Let’s take a simple example of building a regression model using the SASHELP.baseball data to predict the log of salary, in \$ Thousands, earned in 1987 (logSalary). Note that everyone should have access to the SASHELP table when using SAS. A simple model will be tested using these independent variables:

- Years in the Major Leagues (YrMajor)
- Career Home Runs (CrHome)
- Career Hits (CrHits)

Steps for this exercise are detailed below.

1. A Simple PROC TABULATE to identify missing values.

First, we will remove any records that have missing values just to make the exercise simple. We notice that the Log of Salary (logSalary) has 59 missing values in the TABULATE:

```
PROC TABULATE DATA=sashelp.baseball
    NOSEPS MISSING;
    VAR logSalary YrMajor CrHome CrHits;
    TABLE (logSalary YrMajor CrHome CrHits),
        N NMISS / RTS=20 ROW=float;
RUN;
```

Output:

	N	NMiss
Log Salary	263	59
Years in the Major Leagues	322	0
Career Home Runs	322	0
Career Hits	322	0

2. Let’s sample 75% of the non-missing data for the development sample and 25% for the hold out sample which will not be used in the model build.

To generate the sample data sets we are using PROC SURVEYSELECT. Code is shown here:

```
%LET dv=logSalary;
PROC SURVEYSELECT DATA=sashelp.baseball
    (WHERE=(MISSING(logSalary)=0))
    OUT=model_DS SEED=20200621
    SAMPRATE=0.75 OUTALL REP=1;
/* OUTALL - Retains all records. Those selected
get SELECTED=1. Not selected get SELECTED=0 */
RUN;
```

Output from PROC SURVEYSELECT:

Selection Method	Simple Random Sampling
Input Data Set	BASEBALL
Random Number Seed	20200621
Sampling Rate	0.75
Sample Size	198
Selection Probability	0.752852
Sampling Weight	0
Number of Replicates	1
Total Sample Size	198
Output Data Set	MODEL_DS

We generate 2 data sets; DEVELOPMENT (dev) and VALIDATION (val). We could have used the WHERE clauses in a single data set for subsequent steps, but let’s make it simple. Let’s generate the 2 data sets:

```
DATA dev val;
    SET model_DS;
    IF selected THEN OUTPUT dev;
    ELSE OUTPUT val;
RUN;
```

The dev sample records (selected=1) at 75% is sampled at 198. The remaining 65 records (selected=0) are loaded into the val data set. The dev data set will be used to build the model. Then the val data set will be scored with the model generated by the development model, the R-square for the val data will be calculated, and we then can compare performance with the dev data.

3. Generate a model using PROC GLMSELECT and capture the R-Square for the development model using ODS TRACE.

Default selection is a forward method that stops when the SBC criteria is at the minimum.

```
ODS TRACE ON / LABEL;
PROC GLMSELECT DATA=dev;
    MODEL &dv. = YrMajor|CrHome|CrHits @2;
RUN;
ODS TRACE OFF;
QUIT;
```

Each SAS procedures that store ODS OUTPUT tables may generate many tables so you need to take a look at the Log to identify which TABLE has the R-Square information to create the macro variable from the dev data. When I submit a batch job, I was able to get each table name in the LST file listed above each table using:

```
ODS TRACE ON / LISTING;
```

I am using SAS ON DEMAND for ACADEMICS and was not able to get the LISTING option to dump to the OUTPUT (something I should research). The code using ODS TRACE ON / LABEL will list out the name of each table in the Log output.

Log output generated 9 table names from the GLMSELECT code above shows all the tables. I selected the FitStatistics table. Log section:

Output Added:

```
-----
Name:      FitStatistics
Label:     Fit Statistics
Template:  Stat.GLMSELECT.FitStatistics
Path:      GLMSelect.SelectedModel.FitStatistics
Label Path: 'The GLMSelect Procedure'. 'Selected
Model'. 'Fit Statistics'
```

4. Rerun the GLMSELECT code using ODS OUTPUT statement to generate a SAS data set that has the R-Square.

Code:

```
ODS OUTPUT FitStatistics=fits;

PROC GLMSELECT DATA=dev;
  MODEL &dv. = YrMajor|CrHome|CrHits @2;
  OUTPUT OUT=model_resid RESID=residual
         PRED=predicted;
  STORE work.baseball_model;
RUN;

ODS OUTPUT CLOSE;
```

