SAS – Statistics

Arithmetic Mean/Average (sum, max, min, mean/average, range, quartile, percentile)

# Standard Deviation:

# Frequency Distributions

# Cross Tabulations

# T Tests

# Correlation Analysis

# Linear Regression

SAS - Arithmetic Mean

The arithmetic mean is the value obtained by summing value of numeric variables and then dividing the sum with the number of variables. It is also called Average. In SAS arithmetic mean is calculated using **PROC MEANS**. Using this SAS procedure we can find the mean of all variables or some variables of a dataset. We can also form groups and find mean of variables of values specific to that group.

## Syntax

The basic syntax for calculating arithmetic mean in SAS is:

PROC MEANS DATA = DATASET;

CLASS Variables ;

VAR Variables;

Following is the description of parameters used:

* **DATASET** is the name of the dataset used.
* **Variables** are the name of the variable from the dataset.

## Mean of a Dataset

The mean of each of the numeric variable in a dataset is calculated by using the PROC by supplying only the dataset name without any variables.

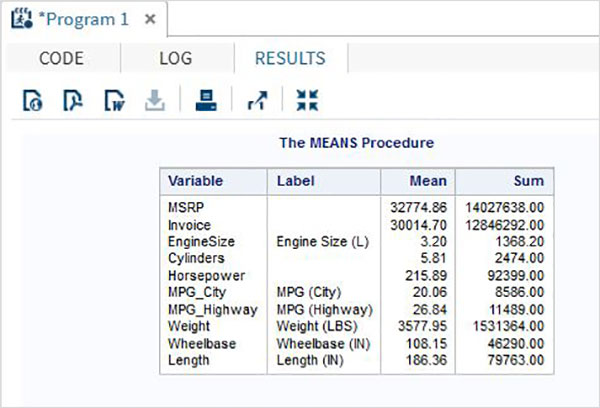
## Example

In the below example we find the mean of all the numeric variables in the SAS dataset named CARS. We specify the maximum digits after decimal place to be 2 and also find the sum of those variables.

PROC MEANS DATA = sashelp.CARS Mean SUM MAXDEC=2;

RUN;

When the above code is executed, we get the following output :



## Mean of Select Variables

We can get the mean of some of the variables by supplying their names in the **var** option.

## Example

In the below we calculate the mean of three variables.

PROC MEANS DATA = sashelp.CARS mean SUM MAXDEC=2 ;

var horsepower invoice EngineSize;

RUN;

When the above code is executed, we get the following output:



## Mean by Class

We can find the mean of the numeric variables by organizing them to groups by using some other variables.

## Example

In the example below we find the mean of the variable horsepower for each type under each make of the car.

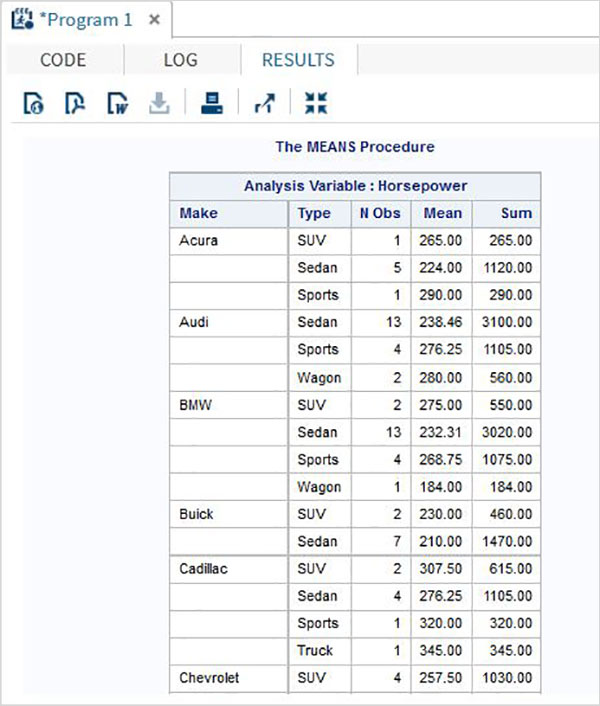
PROC MEANS DATA = sashelp.CARS mean SUM MAXDEC=2;

class make type;

var horsepower;

RUN;

When the above code is executed, we get the following output:



# SAS - Standard Deviation

Standard deviation (SD) is a measure of how varied is the data in a data set. Mathematically it measures how distant or close are each value to the mean value of a data set. A standard deviation value close to 0 indicates that the data points tend to be very close to the mean of the data set and a high standard deviation indicates that the data points are spread out over a wider range of values

In SAS the SD values is measured using PROC MEAN as well as PROC SURVEYMEANS.

