SAS – Visualization

# Histograms

# Bar Charts/column Chart

# Pie Charts

# Scatter Plots

# Box Plots

SAS - Histograms

A Histogram is graphical display of data using bars of different heights. It groups the various numbers in the data set into many ranges. It also represents the estimation of the probability of distribution of a continuous variable. In SAS the **PROC UNIVARIATE** is used to create histograms with the below options.

## Syntax

The basic syntax to create a histogram in SAS is:

PROC UNIVARAITE DATA = DATASET;

HISTOGRAM variables;

RUN;

Following is the description of parameters used:

* **DATASET** is the name of the dataset used.
* **variables** are the values used to plot the histogram.

## Simple Histogram

A simple histogram is created by specifying the name of the variable and the range to be considered to group the values.

## Example

In the below example, we consider the minimum and maximum values of the variable horsepower and take a range of 50. So the values form a group in steps of 50.

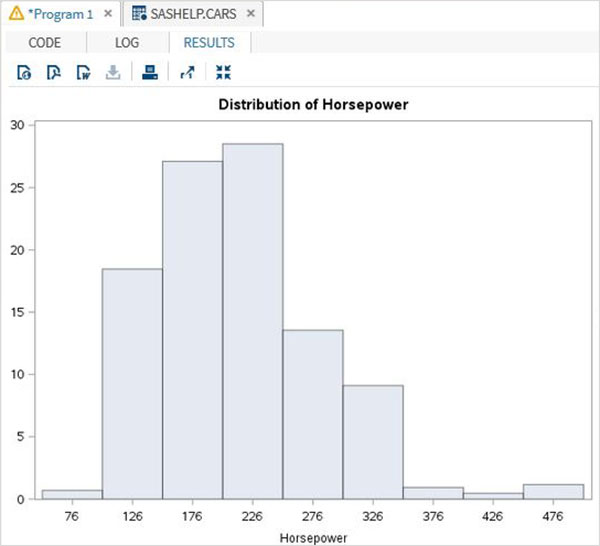
proc univariate data=sashelp.cars;

histogram horsepower

/ midpoints = 176 to 350 by 50;

run;

When we execute the above code, we get the following output:



## Histogram with Curve Fitting

We can fit some distribution curves into the histogram using additional options.

## Example

In the below example we fit a distribution curve with mean and standard deviation values mentioned as EST. This option uses and estimate of the parameters.

proc univariate data=sashelp.cars noprint;

histogram horsepower

/

normal (

mu = est

sigma = est

color = blue

w = 2.5

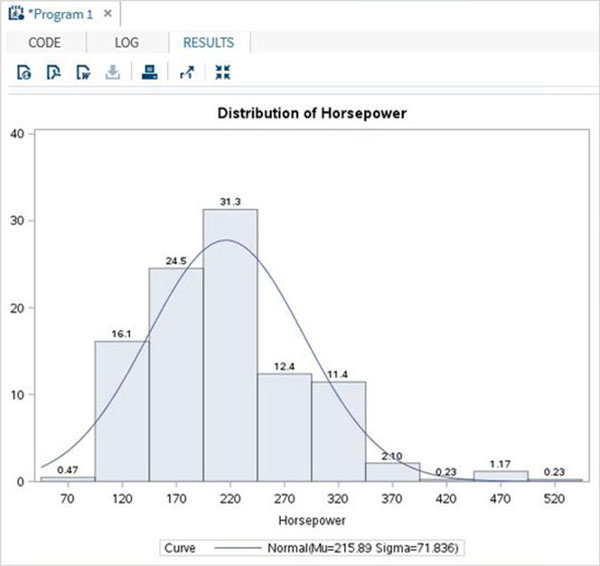
)

barlabel=percent

midpoints = 70 to 550 by 50;

run;

When we execute the above code, we get the following output:



# SAS - Bar Charts

A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable. SAS uses the procedure **PROC SGPLOT** to create bar charts. We can draw both simple and stacked bars in the bar chart. In bar chart each of the bars can be given different colors.

## Syntax

The basic syntax to create a bar-chart in SAS is:

PROC SGPLOT DATA = DATASET;

VBAR variables;

RUN;

Following is the description of parameters used:

* **DATASET** is the name of the dataset used.
* **variables** are the values used to plot the histogram.

## Simple Bar chart

A simple bar chart is a bar chart in which a variable from the dataset is represented as bars.

## Example

The below script will create a bar-chart representing the length of cars as bars.

