

VIEWS

International SAS Programmer Community



News, Hints, Tips and Information for SAS® Users

Issue 56, 2nd Quarter 2019

It is now 2019, and how the SAS world has changed, or has it? This newsletter issue actually contains articles from Sanjay Matange about ODS Graphics, LeRoy Bessler about SAS/GRAPH, and Murphy Choy and Michelle L Cheong about SAS/IML and JMP, which were submitted several years ago, but I still believe are relevant. I've also added some more formats, options and functions that you 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.

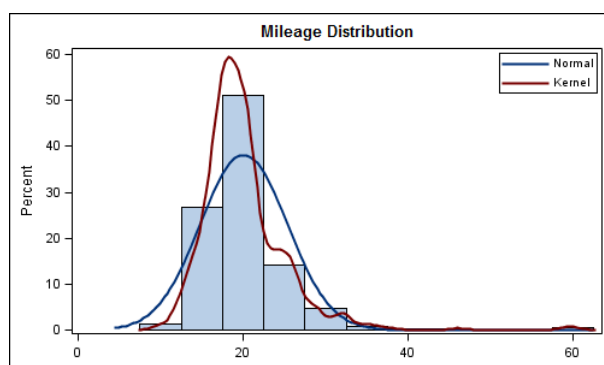
Ask The Consultant

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

Comparative density plots

Q: How do you plot the normal density curves for two classification levels in the same graph?

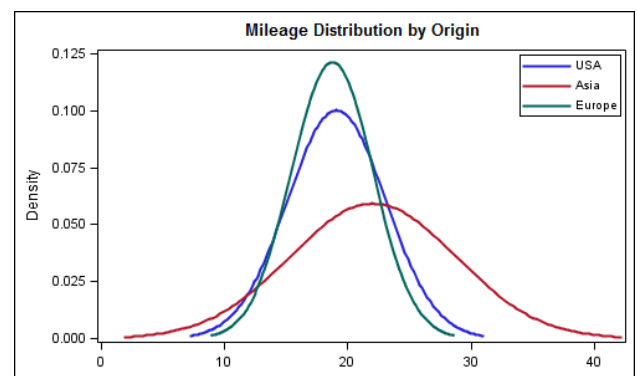
A:[SAS 9.3] We have often seen examples of a distribution plot of one variable using a histogram with normal and kernel density curves. Here is a simple example:



```
TITLE 'Mileage Distribution';  
PROC SGPLOT DATA = sashelp.cars;  
  HISTOGRAM mpg_city;  
  DENSITY mpg_city /  
    TYPE = NORMAL LEGENDLABEL = 'Normal'  
    LINEATTRS = (PATTERN=SOLID);  
  DENSITY mpg_city /  
    TYPE = KERNEL LEGENDLABEL = 'Kernel'  
    LINEATTRS = (PATTERN=SOLID);  
  KEYLEGEND / LOCATION = INSIDE  
    POSITION = TOPRIGHT ACROSS = 1;  
  XAXIS DISPLAY = (NOLABEL);  
run;
```

To compare the distribution by a classifier in the same graph, you can do something similar as long as the classified data is transformed into a multi-column format. Now, you can overlay two (or more) DENSITY curves of different variables in the same way.

In the example below, we have transformed the data for **sashelp.cars** into a multi-column format using the code suggested by Rick Wicklin in his article [Reshape data so that each category becomes a variable](#). The values of **mpg_city** for the three levels of the **origin** variable are transformed into three independent columns. Then, we have used three DENSITY statements to plot the data in one graph. Here is the graph and the code snippet. Full program is included at the bottom.



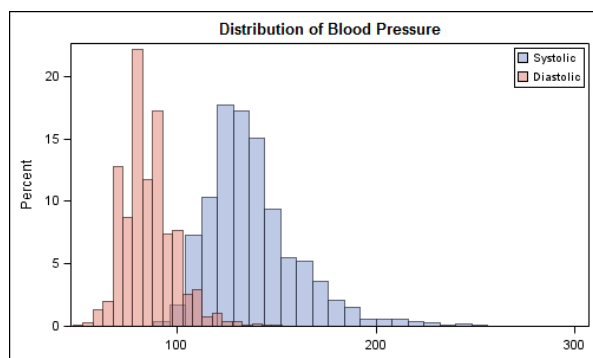
```
/*-Transpose grouped data into multi-variable-*/  
DATA multiVar;  
  KEEP mpg_usa mpg_asia mpg_eur;  
  MERGE sashelp.cars  
    (WHERE = (origin="USA")  
      RENAME = (mpg_city=mpg_usa))  
    sashelp.cars  
    (WHERE = (origin="Asia")  
      RENAME = (mpg_city=mpg_asia))  
    sashelp.cars  
    (WHERE = (origin="Europe")  
      RENAME = (mpg_city=mpg_eur));  
RUN;
```

```

TITLE 'Mileage Distribution by Origin';
PROC SGPLOT DATA = multiVar;
  DENSITY mpg_usa /
    LEGENDLABEL = 'USA'
    LINEATTRS = (PATTERN=SOLID);
  DENSITY mpg_asia /
    LEGENDLABEL = 'Asia'
    LINEATTRS = (PATTERN=SOLID);
  DENSITY mpg_eur /
    LEGENDLABEL = 'Europe'
    LINEATTRS = (PATTERN=SOLID);
  KEYLEGEND / LOCATION = INSIDE
    POSITION = TOPRIGHT ACROSS = 1;
  XAXIS DISPLAY = (NOLABEL);
RUN;

