

CT12

Fun with Figures

PHUSE Connect EU 2025
Miriam Amor Miriam.amor@veramed.co.uk

The logo for Veramed, featuring the word "Veramed" in a bold, sans-serif font. The "Vera" part is in a vibrant pink color, and the ".med" part is in a light blue color. The dot of the period is a small, solid blue circle.

Introduction



Emojis, memes, pictures, diagrams, etc: images are increasingly used in the way we communicate... and clinical trials are not the exception.



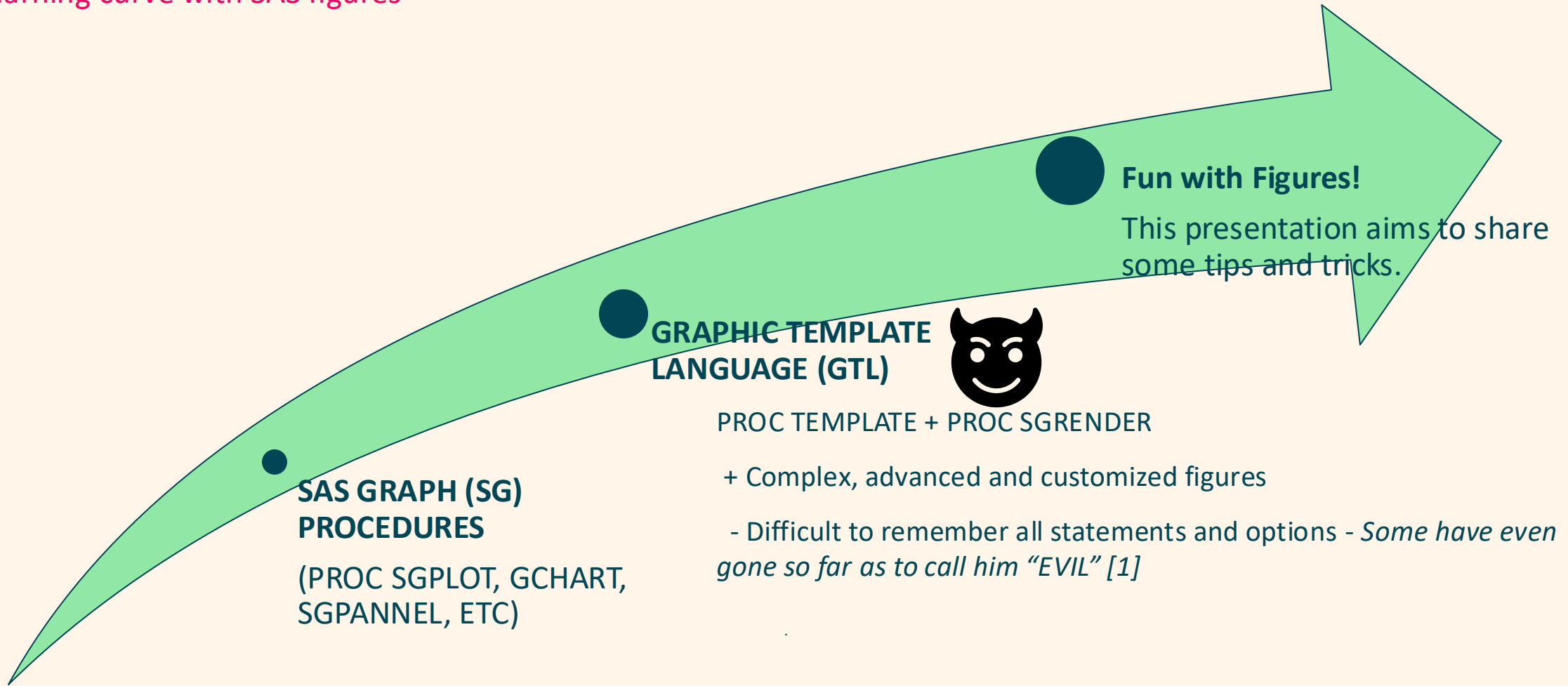
Graphics are not only a powerful tool for data-driven decision making, but also an artistic way of displaying data.



With this new trend, project leads are on the hunt for skilled programmer in SAS.