PROC SQL;

create table CARS1 as

SELECT make,model,type,invoice,horsepower,length,weight

FROM

SASHELP.CARS

WHERE make in ('Audi','BMW')

;

RUN;

proc SGPLOT data=work.cars1;

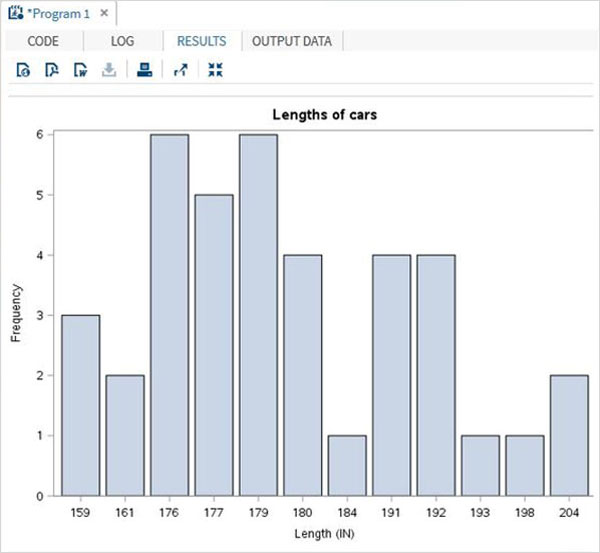
vbar length ;

title 'Lengths of cars';

run;

quit;

When we execute the above code, we get the following output:



## Stacked Bar chart

A stacked bar chart is a bar chart in which a variable from the dataset is calculated with respect to another variable.

## Example

The below script will create a stacked bar-chart where the length of the cars are calculated for each car type. We use the group option to specify the second variable.

proc SGPLOT data=work.cars1;

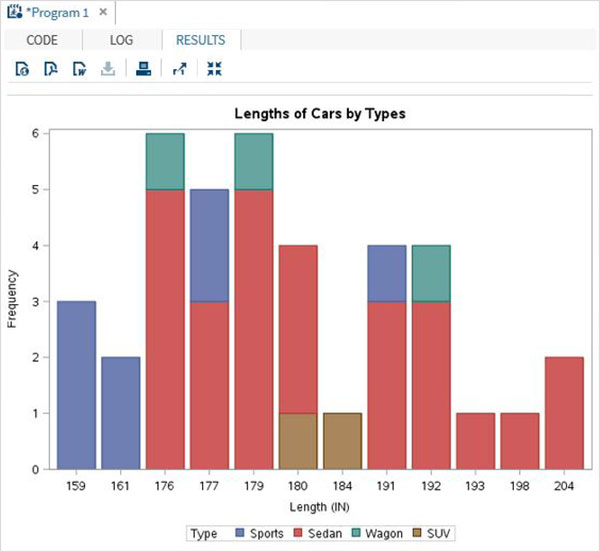
vbar length /group = type ;

title 'Lengths of Cars by Types';

run;

quit;

When we execute the above code, we get the following output:



## Clustered Bar chart

The clustered bar chart is created to show how the values of a variable are spread across a culture.

## Example

The below script will create a clustered bar-chart where the length of the cars is clustered around the car type. So we see two adjacent bars at length 191, one for the car type 'Sedan' and another for the car type 'Wagon'.

proc SGPLOT data=work.cars1;

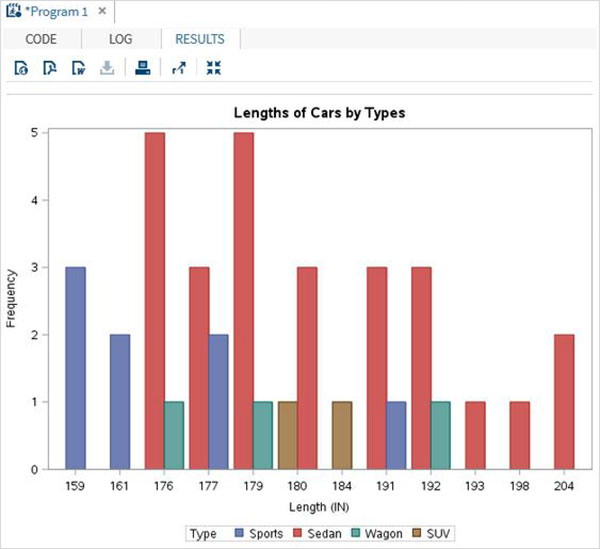
vbar length /group = type GROUPDISPLAY = CLUSTER;

title 'Cluster of Cars by Types';

run;

quit;

When we execute the above code, we get the following output:



# SAS - Pie Charts

A pie-chart is a representation of values as slices of a circle with different colors. The slices are labeled and the numbers corresponding to each slice is also represented in the chart.

