# SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE

NAAC ACCREDITED 'A' GRADE COLLEGE

(ISO 9001:2015 CERTIFIED INSTITUTION)

NERUL, NAVI MUMBAI - 400706

## Certificate

Seat No: 3713538

Certified that VARMA VISHAL VIJAY

Of Class MSC.IT PART-1 has duly completed the practical

course in the subject of RESEARCH IN COMPUTING

during the academic year 2021-22 as per the syllabus

prescribed by the University of Mumbai.

**Subject Teacher** **External Examiner**

**Head of Department** **Principal**

# Index

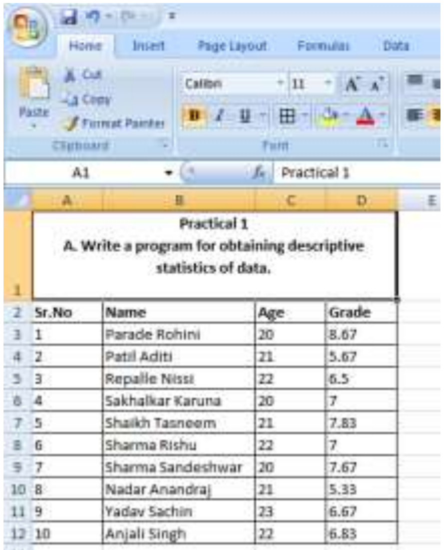
| Sr. No. | Practical | Date | Sign |
|---|---|---|---|
| **1** | A. Write a program for obtaining on descriptive statistics of data.<br>B. Import data from different data sources (from Excel, csv, mysql, sql server, oracle to R/Python/Excel) | | |
| **2** | A. Design a survey form for a given case study, collect the primary data and analyse it<br>B. Perform suitable analysis of given secondary data. | | |
| **3** | A. Perform testing of hypothesis using one sample t-test.<br>B. Perform testing of hypothesis using two sample t-test.<br>C. Perform testing of hypothesis using paired t-test. | | |
| **4** | A. Perform testing of hypothesis using chi-squared goodness-of-fit test.<br>B. Perform testing of hypothesis using chi-squared Test of Independence | | |
| **5** | Perform testing of hypothesis using Z-test. | | |
| **6** | A. Perform testing of hypothesis using one-way ANOVA.<br>B. Perform testing of hypothesis using two-way ANOVA.<br>C. Perform testing of hypothesis using multivariate ANOVA (MANOVA). | | |
| **7** | A. Perform the Random sampling for the given data and analyse it.<br>B. Perform the Stratified sampling for the given data and analyse it. | | |
| **8** | Compute different types of correlation. | | |
| **9** | A. Perform linear regression for prediction.<br>B. Perform polynomial regression for prediction. | | |
| **10** | A. Perform multiple linear regression.<br>B. Perform Logistic regression. | | |

# Practical No 1

## A. <u>Write a program for obtaining descriptive statistics of data.</u>

**Solution:**

The data is usually given in the exam or we can generate our own sample data.

