V is for Venn Diagrams

Kriss Harris, SAS Specialists Limited

ABSTRACT

Would you like to produce Venn diagrams easily? This paper shows how you can produce stunning two-, three-, and four-way Venn diagrams by using the Graph Template Language, in particular the DRAWOVAL and DRAWTEXT statements. From my experience, Venn diagrams have typically been created in the pharmaceutical industry by using Microsoft Excel and Microsoft PowerPoint. Excel is used to first count the numbers in each group, and PowerPoint is used to generate the two- or three-way Venn diagrams. The four-way Venn diagram is largely unheard of. When someone is brave enough to tackle it manually, working out the numbers that should go in each of the 16 groups and entering the right number into the right group is usually done nervously!

INTRODUCTION

Venn diagrams where introduced in 1883 by John Venn (1834-1923), the Hull born philosopher and mathematician. They are a great way to visualize elements that are unique to only one group and simultaneously visualize elements that intersect with other groups. They are symmetrical by nature and the number of groups in a Venn diagram = 2^n (including the group outside the diagram).

Venn diagrams are built into JMP Genomics. The web applications (Oliveros, 2018) and (Hulsen, 2018) can also be used to create Venn diagrams. Sometimes you may need to keep your data within SAS and this paper will demonstrate a macro that can be used in SAS® 9.4 to create Venn diagrams. Unlike the previous paper which shows how to produce Venn diagrams using quantitative data and a specified cut-off (Harris, V is for Venn Diagrams, 2013), this paper shows how to produce Venn diagrams when the data is qualitative.

Essentially there are two sections to this macro. The first section calculates the numbers and percentages for each group of the Venn diagram and the second section produces the Venn diagram. Two types of Venn diagrams can be created: normal and proportional Venn diagrams. If you want to output a normal Venn diagram, where the circles or ellipses are all the same size, then Graph Template Language (GTL) will create the Venn diagram. If you want to output a proportional Venn diagram then the API for Google Charts will produce the Venn diagram; the inspiration for the Google Charts method came from the paper (Kruger, 2011). In the paper "Creating Proportional Venn Diagrams Using Google and SAS®", Kruger shows how the API for Google Charts can produce proportional Venn diagrams. This paper builds Kruger's paper by also calculating the numbers for each group within the Venn diagram, and also allowing the numbers in each group to be displayed on the figure.

MACRO CALL

Program 1 below illustrates how the macro call works for the %VENN macro.

```
%macro venn(

    dsn =                    (1)

  ,proportional = 0       (2)

  ,venn_diagram = 2       (3)

  ,varA =
  ,varB =                 (4)
  ,varC =
  ,varD =
```

```
    ,LabelA = List A
    ,LabelB = List B                5
    ,LabelC = List C
    ,LabelD = List D

    ,percentage = 0    6

    ,prop_n_label = 0    7

    ,out_location = C:\SGF2018\VennDiagram\output    8

    ,outputfilename = Venn diagram    9

);
```

Program 1: Macro Call

(1) Define the data set name. An example data set named VENNDATA has been created in the program and is available for use. The data set is largely based on the SASHELP.CARS data set, but the dataset was modified so that some of the elements would be in the other Venn diagram groups. The information from (Jedi, 2018) was used to convert the data set into SAS datalines so that the data set could easily be created and used within the macro.

(2) To produce a normal Venn diagram set the PROPORTIONAL macro variable value to be 0, and if you want to produce a proportional Venn diagram set the macro variable value to be 1. Normal Venn diagrams are produced by default.

(3) If the PROPORTIONAL macro variable value is set to 0, then you are able to create two,three, or four-way Venn diagrams, by setting the VENN_DIAGRAM macro variable value to be 2, 3, or 4 respectively. For proportional Venn diagrams you are able to create two or three-way Venn diagrams.

Two-way Venn diagrams are produced by default.

(4) The data set variables that contain the list of elements are defined here. For two-way Venn diagrams, values for the macro variables VARA and VARB are required. For three-way Venn diagrams a value for VARC is also required, and for four-way Venn diagrams a value for VARD is also required.

In the example using the VENNDATA data set then the values to be used are MAKEA, MAKEB, MAKEC or MAKED.

(5) The labels for the sets are defined here. The value of the macro variable LABELA, is the label for the data used in the macro variable VARA. This pattern repeats for the other groups.

(6) For normal Venn diagrams, only the numbers are displayed for each set and intersect by default. If you also want percentages to be displayed then set the PERCENTAGE macro variable value to be 1. This option only applies to normal Venn diagrams, i.e. when the PROPORTIONAL macro variable value is set to 0.

(7) For proportional Venn diagrams, no numbers are being displayed for each set and intersect by default. If you want the numbers to be displayed in the legend then set the PROP_N_LABEL macro variable value to be 1. This option only applies to proportional Venn diagrams, i.e. when the PROPORTIONAL macro variable value is set to 0.

(8) Define the path for your output files here.

(9) Define the filename for your .PNG or .HTML file here. This file will show the Venn diagram.

The %VENN macro also outputs data sets in the WORK library named DISTINCT_ELEMENTS, and MERGED_ELEMENTS. The DISTINCT_ELEMENTS data set contains the elements in each group of the

sets. The MERGED_ELEMENTS dataset identifies the elements that belong to each group within the Venn diagram. These dataset can be used to assess the elements that are in each group.

The full %VENN macro code is in the APPENDIX, and is also available at:
http://www.krissharris.co.uk/sgf/2018/1965/VennDiagram.sas

METHOD

For normal Venn diagrams the macro works by counting the number in each group, and then using GTL to produce the Venn Diagrams with the corresponding numbers in each group displayed. The SQL procedure was used to count the elements in each group because sub querying was done so that the elements in one set and not in the other set could be calculated. For example, A∩B'.

For proportional Venn diagrams the numbers in each group were calculated, and were used as values in the Google Chart code which the macro created. When the outputted HTML files is open, a Venn diagram is produced with the help of the visualization API for Google Charts.

GTL FOR NORMAL VENN DIAGRAMS

Firstly a data set named TEST was created, which had two variables X and Y and had values ranging from 1 to 100. This was created so that it could be used as a blank canvas in conjunction with the SCATTERPLOT statement. This will be explained more later.

```
data test;
  do x = 1 to 100;
    y = x;
    output;
  end;
run;
```
Program 2: Program to create TEST data set

Two-way Venn Diagram Example

This paper assumes that you are familiar with GTL. For more details on using GTL please see (Harris, Hands-On Graph Template Language (GTL): Part A, 2017) or (Harris & Watson, Great Time to Learn GTL, 2018).

To produce the Venn diagram the LAYOUT OVERLAY container was used and contains the SCATTERPLOT plot statement, and DRAWOVAL and DRAWTEXT draw statements. The YAXIS and XAXIS label, line, tick marks and tick values are of no interest when producing the Venn diagram, so these are suppressed in the LAYOUT OVERLAY statement by using the option DISPLAY = NONE.

The SCATTERPLOT statement is used to essentially produce the blank canvas using the TEST dataset to enable the placement of the subsequent items for the Venn diagram by hiding the markers using the SIZE = 0 marker attributes options.

The fun part begins with the DRAWOVAL draw statements, which for two, and three-way Venn diagrams can be used to draw circles. The X and Y arguments are where the centre of the oval is going to be drawn. As you may recall the scale that was used for the blank canvas ranges from 0 to 100 for the XAXIS and YAXIS, and the X and Y arguments that are specified map directly to the values on the scale, so in the first example the centre point will be precisely at X = 36 and Y = 50. In this example the WIDTH = 45 and the HEIGHT = 60 for the oval. These are specified because it means that the ratio between the WIDTH and HEIGHT of the oval is 3:4 and this is to compensate for the fact that usually the default graph and also in this case the ratio between the WIDTH and HEIGHT of the entire graph is 4:3. Therefore the ratio that is used for the oval compensates for this and enables circles to be produced instead of ovals.

The DRAWTEXT draw statement is used to display the numbers and group names in the Venn diagram groups.