Introduction

Learning curve with SAS figures



TIP 1: PLOT TWIST - From SG To GTL

The TMPLOUT option

Annotated dataset

function	label	anchor	width	textsize	x1space	y1space	x1	y1	textweight	height
text	Median (95CI)		20	7	datavalue	datavalue	1500	0.87		
text	418(92, NC)		20	7	datavalue	datavalue	1500	0.85		
text	ALL	left	20	7	datavalue	datavalue	950	0.85	bold	
text	...	...	...	...	...	...	...	...	...	
rectangle		topleft	28		datavalue	datavalue	948	0.86		8

```
proc sgplot data=mydata sganno=myannot
  tmplout="C:\Documents\Figures\mypgm.sas";
  step    x=time y=survival/group=stratum;
  scatter x=time y=censored/group=stratum;
  xaxistable numberatrisk / x=timelist
  class=stratum colorgroup=stratum
  valueattrs=(weight=bold);
run;
```

```
proc template;
  define statgraph sgplot;
  beginngraph / collation=binary;
  layout lattice / columnweights=preferred
  rowweights=preferred
  columndatarange=union rowdatarange=union
  columns=1;
  ...
```

TIP 1: PLOT TWIST - From SG To GTL

The TMPLOUT option

Code created by TMPLOUT in C:\Documents\Figures\mypgm.sas

```
proc template;
```

```
define statgraph sgplot;
begingraph / collation=binary;
```

Creates a grid with 1 column and as many rows as cells defined

```
layout lattice / columnweights=preferred rowweights=preferred
columnatarange=union rowatarange=union columns=1;
```

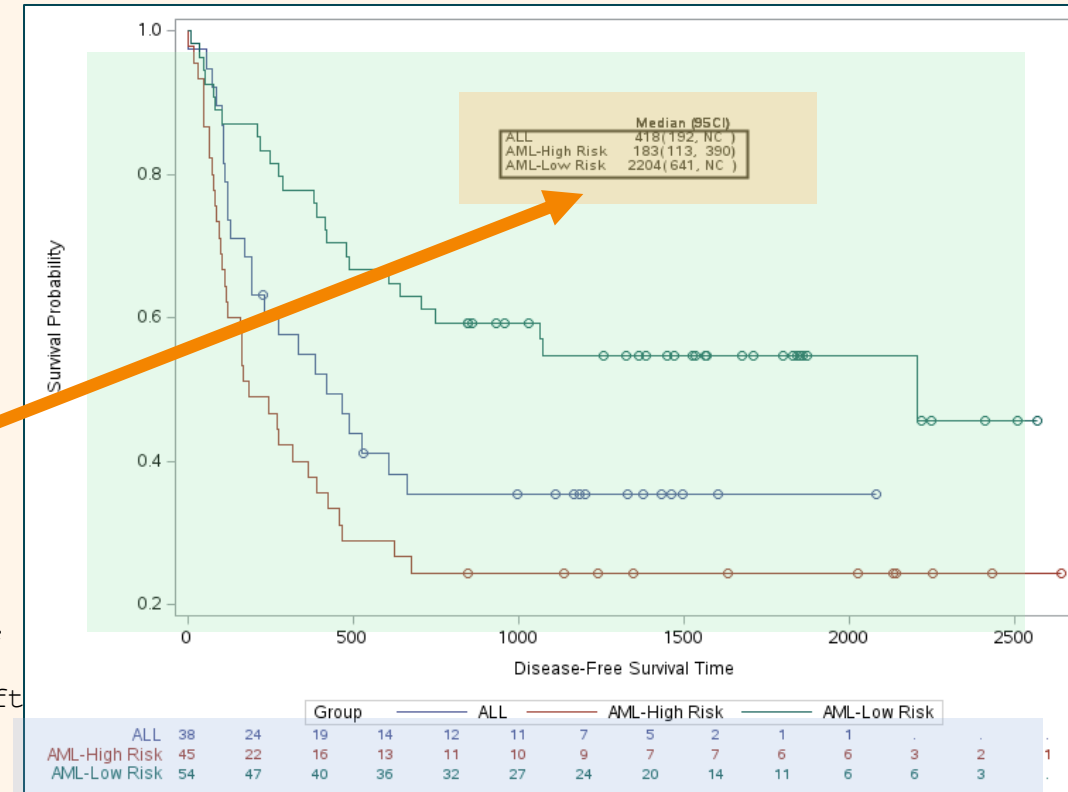
```
layout overlay;
StepPlot X=TIME Y=SURVIVAL / primary=true
Group=STRATUM LegendLabel="Survival Probability" NAME="STEP";
ScatterPlot X=TIME Y=CENSORED / subpixel=off Group=STRATUM
```

```
DrawText TEXTATTRS=( SIZE=7) "Median (95CI)"
/ X=1500 Y=0.9 XSPACE=datavalue YSPACE=datavalue WIDTH=20;
DrawText TEXTATTRS=( SIZE=7) " 418 ( 192, N
/ X=1500 Y=0.85 XSPACE=datavalue YSPACE=datavalue WIDTH=20;
DrawText TEXTATTRS=( SIZE=7 WEIGHT=bold) "ALL"
/ X=950 Y=0.85 XSPACE=datavalue YSPACE=datavalue ANCHOR=left
```

Hard-coding

```
endlayout;
```

```
Layout Overlay / walldisplay=none xaxisopts=(display=none griddisplay=off
displaySecondary=none) x2axisopts=(display=none griddisplay=off displaySecondary=none);
AxisTable Value=NUMBERATRISK X=TIMELIST / labelPosition=min Class=STRATUM ValueAttrs=(
colorGroup=STRATUM Display=(Label);
```