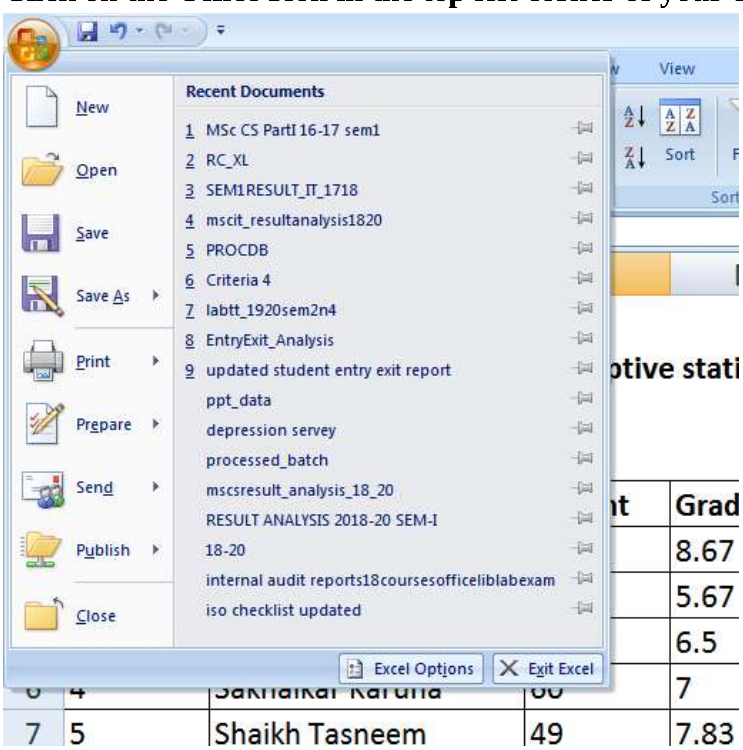


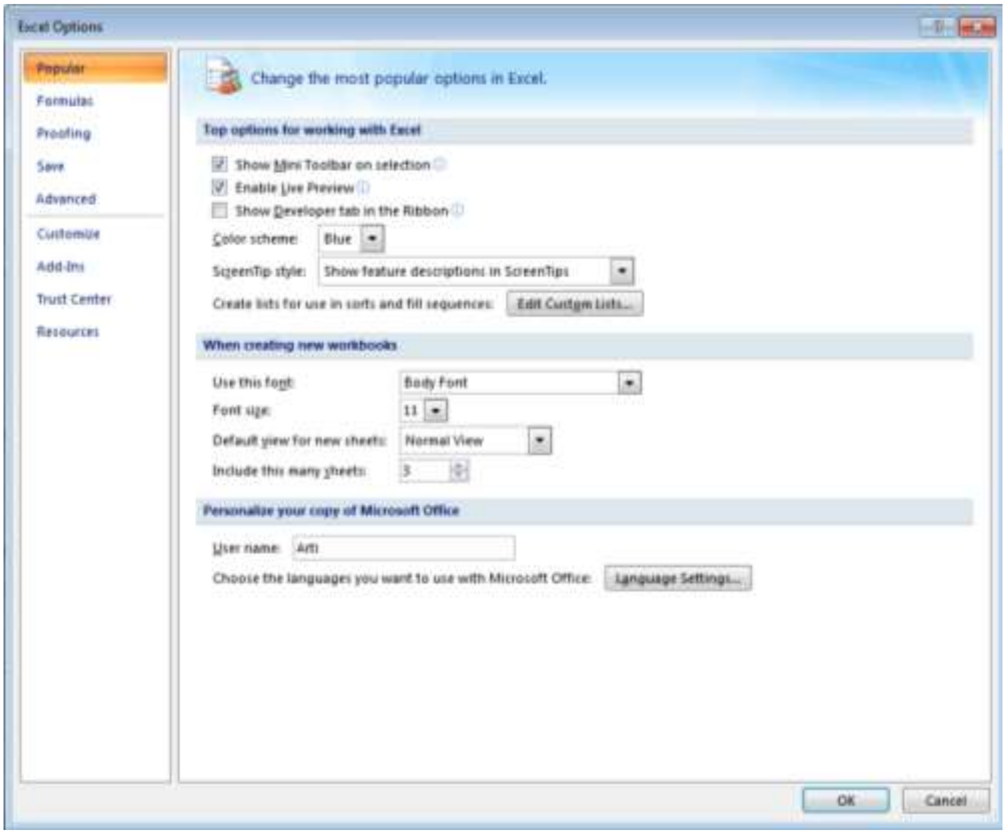
Practical 1
A. Write a program for obtaining descriptive statistics of data.

| Sr.No | Name | Age | Grade |
|---|---|---|---|
| 1 | Parade Rohini | 20 | 8.67 |
| 2 | Patil Aditi | 21 | 5.67 |
| 3 | Repalle Nissi | 22 | 6.5 |
| 4 | Sakhalkar Karuna | 20 | 7 |
| 5 | Shaikh Tasneem | 21 | 7.83 |
| 6 | Sharma Rishu | 22 | 7 |
| 7 | Sharma Sandeshwar | 20 | 7.67 |
| 8 | Nadar Anandraj | 21 | 5.33 |
| 9 | Yadav Sachin | 23 | 6.67 |
| 10 | Anjali Singh | 22 | 6.83 |