Please see Program 3 below for the example GTL code.

```
proc template;
  define statgraph Venn2Way;
    begingraph / drawspace=datavalue;
    /* Plot */
      layout overlay / yaxisopts = (display = NONE) xaxisopts =
        display = NONE);
        scatterplot x=x y=y / markerattrs=(size = 0);

        /* Venn Diagram (Circles) */
        drawoval x=36 y=50 width=45 height=60 /
          display=all fillattrs=(color=red)
          transparency=0.75 WIDTHUNIT= Percent HEIGHTUNIT= Percent;

        drawoval x=63 y=50 width=45 height=60 /
          display=all fillattrs=(color=green)
          transparency=0.75 WIDTHUNIT= Percent HEIGHTUNIT= Percent;

        /* Numbers */
        drawtext "&A" / x=33 y=50 anchor=center;
        drawtext "&AB" / x=50 y=50 anchor=center;
        drawtext "&B" / x=66 y=50 anchor=center;
        /* Percentages */
        %if &percentage = 1 %then %do;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold)
            "&A_per" / x=33 y=45 anchor=center ;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold)
            "&AB_per" / x=50 y=45 anchor=center;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold)
            "&B_per" / x=66 y=45 anchor=center;
        %end;
        /* Labels */
        drawtext "&GroupA" / x=30 y=15 anchor=center width = 30;
        drawtext "&GroupB" / x=70 y=15 anchor=center width = 30;
      endlayout;
    endgraph;
  end;
run;
```

Program 3: GTL Code for Two-Way Venn Diagram

Figure 1 below shows a two-way Venn diagram. From this Venn diagram you can see that there are 42 items in List A, 45 items in List B, and 41 items in the intersection of Lists A and B. The macro call for Figure 1 is below:

```
%venn(dsn= venndata, proportional = 0, venn_diagram = 2, varA=makeA,
  varB=makeB, varC=makeC, varD=makeD, percentage = 0, prop_n_label = 0,
  outputfilename=%str(Figure 1));
```
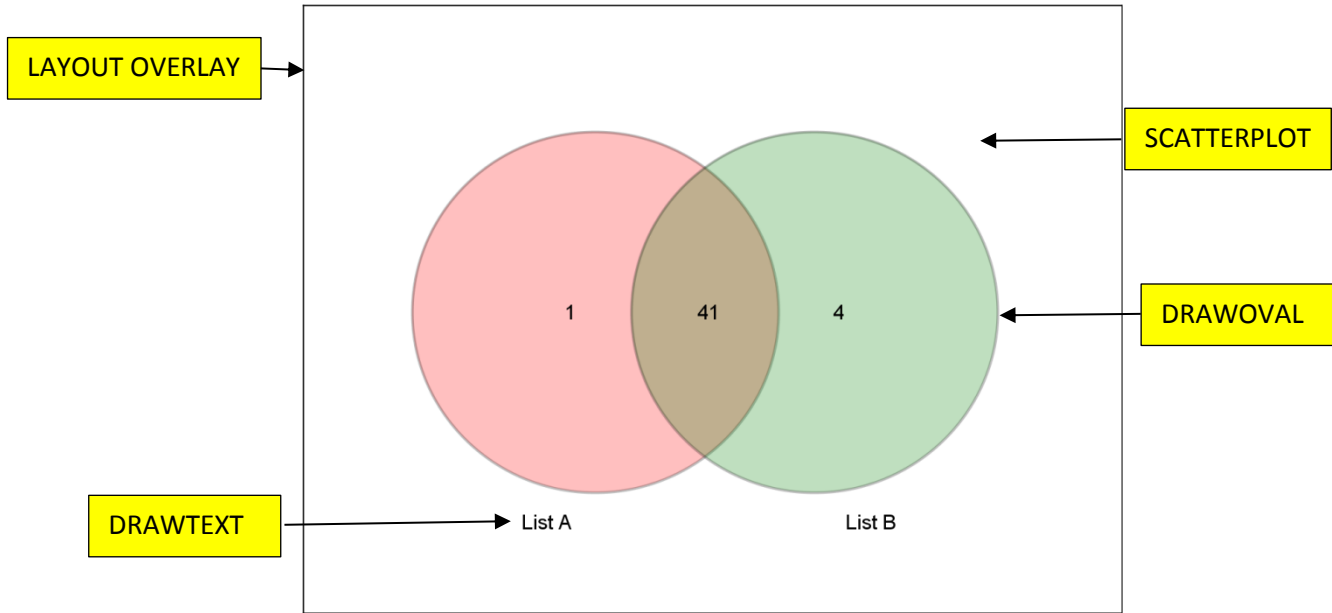
Figure 1: Two-Way Venn Diagram

To also display the percentages on the Venn diagram, set the PERCENTAGE macro variable value to be 1. Figure 2, shows an example of the Venn diagram with percentages, and the macro call for Figure 2 is below:

```
%venn(dsn= venndata, proportional = 0, venn_diagram = 2, varA=makeA,
   varB=makeB, varC=makeC, varD=makeD, percentage = 1, prop_n_label = 0,
   outputfilename=%str(Figure 2));
```
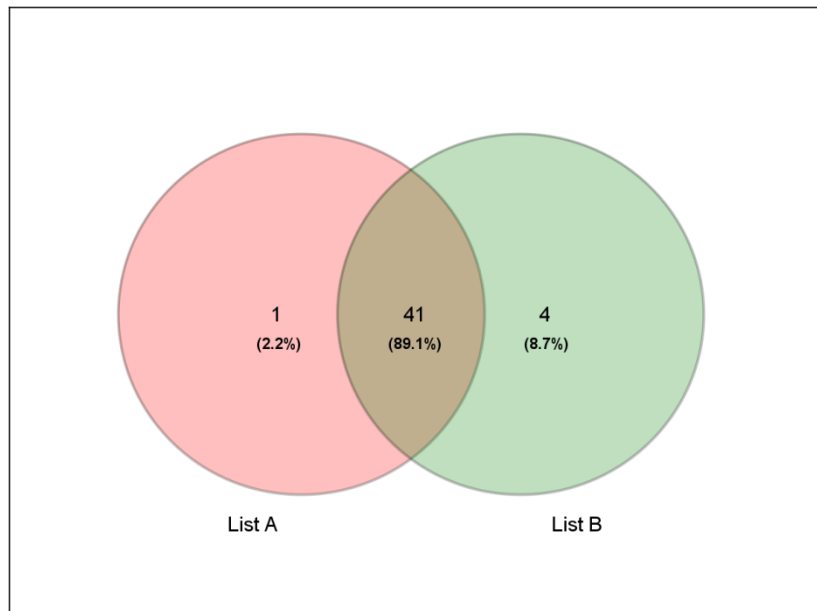

Figure 2: Two-Way Venn Diagram with Percentages

Three and Four-Way Venn Diagrams

To produce three-way Venn diagrams all that is needed is to add the third variable and label to the existing two-way Venn diagram GTL code. Essentially (when the percentages are omitted from the graph) this is an extra DRAWOVAL statement for the third circle, and five more DRAWTEXT statements for the additional numbers in each group. As you can envisage the four-way Venn diagram needs an extra DRAWOVAL statement compared to the three way, and a lot more DRAWTEXT statements. To produce the ellipses the ratio of 1:4 between the WIDTH and the HEIGHT can be used in the DRAWOVAL statement, and the ROTATE argument can be used to position the ellipses in the right direction.

To produce the GTL for the four-way Venn diagram, a lot of trial and error was used to determine the location, size and rotation to use to produce a good diagram. However, there probably is a mathematical way to obtain the exact locations and dimensions for the optimal four-way Venn diagram.

Figure 3 and Figure 4 show the three- and four-way Venn diagrams respectively.