To confirm we have the correct table name and what information to pull to generate the macro variable for the R-Square for the development data we can use PROC CONTENTS and PRINT:

```
TITLE 'FIT TABLE INFO';

PROC CONTENTS DATA=fits;
RUN;

PROC PRINT NOOBS DATA=fits;
RUN;
```

The CONTENTS run showed 7 observations. The PROC PRINT confirmed that the R-Square is in the FitStatistics table with a result of 0.6926:

FIT TABLE INFO

Engine/Host Dependent Information	
Data Set Page Size	131072
Number of Data Set Pages	2
Number of Data Set Repairs	0
Host Created	Linux
Inode Number	936708
Access Permission	rw-r--r--
Owner Name	jonasbilenas0
File Size	384KB
File Size (bytes)	393216

Alphabetic List of Variables and Attributes			
	# Variable	Type	n
	1	Label1	Char
			#

From the code above we also included a STORE statement that can be used to model other data sets. Not all regression procedures have the STORE statement and is also dependent on what version of SAS STATS you are using. From the model STORE created we can score the HOLD OUT data using PROC PLM. There are other ways of scoring new data sets but I find the PLM procedure handy.

Some of the results from the GLMSELECT procedure is shown in the code below:

GLMSELECT

The GLMSELECT Procedure

Data Set	WORK.DEV
Dependent Variable	logSalary
Selection Method	Stepwise
Select Criterion	SBC
Stop Criterion	SBC
Effect Hierarchy Enforced	None

Number of Observations Read	198
Number of Observations Used	198

Dimensions	
Number of Effects	7
Number of Parameters	7

The GLMSELECT Procedure

Data Set	WORK.DEV
Dependent Variable	logSalary
Selection Method	Stepwise
Select Criterion	SBC
Stop Criterion	SBC
Effect Hierarchy	None

Number of Observations Read	198
Number of Observations Used	198

Dimensions	
Number of Effects	7

The GLMSELECT Procedure

Stepwise Selection Summary			
Effect Step Entered	Effect Removed	Number Effects In	SBC
0 Intercept		1	-43.9263

GLMSELECT

The GLMSELECT Procedure

Selected Model

Effects: Intercept CrHits YrMajor*CrHits

You can run the code to see the full GLMSELECT output. I am not a fan of stepwise models but this is just a quick example to test for this article on ODS TRACE and ODS OUTPUT. The resulting R-Square is 0.6926 for the DEV data and can be seen in a full output from GLMSELECT. We will next pull the R-Square variable from the FITS data and generate the macro variable, RSQ_DEV.

5. Create the MACRO variable, RSQ_DEV.

Code:

```
DATA _NULL_;
  SET fits;
  IF Label1='R-Square' THEN DO;
    CALL
      SYMPUTX('RSQ_DEV',ROUND(nValue1, 0.0001));
    STOP;
  END;
RUN;
%PUT RSQ_DEV = &RSQ_DEV.;
```

Log:

```
SYMBOLGEN: Macro variable RSQ_DEV resolves to
0.6926
157
158      %PUT RSQ_DEV = &RSQ_DEV.;
RSQ_DEV = 0.6926
```

6. How do we get the R-Square for the Hold Out sample to compare the R-Square for Development data?

Code to score the hold out records:

```
/* scoring the validation data */
PROC PLM RESTORE=work.baseball_model;
  SCORE DATA=val OUT=validation
    predicted / ILINK;
RUN;

DATA validation;
  SET validation;
  residual = &dv. - predicted;
/* to compare residuals */
RUN;
```

We can get an R-Square by using ODS OUTPUT with PROC CORR. the correlation between the dependent variable and the predicted variable, squared, is the resulting R-Square for the validation data set. You may need to look what TABLES have the correlation using the ODS TRACE, ODS OUTPUT, PROC CONTENTS, and PROC PRINT.

```
ODS OUTPUT PearsonCorr=stuff;

PROC CORR DATA=validation
  NOSIMPLE NOPROB PEARSON;
  VAR predicted;
  WITH &dv.;
  TITLE
    'Correlation between predicted and actual for
Validation data';
RUN;

ODS OUTPUT CLOSE;

DATA _NULL_;
  SET stuff;
  RSQ_VAL = ROUND(predicted**2, 0.0001);
  CALL SYMPUTX("RSQ_VAL",RSQ_VAL);
RUN;

%PUT RSQ_VAL = &RSQ_VAL.;
```