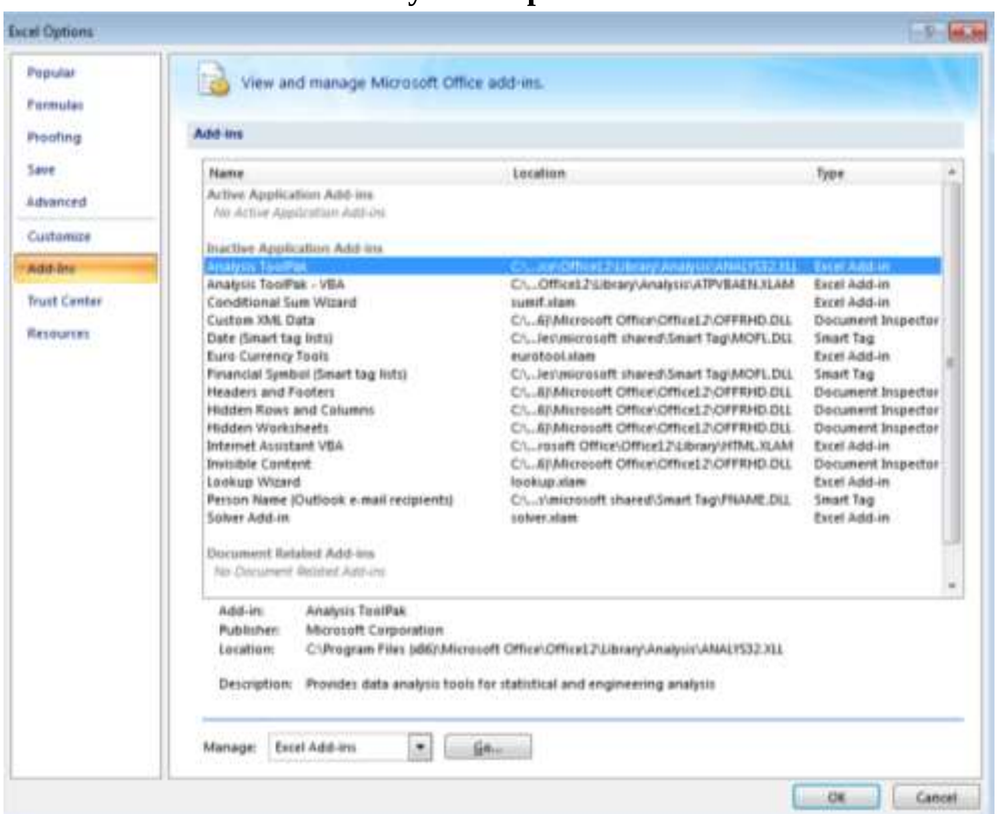
**First, we need to install the Data Analysis pack in Excel. The steps for installation are given below:**