```

We can take this idea further, and create a plot to see the distribution of multiple variables on the same graph using HISTOGRAM and/or DENSITY plots. Here is an example of systolic and diastolic blood pressure from **sashelp.heart**. We have set a TRANSPARENCY level for each plot to be able to see the data:

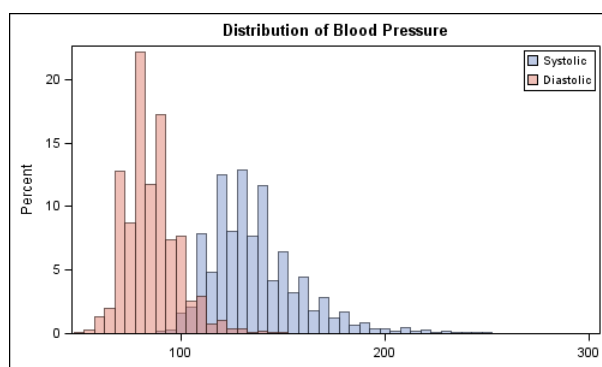


```

TITLE 'Distribution of Blood Pressure';
PROC SGPLOT DATA = sashelp.heart;
  HISTOGRAM systolic /
    FILLATTRS = GRAPHDATA1 NAME = 's'
    LEGENDLABEL = 'Systolic'
    TRANSPARENCY = 0.5;
  HISTOGRAM diastolic /
    FILLATTRS = GRAPHDATA2 NAME = 'd'
    LEGENDLABEL = 'Diastolic'
    TRANSPARENCY = 0.5;
  KEYLEGEND 's' 'd' /
    LOCATION = INSIDE
    POSITION = TOPRIGHT ACROSS = 1;
  XAXIS DISPLAY = (NOLABEL);
RUN;

```

With SAS 9.3, you can set the BINWIDTH for the HISTOGRAM to get a better comparative graph:



```

TITLE 'Distribution of Blood Pressure';
PROC SGPLOT DATA = sashelp.heart;
  HISTOGRAM systolic /
    FILLATTRS = GRAPHDATA1 NAME = 's'
    LEGENDLABEL = 'Systolic'
    TRANSPARENCY = 0.5 BINWIDTH = 5;
  HISTOGRAM diastolic /
    FILLATTRS = GRAPHDATA2 NAME = 'd'
    LEGENDLABEL = 'Diastolic'
    TRANSPARENCY = 0.5 BINWIDTH = 5;
  KEYLEGEND 's' 'd' /
    LOCATION = INSIDE
    POSITION = TOPRIGHT ACROSS = 1;
  XAXIS DISPLAY = (NOLABEL);
RUN;

```

Sanjay Matange

From: <http://blogs.sas.com/content/graphicallyspeaking/>

Why and How To Make a “Tall” Web Graph: Optimizing the Image When Vertical Scrolling Is Necessary and Desirable

Q: How do you make a tall web graph?

A:[SAS 9.2] In *VIEWES News Issue 54*, I departed from my long-avowed opposition to scrolling a “wide” web graph. The ideal web graph is sized so that it fits entirely inside the web browser window. However, if you do not want to break up a very long trend plot with many data points into interlinked subspans of the total date, time, or datetime range (the solution presented in *VIEWES News Issue 55*), then using a sufficiently large **xpixel** count is the solution. If you want horizontal tick values at reasonable intervals, readability is otherwise difficult to achieve.

In *VIEWES News Issue 44*, I presented two subsetting and ranked horizontal bar charts as a communication-effective tool to “Show Them What Is Important”.

In this issue, I step up to the reality of information for which a vertically scrollable web graph is the appropriate solution, but where there is a requirement to show ALL of the information, if only as a web page that you can optionally click to. Furthermore, if your SAS application is an information look-up tool, you must make all of the information available. And you might want to show it in context, i.e. in ranked relationship to the rest of the information, showing all of the data, not just a rank number in a one-line display of only the looked-up value.

The tall image demonstrated here is an ordered horizontal bar chart with subsetting turned off.

A fact of SAS/GRAPH is that it does not necessarily optimize the use of available vertical space, i.e. if your horizontal bar chart consists of only a few bars, they will sit in the vertical middle of the chart image with a lot of unnatural looking white space above and below.