Log:

```
SYMBOLGEN: Macro variable RSQ_VAL resolves to
0.7129
195      %PUT RSQ_VAL = &RSQ_VAL.;
RSQ_VAL = 0.7129
```

7. Insert the R-Squares into distribution plots of residuals as an example of using the 2 macro variables.

```
PROC DATASETS LIB=work NOLIST;
  DELETE min_max_residual;
RUN; QUIT;

PROC APPEND BASE=min_max_residual
  DATA=model_resid (KEEP=residual)
  FORCE;

RUN; QUIT;

PROC APPEND BASE=min_max_residual
  DATA=validation (KEEP=residual)
  FORCE;

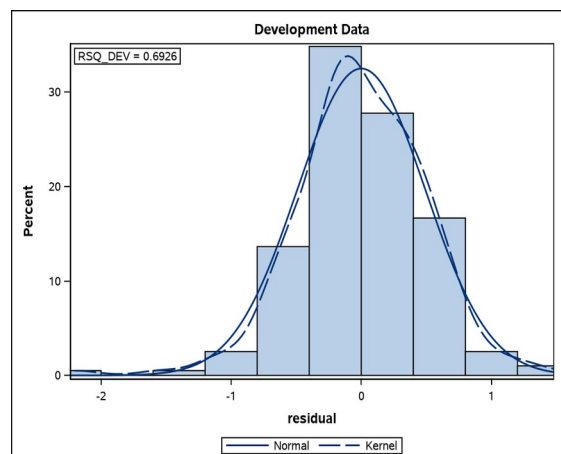
RUN; QUIT;

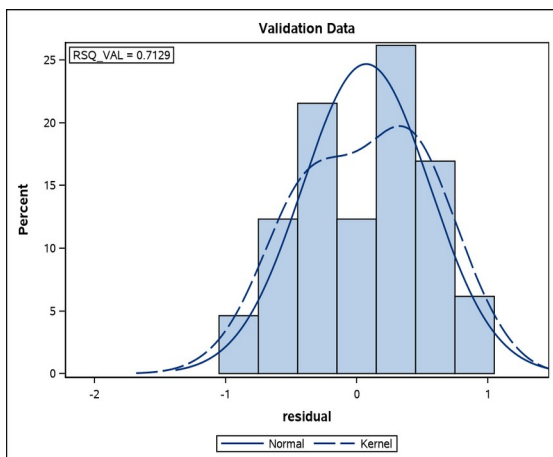
DATA _NULL_;
  SET min_max_residual END=eof;
  RETAIN min_resid max_resid 0;
  min_resid = residual >< min_resid;
  max_resid = residual <> max_resid;
  IF eof THEN DO;
    CALL SYMPUTX('min_resid', min_resid);
    CALL SYMPUTX('max_resid', max_resid);
  END;
RUN;

/* Compare DEV and VAL data residual plots */
PROC SGPLOT DATA=model_resid;
  HISTOGRAM residual;
  DENSITY residual / TYPE=NORMAL;
  DENSITY residual / TYPE=KERNEL;
  TITLE 'Development Data';
  XAXIS MIN=&min_resid. MAX=&max_resid.;
  INSET "RSQ_DEV = &RSQ_DEV." /
    POSITION=TOPLEFT BORDER;
RUN;

PROC SGPLOT DATA=validation;
  HISTOGRAM residual;
  DENSITY residual / TYPE=NORMAL;
  DENSITY residual / TYPE=KERNEL;
  TITLE 'Validation Data';
  XAXIS MIN=&min_resid. MAX=&max_resid.;
  INSET "RSQ_VAL = &RSQ_VAL." /
    POSITION=TOPLEFT BORDER;
RUN;
```

The above code used the DATA _NULL_ option to calculate the minimum and maximum residuals to align the XAXIS of both plots since I noticed the validation data had a better R-square. There may be an outlier in the DEVELOPMENT data. I could have used ODS OUTPUT and TRACE to capture the same results. Output is shown next.





Jonas V Bilenas, Freddie Mac

Annoying SAS Programming Styles: Part 2

Q: How do you remember where the program code came from? Most of the programs I inherit have very little documentation included.