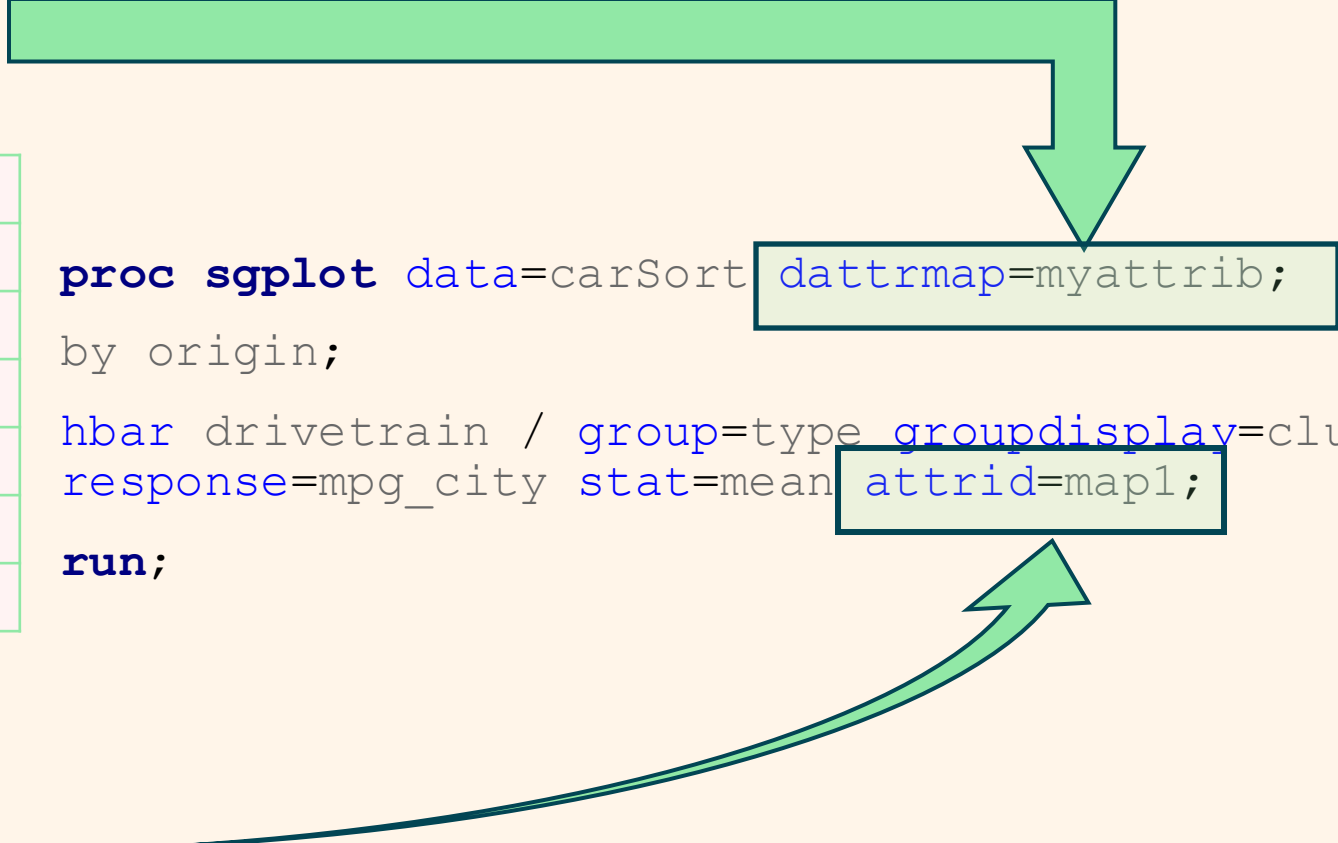


TIP 2: Figures in style

The attributes map

Dataset name: MYATTRIB

id	value	fillcolor	show
map1	Hybrid	blue	ATTRMAP
map1	SUV	yellow	ATTRMAP
map1	Sedan	orange	ATTRMAP
map1	Sports	lipk	ATTRMAP
map1	Truck	green	ATTRMAP
map1	Wagon	purple	ATTRMAP



```
proc sgplot data=carSort datrmap=myattrib;  
by origin;  
hbar drivetrain / group=type groupdisplay=cluster  
response=mpg_city stat=mean attrid=map1;  
run;
```