The macro calls for Figure 3 and Figure 4 are below:

```
* Figure 3 macro call;
%venn(dsn= venndata, proportional = 0, venn_diagram = 3, varA=makeA,
  varB=makeB, varC=makeC, varD=makeD, percentage = 0, prop_n_label = 0,
  outputfilename=%str(Figure 3));

* Figure 4 macro call;
%venn(dsn= venndata, proportional = 0, venn_diagram = 4, varA=makeA,
  varB=makeB, varC=makeC, varD=makeD, percentage = 0, prop_n_label = 0,
  outputfilename=%str(Figure 4));
```
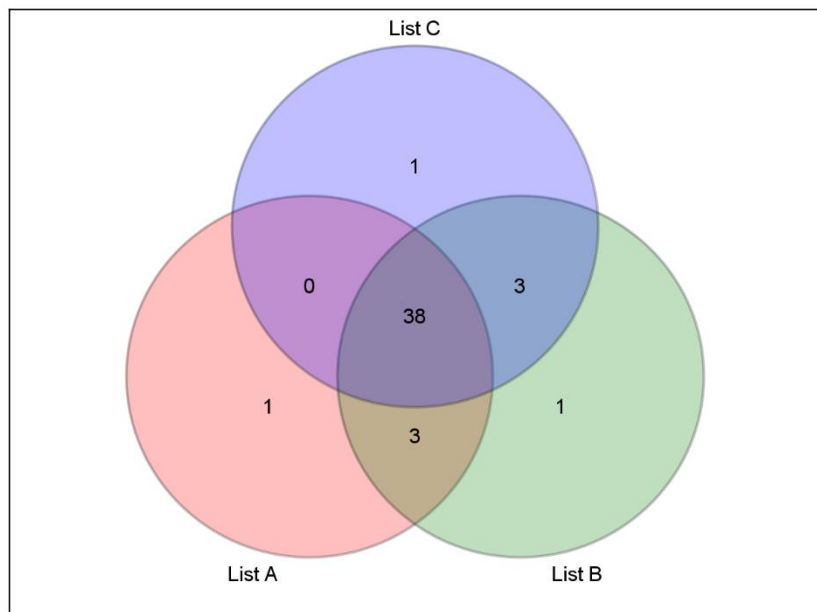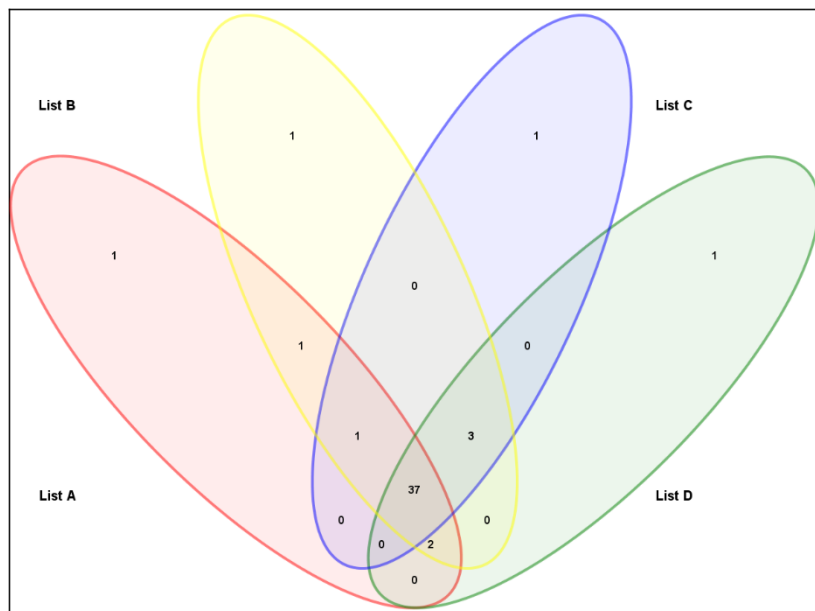

Figure 3: Three-Way Venn Diagram

Figure 4: Four-Way Venn Diagram

The %VENN macro also allows you to produce proportional Venn diagrams. The paper (Kruger, 2011) has more details relating to the method used. The macro in this paper built upon Kruger's brilliant paper, because:

1. The API for Google Charts has been updated since 2011 and so the code that was used by Kruger in 2011, was not working at present.

2. The Venn diagrams which were outputted in Kruger's paper did not show the numbers of elements that each group had, and this was thought to be useful.

3. The method described in Kruger's paper expects you to know the numbers to use as the values for the Google Chart code. This %VENN macro automatically calculates those values for you and places them in the right sections to enable the diagram to be outputted.

Program 4 below was used to create a HTML file which contains an IMG tag. This IMG tag references the Google API and produces the specified proportional Venn diagram. All you need to do is open the HTML file in an internet browser, such as Internet Explorer, Mozilla Firefox or Google Chrome. This is an example program and in the %VENN macro the correct values are populated automatically (assuming you have specified the appropriate data set and variables). Also the data step in Program 4 does not contain any indents on purpose and this is so the correct string can be used as the IMG SRC reference.

```
data _null_;
file "C:\SGF2018\VennDiagram\output\VennGoogle.html";
put '<img src="https://chart.googleapis.com/chart?
cht='"v"'
&chd=t:'"42"','"45"','"0"','"41"'
&chs='"500x500"'
&chco='"ed7878"','"90d190"','"7e7ed3"'
&chdl='"List A"'|'"List B"'"'/>';
run;
```

Program 4: SAS Code for Google Chart

Output 1 shows how the IMG tag looks in the HTML file.

```
<img src="https://chart.googleapis.com/chart?cht=v&chd=t:42,45,0,41&chs=500x500&chco=ed7878,90d190,7e7ed3&chdl=List A|List B"/>
```

Output 1: IMG tag used for Proportional Two-Way Venn Diagram

Figure 5 below shows the proportional two-way Venn diagram. As you may recall, in Figure 2, 89% of the elements in List A intersected with the elements in List B, and this is reflected in the proportional Venn diagram as the two circles overlay a great deal. The macro call for Figure 5 is below:

```
%venn(dsn= venndata, proportional = 1, venn_diagram = 2, varA=makeA,
   varB=makeB, varC=makeC, varD=makeD, percentage = 0, prop_n_label = 0,
   outputfilename=%str(Figure 5));
```


Figure 5: Proportional Two-Way Venn Diagram

In Figure 5, there are no numbers being displayed which show how much elements are in each group. To display this number in the legend then set the PROP_N_LABEL macro variable value to be 1.

Figure 6, shows the Venn diagram with the numbers of each group displayed in the legend. The macro call for Figure 6 is below:

```
%venn(dsn= venndata, proportional = 1, venn_diagram = 2, varA=makeA,
   varB=makeB, varC=makeC, varD=makeD, percentage = 0, prop_n_label = 1,
   outputfilename=%str(Figure 6));
```


Figure 6: Proportional Two-Way Venn Diagram with Number in each Group

Figure 7 below shows an example of a three-way Venn diagram. If you want to add the N numbers in the legend then all you need to do is set the PROP_N_LABEL macro variable value to be 1, as in Figure 6. The macro call for Figure 7 is below:

```
%venn(dsn= venndata, proportional = 1, venn_diagram = 3, varA=makeA,
  varB=makeB, varC=makeC, varD=makeD, percentage = 0, prop_n_label = 0,
  outputfilename=%str(Figure 7));
```
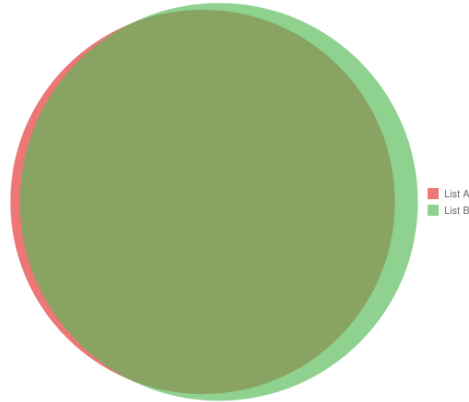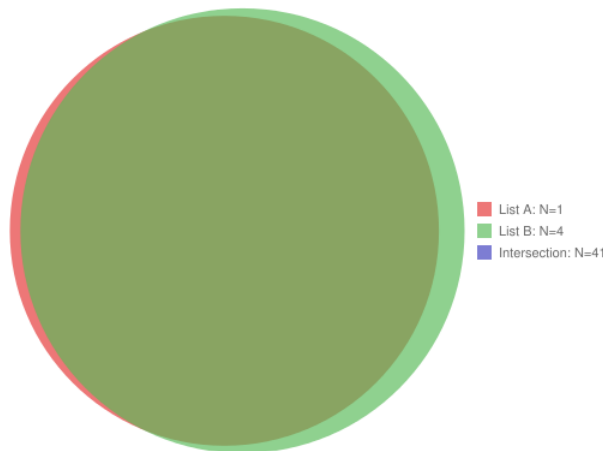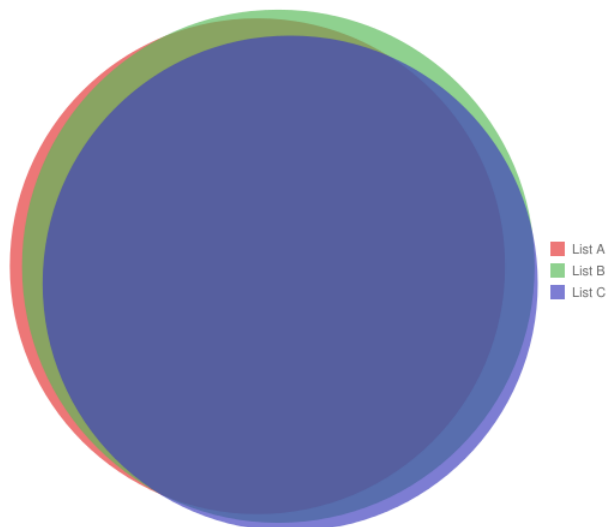

Figure 7: Proportional Three-Way Venn Diagram

CONCLUSION

Venn Diagrams are very useful for visualizing the relationships between groups. This paper has demonstrated how the technique can be implemented using SAS 9.4, with aesthetically pleasing results.