**Click on the Office Icon in the top left corner of your excel window. Click on Excel Options**

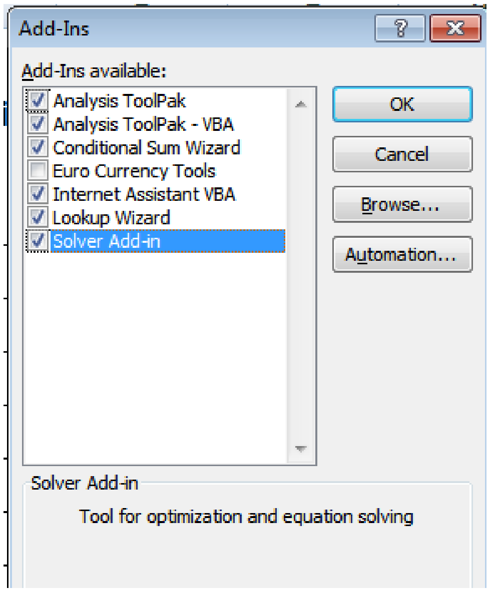




**Click on Add-Ins. Select Analysis Toolpak and click on Go.**

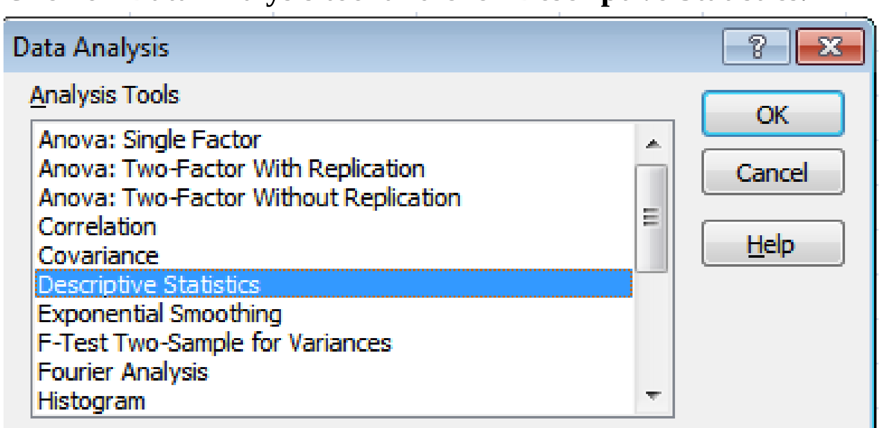


**Select all of the options given in the checkbox. Eurocurrency is optional and can be ignored. Click OK.**

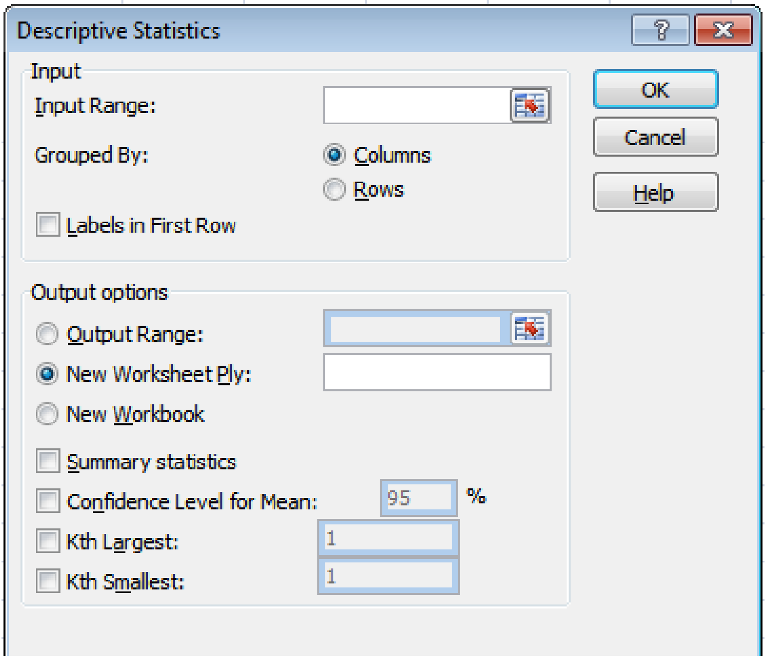