A:[SAS] Although I have been using SAS since the early 1980s, I still see myself as a learner and enjoy using different approaches when writing new code. Often when I encounter a problem, rather than re-invent the wheel I will search the web for a solution. I used to keep a library of these solutions, but for the past few years have started incorporating weblinks directly into the comments of the code. This way, if I come across a problem that had been solved in a previous program, I can go back and find the solution together with the reference. An added advantage is that this makes the program more comprehensible to anyone who might look at it in the future (I hope), and gives credit where it is due. By way of illustration, here are six examples:

1. To produce reports that present data in a way that is easy to digest, but also enables users to do their own simple analyses, I have many programs that produce Excel files using the following ideas.

```

/*****
/**** Ideas taken from:
* Program Name: dks_sgf2018_pgm.sas *
* Author : Devi Sekar *
* Date : 02/20/2018 *
* Purpose : This program creates an excel
workbook with five sheets using the *
* ODS Excel destination and PROC REPORT. The
program uses cars *
* dataset from sashelp library. *
/****
https://www.sas.com/content/dam/SAS/support/en/s
as-global-forum-proceedings/2018/2479-2018.pdf
/****
/**** and 'Tips for Using the ODS Excel
Destination'
/****
https://blogs.sas.com/content/sgf/2017/02/20/tip
s-for-using-the-ods-excel-destination/
/****
/**** For Excel tab colours, see
/****
https://support.sas.com/rnd/base/ods/templateFAQ
/Template_colors.html
/*****/

```

2. To create frozen headers in Excel:

```

/*****/
/**** from https://documentation.sas.com/?
docsetId=odsug&docsetTarget=p0ydi4m5gper2on1mh15
trhsj3hb.htm&docsetVersion=9.4&locale=en#plei7g9
uxm8papnlmmt5z9nzkd9
/**** and https://communities.sas.com/t5/ODS-and-
Base-Reporting/Create-worksheets-with-different-
papersize-settings-using/td-p/171159
/*****/
ODS TAGSETS.EXCELXP FILE="bylines.xls"
STYLE=htmlBlue OPTIONS(SUPPRESS BYLINES='yes'
EMBEDDED_TITLES="yes" SHEET_LABEL="By"
FROZEN_HEADERS="yes");

ODS TAGSETS.EXCELXP CLOSE;
/*****/

```

3. To concatenate several files of similar structure:

```

/*****/
/**** CONCATENATE ****/
/****
https://stats.idre.ucla.edu/sas/modules/concater
ating-data-files-in-sas/ ****/
/*****/
DATA curr.eq5all_current_data;
SET curr.eq500m_current_data
curr.eq503m_current_data
curr.eq506m_current_data
curr.eq512m_current_data;
RUN;
/*****/

```

4. To summarise missing data (this example includes a note to myself):

```

/*****/
/**** Summarise the missing data, from:
/****
https://www.sas.com/content/dam/SAS/en_ca/User
%20Group%20Presentations/TASS/Zdeb-
MissingData.pdf
/**** see also:
/**** https://stats.idre.ucla.edu/sas/faq/how-
can-i-see-the-number-of-missing-values-and-
patterns-of-missing-values-in-a-data-file/
/*****/
PROC FORMAT;
/* Needs to be modified to incorporate coded
missing */
VALUE nm LOW-HIGH = 'X' OTHER = '.';
VALUE $ch ' ' = '.' OTHER = 'X';
RUN;

PROC FREQ DATA=curr.combine_current_data;
TABLE _ALL_ / LIST MISSING NOCUM;
FORMAT _NUMERIC_ nm. _CHARACTER_ $ch.;
RUN;
/*****/

```

5. Special characters:

```
/******  
ODS ESCAPECHAR='^';  
/* from https://www.quanticate.com/blog/4-sas-  
tips-in-clinical-programming */  
%LET sup3=%SYSFUNC(BYTE(179));  
/* superscript 3 for cubed */  
%LET micr=%SYSFUNC(BYTE(181));  
/* micro mu */  
/******
```

6. Count the number of distinct levels of a variable:

```
/******  
/* Use the NLEVELS option with the ODS SELECT  
statement to capture the */  
/* number of levels for a variable. */  
/* from https://support.sas.com/kb/36/898.html  
*/  
ODS SELECT NLEVELS;  
PROC FREQ DATA=tmp.tempsummary NLEVELS;  
TABLES Subject;  
TITLE  
'Number of distinct values for participant';  
RUN;  
/* Use PROC SQL to count the number of levels  
for a variable. */  
PROC SQL;  
CREATE TABLE new AS  
SELECT COUNT(DISTINCT(Subject)) AS Labelxcount  
FROM tmp.tempsummary;  
QUIT;  
/******
```

Norman Williams, University College London (UCL)

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions. Please read the SAS documentation to find out more about them.