This paper is an updated version of a previous Venn diagram poster that was presented in 2013, and the improvements are that the %VENN macro now works with qualitative data, some checks are done on the macro variables values to help to ensure that only valid values are used, and also the macro now (indirectly) produces proportional Venn diagrams.

Further upgrades can be made to allow the macro to be able to select parts of the Venn diagram and then see the elements that are in that group, and to generate proportional Venn diagrams entirely within SAS. Also to automatically reduce the font size if the numbers in the groups are very large.

REFERENCES

Chung, C. Y., & King, J. (2009). IS THIS MACRO PARAMETER BLANK? . *SAS Global Forum 2009.* SAS Global Forum.

Harris, K. (2013). V is for Venn Diagrams. *SAS Global Forum 2013.* San Francisco: SAS Global Forum.

Harris, K. (2017). Hands-On Graph Template Language (GTL): Part A. *SAS Global Forum.* Denver: SAS Global Forum.

Harris, K., & Watson, R. (2018). Great Time to Learn GTL. *PharmaSUG.* Seattle: PharmaSUG.

Hulsen, T. (2018, 3 1). *BioVenn.* Retrieved from http://www.biovenn.nl/index.php

Jedi, S. (2018, 03 01). *Jedi SAS Tricks: The DATA to DATA Step Macro.* Retrieved from SAS Learning Post: https://blogs.sas.com/content/sastraining/2016/03/11/jedi-sas-tricks-data-to-data-step-macro/

Kruger, H. (2011). Creating Proportional Venn Diagrams Using Google and SAS®. *SAS Global Forum 2011.* Las Vegas: SAS Global Forum.

Mengelbier, M. (2008). Simple %str(ER)ROR Checking in Macros . *SAS Global Forum 2008.* San Antonio: SAS Global Forum.

Oliveros, J. C. (2018, 03 01). *Venny 2.1.* Retrieved from http://bioinfogp.cnb.csic.es/tools/venny/index.html

ACKNOWLEDGEMENTS

I would like to thank Adam Taylor for prompting me to start the Venn diagram SAS Macro years ago by simply asking me if it was possible to do Venn diagram calculations in SAS and I would like to thank Mark Lennon for encouraging me to plot the Venn diagrams in SAS (in 2007).

I would also like to thank Adrienne Bonwick for reviewing this paper.

CONTACT INFORMATION

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

Kriss Harris
SAS Specialists Limited
italjet125@yahoo.com
http://www.krissharris.co.uk