TIP 2: Figures in style

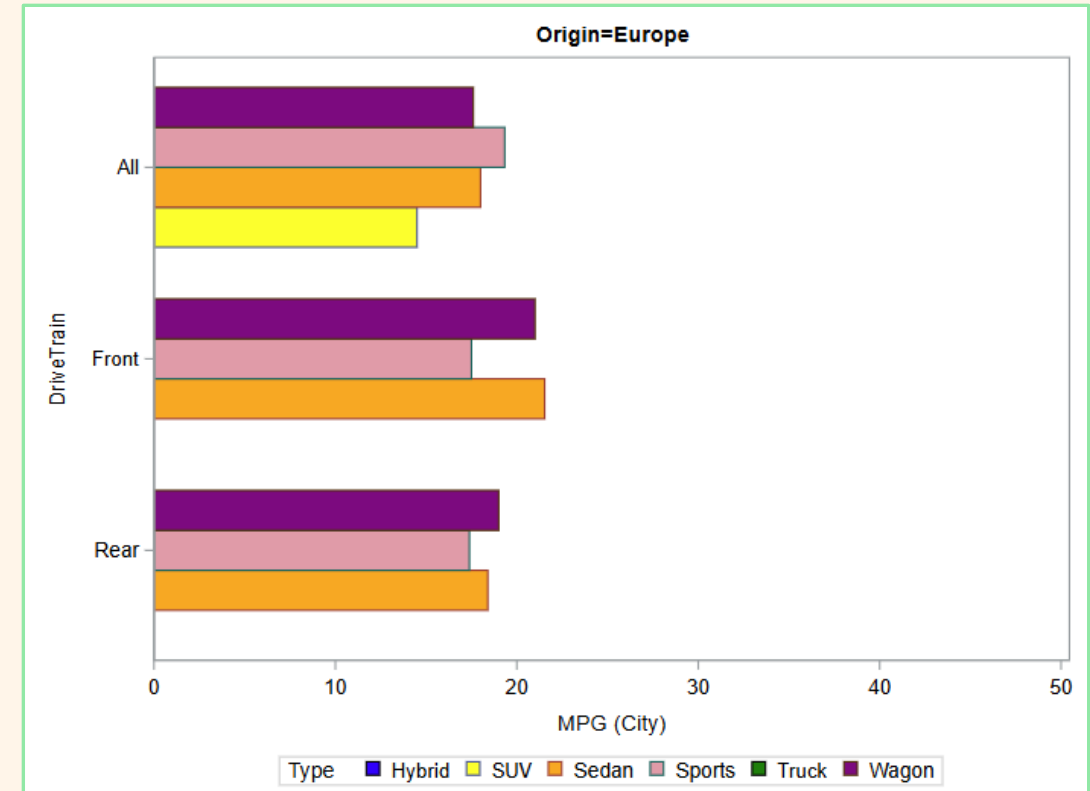
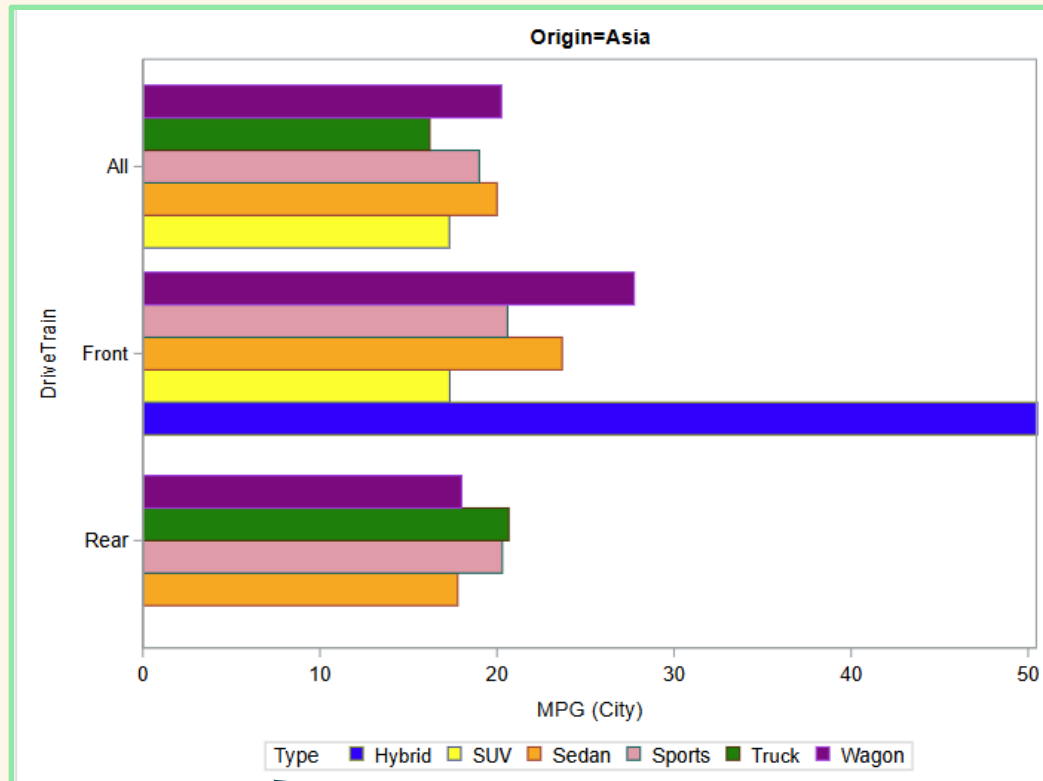
The attributes map

```
proc template;
define statgraph mysgplot;

begingraph / collation=binary;
  DiscreteAttrVar attrvar=MAP1_TYPE var=TYPE attrmap="__ATTRMAP_MAP1";
  DiscreteAttrMap name="__ATTRMAP_MAP1" / discreteLegendEntryPolicy=attrmap;
    Value "Hybrid" / fillattrs=(color=blue);
    Value "SUV" / fillattrs=(color=yellow);
    Value "Sedan" / fillattrs=(color=orange);
    Value "Sports" / fillattrs=( color=lipk);
    Value "Truck" / fillattrs=( color=green);
    Value "Wagon" / fillattrs=( color=purple);
  EndDiscreteAttrMap;
...
layout overlay/ ...
  BarChartParm X='DriveTrain'n Y='_Mean1_MPG_City_'n / primary=true orient=horizontal
  Group=MAP1_TYPE LegendLabel="MPG (City)" NAME="HBAR" groupdisplay=cluster;
endlayout;
```

