

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information
for SAS® Users

Issue 60, 2nd Quarter 2020

This newsletter issue contains an article from LeRoy Bessler about Waterfall Plots, a history of SAS and himself from Anders Skölleremo, and a plea for action from me about annoying SAS programming styles. I have again included some SAS-related events publicised for the next few months, and I've also added some more SAS formats, options and functions that you may not have come across before relating to character sets other than UK and US, 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?

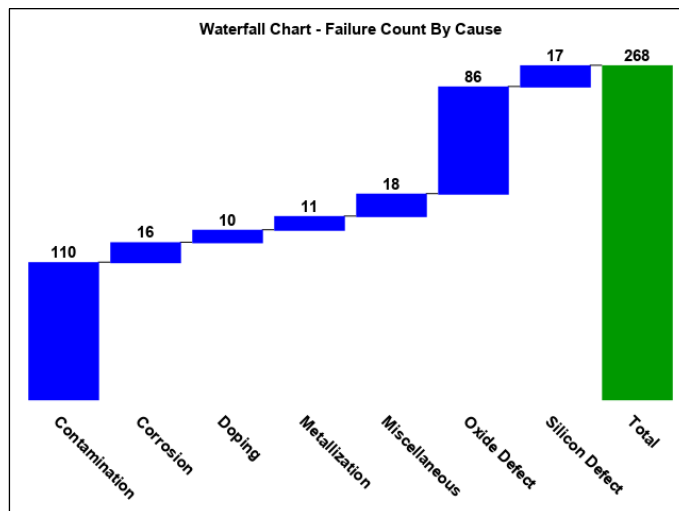
If you have a useful hint or tip, please send it to the Editor, and share it with the VIEWS membership.

The ODS Graphics Waterfall Chart: Improvement, An Alternative, & Real Purpose

For the initial set of examples I used the SASHELP.FAILURE data set which is used in the "About Waterfall Charts" section of the SAS 9.4 ODS Graphics: Procedures Guide, Sixth Edition. After I worked through the first three examples, I learned that this kind of data set is not really the kind of data for which the Waterfall Chart was originally designed, as I will explain later.

All of the examples shown here were created with the ODS Graphics default width of 640 pixels and default height of 480 pixels. These results were shrunk to 55% of their original size upon insertion into this article.

The ODS Graphics Waterfall Chart default is to present the categories in alphabetic order. Making the Total bar a different color was optional, but is often done to emphasize its difference in character from the other bar categories.



The chart was created with this code:

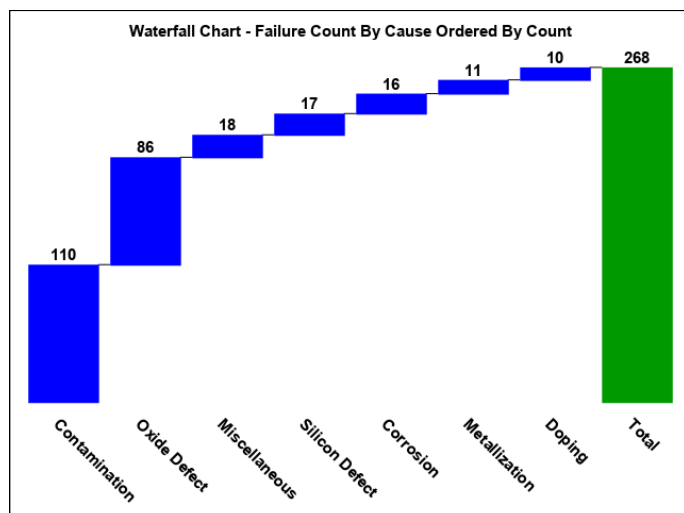
```
ODS RESULTS=OFF;
ODS _ALL_ CLOSE;

ODS LISTING STYLE=LISTING
  GPATH="C:\VIEWS 2020Q2\Results";
ODS GRAPHICS ON / RESET=ALL SCALE=OFF
  /* width=640px height=480px
   * accepting these defaults */
  IMAGENAME="WaterfallChart";
TITLE FONT='Arial/Bold'
  "Waterfall Chart - Failure Count By Cause";

PROC SGPLOT DATA=SASHELP.FAILURE NOBORDER;
  WATERFALL CATEGORY=Cause RESPONSE=Count /
  STAT=SUM NOOUTLINE DATALABEL
  DATALABELATTRS=(FAMILY=ARIAL SIZE=11pt
    /* match the X-axis mark values
     * 7pt is the default */
    WEIGHT=BOLD)
  BASELINEATTRS=(COLOR=WHITE)
  /* DISPLAYBASELINE=OFF is not a
   * Waterfall Chart option */
  FILLATTRS=(COLOR=CX0000FF)
  FINALBARATTRS=(COLOR=CX009900)
  FINALBARTICKVALUE="Total";
XAXIS DISPLAY=(NOLABEL NOLINE NOTICKS)
  VALUEATTRS=(FAMILY=ARIAL SIZE=11pt
    /* 9pt is the default, but it
     * works as well, as 8pt and 7pt
     * also produce titled values,
     * as does 11pt here.
     * 6pt provides horizontal values
     * that are unreadable. */
    WEIGHT=BOLD);
YAXIS DISPLAY=NONE;
RUN;

ODS LISTING CLOSE;
```