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX

```
*---------------------------------------------*;
* Author:      Kriss Harris                   *;
* Email:       ITALJET125@YAHOO.COM           *;
* Description: Venn Diagram Macro             *;
* Purpose:     V is for Venn Diagrams         *;
*              SAS Global Forum 2018          *;
* Paper:       2018-1965                      *;
*---------------------------------------------*;

* Deleting existing datasets;
proc datasets library = work kill;
run;

*Format for percentages;
proc format;
picture pctpic (round)  0 = '009%)' (prefix='(')
                        0<-high='009.9%)' (prefix='(');
run;

/* Data */
data venndata;
  infile datalines dsd truncover;
  input makeA:$100. makeB:$100. makeC:$100. makeD:$100.;
datalines4;
,Ferrari2BCD,Ferrari2BCD,Ferrari2BCD
,Ferrari3BCD,Ferrari3BCD,Ferrari3BCD
,FerrariBCD,FerrariBCD,FerrariBCD
Acura,Acura2,Acura3,Acura4
Audi,Audi,Audi,Audi
BMW,BMW,BMW,BMW
Buick,Buick,Buick,Buick
Cadillac,Cadillac,Cadillac,Cadillac
Chevrolet,Chevrolet,Chevrolet,Chevrolet
Chrysler,Chrysler,Chrysler,Chrysler
```

```
Dodge,Dodge,Dodge,Dodge
Ferrari2ABD,Ferrari2ABD,,Ferrari2ABD
FerrariAB,FerrariAB,,
FerrariABC,FerrariABC,FerrariABC,
FerrariABD,FerrariABD,,FerrariABD
Ford,Ford,Ford,Ford
GMC,GMC,GMC,GMC
Honda,Honda,Honda,Honda
Hummer,Hummer,Hummer,Hummer
Hyundai,Hyundai,Hyundai,Hyundai
Infiniti,Infiniti,Infiniti,Infiniti
Isuzu,Isuzu,Isuzu,Isuzu
Jaguar,Jaguar,Jaguar,Jaguar
Jeep,Jeep,Jeep,Jeep
Kia,Kia,Kia,Kia
Land Rover,Land Rover,Land Rover,Land Rover
Lexus,Lexus,Lexus,Lexus
Lincoln,Lincoln,Lincoln,Lincoln
MINI,MINI,MINI,MINI
Mazda,Mazda,Mazda,Mazda
Mercedes-Benz,Mercedes-Benz,Mercedes-Benz,Mercedes-Benz
Mercury,Mercury,Mercury,Mercury
Mitsubishi,Mitsubishi,Mitsubishi,Mitsubishi
Nissan,Nissan,Nissan,Nissan
Oldsmobile,Oldsmobile,Oldsmobile,Oldsmobile
Pontiac,Pontiac,Pontiac,Pontiac
Porsche,,,Porsche
Porsche,Porsche,Porsche,Porsche
Saab,Saab,Saab,Saab
Saturn,Saturn,Saturn,Saturn
Scion,Scion,Scion,Scion
Subaru,Subaru,Subaru,Subaru
Suzuki,Suzuki,Suzuki,Suzuki
Toyota,Toyota,Toyota,Toyota
Volkswagen,Volkswagen,Volkswagen,Volkswagen
Volvo,Volvo,Volvo,Volvo
;;;;
run;

/* Venn Diagram Macro */

options mprint symbolgen;
%macro venn(

dsn =

,proportional = 0 /* When proportional = 0 then the Venn Diagram is produced using Graph Template Language and
a PNG file is outputted */
                /* When proportional = 1, a .HTML file is created and a Proportional Venn Diagram is produced
using Google  */

,venn_diagram = 2 /* Select whether you want a 2 Way, 3 Way or 4 Way Venn Diagram. EG for 2
                   way enter 2. Valid values are 2,3 and 4 */


,varA = /* The first variable that contains your list of elements, mandatory */
,varB = /* The second variable that contains your list of elements, mandatory */
,varC = /* The third variable that contains your list of elements, mandatory for 3 and 4 way diagrams */
,varD = /* The fourth variable that contains your list of elements, mandatory for 4 way diagrams */

,LabelA = List A /* Define label 1, mandatory */
,LabelB = List B /* Define label 2, mandatory */
,LabelC = List C /* Define label 3, mandatory for 3 and 4 way diagrams */
,LabelD = List D /* Define label 4, mandatory for 4 way diagrams */

,percentage = 0 /* When percentage = 1, the percentages are displayed too on the Venn Diagrams (produced in
SAS) also */

,prop_n_label = 0 /* When prop_n_label = 1, the N numbers in each group and intersection are also displayed in
the legend */

,out_location = C:\SGF2018\VennDiagram\output
/* Define the path for all output files e.g. C:\Venn Diagrams */
,outputfilename = Venn diagram /* Define the filename for the graphic file */

);

%let parameterchecks = 0; * Assuming no paramter input errors;

* Checking if data exists;
```

```
%if %sysfunc(exist(&dsn)) and &dsn ne %then %do;
  proc sql noprint;
    select count(*) into: obs_count
    from &dsn;
  quit;
%end;

%else %do;
  %let obs_count = 0;
%end;

%if (&obs_count <= 0 ) %then %do;
  %put %str(ER)ROR: Please define the dataset to be used.;
  %let parameterchecks = 1;
%end;

%if (&proportional ^=0 and &venn_diagram>3) %then %do;
  %put %str(ER)ROR: The Proportional Venn Diagrams can only be produced with 2 or 3 Way Venn Diagrams.;
  %let parameterchecks = 1;
%end;

%if (&venn_diagram=2 and (&varA = or &varB = )) %then %do;
  %put %str(ER)ROR: Please define your first and second variables for the Venn Diagram;
  %let parameterchecks = 1;
%end;

%if (&venn_diagram=3 and (&varA = or &varB = or &varC = )) %then %do;
  %put %str(ER)ROR: Please define your first, second and third variables for the Venn Diagram;
  %let parameterchecks = 1;
%end;

%if (&venn_diagram=4 and (&varA = or &varB = or &varC = or &varD = )) %then %do;
  %put %str(ER)ROR: Please define your first, second, third and fourth variables for the Venn Diagram;
  %let parameterchecks = 1;
%end;


%if (&parameterchecks > 0 ) %then %goto exit; %* - error has occured, then exit ;
%* - The following macro code is to be executed if not obvious errors are found ... ;

* Obtaining distinct rows;
proc sql;
  create table distinct_groupA as
  select distinct &varA
  from &dsn
  where &varA ne ""
  order by &varA;
quit;

* Obtaining distinct rows;
proc sql;
  create table distinct_groupB as
  select distinct &varB
  from &dsn
  where &varB ne ""
  order by &varB;
quit;

%if &venn_diagram > 2 %then %do;
  * Obtaining distinct rows;
  proc sql;
    create table distinct_groupC as
    select distinct &varC
    from &dsn
    where &varC ne ""
    order by &varC;
  quit;
%end;

%if &venn_diagram > 3 %then %do;
  * Obtaining distinct rows;
  proc sql;
    create table distinct_groupD as
    select distinct &varD
    from &dsn
    where &varD ne ""
    order by &varD;
  quit;
%end;

data merge_distinct;
```

```sas
    merge distinct_groupA distinct_groupB
    %if &venn_diagram > 2 %then %do;
      distinct_groupC
    %end;
    %if &venn_diagram > 3 %then %do;
      distinct_groupD
    %end;
    ;
run;

%if &venn_diagram = 2 %then %do;
  proc sql;
    create table AB as
    select coalesce(a.&varA, b.&varB) as make,
    case when a.&varA = b.&varB then 1 else 0 end as AB
    from distinct_groupA as a full join distinct_groupB as b
    on a.&varA = b.&varB
    order by calculated make;
  quit;

  * Subquery - elements only in A and not in AB;
  proc sql;
    create table A1 as
    select &varA as make, 1 as A1
    from distinct_groupA where &varA not in (select make from AB where AB = 1)
    order by make;
  quit;

  proc sql;
    create table B1 as
    select &varB as make, 1 as B1
    from distinct_groupB where &varB not in (select make from AB where AB = 1)
    order by make;
  quit;
%end;

%if &venn_diagram = 3 %then %do;
  * Three variables;

  data ABC;
    merge distinct_groupA(rename=(&varA=make) in=a) distinct_groupB(rename=(&varB=make)  in=b)
          distinct_groupC(rename=(&varC=make) in=c);
    by make;
    if a and b and c then ABC = 1;
    else ABC = 0;
    where make ne "";
  run;

  * Subquerying now to find the other elements;

  proc sql;
    create table AB1 as
    select coalesce(a.&varA, b.&varB) as make, 1 as AB,
    case when a.&varA = b.&varB then 1 else 0 end as check_AB
    from distinct_groupA as a full join distinct_groupB as b
    on a.&varA = b.&varB
      where calculated check_AB = 1 and calculated make not in (select make from ABC where ABC = 1)
    order by calculated make;
  quit;

  proc sql;
    create table AC1 as
    select coalesce(a.&varA, b.&varC) as make, 1 as AC,
    case when a.&varA = b.&varC then 1 else 0 end as check_AC
    from distinct_groupA as a full join distinct_groupC as b
    on a.&varA = b.&varC
      where calculated check_AC = 1 and calculated make not in (select make from ABC where ABC = 1)
    order by calculated make;
  quit;

  proc sql;
    create table BC1 as
    select coalesce(a.&varB, b.&varC) as make, 1 as BC,
    case when a.&varB = b.&varC then 1 else 0 end as check_BC
    from distinct_groupB as a full join distinct_groupC as b
    on a.&varB = b.&varC
      where calculated check_BC = 1 and calculated make not in (select make from ABC where ABC = 1)
    order by calculated make;
  quit;

* Subquery - elements only in A;
```

```
  proc sql;
    create table A1 as
    select &varA as make, 1 as A1
    from distinct_groupA where &varA not in (select make from AB1 where AB = 1)
                          and &varA not in (select make from AC1 where AC = 1)
                          and &varA not in (select make from ABC where ABC = 1)
    order by make;
  quit;

* Subquery - elements only in B;
  proc sql;
    create table B1 as
    select &varB as make, 1 as B1
    from distinct_groupB where &varB not in (select make from AB1 where AB = 1)