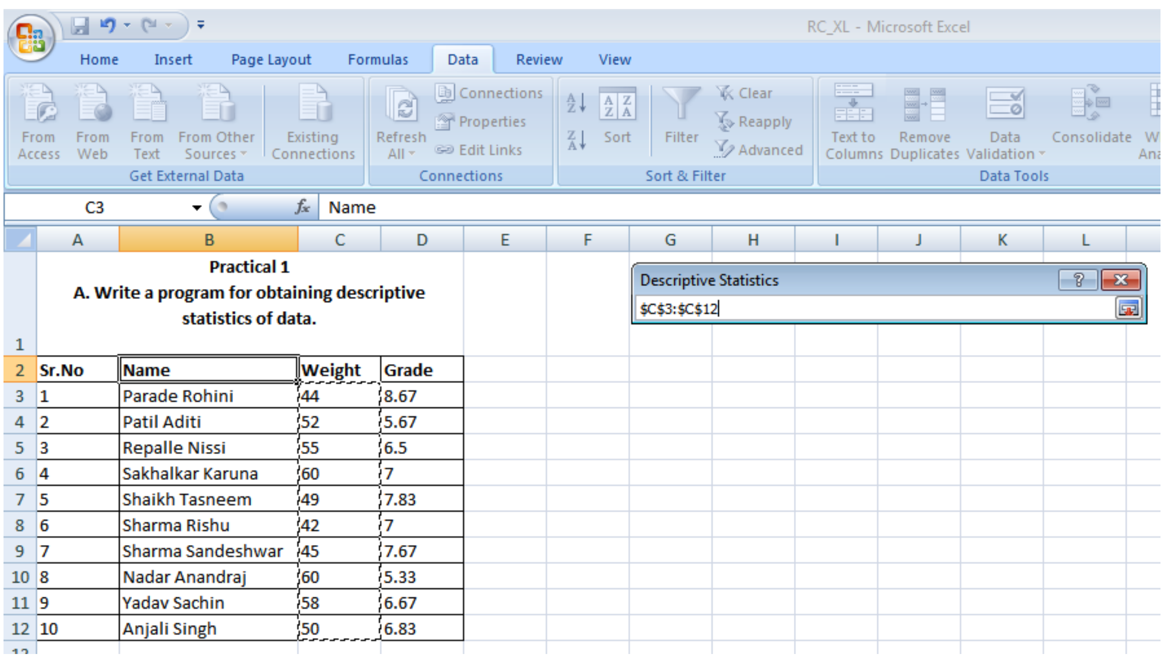

**The data analysis tool is installed successfully.**


**Click on Data Analysis tool and click Descriptive Statistics.**

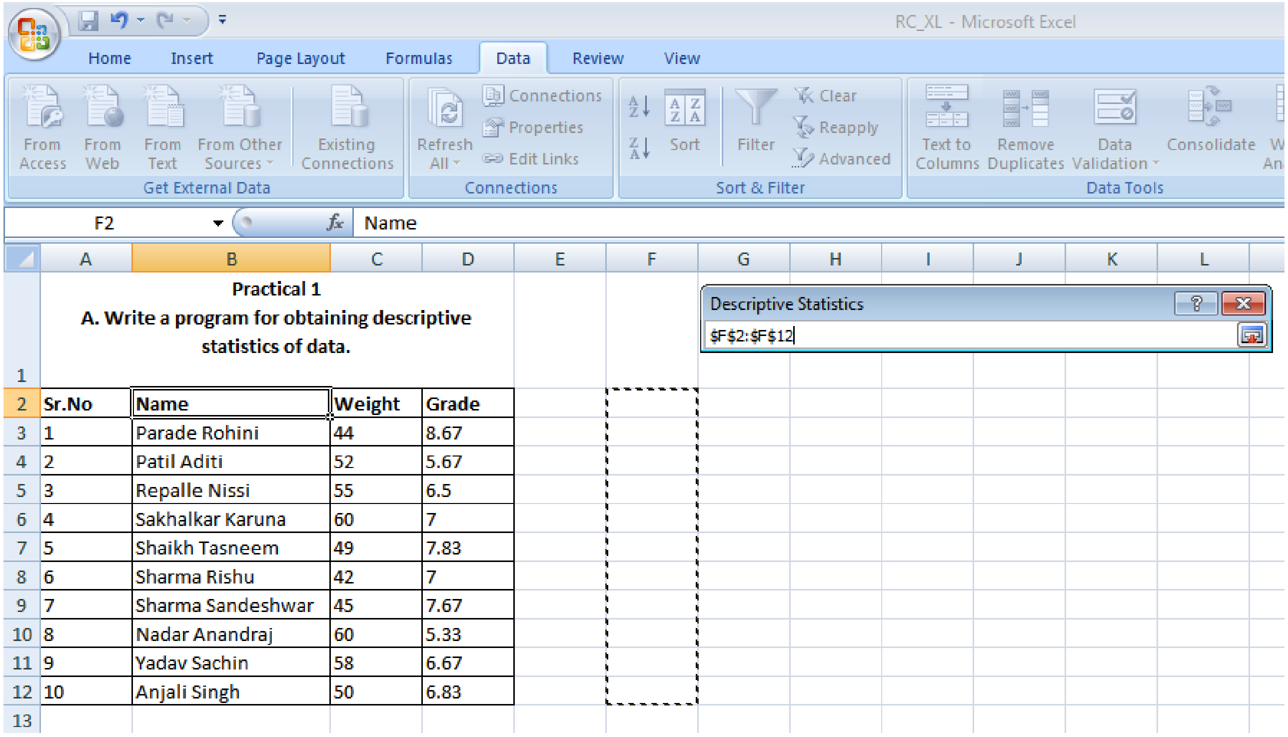


**Select the Input Range as the value for which you have been asked to or need to take statistics. In this example , we will consider the values of weight. Click on the icon besides the input range textbox to select the range.**