Below is the Ranked Waterfall Chart which implements my long-time design principle: “Show them what is important”.



Here is the pre-processing needed to create the chart:

```
PROC SUMMARY DATA=SASHELP.FAILURE NWAY;
  CLASS Cause;
  VAR Count;
  OUTPUT OUT=TotalCountSByCause SUM=;
RUN;

PROC SORT DATA=TotalCountSByCause;
  BY DESCENDING Count;
RUN;

DATA Ranks (DROP=Count);
  LENGTH Rank 8.;
  SET TotalCountSByCause;
  Rank = COMPRESS(_N_);
RUN;

PROC SORT DATA=Ranks;
  BY Cause;
RUN;

PROC SORT DATA=SASHELP.FAILURE
  OUT=FailureCountsByCause;
  BY Cause;
RUN;

DATA MaxLength_Cause;
  RETAIN MaxLength_Cause 0;
  SET TotalCountSByCause END=LastOne;
  MaxLength_Cause = MAX(MaxLength_Cause,
    LENGTH(TRIM(LEFT(Cause))));
  IF LastOne;
  PUT MaxLength_Cause=;
  CALL SYMPUT('MaxLength_Cause',
    MaxLength_Cause);
RUN;

DATA MaxLength_Rank;
  RETAIN MaxLength_Rank 0;
  SET Ranks END=LastOne;
  MaxLength_Rank = MAX(MaxLength_Rank,
    LENGTH(COMPRESS(Rank)));
  IF LastOne;
  CALL SYMPUT('FormatForRank',
    'Z' || COMPRESS(MaxLength_Rank) || '.');
  CALL SYMPUT('LengthOfRankedCause',
    MaxLength_Rank + &MaxLength_Cause + 1);
RUN;
```

```
DATA ToChart (DROP=Rank);
  LENGTH RankedCause $ &LengthOfRankedCause.;
  MERGE FailureCountsByCause Ranks;
  BY Cause;
  RankedCause =
    PUT(Rank, &FormatForRank.) || ':' || Cause;
RUN;

PROC SORT DATA=ToChart OUT=Causes NODUPKEY;
  BY Cause;
RUN;

DATA ToFormat;
  RETAIN FMTNAME 'CauseFormat' TYPE 'C';
  SET Causes
    (RENAME=(RankedCause=START Cause=LABEL));
RUN;

PROC FORMAT CNTLIN=ToFormat;
RUN;

PROC SORT DATA=ToChart (DROP=Cause);
  BY RankedCause;
RUN;
```