In SAS the pie chart is created using **PROC TEMPLATE**which takes parameters to control percentage, labels, color, title etc.

## Syntax

The basic syntax to create a pie-chart in SAS is:

PROC TEMPLATE;

DEFINE STATGRAPH pie;

BEGINGRAPH;

LAYOUT REGION;

PIECHART CATEGORY = variable /

DATALABELLOCATION = OUTSIDE

CATEGORYDIRECTION = CLOCKWISE

START = 180 NAME = 'pie';

DISCRETELEGEND 'pie' /

TITLE = ' ';

ENDLAYOUT;

ENDGRAPH;

END;

RUN;

Following is the description of parameters used:

* **variable** is the value for which we create the pie chart.

## Simple Pie Chart

In this pie chart we take a single variable form the dataset. The pie chart is created with value of the slices representing the fraction of the count of the variable with respect to the total value of the variable.

## Example

In the below example each slice represents the fraction of the type of car from the total number of cars.

PROC SQL;

create table CARS1 as

SELECT make,model,type,invoice,horsepower,length,weight

FROM

SASHELP.CARS

WHERE make in ('Audi','BMW')

;

RUN;

PROC TEMPLATE;

DEFINE STATGRAPH pie;

BEGINGRAPH;

LAYOUT REGION;

PIECHART CATEGORY = type /

DATALABELLOCATION = OUTSIDE

CATEGORYDIRECTION = CLOCKWISE

START = 180 NAME = 'pie';

DISCRETELEGEND 'pie' /

TITLE = 'Car Types';

ENDLAYOUT;

ENDGRAPH;

END;

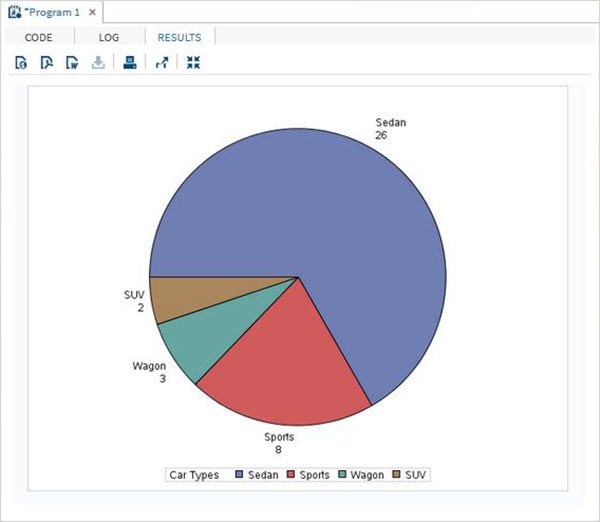
RUN;

PROC SGRENDER DATA = cars1

TEMPLATE = pie;

RUN;

When we execute the above code, we get the following output:



## Pie Chart with Data Labels

In this pie chart we represent both the fractional value as well as the percentage value for each slice. We also change the location of the label to be inside the chart. The style of appearance of the chart is modified by using the DATASKIN option. It uses one of the inbuilt styles, available in the SAS environment.

## Example

PROC TEMPLATE;

DEFINE STATGRAPH pie;

BEGINGRAPH;

LAYOUT REGION;

PIECHART CATEGORY = type /

DATALABELLOCATION = INSIDE

DATALABELCONTENT=ALL

CATEGORYDIRECTION = CLOCKWISE

DATASKIN= SHEEN

START = 180 NAME = 'pie';

DISCRETELEGEND 'pie' /

TITLE = 'Car Types';

ENDLAYOUT;

ENDGRAPH;

END;

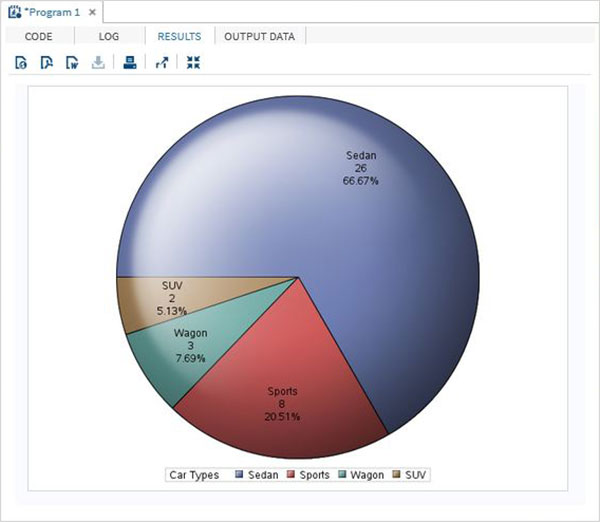
RUN;