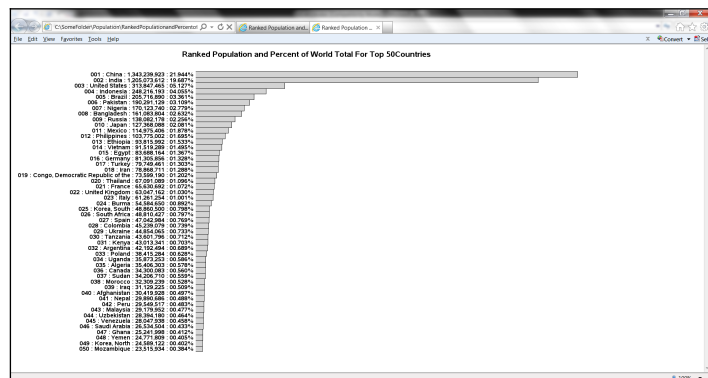
To deliver a tall horizontal bar chart (i.e., a very large number of bars), you obviously need to increase the **xpixel** count. To do this in a production environment, where no iterative loop of trial and adjustment is available, requires what I have long called “software intelligence”, which is my personal favourite grandiose term for data-adaptive program design and construction.

The heart of the solution is to first programmatically determine the number of bars to be delivered, and then use that to dynamically set the **xpixel** count for the GOPTIONS statement.

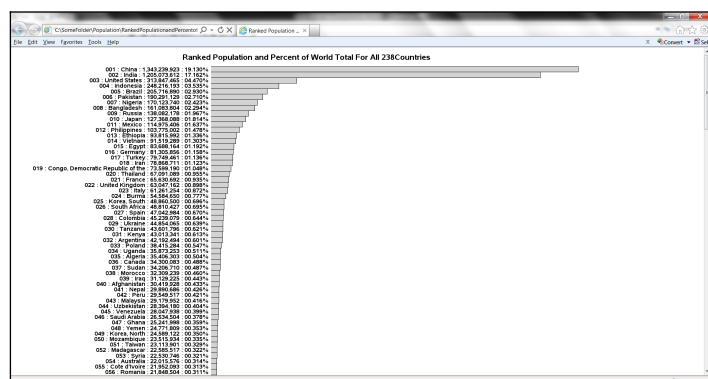
But the key to success is to also dynamically set the length of the vertical axis as a percent of the vertical space. The default handling of available space by SAS/GRAPH created far too much white space between the title and the top bar, no matter what rigmarole I went through after increasing the **ypixel** count. Using axis length to circumvent the default handling of white space was helpfully suggested by SAS Technical Support.

The images were designed to be displayed on a wide screen monitor with resolution 1920 by 1080. The web browser used for these screen captures was Internet Explorer 9, with only these toolbars taking up vertical space: Menu bar and Status bar.

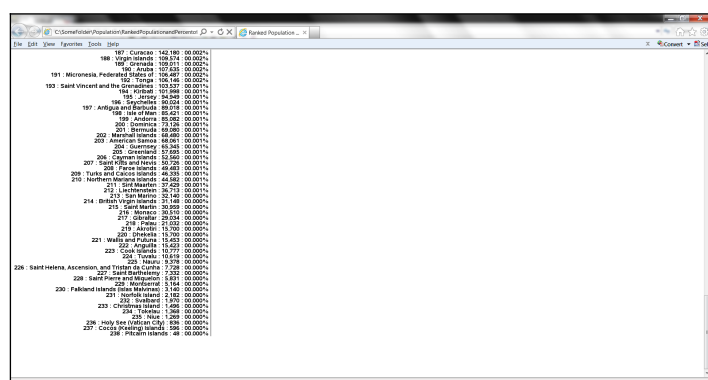
First, here is a web HBAR chart of the Top 50 countries:



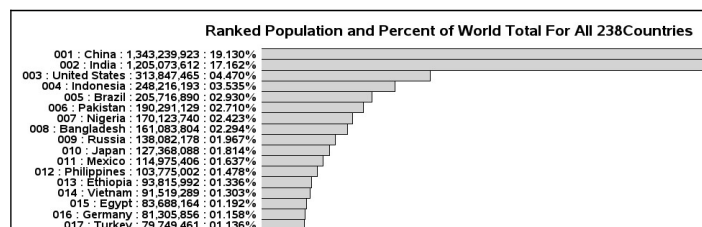
For all 238 countries, here is the top of the tall web chart:



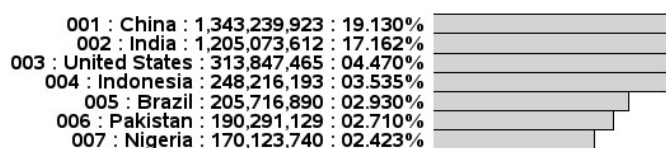
and here is the bottom, reached after scrolling down:



Here is a cropped excerpt of the top of the chart:



and here is an enlargement of part of the excerpt:



From the first two illustrations, you can see that this solution for the tall chart does not deliver the exact same white space between title and top bar as occurs in the subsetted chart, but it does a reasonable job. Also, the value of 0.90 for the macro parameter **DefaultShareOfSpaceForBars** is just an estimate. I did not experiment with other values.