                          and &varB not in (select make from BC1 where BC = 1)
                          and &varB not in (select make from ABC where ABC = 1)
    order by make;
  quit;

* Subquery - elements only in C;
  proc sql;
    create table C1 as
    select &varC as make, 1 as C1
    from distinct_groupC where &varC not in (select make from AC1 where AC = 1)
                          and &varC not in (select make from BC1 where BC = 1)
                          and &varC not in (select make from ABC where ABC = 1)
    order by make;
  quit;
%end;

%if &venn_diagram = 4 %then %do;
  data ABCD;
    merge distinct_groupA(rename=(&varA=make) in=a) distinct_groupB(rename=(&varB=make)  in=b)
          distinct_groupC(rename=(&varC=make) in=c) distinct_groupD(rename=(&varD=make)  in=d);
    by make;
    if a and b and c and d then ABCD = 1;
    else ABCD = 0;
    where make ne "";
  run;

  * Subquery - elements only in ABC;
  proc sql;
    create table ABC1 as
    select coalesce(a.&varA, b.&varB, c.&varC) as make, 1 as ABC,
    case when a.&varA = b.&varB = c.&varC then 1 else 0 end as check_ABC
    from distinct_groupA as a full join distinct_groupB as b
    on a.&varA = b.&varB
      full join distinct_groupC as c
      on a.&varA = c.&varC and b.&varB = c.&varC
    where calculated make ne "" and calculated check_ABC = 1 and calculated make not in (select make from ABCD
where ABCD = 1)
    order by calculated make;
  quit;

  * Subquery - elements only in ABD;
  proc sql;
    create table ABD1 as
    select coalesce(a.&varA, b.&varB, c.&varD) as make, 1 as ABD,
    case when a.&varA = b.&varB = c.&varD then 1 else 0 end as check_ABD
    from distinct_groupA as a full join distinct_groupB as b
    on a.&varA = b.&varB
      full join distinct_groupD as c
      on a.&varA = c.&varD and b.&varB = c.&varD
    where calculated make ne "" and calculated check_ABD = 1 and calculated make not in (select make from ABCD
where ABCD = 1)
    order by calculated make;
  quit;

  * Subquery - elements only in ACD;
  proc sql;
    create table ACD1 as
    select coalesce(a.&varA, b.&varC, c.&varD) as make, 1 as ACD,
    case when a.&varA = b.&varC = c.&varD then 1 else 0 end as check_ACD
    from distinct_groupA as a full join distinct_groupC as b
    on a.&varA = b.&varC
      full join distinct_groupD as c
      on a.&varA = c.&varD and b.&varC = c.&varD
    where calculated make ne "" and calculated check_ACD = 1 and calculated make not in (select make from ABCD
where ABCD = 1)
    order by calculated make;
  quit;
```

```sas
 * Subquery - elements only in BCD;
 proc sql;
   create table BCD1 as
   select coalesce(a.&varB, b.&varC, c.&varD) as make, 1 as BCD,
   case when a.&varB = b.&varC = c.&varD then 1 else 0 end as check_BCD
   from distinct_groupB as a full join distinct_groupC as b
   on a.&varB = b.&varC
     full join distinct_groupD as c
     on a.&varB = b.&varC and a.&varB = c.&varD
   where calculated make ne "" and calculated check_BCD = 1 and calculated make not in (select make from ABCD
where ABCD = 1)
   order by calculated make;
 quit;

 * Subsquerying pair of elements now;
 proc sql;
   create table AB1 as
   select coalesce(a.&varA, b.&varB) as make, 1 as AB,
   case when a.&varA = b.&varB then 1 else 0 end as check_AB
   from distinct_groupA as a full join distinct_groupB as b
   on a.&varA = b.&varB
     where calculated make ne "" and calculated check_AB = 1
                             and calculated make not in (select make from ABC1 where ABC = 1)
                             and calculated make not in (select make from ABD1 where ABD = 1)
                             and calculated make not in (select make from ABCD where ABCD = 1)

   order by calculated make;
 quit;

 proc sql;
   create table AC1 as
   select coalesce(a.&varA, b.&varC) as make, 1 as AC,
   case when a.&varA = b.&varC then 1 else 0 end as check_AC
   from distinct_groupA as a full join distinct_groupC as b
   on a.&varA = b.&varC
     where calculated make ne "" and calculated check_AC = 1
                             and calculated make not in (select make from ABC1 where ABC = 1)
                             and calculated make not in (select make from ACD1 where ACD = 1)
                             and calculated make not in (select make from ABCD where ABCD = 1)
   order by calculated make;
 quit;


 proc sql;
   create table AD1 as
   select coalesce(a.&varA, b.&varD) as make, 1 as AD,
   case when a.&varA = b.&varD then 1 else 0 end as check_AD
   from distinct_groupA as a full join distinct_groupD as b
   on a.&varA = b.&varD
     where calculated make ne "" and calculated check_AD = 1
                             and calculated make not in (select make from ABD1 where ABD = 1)
                             and calculated make not in (select make from ACD1 where ACD = 1)
                             and calculated make not in (select make from ABCD where ABCD = 1)
   order by calculated make;
 quit;


 proc sql;
   create table BC1 as
   select coalesce(a.&varB, b.&varC) as make, 1 as BC,
   case when a.&varB = b.&varC then 1 else 0 end as check_BC
   from distinct_groupB as a full join distinct_groupC as b
   on a.&varB = b.&varC
     where calculated make ne "" and calculated check_BC = 1
                             and calculated make not in (select make from ABC1 where ABC = 1)
                             and calculated make not in (select make from BCD1 where BCD = 1)
                             and calculated make not in (select make from ABCD where ABCD = 1)
   order by calculated make;
 quit;

 proc sql;
   create table BD1 as
   select coalesce(a.&varB, b.&varD) as make, 1 as BD,
   case when a.&varB = b.&varD then 1 else 0 end as check_BD
   from distinct_groupB as a full join distinct_groupD as b
   on a.&varB = b.&varD
     where calculated make ne "" and calculated check_BD = 1
                             and calculated make not in (select make from ABD1 where ABD = 1)
                             and calculated make not in (select make from BCD1 where BCD = 1)
                             and calculated make not in (select make from ABCD where ABCD = 1)
```

```
      order by calculated make;
   quit;

   proc sql;
      create table CD1 as
      select coalesce(a.&varC, b.&varD) as make, 1 as CD,
      case when a.&varC = b.&varD then 1 else 0 end as check_CD
      from distinct_groupC as a full join distinct_groupD as b
      on a.&varC = b.&varD
        where calculated make ne "" and calculated check_CD = 1
                              and calculated make not in (select make from ACD1 where ACD = 1)
                              and calculated make not in (select make from BCD1 where BCD = 1)
                              and calculated make not in (select make from ABCD where ABCD = 1)
      order by calculated make;
   quit;

* Subsquerying single elements now;

* Subquery - elements only in A;
   proc sql;
      create table A1 as
      select &varA as make, 1 as A1
      from distinct_groupA where &varA not in (select make from AB1 where AB = 1)
                            and &varA not in (select make from AC1 where AC = 1)
                            and &varA not in (select make from AD1 where AD = 1)
                            and &varA not in (select make from ABC1 where ABC = 1)
                            and &varA not in (select make from ABD1 where ABD = 1)
                            and &varA not in (select make from ACD1 where ACD = 1)
                            and &varA not in (select make from ABCD where ABCD = 1)
      order by make;
   quit;

* Subquery - elements only in B;
   proc sql;
      create table B1 as
      select &varB as make, 1 as B1
      from distinct_groupB where &varB not in (select make from AB1 where AB = 1)
                            and &varB not in (select make from BC1 where BC = 1)
                            and &varB not in (select make from BD1 where BD = 1)
                            and &varB not in (select make from ABC1 where ABC = 1)
                            and &varB not in (select make from ABD1 where ABD = 1)
                            and &varB not in (select make from BCD1 where BCD = 1)
                            and &varB not in (select make from ABCD where ABCD = 1)
      order by make;
   quit;

* Subquery - elements only in C;
   proc sql;
      create table C1 as
      select &varC as make, 1 as C1
      from distinct_groupC where &varC not in (select make from AC1 where AC = 1)
                            and &varC not in (select make from BC1 where BC = 1)
                            and &varC not in (select make from CD1 where CD = 1)
                            and &varC not in (select make from ABC1 where ABC = 1)
                            and &varC not in (select make from ACD1 where ACD = 1)