TIP 2: Figures in style

The attributes map



Type is displayed in same order as in the attributes dataset

Colours consistent

All categories shown in the legend

Source: [Blum and Duggins, 2021](#) [3]

TIP 3: Dynamic thinking

One template, many figures

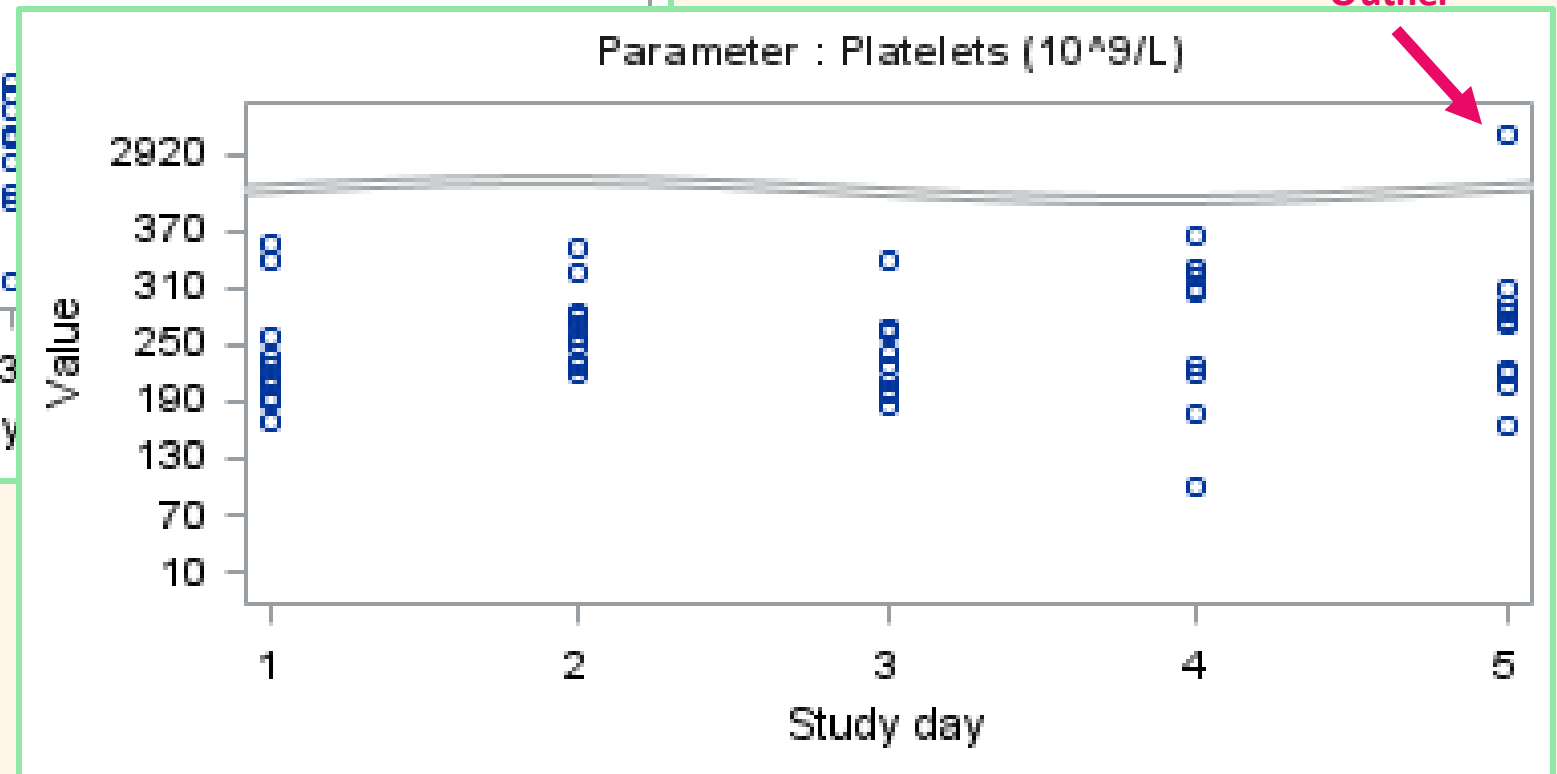
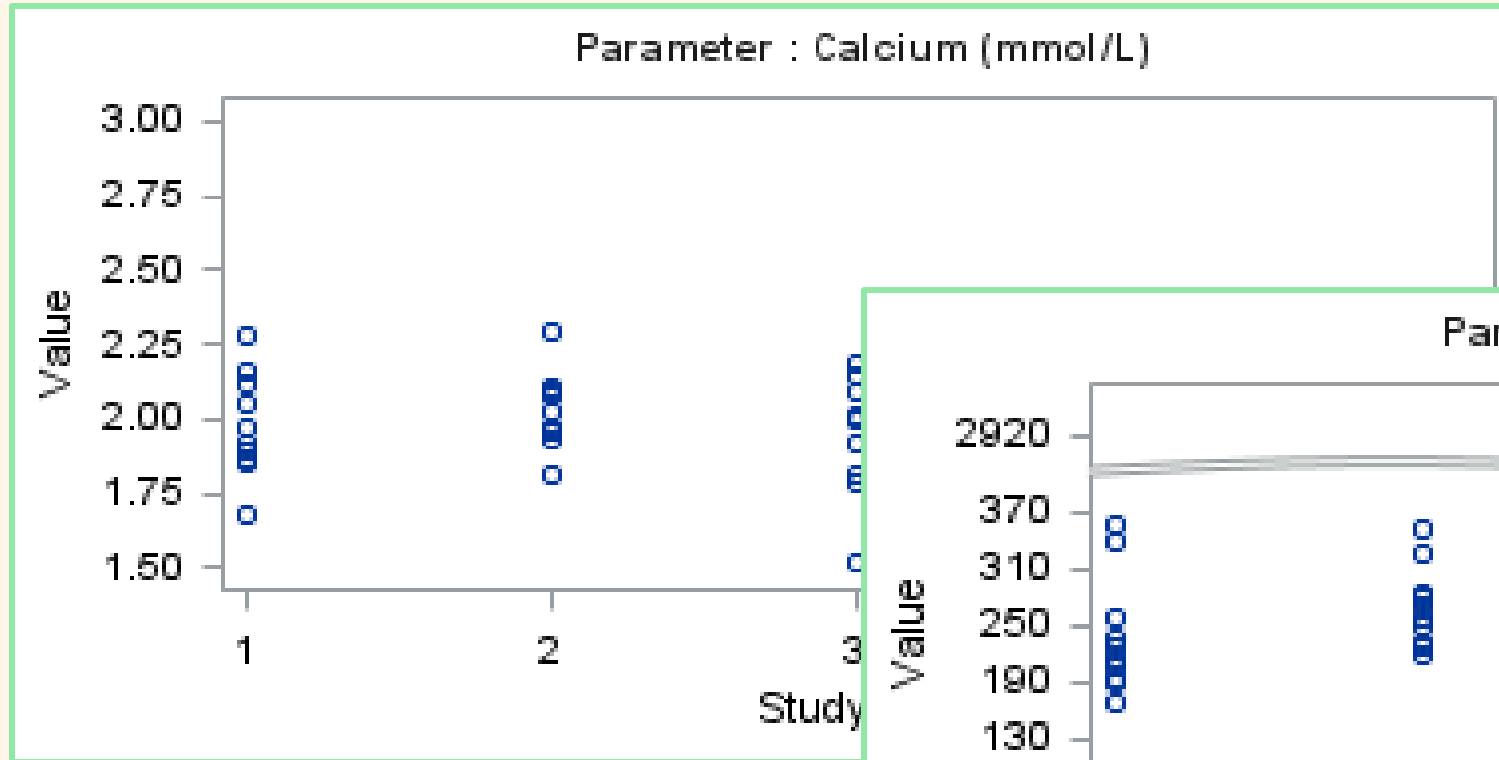
```
proc template;
  define statgraph mytemplate/store=work.templat;
  begingraph;
  dynamic _byval_ _byvar_ _byval2_ _byvar2_; /* Dynamic var definitions: corresponds to the variables and its
                                              values stated in the PROC SRGENDER */
  entrytitle "Parameter : " _byval2_ ;

  if ( _byval2_ = 'Calcium (mmol/L)')
    layout overlay/
      yaxisopts=(label='Value' linearopts=(viewmax=3)) ; /* Max value in y-axis=3*/
      scatterplot x=ady y=aval ;
    endlayout;
  else
    layout overlay/
      yaxisopts=( display=(line ticks tickvalues label) label='Value'
      linearopts=(includeranges=(1-400 2900-2950) )) ; /* Includes an axis-break*/
      scatterplot x=ady y=aval ;
    endlayout;
  endif;
endgraph;
end;
run;
```

```
proc sgrender template=mytemplate data=mydata;
  by paramn param; /* _byvar1 & _byvar2_*/
run;
```