In any case, for more title lines and/or some footnote(s), and/or for a different **TITLE** height and **HTEXT** value, the **DefaultShareOfSpaceForBars** would have to be adjusted. However, this parameter, to be optimized, is obviously some function of the number of titles and footnotes used, their height, and the default text height specified with **HTEXT**. This moderate amount complexity and a priori ambiguity is not nice, but the solution, once coded to fit the specifications of the text parts of the image, can be used reliably in production to deliver satisfactory results, no matter what the number of bars might be from run to run of the application. Furthermore, an ambitious programmer could easily enhance the macro to work with additional parameters for **NumberOfTitleLines**, **TitleHeight**, **NumberOfFootnoteLines**, and **FootnoteHeight**. If, as I like, titles and footnotes were to be rendered with the same height, then the number of extra parameters needed would be only two.

Here is the code used to create the web pages (for a way to provide hyperlinks between the web pages, see for example my article in *IEWS News Issue 55*):

```
%MACRO TallImageFileForHBARchart(
  data=
  ,RptPath=
  ,TitlePrefix=
  ,TitleSuffix=
  ,CountryCount=
  ,BarLabelVar=
  ,BarValueVar=
  ,BarValueFormat=
  ,BarColor=lightgrey
  ,TitleFont='Albany AMT/Bold'
  ,TitleHeight=16 PT
  ,TextFont='Albany AMT/Bold'
  ,TextHeight=12 PT
  ,BarCountMaxForDefaultYpixels=50
  ,DefaultShareOfSpaceForBars=0.90
  ,X_pixels=1600
  ,DefaultYpixels=850
);

%IF %UPCASE(&CountryCount.) NE ALL %THEN %DO;

  PROC SORT DATA = &data.;
    BY DESCENDING &BarValueVar.;
```

```

RUN;

OPTIONS OBS=&CountryCount.;

%END;

DATA ToProcess;
  SET &data.;
RUN;

PROC SQL NOPRINT;
  SELECT COUNT(&BarLabelVar.)
         ,SUM(&BarValueVar.)
  INTO :BarCount
       ,:BarValueTotal
  FROM ToProcess;
QUIT;

OPTIONS OBS=MAX;

DATA ToPrep (KEEP=BarLabel &BarValueVar.);
  LENGTH BarLabel $256 BarLabelSuffix $7;
  RETAIN MaxLabelLength 0;
  SET ToProcess END=LastOne;
  BarLabelSuffix =
    TRIM(LEFT(PUT(((&BarValueVar. /
                    &BarValueTotal.) * 100),6.3))) || '%';
  IF LENGTH(BarLabelSuffix) EQ 6
    THEN BarLabelSuffix =
      '0' || BarLabelSuffix;
  BarLabel =
    TRIM(LEFT(PUT(_N_,z3.))) || ' : ' ||
    TRIM(LEFT(&BarLabelVar.)) || ' : ' ||
    TRIM(LEFT(PUT(&BarValueVar.,
                  &BarValueFormat.))) ||
    ' : ' || BarLabelSuffix;
  MaxLabelLength =
    MAX(MaxLabelLength, LENGTH(BarLabel));
  OUTPUT;
  IF LastOne;
  CALL SYMPUT('LabelLength', MaxLabelLength);
RUN;

DATA ToChart;
  LENGTH BarLabel $&LabelLength.;
  SET ToPrep;
RUN;

%LET BarCount = %TRIM(%left(&BarCount.));

GOPTIONS RESET=ALL;

GOPTIONS XPIXELS=&X_pixels.
         YPIXELS=&DefaultYpixels.;

GOPTIONS FTEXT=&TextFont. HTEXT=&TextHeight.;

%IF %EVAL(&BarCount. GT
          &BarCountMaxForDefaultYpixels.)
%THEN %DO;

  %LET DefaultYpixelsForDefaultBarCount =
    %SYSFUNC(CEIL(%SYSEVALF(&DefaultYpixels. *
                            &DefaultShareOfSpaceForBars.)));

  %LET ReserveYpixelsForTtlsFtnntsEtc =
    %EVAL(&DefaultYpixels. -
          &DefaultYpixelsForDefaultBarCount.);

  %LET YpixelsPerBar = %SYSEVALF(
    &DefaultYpixelsForDefaultBarCount. /
    &BarCountMaxForDefaultYpixels.);

  %LET YpixelsForAllBars =
    %SYSFUNC(CEIL(%SYSEVALF(&BarCount. *
                            &YpixelsPerBar.)));

```

```

%LET Y_pixels =
  %SYSEVALF(&YpixelsForAllBars. +
            &ReserveYpixelsForTtlsFtnntsEtc.);

GOPTIONS YPIXELS=&Y_pixels.;

%LET PercentOfYpixelsForAxis =
  %SYSFUNC(FLOOR(%SYSEVALF(100 -
    %SYSEVALF(%SYSEVALF(
      &ReserveYpixelsForTtlsFtnntsEtc.
      / &Y_pixels.) * 100))));

%END;

PROC CATALOG CAT = work.gseg KILL;
RUN;
QUIT;