                            and &varC not in (select make from BCD1 where BCD = 1)
                            and &varC not in (select make from ABCD where ABCD = 1)
      order by make;
   quit;


* Subquery - elements only in D;
   proc sql;
      create table D1 as
      select &varD as make, 1 as D1
      from distinct_groupD where &varD not in (select make from AD1 where AD = 1)
                            and &varD not in (select make from BD1 where BD = 1)
                            and &varD not in (select make from CD1 where CD = 1)
                            and &varD not in (select make from ABD1 where ABD = 1)
                            and &varD not in (select make from ACD1 where ACD = 1)
                            and &varD not in (select make from BCD1 where BCD = 1)
                            and &varD not in (select make from ABCD where ABCD = 1)
      order by make;
   quit;
%end;

%if &venn_diagram = 2 %then %do;
   data merged_elements;
      merge AB A1 B1;
      by make;
      where make ne "";
```

```
      array elements (*) _numeric_;
      do i = 1 to dim(elements);
        if elements(i) = . then elements(i) = 0;
      end;
      id = _n_;
      drop i;
    run;
%end;

%if &venn_diagram = 3 %then %do;
  data merged_elements;
    merge ABC AB1 AC1 BC1 A1 B1 C1;
    by make;
    where make ne "";
    array elements (*) _numeric_;
    do i = 1 to dim(elements);
      if elements(i) = . then elements(i) = 0;
    end;
    drop check: i;
    id = _n_;
  run;
%end;

%if &venn_diagram = 4 %then %do;
  data merged_elements;
    merge ABCD BCD1 ACD1 ABD1 ABC1 AB1 AC1 AD1 BC1 BD1 CD1 A1 B1 C1 D1;
    by make;
    where make ne "";
    array elements (*) _numeric_;
    do i = 1 to dim(elements);
      if elements(i) = . then elements(i) = 0;
    end;
    drop check: i;
    id = _n_;
  run;
%end;

proc univariate data = merged_elements noprint;
  var AB A1 B1
  %if &venn_diagram > 2 %then %do;
    ABC AC BC C1
  %end;
  %if &venn_diagram > 3 %then %do;
    ABCD ABD ACD BCD AD BD CD D1
  %end;
  ;
  output out = data_sum sum = sum_AB sum = sum_A1 sum = sum_B1
  %if &venn_diagram > 2 %then %do;
    sum = sum_ABC sum = sum_AC sum = sum_BC sum = sum_C1
  %end;
  %if &venn_diagram > 3 %then %do;
    sum = sum_ABCD sum = sum_ABD sum = sum_ACD
    sum = sum_BCD sum = sum_AD sum = sum_BD
    sum = sum_CD sum = sum_D1
  %end;
  ;
run;

/* Counting the number in the universal set */
proc sql noprint;
  create table id_count as
  select count(id) as count_id
  from merged_elements;
quit;
/* Counting the number inside the union */
data data_sum2;
  set data_sum;
  totalinside = sum(sum_AB, sum_A1, sum_B1
  %if &venn_diagram > 2 %then %do;
    ,sum_ABC, sum_AC, sum_BC, sum_C1
  %end;
  %if &venn_diagram > 3 %then %do;
    ,sum_ABCD, sum_ABD, sum_ACD, sum_BCD, sum_AD,
    sum_BD, sum_CD, sum_D1
  %end;
  );
run;


/* Assigning the sums to macro variables */
proc sql noprint;
```

```
   select strip(put(sum_A1, best.)), strip(put(sum_B1, best.)), strip(put(sum_AB, best.)) into :A, :B, :AB
   from data_sum2;
quit;

* Calculating percentages;
proc sql noprint;
   select put(sum_A1/totalinside*100, pctpic.), put(sum_B1/totalinside*100, pctpic.),
put(sum_AB/totalinside*100, pctpic.) into :A_per, :B_per, :AB_per
   from data_sum2;
quit;

%if &venn_diagram > 2 %then %do;
   proc sql noprint;
      select strip(put(sum_C1, best.)), strip(put(sum_AC, best.)), strip(put(sum_BC, best.)), strip(put(sum_ABC,
best.)) into :C, :AC, :BC, :ABC
      from data_sum2;
   quit;

* Calculating percentages;
   proc sql noprint;
      select put(sum_C1/totalinside*100, pctpic.), put(sum_AC/totalinside*100, pctpic.),
put(sum_BC/totalinside*100, pctpic.),  put(sum_ABC/totalinside*100, pctpic.)
         into :C_per, :AC_per, :BC_per, :ABC_per
      from data_sum2;
   quit;
%end;
%if &venn_diagram > 3 %then %do;
   proc sql noprint;
      select strip(put(sum_D1, best.)), strip(put(sum_AD, best.)), strip(put(sum_BD, best.)), strip(put(sum_CD,
best.)),
         strip(put(sum_ABD, best.)), strip(put(sum_ACD, best.)), strip(put(sum_BCD, best.)), strip(put(sum_ABCD,
best.))
         into :D, :AD, :BD, :CD, :ABD, :ACD, :BCD, :ABCD
      from data_sum2;
   quit;

   proc sql noprint;
      select put(sum_D1/totalinside*100, pctpic.), put(sum_AD/totalinside*100, pctpic.),
put(sum_BD/totalinside*100, pctpic.), put(sum_CD/totalinside*100, pctpic.),
         put(sum_ABD/totalinside*100, pctpic.), put(sum_ACD/totalinside*100, pctpic.),
put(sum_BCD/totalinside*100, pctpic.), put(sum_ABCD/totalinside*100, pctpic.)
         into :D_per, :AD_per, :BD_per, :CD_per, :ABD_per, :ACD_per, :BCD_per, :ABCD_per
      from data_sum2;
quit;
%end;

%if &proportional = 0 %then %do;

   /* The rest of the macro needs to be done seperately for 2, 3 and 4
   way plots */
   data test;
      do x = 1 to 100;
      y = x;
      output;
      end;
   run;

   /*************** 2 WAY VENN DIAGRAMS ***************/
   %if &venn_diagram=2 %then %do;
      proc template;
         define statgraph Venn2Way;
            begingraph / drawspace=datavalue;
            /* Plot */
            layout overlay / yaxisopts = (display = NONE) xaxisopts = (display = NONE);
               scatterplot x=x y=y / markerattrs=(size = 0);
               /* Venn Diagram (Circles) */
               drawoval x=36 y=50 width=45 height=60 /
                  display=all fillattrs=(color=red)
                  transparency=0.75 WIDTHUNIT= Percent HEIGHTUNIT= Percent;
               drawoval x=63 y=50 width=45 height=60 /
                  display=all fillattrs=(color=green)
                  transparency=0.75 WIDTHUNIT= Percent HEIGHTUNIT= Percent;
               /* Numbers */
               drawtext "&A" / x=33 y=50 anchor=center;
               drawtext "&AB" / x=50 y=50 anchor=center;
               drawtext "&B" / x=66 y=50 anchor=center;

                      /* Percentages */
               %if &percentage = 1 %then %do;
                  drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&A_per" / x=33 y=45 anchor=center ;
                  drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&AB_per" / x=50 y=45 anchor=center;
```

```
            drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&B_per" / x=66 y=45 anchor=center;
        %end;

        /* Labels */
        drawtext "&LabelA" / x=30 y=15 anchor=center width = 30;
        drawtext "&LabelB" / x=70 y=15 anchor=center width = 30;
      endlayout;
      endgraph;
    end;
  run;
%end;
/*************** 3 WAY VENN DIAGRAMS ***************/
%if &venn_diagram = 3 %then %do;
  proc template;
    define statgraph Venn3Way;
      begingraph / drawspace=datavalue;
        /* Plot */
      layout overlay / yaxisopts = (display = NONE) xaxisopts = (display = NONE);
        scatterplot x=x y=y / markerattrs=(size = 0);
       /* Venn Diagram (Circles) */
        drawoval x=37 y=40 width=45 height=60 /
          display=all fillattrs=(color=red)
          transparency=0.75 WIDTHUNIT= Percent HEIGHTUNIT= Percent;
        drawoval x=63 y=40 width=45 height=60 /
          display=all fillattrs=(color=green)
          transparency=0.75 WIDTHUNIT= Percent HEIGHTUNIT= Percent;
        drawoval x=50 y=65 width=45 height=60 /
          display=all fillattrs=(color=blue)
          transparency=0.75 WIDTHUNIT= Percent HEIGHTUNIT= Percent;
        /* Numbers */
        drawtext "&A" / x=32 y=35 anchor=center;
        drawtext "&AB" / x=50 y=30 anchor=center;
        drawtext "&B" / x=68 y=35 anchor=center;
        drawtext "&ABC" / x=50 y=50 anchor=center;
        drawtext "&AC" / x=37 y=55 anchor=center;
        drawtext "&BC" / x=63 y=55 anchor=center;
        drawtext "&C" / x=50 y=75 anchor=center;

        /* Percentages */
        %if &percentage = 1 %then %do;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&A_per" / x=32 y=32 anchor=center;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&AB_per" / x=50 y=27 anchor=center;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&B_per" / x=68 y=32 anchor=center;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&ABC_per" / x=50 y=47 anchor=center;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&AC_per" / x=37 y=52 anchor=center;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&BC_per" / x=63 y=52 anchor=center;
          drawtext textattrs = GRAPHVALUETEXT(size = 8pt weight = bold) "&C_per" / x=50 y=72 anchor=center;
        %end;

                /* Labels */
        drawtext "&LabelA" / x=30 y=7 anchor=center width = 30;
        drawtext "&LabelB" / x=70 y=7 anchor=center width = 30;
        drawtext "&LabelC" / x=50 y=98 anchor=center width = 30;
      endlayout;
      endgraph;
    end;
  run;
%end;
/*************** 4 WAY VENN DIAGRAMS ***************/
%if &venn_diagram = 4 %then %do;
  proc template;