## Using PROC MEANS

To measure the SD using **proc means** we choose the STD option in the PROC step. It brings out the SD values for each numeric variable present in the data set.

## Syntax

The basic syntax for calculating standard deviation in SAS is:

PROC means DATA = dataset STD;

Following is the description of the parameters used:

* **Dataset** is the name of the dataset.

## Example

In the below example we create the data set CARS1 form the CARS data set in the SASHELP library. We choose the STD option with the PROC means step.

PROC SQL;

create table CARS1 as

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

FROM

SASHELP.CARS

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

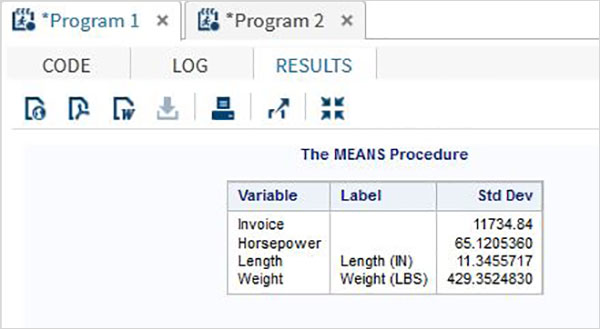
;

RUN;

proc means data=CARS1 STD;

run;

When we execute the above code it gives the following output:



## Using PROC SURVEYMEANS

This procedure is also used for measurement of SD along with some advance features like measuring SD for categorical variables as well as provide estimates in variance.

## Syntax

The syntax for using PROC SURVEYMEANS is:

PROC SURVEYMEANS options statistic-keywords ;

BY variables ;

CLASS variables ;

VAR variables ;

Following is the description of the parameters used :

* **BY** indicates the variables used to create groups of observations.
* **CLASS** indicates the variables used for categorical variables.
* **VAR** indicates the variables for which SD will be calculated.

## Example

The below example describes the use of class option which creates the statistics for each of the values in the class variable.

proc surveymeans data=CARS1 STD;

class type;

var type horsepower;

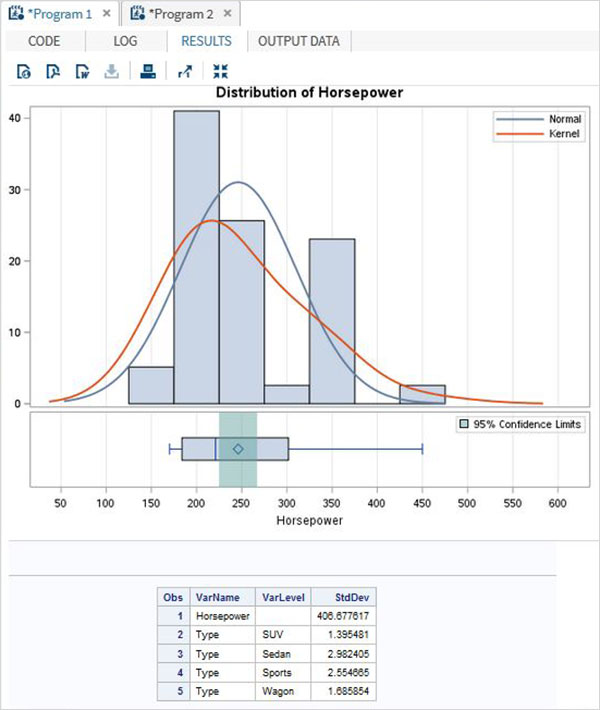
ods output statistics=rectangle;

run;

proc print data=rectangle;

run;

When we execute the above code it gives the following output:



## Using BY option

The below code gives example of BY option. In it the result is grouped for each value in the BY option.

## Example

proc surveymeans data=CARS1 STD;

var horsepower;

BY make;

ods output statistics=rectangle;

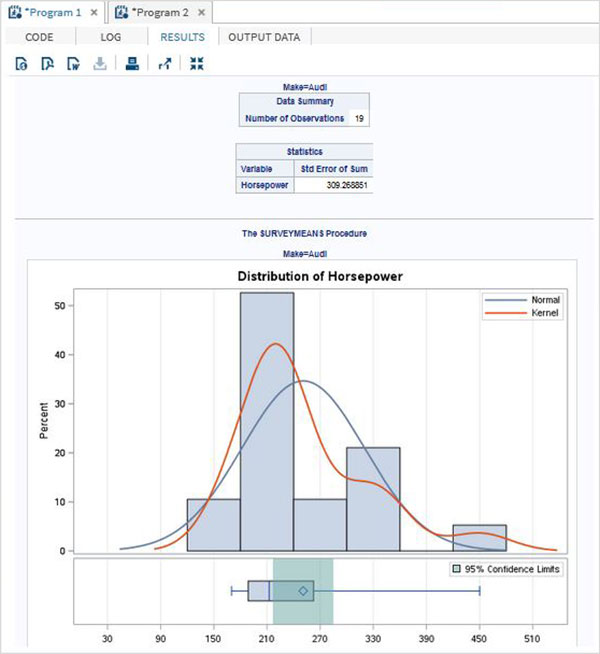
run;

proc print data=rectangle;

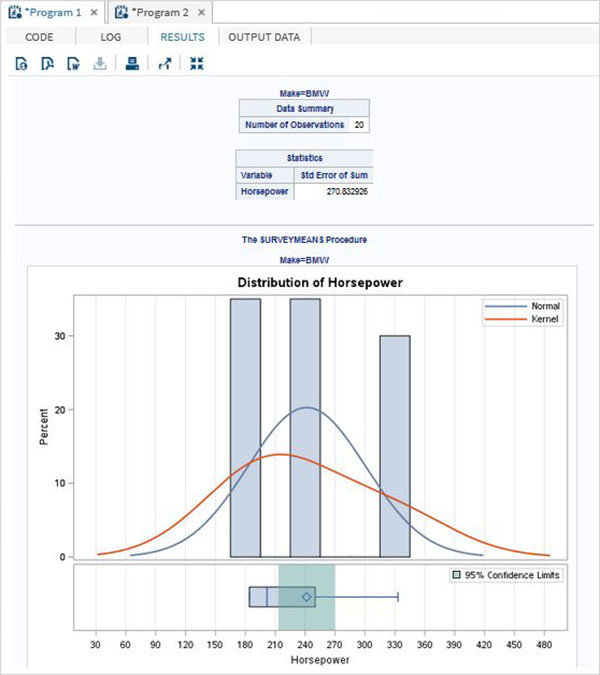
run;

When we execute the above code it gives the following output:

### Result for make = "Audi"



### Result for make = "BMW"



# SAS - Frequency Distributions

A frequency distribution is a table showing the frequency of the data points in a data set. Each entry in the table contains the frequency or count of the occurrences of values within a particular group or interval, and in this way, the table summarizes the distribution of values in the sample.

SAS provides a procedure called **PROC FREQ** to calculate the frequency distribution of data points in a data set.

## Syntax

The basic syntax for calculating frequency distribution in SAS is:

PROC FREQ DATA = Dataset ;

TABLES Variable\_1 ;

BY Variable\_2 ;

Following is the description of the parameters used:

* **Dataset** is the name of the dataset.
* **Variables\_1** is the variable names of the dataset whose frequency distribution needs to be calculated.
* **Variables\_2** is the variables which categorised the frequency distribution result.

## Single Variable Frequency Distribution

We can determine the frequency distribution of a single variable by using PROC FREQ. In this case the result will show the frequency of each value of the variable. The result also shows the percentage distribution, cumulative frequency and cumulative percentage.

## Example

In the below example we find the frequency distribution of the variable horsepower for the dataset named CARS1 which is created form the library SASHELP.CARS. We can see the result divided into two categories of results. One for each make of the car.

PROC SQL;

create table CARS1 as

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

FROM

SASHELP.CARS

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

;

RUN;

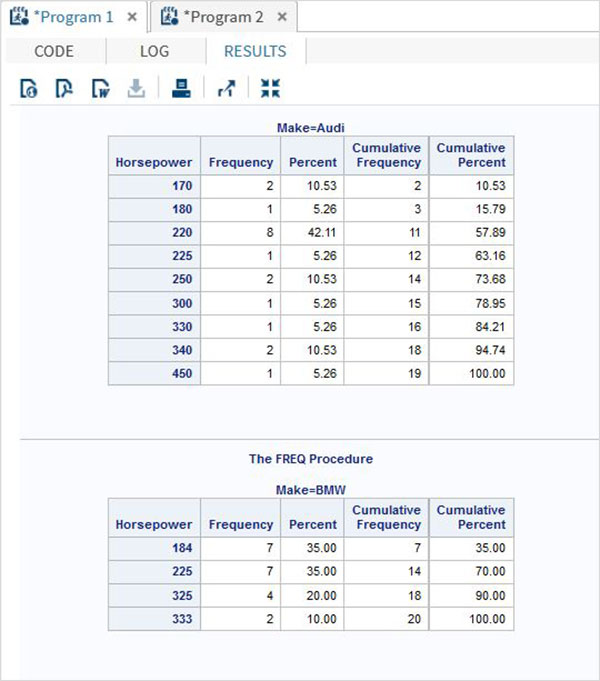
proc FREQ data=CARS1 ;

tables horsepower;

by make;

run;

When the above code is executed, we get the following result:



## Multiple Variable Frequency Distribution

We can find the frequency distributions for multiple variables which groups them into all possible combinations.

## Example

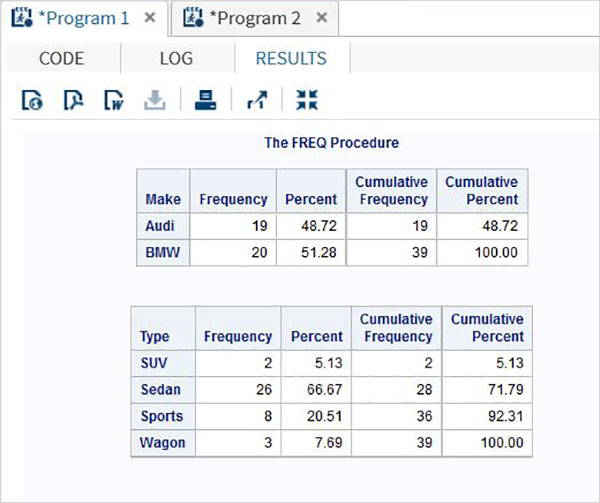
In the below example we calculate the frequency distribution for the make of a car for grouped by car type and also the frequency distribution of each type of car grouped by each make.

proc FREQ data=CARS1 ;

tables make type;

run;

When the above code is executed, we get the following result:



## Frequency Distribution with Weight

With the weight option we can calculate the frequency distribution biased with the weight of the variable. Here the value of the variable is taken as the number of observations instead of the count of value.

## Example

In the below example we calculate the frequency distribution of the variables make and type with weight assigned to horsepower.

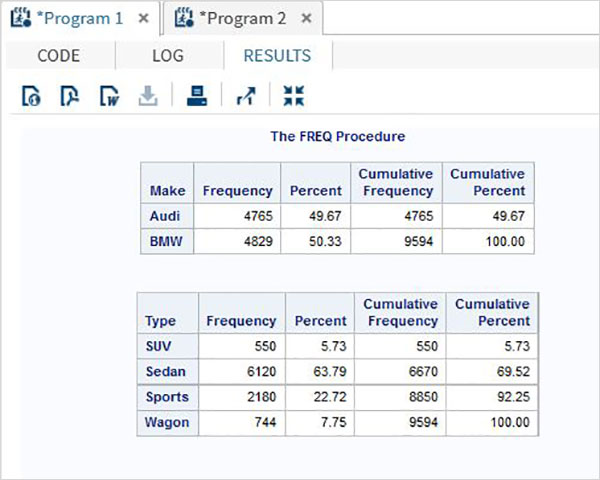
proc FREQ data=CARS1 ;

tables make type;

weight horsepower;

run;

When the above code is executed, we get the following result:



# SAS - Cross Tabulations

Cross tabulation involves producing cross tables also called contingent tables using all possible combinations of two or more variables. In SAS it is created using **PROC FREQ** along with the **TABLES** option. For example - if we need the frequency of each model for each make in each car type category, then we need to use the TABLES option of PROC FREQ.

## Syntax

The basic syntax for applying cross tabulation in SAS is:

PROC FREQ DATA = dataset;

TABLES variable\_1\*Variable\_2;

Following is the description of the parameters used:

* **Dataset** is the name of the dataset.
* **Variable\_1 and Variable\_2** are the variable names of the dataset whose frequency distribution needs to be calculated.

## Example

Consider the case of finding how many car types are available under each car brand from the dataset cars1 which is created form SASHELP.CARS as shown below. In this case we need the individual frequency values as well as the sum of the frequency values across the makes and across the types. We can observer that the result shows values across the rows and the columns.

PROC SQL;

create table CARS1 as

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

FROM

SASHELP.CARS

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

;

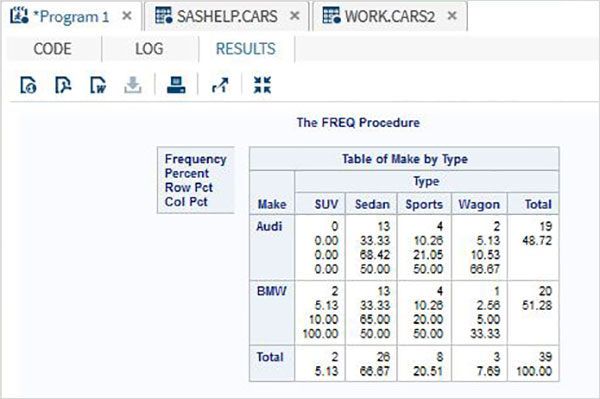
RUN;

proc FREQ data=CARS1 ;

tables make\*type;

run;

When the above code is executed, we get the following result:



## Cross tabulation of 3 Variables

When we have three variables we can group 2 of them and cross tabulate each of these two with the third varaible. So in the result we have two cross tables.

## Example

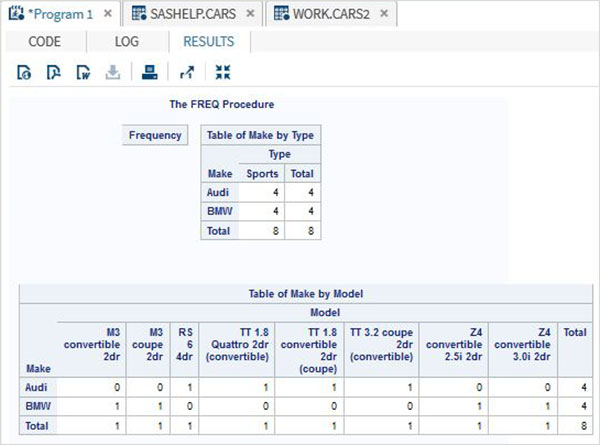
In the below example we find the frequency of each type of car and each model of car with respect to the make of the car. Also we use the nocol and norow option to avoid the sum and percentage values.

proc FREQ data=CARS2 ;

tables make \* (type model) / nocol norow nopercent;

run;

When the above code is executed, we get the following result:



## Cross tabulation of 4 Variables

With 4 variables, the number of paired combinations increases to 4. Each variable from group 1 is paired with each variable of group 2.

## Example

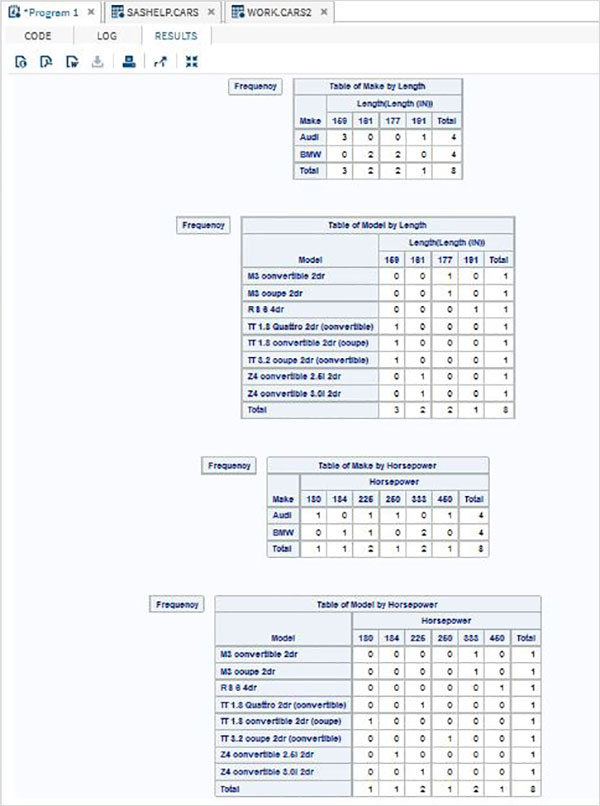
In the below example we find the frequency of length of the car for each make and each model. Similarly the frequency of horsepower for each make and each model.

proc FREQ data=CARS2 ;

tables (make model) \* (length horsepower) / nocol norow nopercent;

run;

When the above code is executed, we get the following result:



# SAS - T Tests

The T-tests are performed to compute the confidence limits for one sample or two independent samples by comparing their means and mean differences. The SAS procedure named **PROC TTEST** is used to carry out t tests on a single variable and pair of variables.

## Syntax

The basic syntax for applying PROC TTEST in SAS is:

PROC TTEST DATA = dataset;

VAR variable;

CLASS Variable;

PAIRED Variable\_1 \* Variable\_2;

Following is the description of the parameters used:

* **Dataset** is the name of the dataset.
* **Variable\_1 and Variable\_2** are the variable names of the dataset used in t test.

## Example

Below we see one sample t test in which find the t test estimation for the variable horsepower with 95 percent confidence limits.

PROC SQL;

create table CARS1 as

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

FROM

SASHELP.CARS

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

;

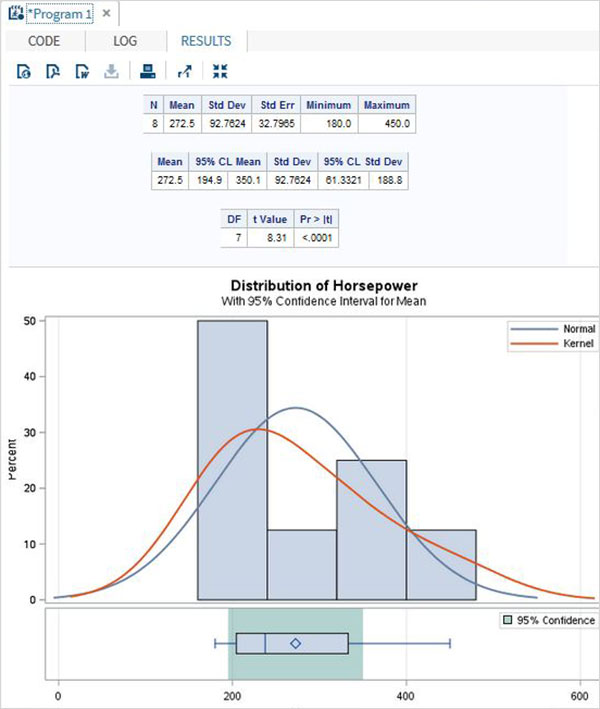
RUN;

proc ttest data=cars1 alpha=0.05 h0=0;

var horsepower;

run;

When the above code is executed, we get the following result:



## Paired T Test

The paired T Test is carried out to test if two dependent variables are statistically different from each other or not.

## Example

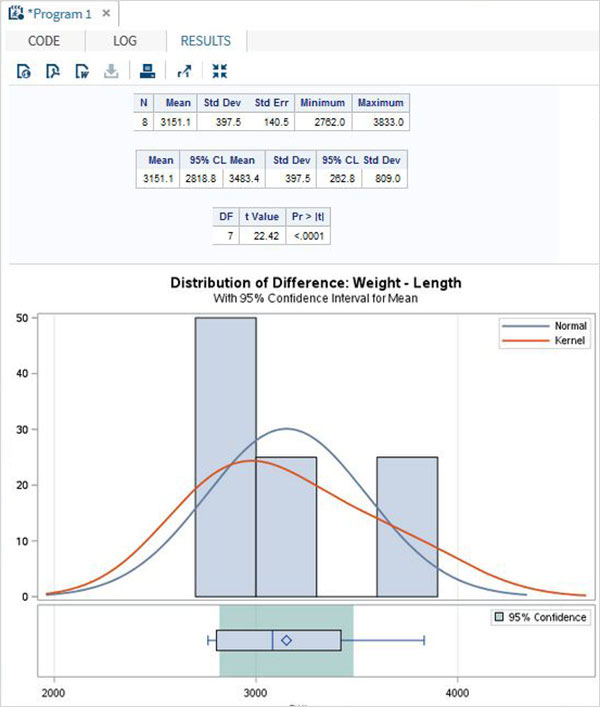
As length and weight of a car will be dependent on each other we apply the paired T test as shown below.

proc ttest data=cars1 ;

paired weight\*length;

run;

When the above code is executed, we get the following result:



### Two sample t-test

This t-test is designed to compare means of same variable between two groups.

## Example

In our case we compare the mean of the variable horsepower between the two different makes of the cars("Audi" and "BMW").

proc ttest data=cars1 sides=2 alpha=0.05 h0=0;

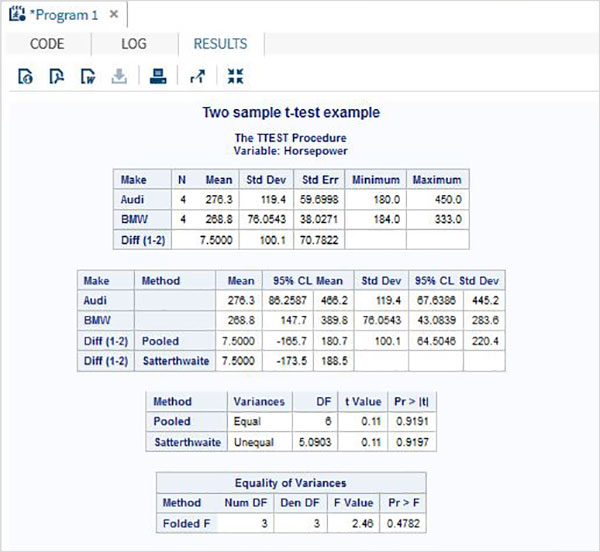
title "Two sample t-test example";

class make;

var horsepower;

run;

When the above code is executed, we get the following result:



# SAS - Correlation Analysis

Correlation analysis deals with relationships among variables. The correlation coefficient is a measure of linear association between two variables.Values of the correlation coefficient are always between -1 and +1. SAS provides the procedure **PROC CORR** to find the correlation coefficients between a pair of variables in a dataset.

## Syntax

The basic syntax for applying PROC CORR in SAS is:

PROC CORR DATA = dataset options;

VAR variable;

Following is the description of the parameters used:

* **Dataset** is the name of the dataset.
* **Options** is the additional option with procedure like plotting a matrix etc.
* **Variable**is the variable name of the dataset used in finding the correlation.

## Example

Correlation coefficients between a pair of variables available in a dataset can be obtained by use their names in the VAR statement. In the below example we use the dataset CARS1 and get the result showing the correlation coefficients between horsepower and weight.

PROC SQL;

create table CARS1 as

SELECT invoice,horsepower,length,weight

FROM

SASHELP.CARS

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

;

RUN;

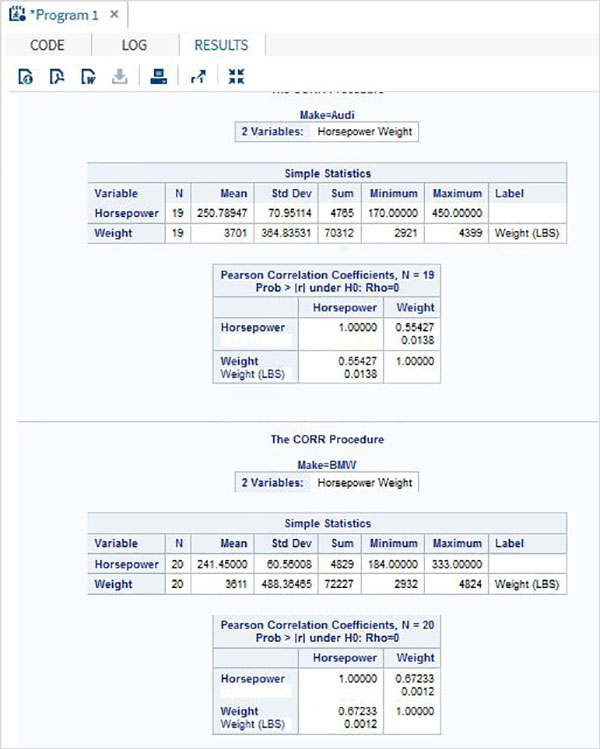
proc corr data=cars1 ;

VAR horsepower weight ;

BY make;

run;

When the above code is executed, we get the following result:



## Correlation Between All Variables

Correlation coefficients between all the variables available in a dataset can be obtained by simply applying the procedure with the dataset name.

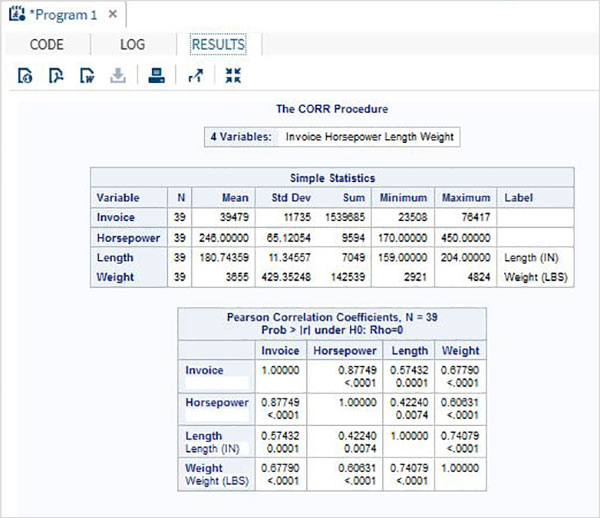
## Example

In the below example we use the dataset CARS1 and get the result showing the correlation coefficients between each pair of the variables.

proc corr data=cars1 ;

run;

When the above code is executed, we get the following result:



## Correlation Matrix

We can obtain a scatterplot matrix between the variables by choosing the option to plot matrix in the PROC statement.

## Example

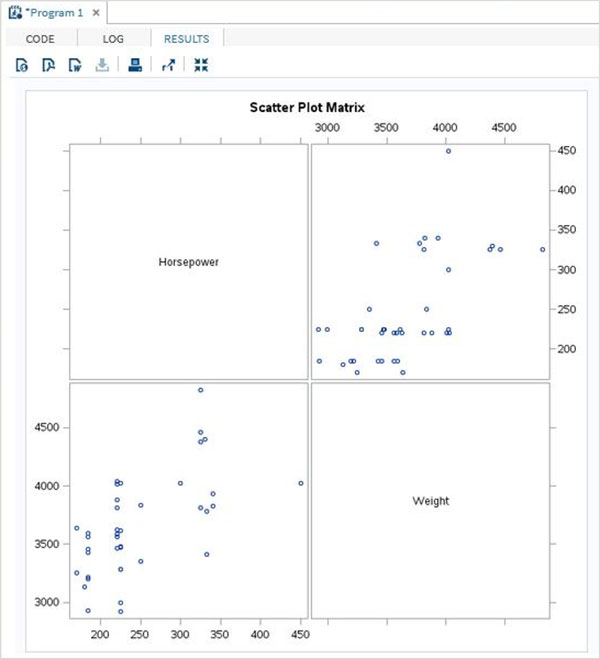
In below example we get the matrix between horsepower and weight.

proc corr data=cars1 plots=matrix ;

VAR horsepower weight ;

run;

When the above code is executed, we get the following result:



# SAS - Linear Regression

Linear Regression is used to identify the relationship between a dependent variable and one or more independent variables. A model of the relationship is proposed, and estimates of the parameter values are used to develop an estimated regression equation.

Various tests are then used to determine if the model is satisfactory. If it is then, the estimated regression equation can be used to predict the value of the dependent variable given values for the independent variables. In SAS the procedure **PROC REG** is used to find the linear regression model between two variables.

## Syntax

The basic syntax for applying PROC REG in SAS is:

PROC REG DATA = dataset;

MODEL variable\_1 = variable\_2;

Following is the description of the parameters used:

* **Dataset** is the name of the dataset.
* **variable\_1 and variable\_2**are the variable names of the dataset used in finding the correlation.

## Example

The below example shows the process to find the correlation between the two variables horsepower and weight of a car by using PROC REG. In the result we see the intercept values which can be used to form the regression equation.

PROC SQL;

create table CARS1 as

SELECT invoice,horsepower,length,weight

FROM

SASHELP.CARS

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

;

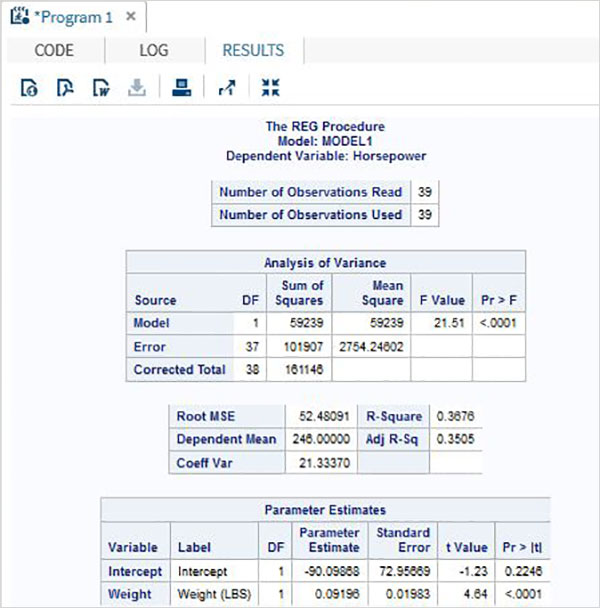
RUN;

proc reg data=cars1;

model horsepower= weight ;

run;

When the above code is executed, we get the following result:



The above code also gives the graphical view of various estimates of the model as shown below. Being an advanced SAS procedure it simply does not stop at giving the intercept values as the output.