```

The PROC step above permits reuse of a graph file name in reruns during the same SAS or Enterprise Guide session. However, the first time that the PROC step runs, it causes an ERROR message because **work.gseg** does not yet exist.

```

%LET Title = &TitlePrefix;

%IF %UPCASE(&CountryCount.) EQ ALL
%THEN %LET Title = &Title.%STR(All)&BarCount.;
%ELSE %LET Title = &Title.%STR(Top)&BarCount.;

%LET Title = &Title.&TitleSuffix.;

%LET FileName =
  %SYSFUNC(COMPRESS(&Title.,' '));

OPTIONS NOCENTER;
FOOTNOTE;
ODS NORESULTS;
ODS LISTING CLOSE;
ODS HTML PATH="&RptPath." (URL=NONE)
  STYLE=MINIMAL GTITLE GFOOTNOTE
  BODY="&FileName..html" (TITLE="&Title.");

TITLE1 FONT=&TitleFont. HEIGHT=&TitleHeight.
  "&Title.";

PROC GCHART DATA = ToChart;
  PATTERN1 COLOR=&BarColor. VALUE=SOLID;
  AXIS1 LABEL=NONE MAJOR=NONE MINOR=NONE
  STYLE=0 VALUE=(JUSTIFY=RIGHT)
%IF %EVAL(&BarCount. GT 50) %THEN %DO;
  LENGTH=&PercentOfYpixelsForAxis. PERCENT
%END;

  ;
  AXIS2 LABEL=NONE MAJOR=NONE MINOR=NONE
  STYLE=0 VALUE=NONE;

  HBAR BarLabel / SUMVAR=&BarValueVar.
  LEVELS=ALL DESCENDING NOSTATS NOFRAME
  MAXIS=AXIS1 RAXIS=AXIS2
  SPACE=0 COUTLINE=black
  NAME="HBar&CountryCount."
  DESCRIPTION=' ';

RUN;
QUIT;

ODS HTML CLOSE;
ODS LISTING;

%MEND TallImageFileForHBARchart;

OPTIONS MPRINT MPRINTNEST;

LIBNAME DataLib "C:\VIEWS56";

```

```
%TallImageFileForHBarChart(
  data=DataLib.PopulationByCountryPerWFB2012
  ,CountryCount=50
  ,RptPath=C:\VIEWS56\Population
  ,TitlePrefix = Ranked Population and Percent
    of World Total For
  ,TitleSuffix=Countries
  ,BarLabelVar=Country
  ,BarValueVar=Population
  ,BarValueFormat=comma13.);
```

```
%TallImageFileForHBarChart(
  data=DataLib.PopulationByCountryPerWFB2012
  ,CountryCount=All
  ,RptPath=C:\VIEWS56\Population
  ,TitlePrefix=Ranked Population and Percent of
    World Total For
  ,TitleSuffix=Countries
  ,BarLabelVar=Country
  ,BarValueVar=Population
  ,BarValueFormat=comma13.);
```

To generate the tall HBAR chart, it is necessary to add options LEVELS=ALL. Otherwise the graph is not generated, and this message is issued:

```
ERROR: There were 238 midpoints generated from
the data. Specify levels=all to display the
graph.
```

Under normal circumstances, the option is unneeded.

For both cases, Top 50 Countries and All 238 Countries, the SAS log contains this false alert:

```
WARNING: The left vertical axis labeled BarLabel
could not be fit as specified. The axis values
will overwrite.
```

For the case of the tall HBAR chart, the SAS log contains an additional false assertion:

```
WARNING: The left vertical axis labeled BarLabel
was too large as specified. Specify LENGTH=94
PERCENT.
```

OPTIONS MPRINT forces a listing of the dynamically customized AXIS statement in the SAS log, showing that my algorithm actually provided more space for the bar labels:

```
AXIS1 LABEL=NONE MAJOR=NONE MINOR=NONE
STYLE=0 VALUE=(JUSTIFY=RIGHT) LENGTH=97
PERCENT;
```

The input data set used in this article is available from the author via email request.

For an ODS Graphics version of this tool, with expanded function to create interlinked web pages between different views of the data, please see the section TOOL 5. THE UpTo4WaysRankingHbarChartsToHTML MACRO in “Data Visualization Power Tools: Expedite the Easy, Implement the Difficult, or Handle Big Data”, which can be found in the web folder www.pharmasug.org/proceedings/2013/DG in the PDF file [PharmaSUG-2013-DG11.pdf](#).

LeRoy Bessler

Did You Know?

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

Analytical Hierarchical Process (AHP) in JMP and SAS/IML

Abstract

Analytical Hierarchical Process is one of the leading techniques in Multi-criteria Decision Analysis field. It is first developed as a structured approach for corporate to manage complex decision making scenario with reasonable level of accuracy. AHP is a very simple computation approach which can be programmed in several computer languages. In this paper, we shall examine the theoretical foundations of AHP as well as the application of AHP in a variety of scenarios. Two simple macros in JMP and SAS/IML will be included in the appendix. structured approach for corporate to manage complex decision making scenario with reasonable level of accuracy. AHP is a very simple computation approach which can be programmed in several computer languages. In this paper, we shall examine the theoretical foundations of AHP as well as the application of AHP in a variety of scenarios. Two simple macros in JMP and SAS/IML will be included in the appendix.