TIP 3: Dynamic thinking

One template, many figures

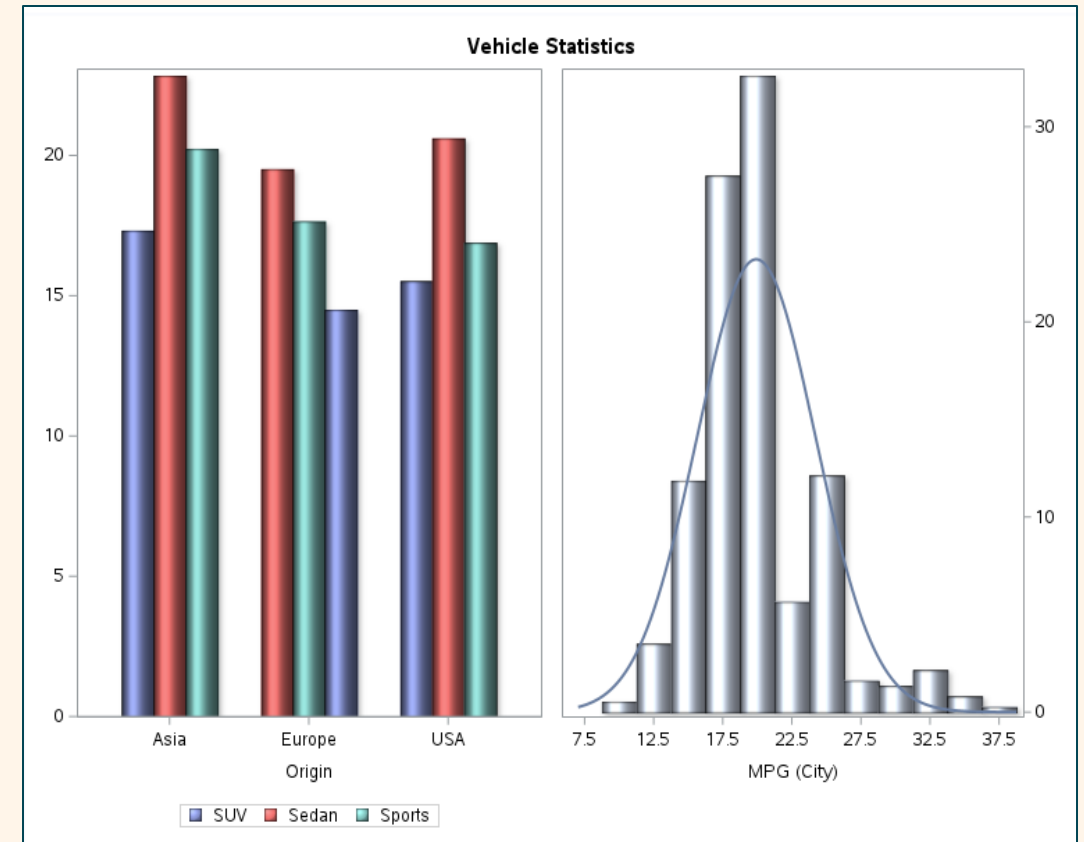


TIP 4: Mixing it up

Creating two figures in one panel



```
proc template;  
  define statgraph pannels;  
    begingraph;  
    entrytitle "Vehicle Statistics";  
    layout lattice / columns=2;  
  
    {  
      layout overlay /  
        barchart x=origin y=mpg_city /stat=mean  
        group=type groupdisplay=cluster  
      endlayout;  
  
      {  
        layout overlay /  
          histogram mpg_city ;  
          densityplot mpg_city;  
        endlayout;  
      }  
    }  
  ...  
end;
```