[SAS] Believe or not the **MMDDYYw.** format and informat have never appeared in any previous issues, even though they have existed in SAS since before version 5! One reason, I suspect, is that programmers in the UK rarely use the format willingly, as it creates US-specific dates. However, the informat is probably more widely used when reading US-centric data.

[SAS] Another “ancient” omission has been the System Option **CENTER**, which, with its reverse **NOCENTER**, is probably used daily by all SAS programmers either deliberately or by default. Centring applies to the whole output, and not to the individual columns, so **NOCENTER** can be useful if the output is to be re-read into SAS or another application, as the text will always start at the left margin.

[SAS 8] **COALESCE()** was introduced in PROC SQL in SAS version 6 to select the first non-missing value in a list of arguments. In PROC SQL it can take either all numeric or all character variables and values. It was then introduced into Data Steps in SAS version 8, but there **COALESCE()** can only accept numeric variables and values. To process character variables and values you need to use the **COALESCEC()** function instead.

In Brief

- Back issues of *VIEWS News* are available from the VIEWS UK forum at **blog.hollandnumerics.org.uk**. Please register on the blog, and then request free access to the VIEWS UK forum.
- The SAS Professional Forum group on LinkedIn, which currently has over 34,800 members. Everyone is welcome to request to join, but discussions here are restricted to SAS-related topics: **www.linkedin.com/groups?gid=70702**.
- The SAS Opportunities group on LinkedIn, which currently has over 11,900 members. Everyone is welcome to request to join, but discussions here are also restricted to SAS-related topics: **www.linkedin.com/groups?gid=805217**.
- In the UK and Ireland the SUGUKI group had been holding regular meetings in London, and also occasional meetings in Bristol, Dublin, Edinburgh, Leeds and Manchester. Unfortunately lockdown has necessitated running webinars instead. For more details and to register for future events please go to **meetup.com/SUGUKI**.
- WPS Community Edition provides users with free access to WPS Analytics software including a SAS language compiler with coding and workflow tools on unlimited data volumes. A Community Edition licence key will be issued that will activate the software for up to six months. At the end of this term, users may be invited to renew their licence and to upgrade to the latest available version of the software. No technical support service is offered with the Community Edition. Download from **www.worldprogramming.com/try-or-buy/editions**. Note that the software is made available without warranty or indemnity, and is not made available to US residents.

Diary

Are you organising an event that would be of interest to the VIEWS readership? Let us know as we are interested in all non-profit making events related to SAS.

November 2020

8-11 PhUSE EU Connect Online

The Pharmaceutical Users Software Exchange (PhUSE) conference is now the premier programming event for the pharmaceutical industry in Europe. This will be a virtual conference. See **phuse.eu** for more details.

April 2021

19-22 Virtual SAS Global Forum

For more details about times and speakers please go to the SAS Global Forum web site at **www.sasglobalforum.org**.

May 2021

23-26 PharmaSUG US, Baltimore, MA, USA

The 2020 conference was cancelled due to COVID-19. See the PharmaSUG web site at **pharmasug.org** for further details.

August 2021

27-28 PharmaSUG China, Shanghai, China

The 2020 conference was cancelled due to COVID-19. See the PharmaSUG China web site at **pharmasug.org/china** for further details.

Contacts

Jonas V Bilenas,
Freddie Mac

Web: jonasbilenas.com,
www.linkedin.com/in/jonasbilenas/

LeRoy Bessler

Email: le_roy_bessler@wi.rr.com

Norman Williams,
University College
London (UCL)

Email: norman.williams@ucl.ac.uk
Web: www.linkedin.com/in/norman-williams-662977151/

SAS UK

Tel: +44-(0)1628-486933
Web: www.sas.com/uk

IEWS

Web Site

Web: blog.hollandnumerics.org.uk

Newsletter Editor

Email:
views-uk@hollandnumerics.org.uk

***IEWS News** is published quarterly for the IEWS International SAS Programmer Community by Holland Numerics Limited.*