PROC SGRENDER DATA = cars1

TEMPLATE = pie;

RUN;

When we execute the above code, we get the following output:



## Grouped Pie Chart

In this pie chart the value of the variable presented in the graph is grouped with respect to another variable of the same data set. Each group becomes one circle and the chart has as many concentric circles as the number of groups available.

## Example

In the below example we group the chart with respect to the variable named "Make". As there are two values available ("Audi" and "BMW") so we get two concentric circles each representing slices of car types in its own make.

PROC TEMPLATE;

DEFINE STATGRAPH pie;

BEGINGRAPH;

LAYOUT REGION;

PIECHART CATEGORY = type / Group = make

DATALABELLOCATION = INSIDE

DATALABELCONTENT=ALL

CATEGORYDIRECTION = CLOCKWISE

DATASKIN= SHEEN

START = 180 NAME = 'pie';

DISCRETELEGEND 'pie' /

TITLE = 'Car Types';

ENDLAYOUT;

ENDGRAPH;

END;

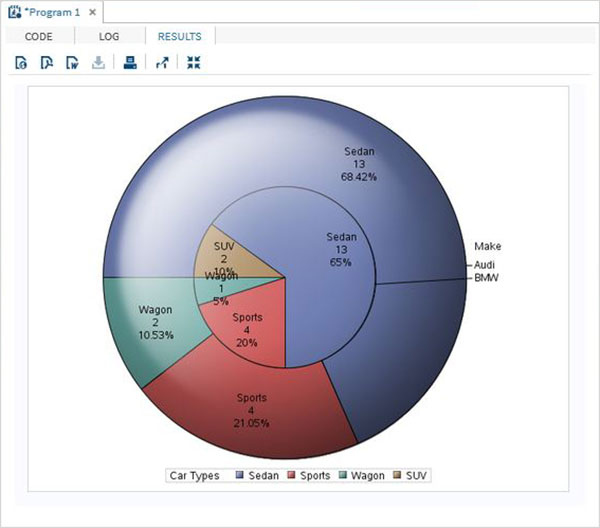
RUN;

PROC SGRENDER DATA = cars1

TEMPLATE = pie;

RUN;

When we execute the above code, we get the following output :



# SAS - Scatter Plots

A scatterplot is a type of graph which uses values from two variables plotted in a Cartesian plane. It is usually used to find out the relationship between two variables. In SAS we use **PROC SGSCATTER** to create scatterplots.

Please note that we create the data set named CARS1 in the first example and use the same data set for all the subsequent data sets. This data set remains in the work library till the end of the SAS session.

## Syntax

The basic syntax to create a scatter-plot in SAS is:

PROC sgscatter DATA=DATASET;

PLOT VARIABLE\_1 \* VARIABLE\_2

/ datalabel = VARIABLE group = VARIABLE;

RUN;

Following is the description of parameters used:

* **DATASET** is the name of data set.
* **VARIABLE** is the variable used from the dataset.

## Simple Scatterplot

In a simple scatterplot we choose two variables form the dataset and group them with respect a third variable. We can also label the data. The result shows how the two variables are scattered in the Cartesian plane.

## Example

PROC SQL;

create table CARS1 as

SELECT make,model,type,invoice,horsepower,length,weight

FROM

SASHELP.CARS

WHERE make in ('Audi','BMW')

;

RUN;

TITLE 'Scatterplot - Two Variables';

PROC sgscatter DATA=CARS1;

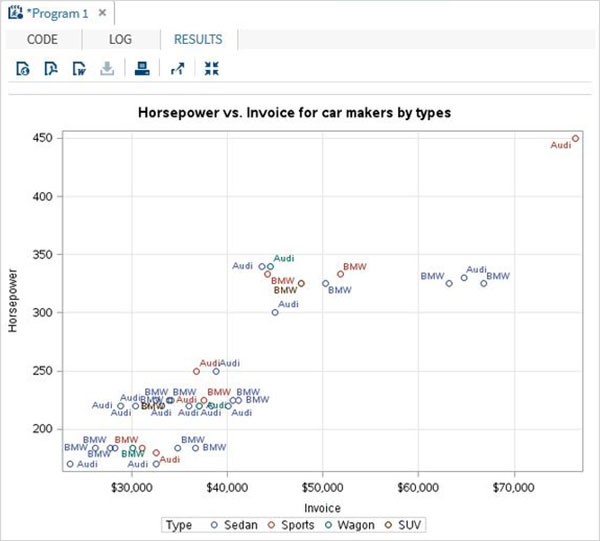
PLOT horsepower\*Invoice

/ datalabel = make group = type grid;

title 'Horsepower vs. Invoice for car makers by types';

RUN;

When we execute the above code, we get the following output:



## Scatterplot with Prediction

we can use an estimation parameter to predict the strength of correlation between by drawing an ellipse around the values. We use the additional options in the procedure to draw the ellipse as shown below.

## Example

proc sgscatter data =cars1;

compare y = Invoice x =(horsepower length)

/ group=type ellipse =(alpha =0.05 type=predicted);

title

'Average Invoice vs. horsepower for cars by length';

title2

'-- with 95% prediction ellipse --'

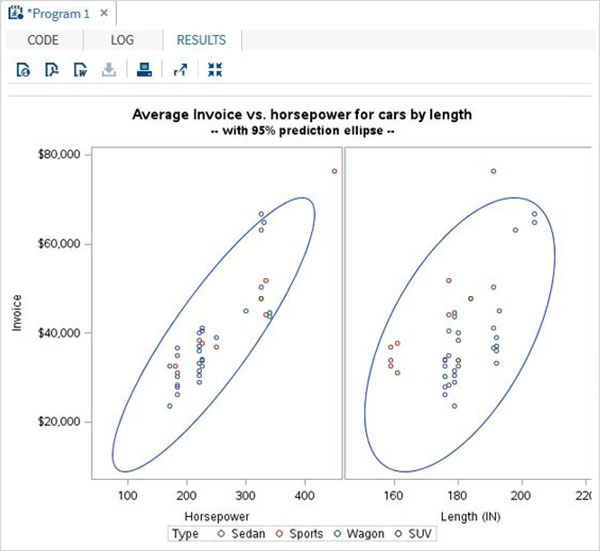
;

format

Invoice dollar6.0;

run;

When we execute the above code, we get the following output:



## Scatter Matrix

We can also have a scatterplot involving more than two variables by grouping them into pairs. In the example below we consider three variables and draw a scatter plot matrix. We get 3 pairs of resulting matrix.

## Example

PROC sgscatter DATA=CARS1;

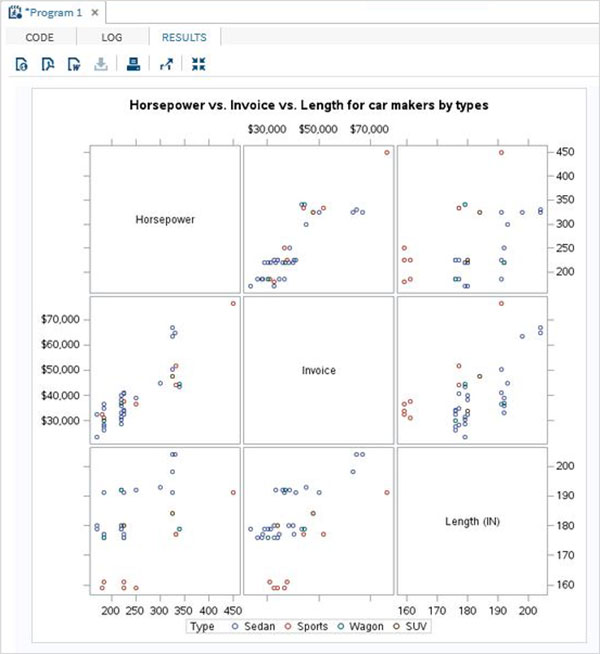
matrix horsepower invoice length

/ group = type;

title 'Horsepower vs. Invoice vs. Length for car makers by types';

RUN;

When we execute the above code, we get the following output:



# SAS - Box Plots

A Boxplot is graphical representation of groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles. The bottom and top of the box are always the first and third quartiles, and the band inside the box is always the second quartile (the median). In SAS a simple Boxplot is created using **PROC SGPLOT** and paneled boxplot is created using **PROC SGPANEL**.