The chart was created with this ODS Graphics code:

```
ODS RESULTS=OFF;
ODS _ALL_ CLOSE;

ODS LISTING STYLE=LISTING
  GPATH="C:\VIEWS_2020Q2\Results";
ODS GRAPHICS ON / RESET=ALL SCALE=OFF
  IMAGENAME="RankedWaterfallChart";
TITLE FONT='Arial/Bold'
  "Waterfall Chart - Failure Count By Cause
  Ordered By Count";

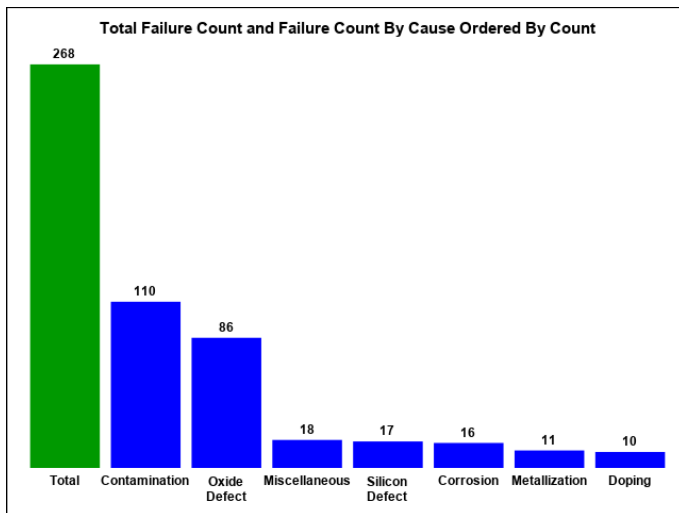
PROC SGPLOT DATA=ToChart NOBORDER;
  WATERFALL CATEGORY=RankedCause
    RESPONSE=Count /
  STAT=SUM NOOUTLINE DATALABEL
  DATALABELATTRS=(FAMILY=ARIAL SIZE=11pt
    WEIGHT=BOLD)
  BASELINEATTRS=(COLOR=WHITE)
  FILLATTRS=(COLOR=CX0000FF)
  FINALBARATTRS=(COLOR=CX009900)
  FINALBARTICKVALUE="Total";
  XAXIS DISPLAY=(NOLABEL NOLINE NOTICKS)
  VALUEATTRS=(FAMILY=ARIAL SIZE=11pt
    WEIGHT=BOLD);
  YAXIS DISPLAY=NONE;
  FORMAT RankedCause $CauseFormat.;
RUN;

ODS LISTING CLOSE;
```

This is much more code than I would like to be necessary to achieve this important design objective: ranked order for the bars. If you can find a simpler way to do this, please share your solution in a future *VIEWS* Newsletter, or email it to me.

My easy alternative to the Ranked Waterfall Chart is a Ranked Vertical Bar Chart with a Total Bar. To get a green Total bar in emulation of the Waterfall Chart design requires use of an ODS Graphics attribute map. If you give up that requirement, creating this chart is even simpler.

Here, the SGPlot procedure VBAR statement processing decided to line-break multi-word X-axis values, and the 9pt font was able to deliver horizontal text. The Waterfall Chart processing refused to line-break the multi-word values, and can deliver horizontal values only at 6pt or smaller for this data set.



I can see no advantage in a Ranked Waterfall Chart over this chart. This solution also delivers the Total Bar and, unlike the Waterfall Chart, makes reliable visual comparison of the constituent bars easy since they are side by side, rather than running away from each other up the waterfall. Why is it called a Waterfall Chart?