Introduction

Multi-criteria Decision Analysis is a very important field of study in decision analytics. With the advances of business in many areas and the beginning of inter-departmental operations, many project decision making structures have been demonstrated to be extremely ineffective in evaluating the decision choices. This led to the development of several decision analytical techniques which led to goal programming as well as data envelopment analysis. One of the most important techniques developed in this field is AHP by Thomas L.Saaty.

Theoretical Basis of AHP

The key step to AHP is the concept of measurements. As measurements can be inaccurate, this inaccuracy will have serious repercussions in different areas. One of the most common inaccuracies in measurements is the fact that a human preference is not linear in nature which is otherwise indicated when we are collecting data. Even with the presence of ordinal scales, this problem with scales has always been an interesting problem which is commonly encountered by many analysts in their daily routines. To ensure consistencies, preferences in AHP are indicated by various levels of pair wise comparisons in which they are inverses of one another. There are some strong reasons to this arrangement in which Thomas explains in many of his papers on the applicability of his model (Saaty, 1990).

One key point about the use of the preference matrix is the fallibility of numerical standard measures when the objective changed. To prevent this from happening, an initial standard for comparison is designed at the beginning of the analysis to ensure that the measurement is relevant. Much like the standard scales of weights, temperature and energy, there has to be a property to which the scale is measuring. However, there exists another type of scale more commonly known as the relative scale or measurement. This is useful for situation where there can be no standard scales such as the futile attempt to measure love or anger. A useful trait of the relative scales is the ease of conversion to a standard scale.

For AHP, the key sources for a relative scale come from

comparison, the most effective approach to understanding and isolating the effects of a properties is to only introduce pair wise comparisons for that single property across all cases. With this basis in mind, we can start examining the matrices for use in AHP.

Assuming there are n states of comparisons which each one has their own judgment value of $J_1, \dots, J_n$, we can now form a simple matrix highlighting the various states as shown below. The values in the matrix have corresponding inverses in the opposite wing which highlight the symmetry of the weighting. This approach has the added advantage of providing us a direct solution as well as excellent theoretical interpretation.

$$\begin{matrix} & J_1 & J_2 & \cdot & \cdot & \cdot & J_n \\ \begin{matrix} J_1 \\ J_2 \\ \cdot \\ \cdot \\ \cdot \\ J_n \end{matrix} & \begin{pmatrix} w_1/w_1 & w_1/w_2 & \cdot & \cdot & \cdot & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \cdot & \cdot & \cdot & w_2/w_n \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ w_n/w_1 & w_n/w_2 & \cdot & \cdot & \cdot & w_n/w_n \end{pmatrix} & \begin{pmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{pmatrix} \end{matrix} = n \cdot \begin{pmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{pmatrix}$$

Assuming that the end result is nw , then we can say that n is the eigenvalue of J while w is the eigenvector of J . Given that the rank of A is one as a result of the duplicity of the rows beyond the first, the eigenvalues will be zero beside one. The sum of the eigenvalues of the matrix is equal to the sum of the diagonal elements also known as trace. Thus, n will be the principal eigenvalue of J . The solution is made up of positive entries which are unique to that particular multiplier. To ensure uniqueness of w , we typically normalize the entries by their sum. Thus the solution is any normalized column from J . As a result of the inverses, the matrix is also consistent.

Entries to the matrix

One important point to note for AHP is that while the quantitative variables are comparable, the qualitative variables are not. To reduce them to quantitative values, one can always substitute them with actual values perceived by the person in terms of importance. Below is the list of values used typically in a conversion process.

Intensity of Importance	Description
1	Equal in importance
2	Intermediate between the above and below
3	Moderate Importance of one over the other
4	Intermediate between the above and below
5	Essential or Strong Importance
6	Intermediate between the above and below
7	Very Strong Importance
8	Intermediate between the above and below
9	Most important

Approaches to AHP: JMP vs SAS/IML

Given that AHP is all about matrix manipulations, it will be hard to conceive solving the AHP problem without the help of some matrix manipulation languages. To achieve that in base SAS, one has to rely on the use of SAS/IML libraries to calculate the

matrix. Most fortunately, SAS/IML library has an inbuilt function to extract eigenvalues and eigenvectors from the matrix.