    define statgraph Venn4Way;
      begingraph / drawspace=datavalue;
        /* Plot */
      layout overlay / yaxisopts = (display = NONE) xaxisopts = (display = NONE);
        scatterplot x=x y=y / markerattrs=(size = 0);
        /* Venn Diagram (Ellipses) */
        drawoval x=28 y=39 width=26 height=100 / display=all fillattrs=(color=red transparency=0.85)
          outlineattrs=(color=red) transparency=0.50 WIDTHUNIT= Percent HEIGHTUNIT= Percent rotate = 45 ;
        drawoval x=72 y=39 width=26 height=100 / display=all fillattrs=(color=green transparency=0.85)
          outlineattrs=(color=green) transparency=0.50 WIDTHUNIT= Percent HEIGHTUNIT= Percent rotate = 315 ;
        drawoval x=57 y=54 width=26 height=100 / display=all fillattrs=(color=blue transparency=0.85)
          outlineattrs=(color=blue) transparency=0.50 WIDTHUNIT= Percent HEIGHTUNIT= Percent rotate = 335 ;
        drawoval x=43 y=54 width=26 height=100 / display=all fillattrs=(color=yellow transparency=0.85)
          outlineattrs=(color=yellow) transparency=0.50 WIDTHUNIT= Percent HEIGHTUNIT= Percent rotate = 25 ;
        /* Numbers */
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&A" / x=13 y=60 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&B" / x=35 y=80 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&C" / x=65 y=80 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&D" / x=87 y=60 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&AB" / x=36 y=45 anchor=center;
```

```
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&AC" / x=41 y=16 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&AD" / x=50 y=6 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&BC" / x=50 y=55 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&BD" / x=59 y=16 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&CD" / x=64 y=45 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&ABC" / x=43 y=30 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&BCD" / x=57 y=30 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&ACD" / x=46 y=12 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&ABD" / x=52 y=12 anchor=center;
        drawtext textattrs = GRAPHVALUETEXT(size = 6pt weight = bold) "&ABCD" / x=50 y=21 anchor=center;

        /* Labels */
        drawtext textattrs = GRAPHVALUETEXT(size = 7pt weight = bold) "&LabelA" / x=6 y=20 anchor=center
width = 30;
        drawtext textattrs = GRAPHVALUETEXT(size = 7pt weight = bold) "&LabelB" / x=6 y=85 anchor=center
width = 30;
        drawtext textattrs = GRAPHVALUETEXT(size = 7pt weight = bold) "&LabelC" / x=82 y=85 anchor=center
width = 30;
        drawtext textattrs = GRAPHVALUETEXT(size = 7pt weight = bold) "&LabelD" / x=82 y=20 anchor=center
width = 30;
      endlayout;
      endgraph;
    end;
  run;
  %end;

  ********* Outputting the graph *********;
    ods graphics on / reset = all border = off width=16cm height=12cm imagefmt = png imagename =
"&outputfilename";
    ods listing gpath = "&out_location" image_dpi = 200;
    proc sgrender data=test template=Venn&venn_diagram.Way;
    run;
    ods listing close;
    ods graphics off;
  ****** Finish outputting the graph *****;

%end;

* Producing Google Charts now;

%if &proportional ^= 0 %then %do;

  %if &venn_diagram = 2 %then %do;
    * Need to use the Set Totals and Intercept Totals;
    %let totalA = %eval(&A + &AB);
    %let totalB = %eval(&B + &AB);
    %let totalC = 0;

    %let totalAB = &AB;

    %if &prop_n_label ^= 1 %then %do;
      %let lab1 = %str(&LabelA);
      %let lab2 = %str(&LabelB);

      %macro VennChart (size, type, col1, col2, col3, dat1, dat2, dat3, dat4, lab1, lab2);
      data _null_;
        file "&out_location\&outputfilename..html";
        put '<img src="https://chart.googleapis.com/chart?
cht='"&type."'
&chd=t:'"&dat1."','"&dat2."','"&dat3."','"&dat4."'
&chs='"&size."'
&chco='"&col1."','"&col2."','"&col3."'
&chdl='"&lab1."'|'"&lab2."'"'/>';
      run;
      %mend VennChart;
      %VennChart (500x500,v,ed7878,90d190,7e7ed3,&totalA.,&TotalB.,&TotalC.,&TotalAB.,&lab1,&lab2);
    %end;

    * Displaying N number in the legend if appropriate;
    %if &prop_n_label = 1 %then %do;
      %let lab1 = %str(&LabelA: N=&A);
      %let lab2 = %str(&LabelB: N=&B);
      %let lab3 = %str(Intersection: N=&AB);

      %macro VennChart2 (size, type, col1, col2, col3, dat1, dat2, dat3, dat4, lab1, lab2, lab3);
      data _null_;
        file "&out_location\&outputfilename..html";
        put '<img src="https://chart.googleapis.com/chart?
cht='"&type."'
&chd=t:'"&dat1."','"&dat2."','"&dat3."','"&dat4."'
&chs='"&size."'
```

```
&chco='"&col1."','"&col2."','"&col3."'
&chdl='"&lab1."'|'"&lab2."'|'"&lab3."'"/>';
      run;
      %mend VennChart2;
      %VennChart2 (500x500,v,ed7878,90d190,7e7ed3,&totalA.,&TotalB.,&TotalC.,&TotalAB.,&lab1,&lab2, &lab3);
    %end;
  %end;


  %if &venn_diagram = 3 %then %do;
    * Need to use the Set Totals and Intercept Totals;
    %let totalA = %eval(&A + &AB + &AC + &ABC);
    %let totalB = %eval(&B + &AB + &BC + &ABC);
    %let totalC = %eval(&C + &AC + &BC + &ABC);

    %let totalAB = %eval(&AB + &ABC);
    %let totalAC = %eval(&AC + &ABC);
    %let totalBC = %eval(&BC + &ABC);

    %if &prop_n_label ^= 1 %then %do;

      %let lab1 = %str(&LabelA);
      %let lab2 = %str(&LabelB);
      %let lab3 = %str(&LabelC);

      %macro VennChart3 (size, type, col1, col2, col3, dat1, dat2, dat3, dat4, dat5, dat6, dat7, lab1, lab2,
lab3);
      data _null_;
        file "&out_location\&outputfilename..html";
        put '<img src="https://chart.googleapis.com/chart?
cht='"&type."'
&chd=t:'"&dat1."','"&dat2."','"&dat3."','"&dat4."','"&dat5."','"&dat6."','"&dat7."'
&chs='"&size."'
&chco='"&col1."','"&col2."','"&col3."'
&chdl='"&lab1."'|'"&lab2."'|'"&lab3."'"/>';
      run;
      %mend VennChart3;
      %VennChart3
(500x500,v,ed7878,90d190,7e7ed3,&totalA.,&TotalB.,&TotalC.,&TotalAB.,&TotalAC.,&TotalBC.,&ABC.,&lab1,&lab2,
&lab3);

    %end;

    * Displaying N number in the legend if appropriate;
    %if &prop_n_label = 1 %then %do;
      %let lab1 = %str(&LabelA: N=&A);
          %let lab2 = %str(&LabelB: N=&B);
      %let lab3 = %str(&LabelC: N=&C);
          %let lab4 = %str(Intersection of &LabelA and &LabelB: N=&AB);
          %let lab5 = %str(Intersection of &LabelA and &LabelC: N=&AC);
          %let lab6 = %str(Intersection of &LabelB and &LabelC: N=&BC);
          %let lab7 = %str(Intersection of &LabelA and &LabelB and &LabelC: N=&ABC);

      %macro VennChart4 (size, type, col1, col2, col3, dat1, dat2, dat3, dat4, dat5, dat6, dat7, lab1, lab2,
lab3, lab4, lab5, lab6, lab7);
      data _null_;
        file "&out_location\&outputfilename..html";
        put '<img src="https://chart.googleapis.com/chart?
cht='"&type."'
&chd=t:'"&dat1."','"&dat2."','"&dat3."','"&dat4."','"&dat5."','"&dat6."','"&dat7."'
&chs='"&size."'
&chco='"&col1."','"&col2."','"&col3."'
&chdl='"&lab1."'|'"&lab2."'|'"&lab3."'|'"&lab4."'|'"&lab5."'|'"&lab6."'|'"&lab7."'"/>';
      run;
      %mend VennChart4;
      %VennChart4
(500x500,v,ed7878,90d190,7e7ed3,&totalA.,&TotalB.,&TotalC.,&TotalAB.,&TotalAC.,&TotalBC.,&ABC.,&lab1,&lab2,&lab
3,&lab4,&lab5,&lab6,&lab7 );
    %end; %* End of prop_n_label = 1 check;
  %end; %* End of venn_diagram = 4 check;
%end; %* End of proportional ne 0 check;

* Clean up;
data distinct_elements;
  set merge_distinct;
run;

proc datasets library = work memtype=data noprint;
  save venndata distinct_elements Merged_elements;
run;
quit;
```

```
%exit: %mend venn; %* End of Macro;

* Figure 1;
%venn(dsn= venndata, proportional = 0, venn_diagram = 2, varA=makeA, varB=makeB, varC=makeC, varD=makeD,
percentage = 0, prop_n_label = 0, outputfilename=%str(Figure 1));

* Figure 2;
%venn(dsn= venndata, proportional = 0, venn_diagram = 2, varA=makeA, varB=makeB, varC=makeC, varD=makeD,
percentage = 1, prop_n_label = 0, outputfilename=%str(Figure 2));

* Figure 3;
%venn(dsn= venndata, proportional = 0, venn_diagram = 3, varA=makeA, varB=makeB, varC=makeC, varD=makeD,
percentage = 0, prop_n_label = 0, outputfilename=%str(Figure 3));

* Figure 4;
%venn(dsn= venndata, proportional = 0, venn_diagram = 4, varA=makeA, varB=makeB, varC=makeC, varD=makeD,
percentage = 0, prop_n_label = 0, outputfilename=%str(Figure 4));

* Figure 5;
%venn(dsn= venndata, proportional = 1, venn_diagram = 2, varA=makeA, varB=makeB, varC=makeC, varD=makeD,
percentage = 0, prop_n_label = 0, outputfilename=%str(Figure 5));

* Figure 6;
%venn(dsn= venndata, proportional = 1, venn_diagram = 2, varA=makeA, varB=makeB, varC=makeC, varD=makeD,
percentage = 0, prop_n_label = 1, outputfilename=%str(Figure 6));

* Figure 7;
%venn(dsn= venndata, proportional = 1, venn_diagram = 3, varA=makeA, varB=makeB, varC=makeC, varD=makeD,
percentage = 0, prop_n_label = 0, outputfilename=%str(Figure 7));
```