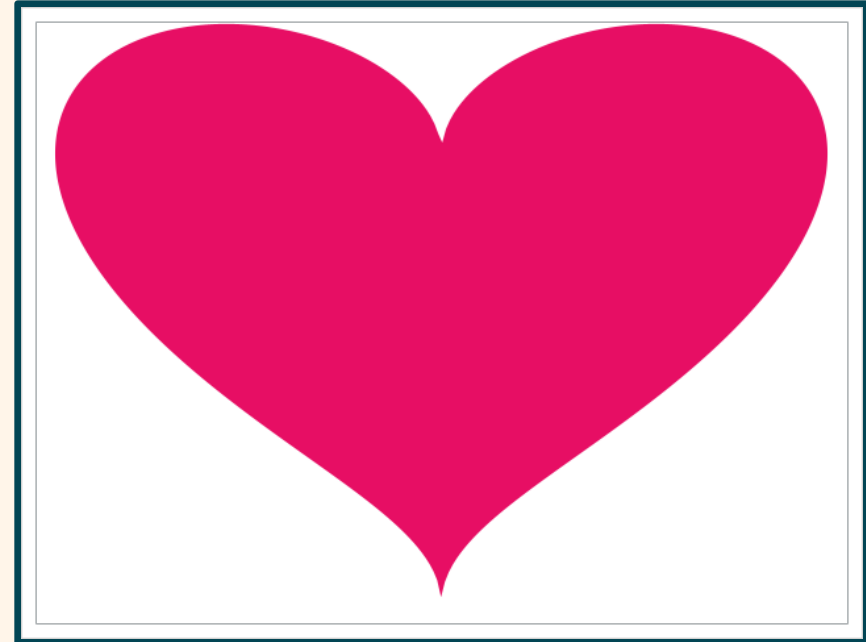


TIP 5: From maths to art

Linking the dots



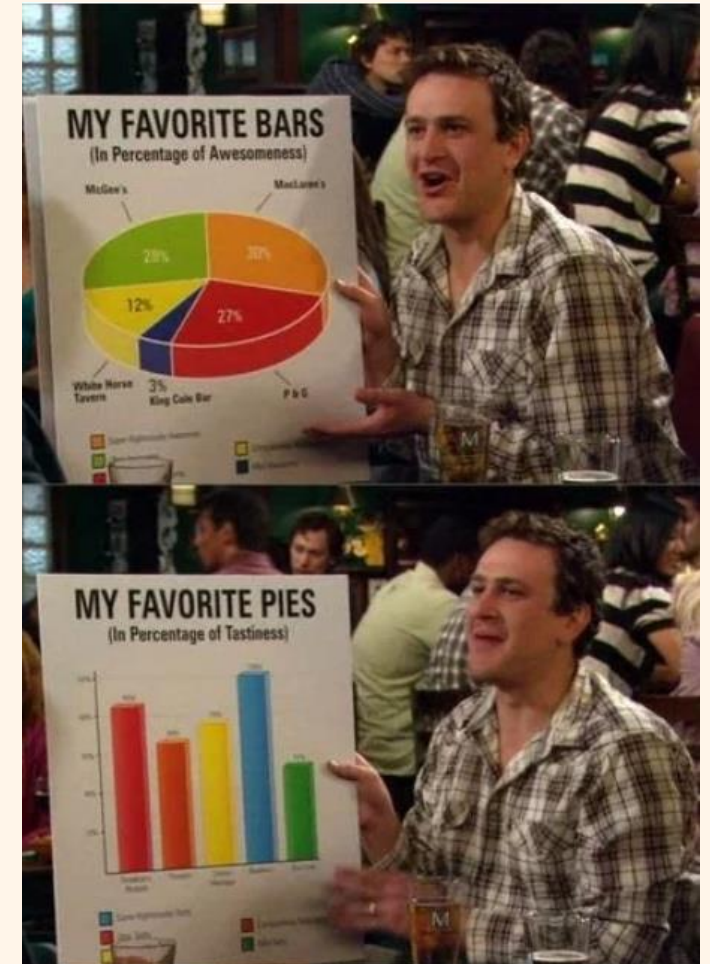
$$\begin{cases} x = \cos(t) \\ y = \sin(t) + \sqrt{|\cos(t)|} \end{cases}; t \in [-\pi, \pi]$$



```
/*Derive data points x1 and y1*/  
data mydata;  
  id='poll';  
  do t=-3.1416 to 3.1416 by 0.01;  
    x1=cos(t);  
    y1=sin(t) + sqrt(abs(cos(t)));  
    output;  
  end;  
run;  
  
/*Create the figure using polygon option */  
proc sgplot data=mydata;  
  polygon x=x1 y=y1 id=id/ fill fillattrs=(color=CXE70E64);  
  xaxis values=(-1 to 1) display=none;  
  yaxis values=(-1 to 2) display=none;  
run;
```

Conclusions

- Figures in SAS doesn't have to be intimidating.
- It can be a creative process when one knows the right tools:
 - Reveal hidden GTL code with **tplout** option
 - Ensure consistent styling with **attribute maps**
 - Create grids with **layout lattice**
 - Build reusable templates with **dynamic variables**
 - Visualize math with **polygon** statement



Source: HIMYM TV Show.



Any Questions?