The chart was created with this code:

```
PROC SQL NOPRINT;
    SELECT SUM(Count) INTO :TOTAL
    FROM SASHELP.FAILURE;
QUIT;

DATA ToChart;
    SET SASHELP.FAILURE END=LastOne;
    OUTPUT;
    IF LastOne;
    Cause='Total';
    Count = &TOTAL.;
    OUTPUT;
RUN;

PROC SORT DATA=ToChart OUT=ForAttrMap NODUPKEY;
    BY CAUSE;
RUN;

DATA CauseAttrMap;
    KEEP ID Value FillColor;
    SET ForAttrMap;
    LENGTH ID $ 5 Value $ 16 FillColor $ 8;
    RETAIN ID 'Cause';
    IF Cause EQ 'Total'
    THEN FILLCOLOR='CX009900';
    ELSE FILLCOLOR='CX0000FF';
    Value = Cause;
RUN;

ODS RESULTS OFF;
ODS _ALL_ CLOSE;

ODS LISTING STYLE=LISTING
    GPATH="C:\VIEWS_2020Q2\Results";
ODS GRAPHICS ON / RESET=ALL SCALE=OFF
    IMAGENAME="RankedVerticalBarChartWithTotalBar";
TITLE FONT='Arial/Bold'
    "Total Failure Count and Failure Count By
    Cause Ordered By Count";

PROC SGPLOT DATA=ToChart NOBORDER
    DATTRMAP=CauseAttrMap NOAUTOLEGEND;
    VBAR Cause /
    RESPONSE=Count CATEGORYORDER=RESPDESC
```

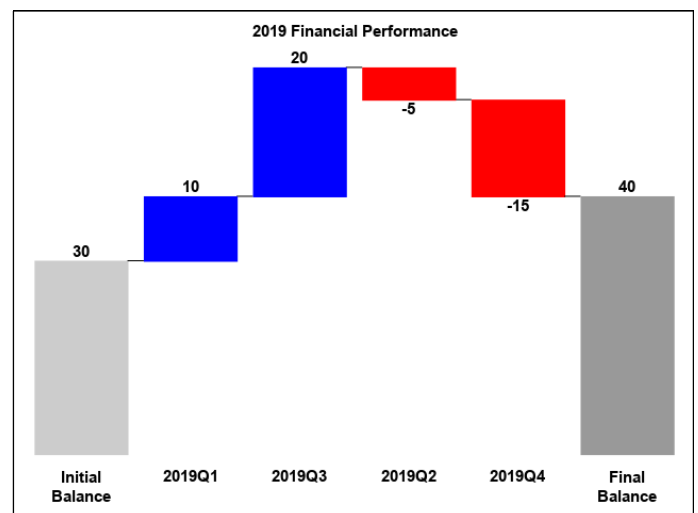
```
STAT=SUM GROUP=Cause ATTRID=Cause
DATALABEL
DATALABELATTRS=(FAMILY=ARIAL SIZE=9pt
/* match the tick mark values
7pt is the default */
WEIGHT=BOLD)

DISPLAYBASELINE=OFF
NOOUTLINE;
XAXIS DISPLAY=(NOLABEL NOLINE NOTICKS)
VALUEATTRS=(FAMILY=ARIAL SIZE=9pt
/* this is the default */
WEIGHT=BOLD);
YAXIS DISPLAY=NONE;
RUN;

ODS LISTING CLOSE;
```

After having made this journey exploring the Waterfall Chart and alternatives, I was curious how it originated. So, I searched Google for “who invented the Waterfall Chart”. At https://en.wikipedia.org/wiki/Waterfall_chart I learned this: “A waterfall chart is a form of data visualization that helps in understanding the cumulative effect of sequentially introduced positive or negative values. These intermediate values can either be time based or category based.”. In other words, the Waterfall Chart is intended to show the evolution of one measure, not to display a collection of categories with all positive response values, as when using the data set above.

So I made up some data. In a Waterfall Chart you can customize the initial bar, final bar, both bars, or neither. Here is what I call a Temporal Change Waterfall Chart:



The WATERFALL statement processing was willing to line-break the two-word X-axis values for this chart, but was unwilling to do so for two-word X-axis values in the first Waterfall Chart. The decision probably depends on the lengths of other values present for the X axis.

Colour-coding the intermediate bars for positive or negative RESPONSE values is optional, but helpful. It is easier to detect than a minus sign, and the signal is immediate.

The chart was created with this code:

```
DATA DataInput;
  LENGTH INorOUT $ 14 PlurOrMinusDollars 3;
  INorOUT = '2019Q1';
  PlurOrMinusDollars = 10;
  OUTPUT;
  INorOUT = '2019Q3';
  PlurOrMinusDollars = 20;
  OUTPUT;
  INorOUT = '2019Q2';
  PlurOrMinusDollars = -5;
  OUTPUT;
  INorOUT = '2019Q4';
  PlurOrMinusDollars = -15;
  OUTPUT;
RUN;

DATA ToChart;
  LENGTH MinusOneOrPlusOne 3;
  SET DataInput;
  IF PlurOrMinusDollars LT 0
    THEN MinusOneOrPlusOne = '-1';
    ELSE MinusOneOrPlusOne = '+1';
RUN;

ODS RESULTS=OFF;
ODS _ALL_ CLOSE;

ODS LISTING STYLE=LISTING
  GPATH="C:\VIEWS_2020Q2\Results";
ODS GRAPHICS ON / RESET=ALL SCALE=OFF
  IMAGENAME="TemporalChangeWaterfallChart";
TITLE FONT='Arial/Bold'
  /* HEIGHT=11pt is the default */
  "2019 Financial Performance";

PROC SGPLOT DATA=ToChart NOBORDER NOAUTOLEGEND;
  WATERFALL CATEGORY=INorOUT
    RESPONSE=PlurOrMinusDollars /
  COLORRESPONSE=MinusOneOrPlusOne
  COLORMODEL=(CXFF0000 CX0000FF)
  /* Red and Blue, NOT Red and Green, the
    regrettably popular "traffic lighting",
    indistinguishable for the commonest form
    of color blindness */
  STAT=SUM DATALABEL
  DATALABELATTRS=(FAMILY=Arial SIZE=11pt
    WEIGHT=Bold)
  BASELINEATTRS=(COLOR=WHITE) NOOUTLINE
  INITIALBARATTRS=(COLOR=CXCCCCC)
  INITIALBARTICKVALUE="Initial Balance"
  INITIALBARVALUE=30
  FINALBARATTRS=(COLOR=CX999999)
  FINALBARTICKVALUE="Final Balance";
  XAXIS DISPLAY=(NOLABEL NOLINE NOTICKS)
  VALUEATTRS=(FAMILY=Arial SIZE=11pt
    WEIGHT=Bold);
  YAXIS DISPLAY=NONE;
RUN;

ODS LISTING CLOSE;
```

LeRoy Bessler, Visual Data Insights™

A kind of answer to the question: “Who is Anders Skölleremo?”

When I was 21 years old, I met a funny girl, with a funny name “Skölleremo”. You get used to the name, she said, when I asked her about her name. She was laughing a lot. I did like that, and needed it. I got used to both her, and to the funny name. We got

married a few years later. It also turned out that my name “Anders Nilsson” was far too common. I got mixed up with other persons, like John Smith in England. So I changed my last name into Skölleremo. No problem. I really liked to have a unique name. “Skölleremo of Sweden”.

I have one Bachelor Degree in Physics, and (nowadays) almost two in Applied Mathematics. At the University of Uppsala I got a Doctors Degree in Atomic and Molecular Physics. I did not like the experimental work, but rather wanted to do computer calculations, to try to improve the instruments. About 15-20 man years were used for every new generation of the instruments. My work was rather “Applied Mathematics or Computer Science, with Applications”.

I wrote a paper, which was described as “Solving an elliptic boundary value problem, as an initial hyperbolic value problem”. If you think that you understand that, you are quite wrong. But it was great and worked and was very safe. Even my wife’s professor in Applied Mathematics Professor Kreiss liked it, after a rather “loud” discussion with me for more than half an hour. He was really a great guy. I did like him a lot.

To work in “Social Democratic” Sweden was not very easy. Especially if you are a programmer who loves to document and to understand what you are doing. I found lots and lots of really serious errors in computer programs and computer environments. I was not blamed for any mistakes, but several companies almost filed for bankruptcy.

My first job after University was entirely hopeless. However I did have a funny and really great experience. I, the theoretical PhD and programmer, should help a user who had really a minimum of school education. To everyone’s surprise we got on extremely well. We almost spoke the same language from the start. I learned a lot from him, including some useful “bad words in Swedish”.

I started to use SAS in 1981. SAS was great to work with. I was soon a kind of expert. I attended the SEUGI meeting 1984 in Amsterdam. Jim Goodnight was there. It was really great. I told Jim “I had to make my name tag using Disspla from ISSCO. SAS/GRAPH does not write the name correctly.”. He said “We have to fix that”. Great.



Jim Goodnight and Anders Skölleremo in Hamburg at SEUGI 1997

I did like SAS very much. But I wanted SAS to be much more strict, in language, documentation, version handling, testing, etc. At my first SAS job, they had not understood the

need to backup data. The result was a “perhaps bankruptcy” situation. But I solved that. I did a lot of SAS teaching and problem solving for my users, at work and as a consultant. This gave me much input to suggestions and requests to SAS Institute. When SAS asked for user input in SASWare Ballot it was great fun to provide opinions and feedback.

The arrival of PC and PC SAS was great. But PCs are ASCII and mainframes are EBCDIC. So the problem of getting the Swedish National Characters (ÅÄÖåäö) down to the PC in the right way was a challenge. I wrote several papers about this. See SUGS, SEUGI and later SUGI Proceedings.

But we used OS/2. The backslash character was interpreted in the wrong way, so tables were downloaded to some area whose name you had never heard of before. But I had access to the code for the patch (= editing in the mainframe load module) from SAS Institute that should correct the NLS. So I modified the patch. My third attempt was OK.

I sent the modified patch to the Swedish SAS Institute, with the letter. “This patch solves the backslash problem. It has been thoroughly tested and documented.”. But somehow, I was not always so popular at the local Swedish office.

On mainframe SAS, tables are stored together like in a wardrobe. From outside you only see the name and size of the wardrobe, not what tables it contains. That was a common question from many users. So I wrote a small command (SASTAB) for IBM mainframe, which runs SAS in non-interactive mode, non-DMS mode, and displays the needed information. This command was very popular on many installations in Stockholm.

Besides you want to sort your data according to NLS on mainframe and PC, and later also to use BY variables that contain NLS characters. You have sort of “ghost situation”. If you do it the wrong way you get SAS programs that will work some days, and perhaps not other days, depending on exactly where in your data that you have the NLS characters. If you are obstinate (I sometimes REALLY am!), then you try to solve all the problems around NLS. So I wrote a paper in SAS Observations about this problems.

In one SAS version on Windows, the keyboard characters were not translated in the correct way to ASCII characters, but I solved that too. Printing was important to my customers. SAS Institute had not understood that. The issue is to display the information in the SAS page on printed paper page, without any losses. The solution was described in my poster “The SAS page and all the other pages you have to consider” at a SEUGI meeting. I did like the title! Problem solved.

Was I working with the SAS System? Was I a SAS specialist? Both Yes and No. Rather a “Problem Solver” in a computer environment with SAS. A lot of the extra work was made in paid overtime and my private extra hours, and as an extra paid consultant. But it was fun. Jim Goodnight and Art Cooke and Phil Winters and Paul Kent and many others (Wallace and Wallace) did like my long lists of new facilities and functions that I wanted. It took several years, but I think that most of all my requests have been implemented.

SAS Institute did not seem to test the software in a good way. We had a problem with SAS/CONNECT. It was like “You buy goods at the supermarket (IBM mainframe) and put it in a big lorry. But the lorry does not go away. It waits until it is full. Finally, when you have bought so many goods that the lorry is full, then the lorry goes away. So you get, at your home (PC)

perhaps five times more that you need.”. The change of the default Block size solved all that. The bug report took me several hours to write, to get it exactly right.

At a “beer party” at a SEUGI meeting, I said that discussions about the SAS software was very important, like the discussions at a British inn. Some persons had been discussing Merge and the IN option and problems with it, so they thought that I was talking about that. In my optimistic mode, perhaps inspired by the beer, I promised to write about MERGE and the IN option. I had used the MERGE statement several times, sometimes with strange results. It took me some time. The result was some papers at SUGS and SEUGI Meetings, and finally a paper in SAS Observations “The Importance of the IN”. I did like the title!

I think that this was the very first correct documentation of MERGE and the IN data set option, when used to achieve many-to-many merge. I made the basic work during Easter holiday in Copenhagen, where I was working as a SAS consultant. On several occasions I just had to lay down, doing nothing, thinking nothing. My head was overheated. But the end result was good.

During my holidays I used to walk in the Swedish mountains. Names like Sarek and Akka (“Grandmother”) were very natural to me. I liked it a lot! Good for me. You can trust people up there, most of the time. The problem was that I always had to leave the civilized world in the mountains, to return to the sometimes very uncivilized world in Stockholm and all the bears, wolves and snakes.

I used to walk with my father. One year he was getting tired from age and long walks perhaps. He fell down in a crevasse on a glacier. I got him up, thanks to British mountaineers, Roger Mason and Wendy Kirk from the Geology Department of London University, who were in the neighbourhood. I wrote a story “A beautiful day on the glacier....” about that, and published it.

Life as a SAS programmer was tough. My job was really to enable other persons to work in a better way. Some of my bosses did understand that, after some discussions. For a while I took a job as a Swedish actuary, studied some courses in Life Probability – about this life, the sick life and perhaps also the next life. And also quite a lot of ordinary Mathematical Statistics. To transform the really big registers of life insurance takers into SAS tables, I wrote very complex and big SAS programs. Initial checking verified that the selection criteria were correct. A Data Step of 100 pages code, including a lot of documentation of course. But extremely complex.

Some years later I worked with the analysis of IBM mainframe and downfalls. Wrote a paper: “An analysis of chained independent stochastic variables - median and standard deviation”. I showed it to my boss. Did he understand any of it? I saw the paper the other day.

On my last job, when I was 65+, I should verify some very interesting MS results, obtained by a medical student. He and his professor wanted to publish the results, obtained using the menu system in SPSS. Well, I am a strict, slightly conservative SAS programmer. Quite a lot of code was needed to correct the input data. Finally, at last, the statistical analysis only showed very ordinary results. “If you get sick early in life, you get the treatment that is available at that time. If you get sick later in life, you can get more modern medicine and better treatment.”. Three months programming to obtain these results, and nothing to publish.

I am retired now. Life is quite good. Both my age and my IQ are 72+. Perhaps I will write a paper about Merge, trying to explain what Merge really does. I call this XMERGE, an extended version of Merge. I think about it now in Corona time, when I do my daily one hour walk, almost alone.

I write a lot about my walks in the Swedish mountains. Not every walk (56 in total) was without (almost) accidents. It all worked out well, but afterwards the thoughts and dreams come to you. I have many, many photos from all my walks, which I use when I try to paint.

Anders Skölleremo, Retired

Ask The Consultant

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

Annoying SAS Programming Styles: Part 1

Q: What do you find annoying when updating SAS programs written by other programmers?

A:[SAS 9.4] I work in the pharmaceutical sector where program reuse is the norm, so most programmers will at some time have to update existing SAS code. What they find is a myriad of different programming styles, some easy to read and update and some less so. This answer will only touch on some of the problems I have found.

The first style that affects code readability is their use, or rather random or no use, of indentation within Data Steps and DO END clauses. Having no consistent indenting of statements can hide the boundaries of related programming statements, which will hinder any attempts to debug the processing.

The second style, introducing unnecessary whitespace into SAS statements, can also affect code readability. My eyes are automatically drawn to double blanks, but I suspect other programmers may not be affected by them at all.

The third style, it seems to me anyway, is probably just laziness. I always close a processing step, whether a Data Step or SAS procedure, with a RUN or QUIT statement. However, I've noticed recently that more and more RUN statements are being omitted between Data Steps and PROC SORT steps, as if the programmer really wanted the code to be some form of SQL SELECT and ORDER BY construct. Unfortunately, a different programmer updating such a program may not realise the link between the adjacent steps, and add another step in-between that breaks that association!

Oddly enough an almost complete solution to all 3 annoying styles is now readily available to many of these programmers. SAS Enterprise Guide and SAS Studio (which is also in SAS University Edition) both include Code Formatting shortcuts that will tidy up at least some of these annoying styles.

For example, here is some annoying SAS code:

```
data    fred ; set    sashelp.class ;
do      i=1    to    10; output ; end;
proc sort;  by    height i ;
run;
```

Until SAS Enterprise Guide version 7.1 using [Ctrl+I], and from version 8.1 using [Ctrl+Shift+B] or clicking the icon , the code above will be reformatted to:

```
data    fred;
set      sashelp.class;
do      i=1    to    10;
    output;
end;

proc sort;
    by    height i;
run;
```

This fixes the first style, and highlights the issue in the last style, so you can see more easily where the Data Step ends and the PROC SORT begins and the possibility of a missing SAS statement there, but the extra whitespace from the second style is only removed at the end of the SAS statements.

From SAS Studio version 3.71 clicking the icon  to Format Code will generate the code below:

```
data fred;
  set sashelp.class;

  do i=1 to 10;
    output;
  end;

proc sort;
  by height i;
run;
```

This fixes the first and second styles, and highlights the issue in the last style, so you can see more easily where the Data Step ends and the PROC SORT begins and the possibility of a missing SAS statement there. Therefore I think I prefer SAS Studio for this!

You may have noticed that I've entitled this article as Part 1. This is not because I have more annoying programming styles to discuss (although I do!), but to give the newsletter readers an opportunity to contribute some of their own discoveries and solutions in future issues of *VIEWS News*. So I am looking forward to including more "annoying" stories!

Philip R Holland (Newsletter Editor)

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 9.2] The following informats now support Arabic text. **\$LOGVSw**, which reads a character string that is in left-to-right logical order and then converts it to visual order, **\$LOGVSRw**, which reads a character string that is in right-to-left logical order and then converts it to visual order, **\$VSLOGw**, which reads a character string that is in visual order and then converts it to left-to-right logical order, and **\$VSLOGRw**, which reads a character string that is in visual order and then converts it to right-to-left logical order.

[SAS 9.1] The **ENCODING=** Data Step option overrides the encoding to use for reading or writing a SAS data set. When used with **FILENAME**, **%INCLUDE**, **EMAIL**, **FILE**, **INFILE**

and ODS statements this file option overrides and transcodes the encoding for input or output processing of external files. It can also be used as **-ENCODING** in SAS configuration files to specify the default encoding for the SAS session.

[SAS 9.2] The function **KPROPDATA()** converts an input data string to the current SAS session, or specified, encoding, removing or replacing unprintable characters.

In Brief

- Back issues of *VIEWES News* are available from the VIEWES UK forum at **blog.hollandnumerics.org.uk**. Please register on the blog, and then request free access to the VIEWES UK forum.
- The SAS Professional Forum group on LinkedIn, which currently has over 34,200 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 11,350 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 has 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.

Diary

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

June 2020

16 Virtual SAS Global Forum

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

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 SAS Global Forum, Las Vegas, NV, USA

For more details 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

Anders Skölleremo

Email: **anders.skolleremo@one.se**
Web: **www.linkedin.com/in/anders-skolleremo-a497964/**

LeRoy Bessler

Email: **le_roy_bessler@wi.rr.com**

SAS UK

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

VIEWES

Web Site

Web: **blog.hollandnumerics.org.uk**

Newsletter Editor

Email: **views-uk@hollandnumerics.org.uk**

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