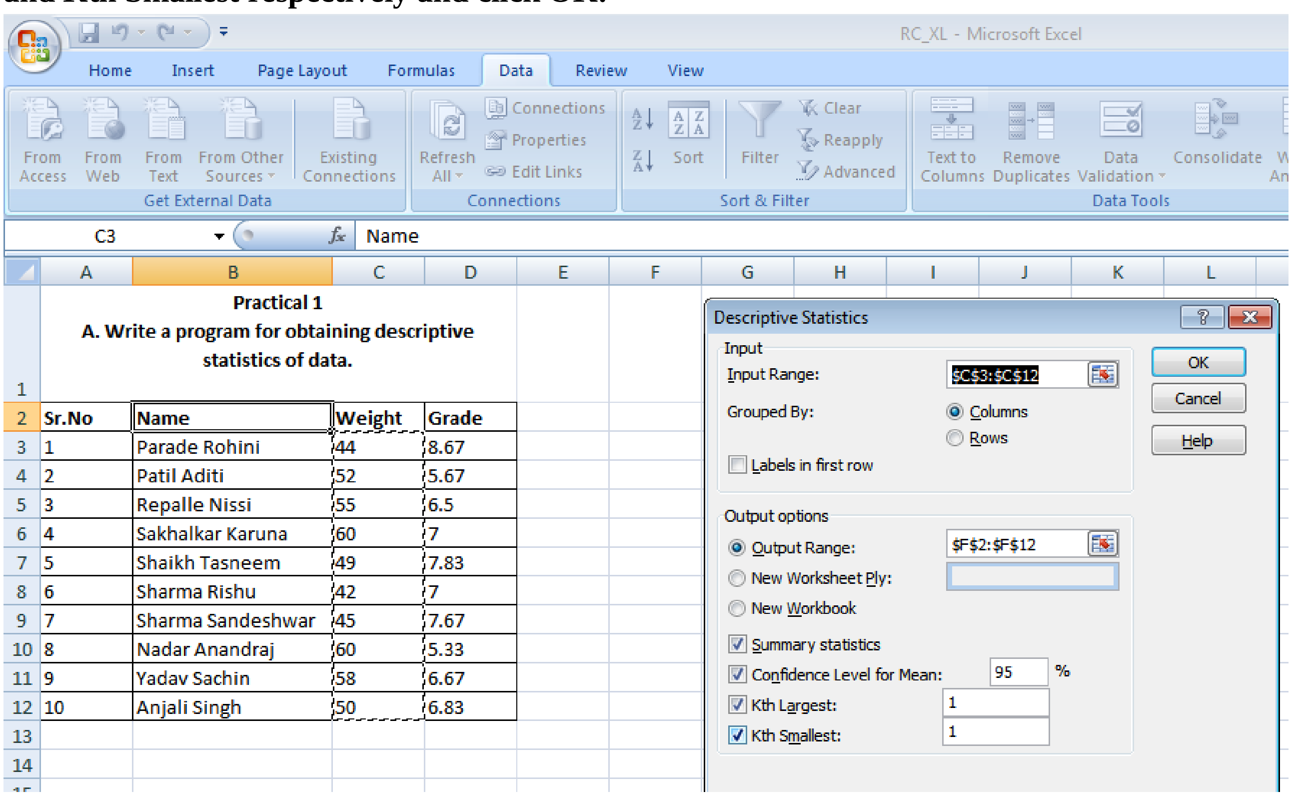


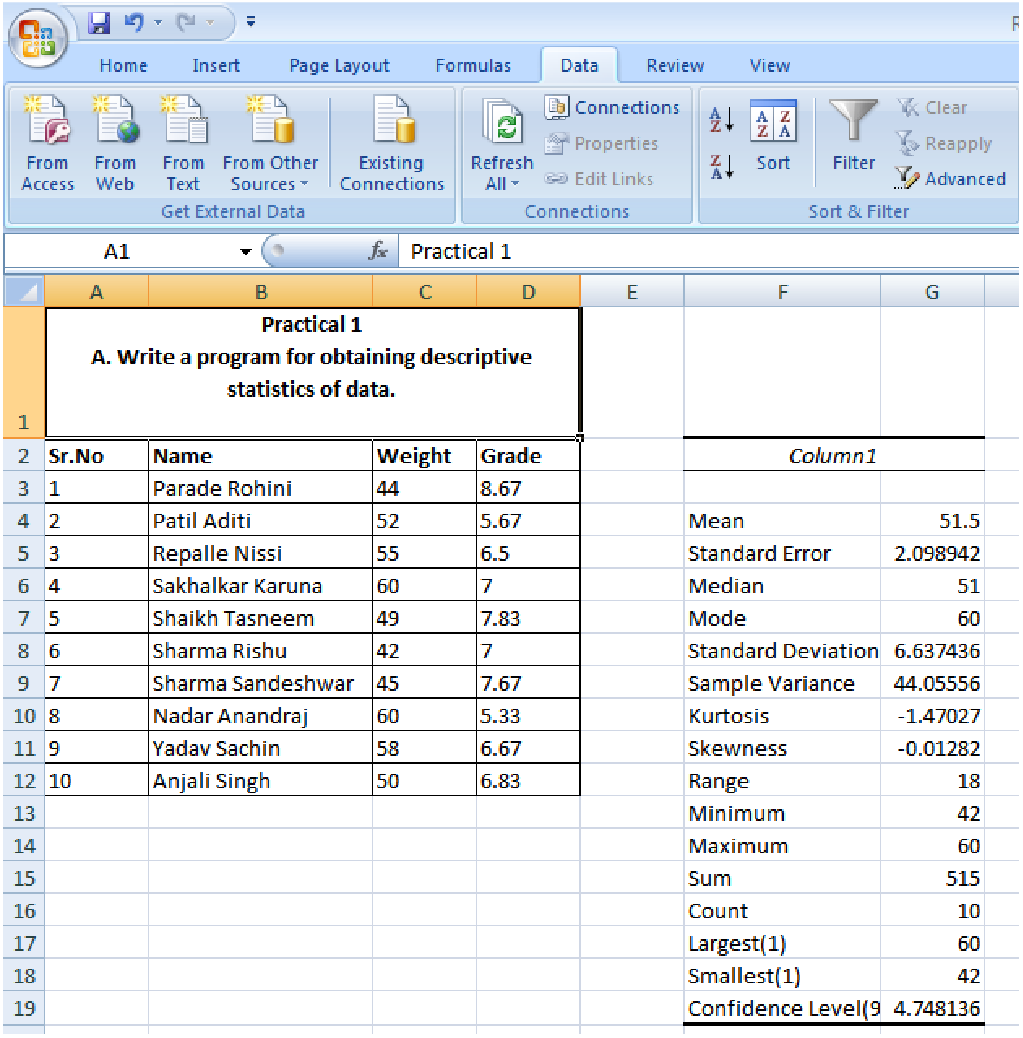
**Next, Select the output range i.e. the cells where you want the output to be displayed.**




**Select all 4 checkboxes namely Summary Statistics, Confidence Level for mean, Kth largest and Kth Smallest respectively and click OK.**



**The output is given is as shown below. It describes various statistics related to your sample data.**




Practical 1
A. Write a program for obtaining descriptive statistics of data.

| Sr.No | Name | Weight | Grade |
|---|---|---|---|
| 1 | Parade Rohini | 44 | 8.67 |
| 2 | Patil Aditi | 52 | 5.67 |
| 3 | Repalle Nissi | 55 | 6.5 |
| 4 | Sakhalkar Karuna | 60 | 7 |
| 5 | Shaikh Tasneem | 49 | 7.83 |
| 6 | Sharma Rishu | 42 | 7 |
| 7 | Sharma Sandeshwar | 45 | 7.67 |
| 8 | Nadar Anandraj | 60 | 5.33 |
| 9 | Yadav Sachin | 58 | 6.67 |
| 10 | Anjali Singh | 50 | 6.83 |

| *Column1* | |
|---|---|
| Mean | 51.5 |
| Standard Error | 2.098942 |
| Median | 51 |
| Mode | 60 |
| