**LAB : 1**

**Aim :** To study basics of dataset, method of statistics and analysis of iris data set using pandas library in ipython.

**Reqiurements :** Jupiter notebook.

**Theory :**

**1. Basic Defintions :**

* **Dataset:** Dataset is a collection of related sets of information that is composed of separate elements but can be manipulated as a unit by a computer. In other words, data set is a combination of data having common features
* **Features of Dataset :** A [**feature**](https://www.datarobot.com/wiki/feature/) is a measurable property of the object you’re trying to analyze. In datasets, features appear as columns.
* **Inferential statistics :** inferential Statistics are produced by more complex mathematical calculations, and allow us to infer trends and make assumptions and predictions about a population based on a study of a sample taken from it. Inferential statistics is one of the two main branches of statistics.Inferential statistics use a random sample of data taken from a population to describe and make inferences about the population. Inferential statistics are valuable when examination of each member of an entire population is not convenient or possible. For example, to measure the diameter of each nail that is manufactured in a mill is impractical. You can measure the diameters of a representative random sample of nails. You can use the information from the sample to make generalizations about the diameters of all of the nails.
* **Descriptive analysis :**In Descriptive Statistics your are describing, presenting, summarizing and organizing your data (population), either through numerical calculations or graphs or tables. Descriptive analysis contrasts with *inferential statistics* which draws conclusions about a population based on a sample of data. Descriptive analysis also contrasts with *predictive analytics*where the focus is on extrapolating from the past to predict future trends and outcomes. And descriptive analysis contrasts with prescriptive analytics, which takes prediction a step further to not only predict what will happen, but also recommend a course of action based on the most rational response to the distribution of future events.
* **ARITHMETIC MEAN** : Sum of all the observations and divided by the number of observations results the arithmetic mean. Symbolically, the arithmetic mean, also called simply mean is

x = ∑x/n , where x is simple mean.

* **MEDIAN** Median is defined as the value of the middle item (or the mean of the values of the two middle items) when the data are arranged in an ascending or descending order of magnitude. Thus, in an ungrouped frequency distribution if the n values are arranged in ascending or descending order of magnitude, the median is the middle value if n is odd. When n is even, the median is the mean of the two middle values.
* **MODE :** The mode is another measure of central tendency. It is the value at the point around which the items are most heavily concentrated.
* **STANDARD DEVIATION** : The standard deviation is similar to the mean deviation in that here too the deviations are measured from the mean. The *standard deviation* of a population of values is computed as:

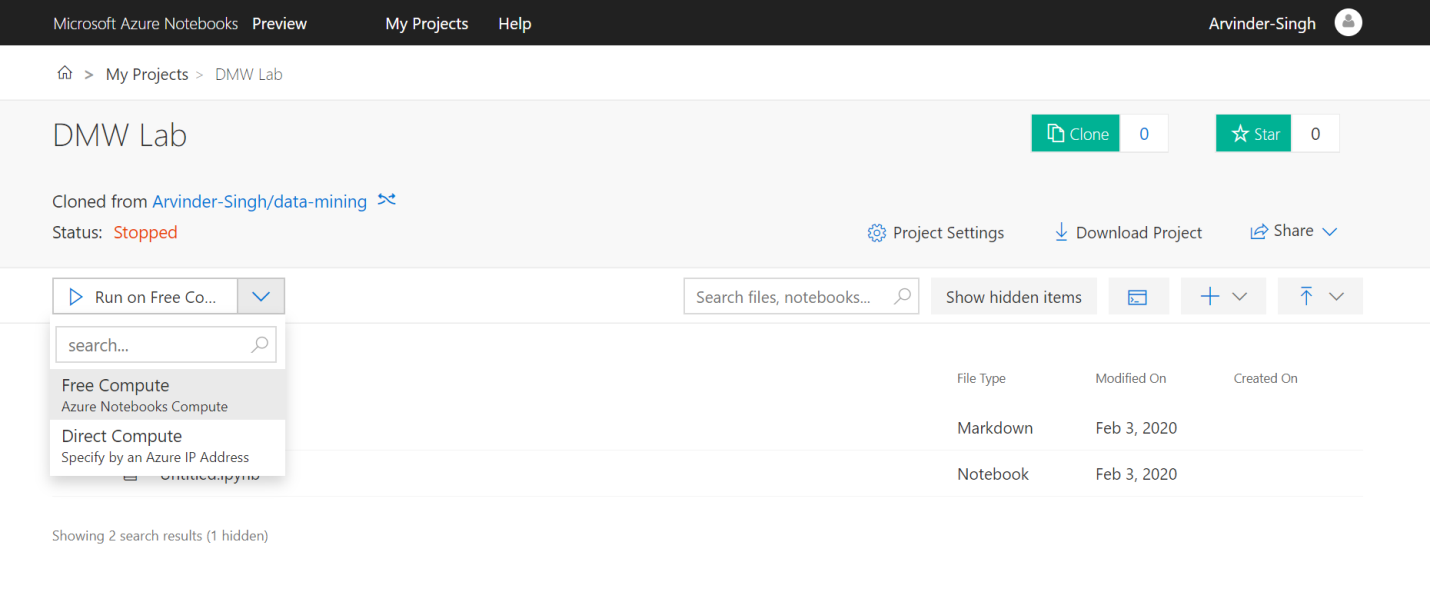
 = [(xi-)2/N]1/2

* **Correlation :** The correlation between variables is a measure of the nature and degree of association between the variables.
* **Variance :** This mean of the squared deviations is known as the variance. Symbolically,

Var (X) = 2 = (xi-)2/N

**Procedure :**

1. Open Jupiter notebook. Here we have used Microsoft Azure Notebook to access Jupiter notebook.
2. Create a new project and open it .
3. To start working, click on “Free Compute” .



1. In Now select Python 3.6 in Notebook section and write following code on interface provided.

**CODE:**

import pandas as pd

df=pd.read\_csv("https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/639388c2cbc2120a14dcf466e85730eb8be498bb/iris.csv")

print(df.shape)

print(df.info)

df.mean()

df.median()

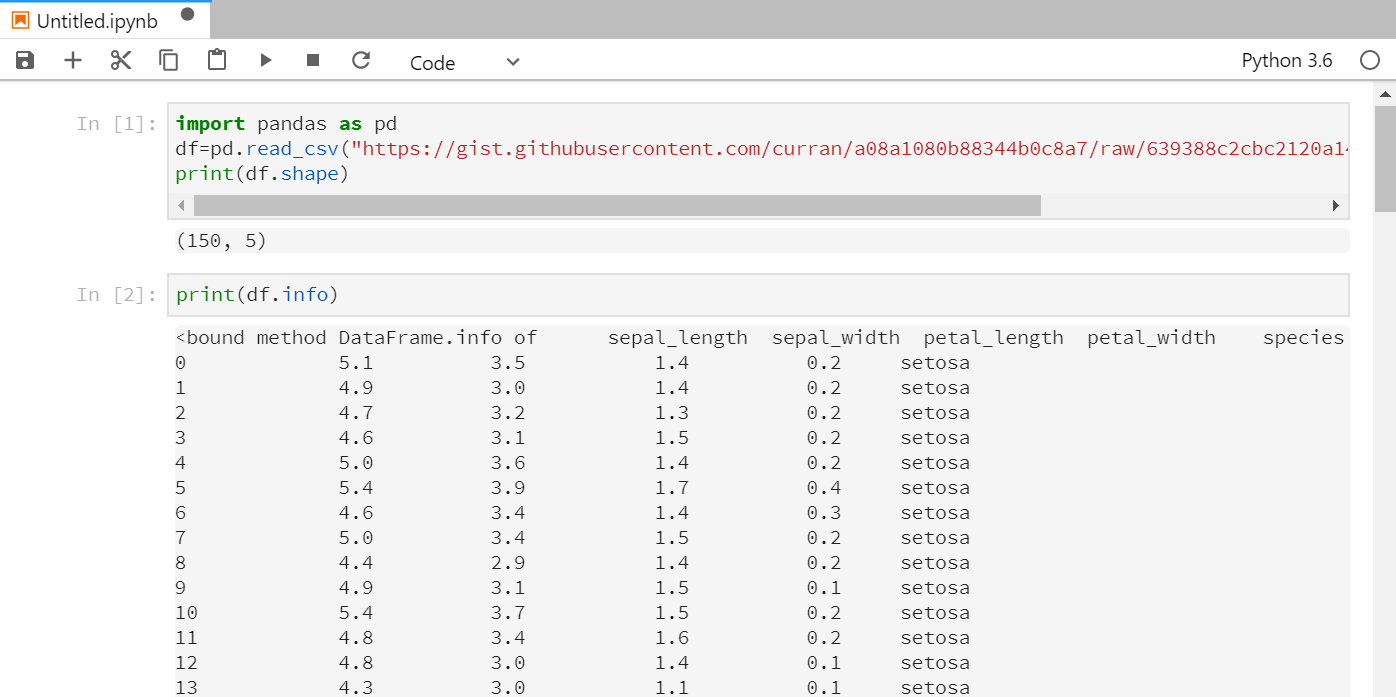
df.mode()

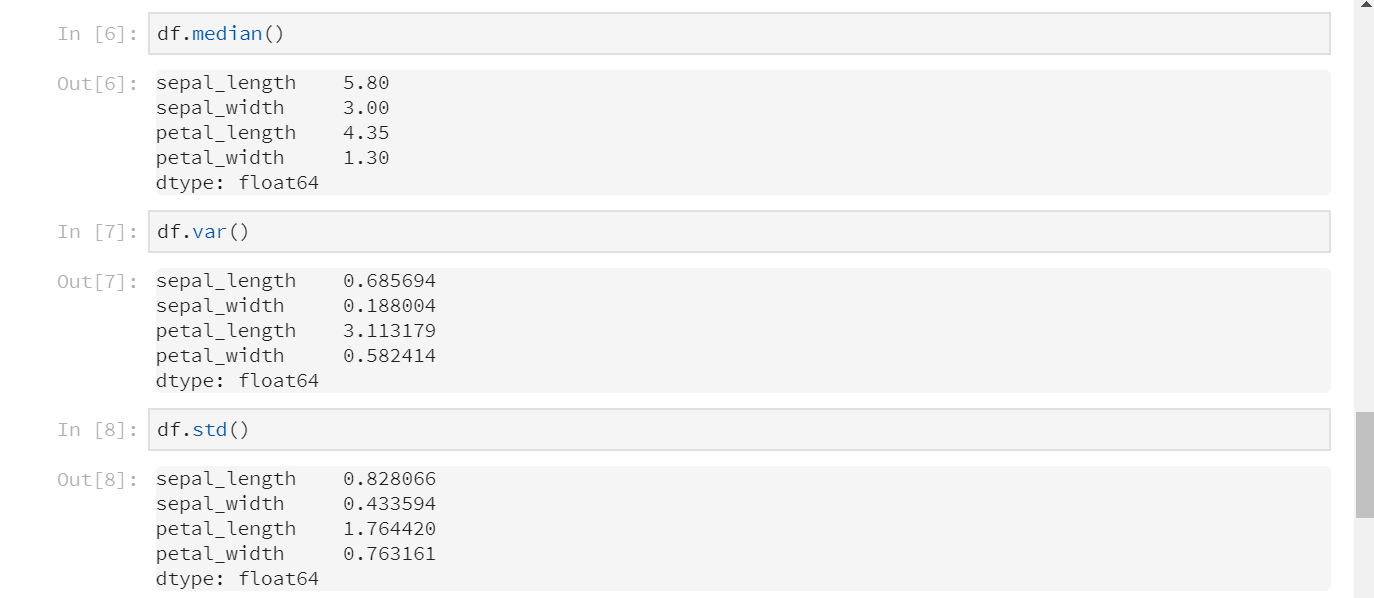
df.std()

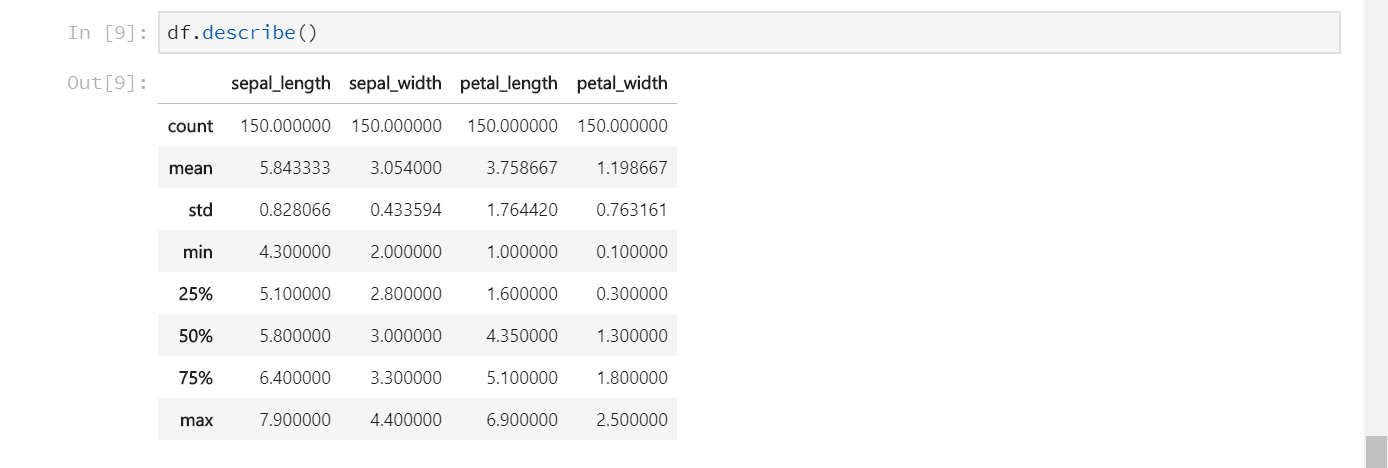
df.var()

df.describe()

1. The output will appear as :



** **

****

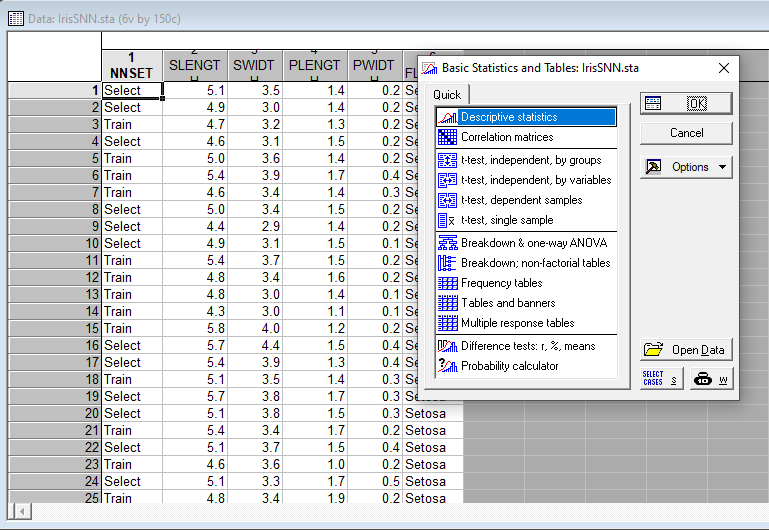
**LAB : 2**

**Aim :** To study and implement Descriptive Analysis using statistica.

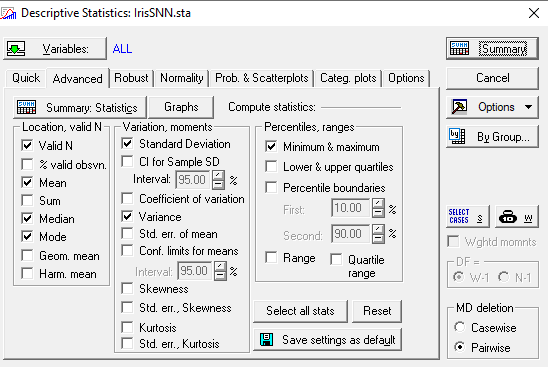
**Reqiurements :** Statistica, iris data set (irirs.csv).

**Procedure :**

1. Load the dataset ( here IrisSNN.sta is used as dataset) on which the analysis is to be performed.
2. Choose “Basic statistics/Tables” option from Statistics menu in menu bar. Following window will apper after this.

****

1. Choose Descriptive statistcs and click OK.
2. Select from the various options available under advanced tab and make chioces as per requirement.



1. Click on Summary tab to view the result result in tabular form

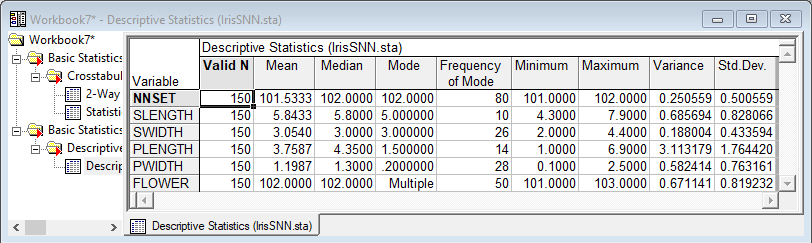
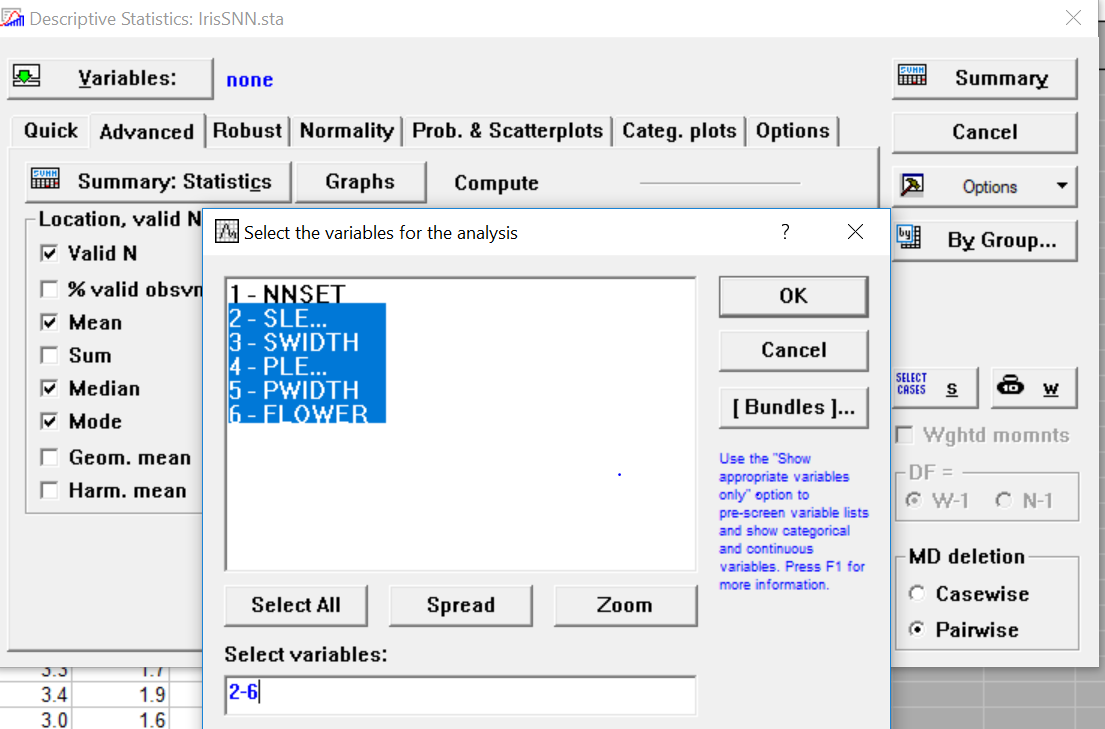
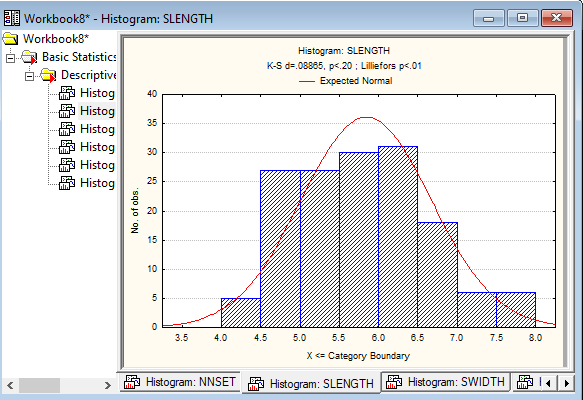
****

Table showing mean, median, mode, variance and standard deviation of the dataset across various attributes like sepal length, sepal width, etc.

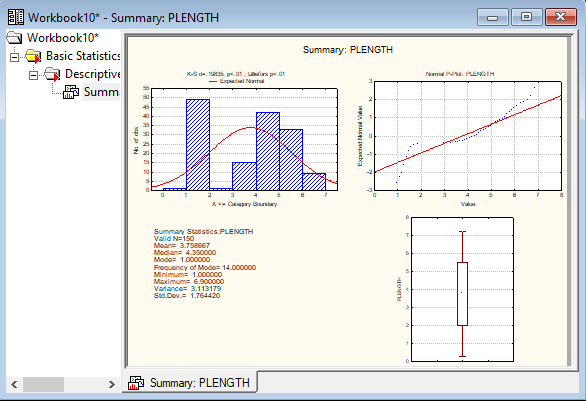
1. We can also choose Graphs tab and select the variables as per requirement to get results in graphical form.

****

various attributes like sepal length, sepal width, etc.



Representation of sepal length in the form of a histogram



Detailed summary of all the flowers based upon petal length

**LAB : 3**

**Aim :** To study and implement

* various distance methods (Euclidean distance, Weighted Euclidean distance, Manhattan distance) and implimenting them in a programming language.
* CHI-Square method in a programming language and perform CHI-Square method on Smoking Habits dataset (SmokingHabts.sta) in Statistica.

**Reqiurements :**Sublime Text editor, GCC compiler, Statistica, SmokingHabts.sta data set (SmokingHabts.sta).

**Procedure :**

1. **Code to implement** Euclidean distance, Weighted Euclidean distance, Manhattan distance

#include <bits/stdc++.h>

#include<cmath>

using namespace std;

double Euclidean(double \*x,double \*y,int n){

int i;

double dis;

for(i=0;i<n;i++){

dis+=(x[i]-y[i])\*(x[i]-y[i]);

}

return sqrt(dis);

}

double WeightedEuclidean(double \*x,double \*y,int n){

int i;

double dis;

for(i=0;i<n;i++){

cout<<"Enter the weight for "<<i+1<<"th attribute- ";

int w;

cin>>w;

dis+=w\*(x[i]-y[i])\*(x[i]-y[i]);

}

return sqrt(dis);

}

double Manhattan(double \*x,double \*y,int n){

int i;

double dis;

for(i=0;i<n;i++){

dis+=abs(x[i]-y[i]);

}

return dis;

}

int main()

{

cout << "Enter no of itemset :";

int n;cin >> n;

double x[n],y[n];

cout << "Enter data for X :";

for (int i = 0; i < n; i++)

{ cin >> x[i]; }

cout << "Enter data for Y :";

for (int i = 0; i < n; i++){

cin >> y[i];

} ;

int ch;

cout<<"\n1. Display Data\n2. Eucledean distance.\n3. Manhatten Distance\n4. Weighted Eucledean distance\n5. Press 5 to exit\n\nEnter a valid option ";

cin>>ch;

while(ch!=5){

switch(ch){

case 0:

break;

case 1:

cout<<"\n\n"<<setw(10)<<"X"<<setw(10)<<"Y";

for(int p=0;p<n;p++)

cout<<"\n"<<setw(10)<<x[p]<<setw(10)<<y[p];

cout<<"\n";

break;

case 2:

cout<<"Required distance "<<Euclidean(x,y,n)<<"\n";

break;

case 3:

cout<<"Required distance "<<Manhattan(x,y,n)<<"\n";

break;

case 4:

cout<<"Required distance "<<WeightedEuclidean(x,y,n)<<"\n";

break;

case 5:

break;

default:

cout<<"Enter a valid choice";

break;

}

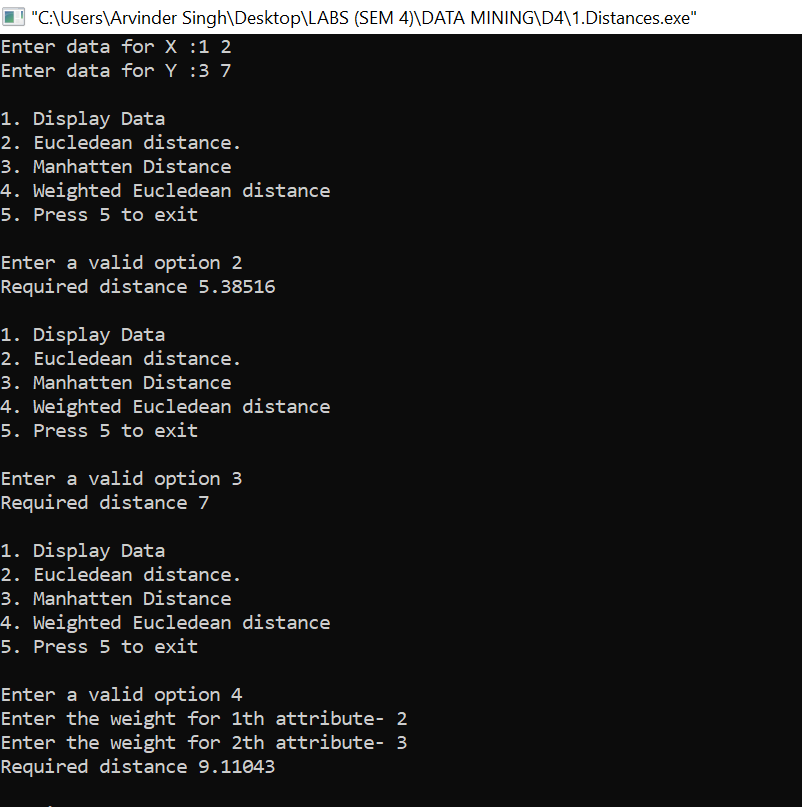
cout<<"\n1. Display Data\n2. Eucledean distance.\n3. Manhatten Distance\n4. Weighted Eucledean distance\n5. Press 5 to exit\n\nEnter a valid option ";

cin>>ch;

}

return 0;

}



1. **Code to implement** CHI-Square method

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int n;

cout<<"CHI-Square analysis for two Variables\n\n";

cout<<"Enter the no. of attributes\n";

cin>>n;

double arr[n+1][3];

cout<<"Enter the observed values\n";

int i;

for(i=0;i<n;i++){

cin>>arr[i][0]>>arr[i][1];

arr[n][0]+=arr[i][0];

arr[n][1]+=arr[i][1];

arr[i][2]+=arr[i][0]+arr[i][1];

}

arr[n][2]=arr[n][0]+arr[n][1];

double exp[n][2];

for(i=0;i<n;i++){

exp[i][0]=arr[i][2]\*arr[n][0]/arr[n][2];

exp[i][1]=arr[i][2]\*arr[n][1]/arr[n][2];

}

double chi\_square=0;

for(i=0;i<n;i++){

chi\_square+=((arr[i][0]-exp[i][0]))\*((arr[i][0]-exp[i][0]))/exp[i][0];

chi\_square+=((arr[i][1]-exp[i][1]))\*((arr[i][1]-exp[i][1]))/exp[i][1];

}

cout<<"Enter the tabular value\n";

double t;

cin>>t;

cout<<"Value of CHI Square is: "<<chi\_square<<"\n";

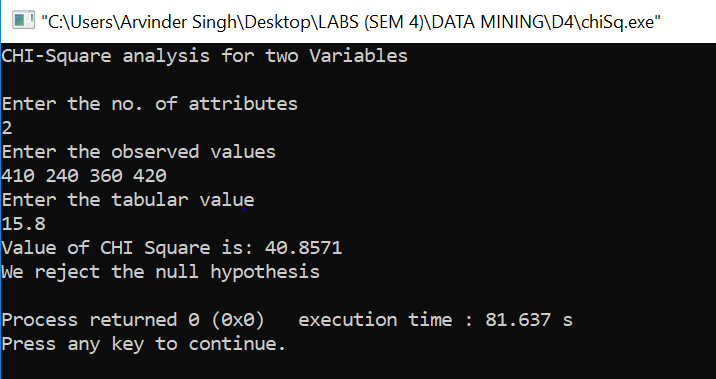
if(t<chi\_square)

cout<<"We reject the null hypothesis\n";

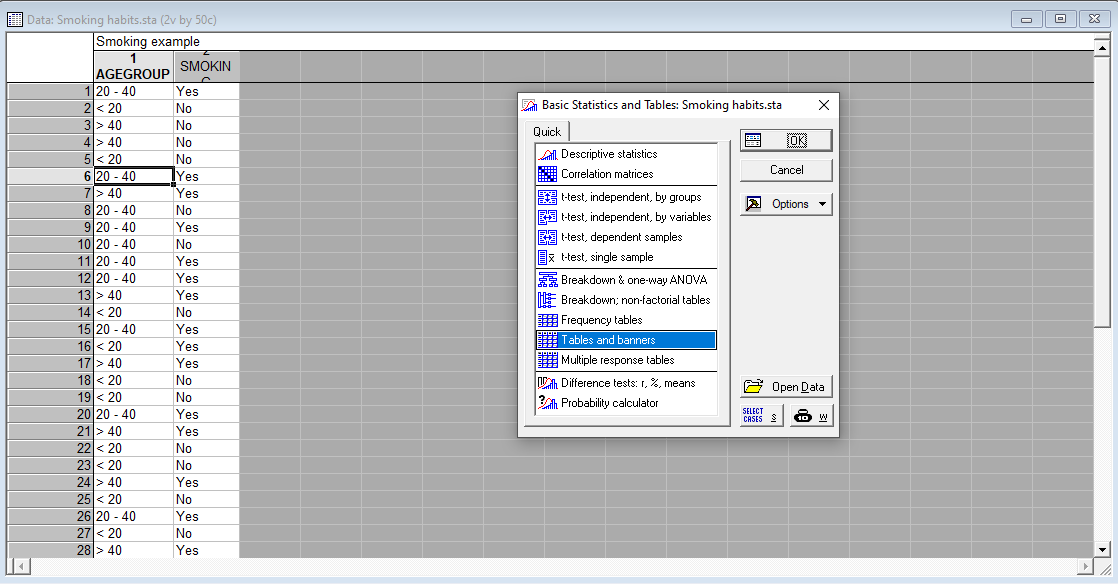
else cout<<"The null hypothesis is correct\n";

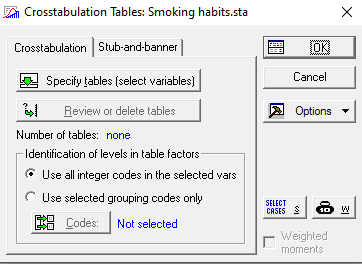
return 0;

}

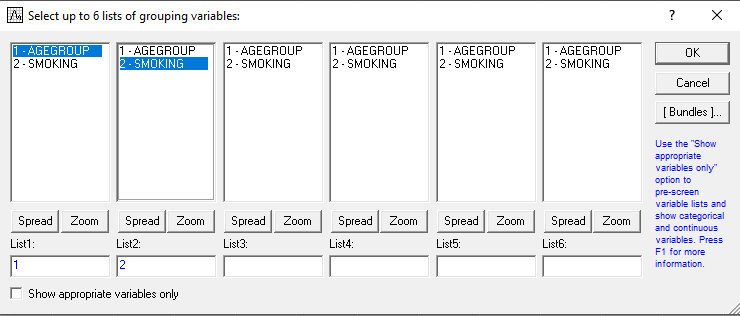


1. **Implimentation of** CHI-Square method on Statistca
2. Select the dataset and then, tables/banners option as shown:-

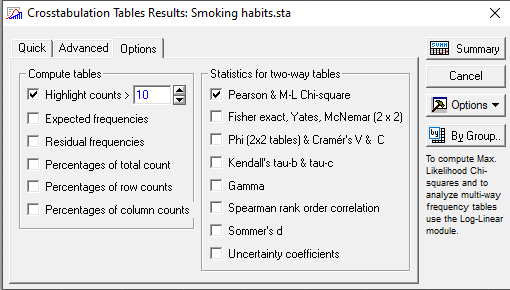


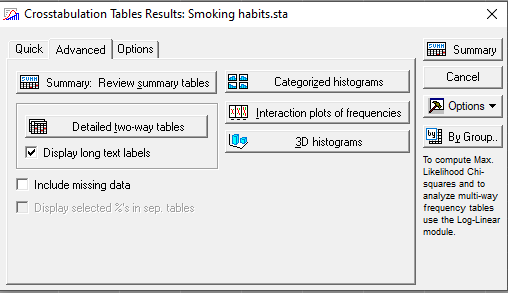


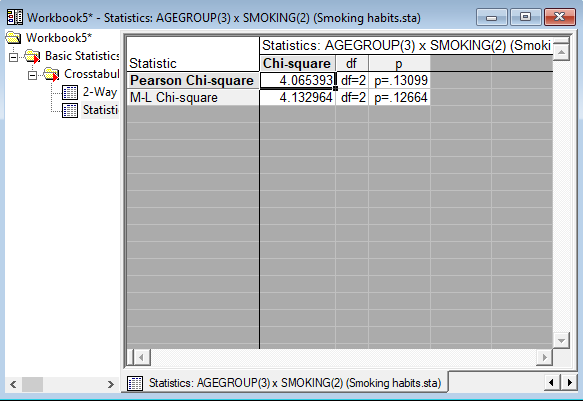
1. Select the list one and list two variables:-



1. Select Pearson & M-L Chi-square:-







Result