To derive the eigenvalues and eigenvectors from a matrix, one only needs to call the Call Eigen function. However, before we start the process, let us first fill up the inverses of the matrix correctly. This can be done in a simple loop.

```
/*DECLARE THE DATA TO BE USED*/
USE X;
/*READ THE DATA*/
READ ALL INTO WORKMAT;
/*FILL IN THE INVERSES*/
DO I = 1 TO NCOL(WORKMAT);
  DO J = 1 TO NROW(WORKMAT);
    IF MISSING(WORKMAT[I, J]) THEN
      WORKMAT[I, J] = 1 / WORKMAT[J, I];
  END;
END;
```

After the above steps, all the inverses required will have been filled in. Let us get the eigenvectors and eigenvalues from the matrix.

```
CALL EIGEN(X3, X4, WORKMAT);
```

It is important to noted that X3 contains the eigenvalues while X4 contains the eigenvector. Due to the manner of extraction, complex roots might be encountered. To obtain the right answer, one can use some simple manipulations to obtain the required results as shown below.

```
EGVAL = X3[1,1];
PRINT EGVAL;
EGVEC = X4[,1];
K = EGVEC[+];
EGVEC = EGVEC / K;
PRINT EGVEC;
```

Extracting the eigenvector and eigenvalue only answers part of the question. One of the key measure is the consistency ratio of the matrix. This indicator answers the question whether the system is consistent and that the results and decisions derived make sense. To answer that, we can apply the formula for deriving the ratio. One key note is that the ratio involved an adjustment factor which is presented by L.Saaty. In this code, we have presented the value up to a matrix of 10 X 10.

```
CI = (EGVAL -
      NCOL(WORKMAT)) / (NCOL(WORKMAT) - 1);
PRINT CI;

IF NCOL(WORKMAT) = 1 THEN AR = 0;
IF NCOL(WORKMAT) = 2 THEN AR = 0;
IF NCOL(WORKMAT) = 3 THEN AR = 0.58;
IF NCOL(WORKMAT) = 4 THEN AR = 0.9;
IF NCOL(WORKMAT) = 5 THEN AR = 1.12;
IF NCOL(WORKMAT) = 6 THEN AR = 1.24;
IF NCOL(WORKMAT) = 7 THEN AR = 1.32;
IF NCOL(WORKMAT) = 8 THEN AR = 1.41;
IF NCOL(WORKMAT) = 9 THEN AR = 1.45;
IF NCOL(WORKMAT) = 10 THEN AR = 1.49;

CR = CI / AR;
PRINT CR;
IF CR <= 0.1 THEN
  PRINT "CONSISTENCY ACHIEVED";
```

```
ELSE PRINT "INCONSISTENCY FOUND";
```

If the consistency ratio is determined to be less than 0.1, then the conclusion from the matrix derived is acceptable, otherwise it will be rejected. Let us examine the output from the model.

```
The SAS System      14:03 Wednesday, March 23, 2011    2
EGVAL
3.2469546

EGVEC
0.2330592
0.0544537
0.712487

CI
0.1294773

CR
0.2128919

INCONSISTENCY FOUND
```

From the options, we can see that the CR does not hold up well. Thus the values in the eigenvector cannot be used to determine the decisions. Let us examine another case.

```
The SAS System      14:03 Wednesday, March 23, 2011    4
EGVAL
3.0857667

EGVEC
0.6738106
0.1006539
0.2255355

CI
0.0428833

CR
0.0739368

CONSISTENCY ACHIEVED
```

In this case, consistency is achieved and the highest is given to the choice corresponding to the first row of the eigenvector. Thus SAS/IML provides a comprehensive and direct calculation of AHP. However, AHP in the case of JMP is extremely different.

JMP has the same capabilities as base SAS although it is notably weaker in some aspects compared to base SAS. JMP has an inbuilt matrix manipulation language which users can directly access using JMP script language. However, as it has a reduced library, JMP does not have the complex Eigen-decomposition ability of SAS. To achieve AHP, we will have to turn to the normalization approach to calculation of the eigenvectors and eigenvalues. This approach basically implies that the matrix should be normalized by dividing with the column sums. After this step, the average for each row is then used as the eigenvector value for that row. Multiplying the original matrix with the eigenvector, we will obtain a vector containing the values which can be used to derive the eigenvalue. By dividing each entries with the corresponding eigenvector value and summing them up, we will obtain the eigenvalue. Subsequent calculation of CI and CR will be as per the SAS/IML steps.

Below is the JMP script that will achieve this.

```
AHP = function(
```

```
{InputMat},
InputMat2 = InputMat;
colsumval = j(1, nrow(InputMat2), 0);
for(i=1, i<=nrow(InputMat2), i++,
for(j=1, j<=ncol(InputMat2), j++,
if(ismissing(InputMat2[i, j]),
InputMat2[i, j]=1/InputMat2[j, i]);
);
);
InputMat3 = InputMat2;
for(i=1, i<=ncol(InputMat2), i++,
colsumval[1, i] = sum(InputMat2[0, i]);
);
print(InputMat2);
for(i=1, i<=nrow(InputMat2), i++,
for(j=1, j<=ncol(InputMat2), j++,
InputMat2[i, j]=
InputMat2[i, j]/colsumval[1, j];
)
);
evect = j(nrow(InputMat2), 1, 0);
for(i=1, i<=nrow(InputMat2), i++,
evect[i, 1] =
sum(InputMat2[i, 0])/nrow(InputMat2);
);
evalue = InputMat3*evect;
CI = (sum(evalue:/evect)/
ncol(InputMat2)-
ncol(InputMat2))/
(ncol(InputMat2)-1);
print(InputMat2);
print(InputMat3);
print(colsumval);
print(evect);
print(evalue);
print(CI);
);
```

Conclusion

AHP is a powerful tools in implementing complex decision. SAS/IML and JMP both provide simple and implementation solution to using AHP with the inbuilt capabilities of the software.

Murphy Choy & Michelle L. Cheong, School of Information System, SMU, Singapore

Invertibility of matrices in SAS/IML

Abstract

Invertibility of matrices is a fundamental idea in linear algebra and something that is widely studied by college students. However, the convenience of statistical and mathematical software has removed much of the needs for matrix manipulations that dominates much of the workings of linear regressions. In this paper, we seeks to explain some of the most fundamental ideas in invertibility of matrices and how they can be detected in SAS/IML before inverses are done on them.

Introduction

Invertibility of matrices has been an extremely well studied area in linear algebra and most of the existing matrix properties and ideas has been formulated before the advances in computing power. The development in many areas of mathematics and statistics has been propelled forward by the advances in computing power combined with matrix manipulation. Items such as eigenvalues and eigenvectors as well as singular value

decomposition have been the products of matrix operations and fundamentals.

Inverses? What are they?

There are two types of inverses, namely inverses to non-square matrix and inverses to square matrix. These two type of matrix are calculated in different ways and has totally different properties. To begin with the simpler one, let us have a closer look at the non-square matrix. For the non-square matrices, one of the key issues surrounding them is the fact that they are not truly invertible as one of the key conditions for invertibility is a square matrix. However these matrices can have either a left inverse or right inverse. As such, they are considered by the author to have a form of inverse if that is the version needed. To understand whether such an inverse exist, the approach is to estimate the rank of the matrix. Rank of a matrix is defined as the number of linearly independent rows or columns in the matrix. The rank for a non-square matrix $m \times n$ is the lower value of the two dimensions. Any matrix with rank value equal to the maximum possible rank is called a full rank matrix, otherwise it will be addressed as a rank deficient matrix. To obtain the ranks of a matrix in SAS/IML, one can use the following function as given in the SAS/IML documentation.

```
rank = round(trace(ginv(a) * a));
```

For the case of a square matrix, there is one other method to determine the invertibility of the matrix beside the ranks. One could calculate the determinant of the matrix and should the determinant be zero, it means that the matrix in question is singular and thus cannot be inverted.

```
deta = det(a);
```

Conclusion

Invertibility is something that SAS/IML can check easily with the use of some simple functions.

Murphy Choy, School of Information Science, SMU, Singapore

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions.

[SAS 9.4M2] The new **BESTDOTX.** format specifies that SAS choose the best notation and use a dot as a decimal separator. This format is independent of the current locale.

[SAS 8] Do you ever want to create a SAS data set on your current SAS platform, but send that file to be read on a different platform? By default SAS creates data sets in the format specific to your current platform, including whether the platform is 32-bit or 64-bit. The **OUTREP=** data set and library option allows you to specify a different platform, such as WINDOWS, HP_UX, LINUX, MVS, etc., so that the data set and whole library for data sets are created appropriately.

[SAS 9.2] The **DATDIF()** and **YRDIF()** functions calculate day and year differences using a variety of different day count conventions beyond what is used in the **INTCK()** function.

In Brief

- Back issues of *VIEWS News* are available from the **VIEWS UK forum** at blog.hollandnumerics.org.uk.
- The SAS Professional Forum group on LinkedIn, which currently has over 33,450 members. Everyone is welcome to join, but discussions are restricted to SAS-related topics: www.linkedin.com/groups?gid=70702

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.

August 2019

30-31 **PharmaSUG China**, Shanghai, China

See the PharmaSUG web site at www.pharmasug.org for further details.

November 2019

10-13 **PhUSE Connect EU**, Amsterdam, Netherlands

The Pharmaceutical Users Software Exchange (PhUSE) conference is the premier programming event for the pharmaceutical industry in Europe. See www.phuse.eu for more details.

March 2020

8-11 **PhUSE Connect US**, Orlando, FL, USA

The Pharmaceutical Users Software Exchange (PhUSE) conference is now the premier programming event for the pharmaceutical industry in the USA. See www.phuse.eu for more details.

29 March-1 April **SAS Global Forum**, Washington, DC, USA

The SAS Global Forum (SGF, formerly called SUGI) is held each year and focuses on the technical aspects of SAS software. See www.sasglobalforum.org for more details.

May 2020

10-13 **PharmaSUG US**, San Francisco, CA, USA

See the PharmaSUG web site at www.pharmasug.org for further details.

Contacts

LeRoy Bessler Email: le_roy_bessler@wi.rr.com

Murphy Choy Email: goladin@gmail.com

Sanjay Matange Email: SanjayMatange@gmail.com

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

VIEWS

Web Site Web: blog.hollandnumerics.org.uk

Newsletter Editor Email: views-uk@hollandnumerics.org.uk

***VIEWS News** is published quarterly by VIEWS International SAS Programmer Community. VIEWS is a not-for-profit organisation.*