Please note that we create the data set named CARS1 in the first example and use the same data set for all the subsequent data sets. This data set remains in the work library till the end of the SAS session.

## Syntax

The basic syntax to create a boxplot in SAS is:

PROC SGPLOT DATA=DATASET;

VBOX VARIABLE / category = VARIABLE;

RUN;

PROC SGPANEL DATA=DATASET;;

PANELBY VARIABLE;

VBOX VARIABLE> / category = VARIABLE;

RUN;

Following is the description of parameters used:

* **DATASET** is the name of the dataset used.
* **VARIABLE** is the value used to plot the Boxplot.

## Simple Boxplot

In a simple Boxplot we choose one variable from the data set and another to form a category. The values of the first variable are categorized in as many number of groups as the number of distinct values in the second variable.

## Example

In the below example we choose the variable horsepower as the first variable and type as the category variable. So we get boxplots for the distribution of values of horsepower for each type of car.

PROC SQL;

create table CARS1 as

SELECT make,model,type,invoice,horsepower,length,weight

FROM

SASHELP.CARS

WHERE make in ('Audi','BMW')

;

RUN;

PROC SGPLOT DATA=CARS1;

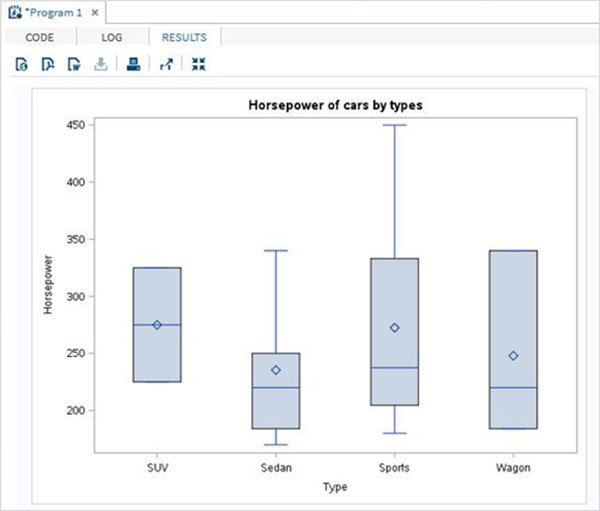
VBOX horsepower

/ category = type;

title 'Horsepower of cars by types';

RUN;

When we execute the above code, we get the following output:



## Boxplot in Vertical Panels

We can divide the Boxplots of a variable into many vertical panels(columns). Each panel holds the boxplots for all the categorical variables. But the boxplots are further grouped using another third variable which divides the graph into multiple panels.

## Example

In the below example we have paneled the graph using the variable 'make'. As there are two distinct values of 'make' so we get two vertical panels.

PROC SGPANEL DATA=CARS1;

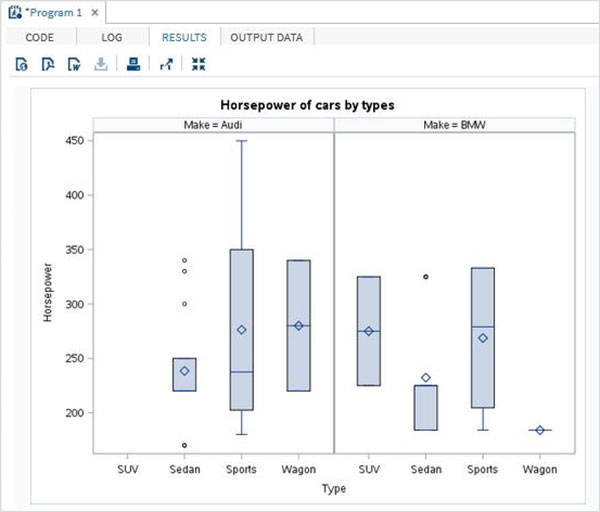
PANELBY MAKE;

VBOX horsepower / category = type;

title 'Horsepower of cars by types';

RUN;

When we execute the above code, we get the following output:



## Boxplot in Horizontal Panels

We can divide the Boxplots of a variable into many horizontal panels(rows). Each panel holds the boxplots for all the categorical variables. But the boxplots are further grouped using another third variable which divides the graph into multiple panels. In the below example we have paneled the graph using the variable 'make'. As there are two distinct values of 'make' so we get two horizontal panels.

PROC SGPANEL DATA=CARS1;

PANELBY MAKE / columns = 1 novarname;

VBOX horsepower / category = type;

title 'Horsepower of cars by types';

RUN;

When we execute the above code, we get the following output :

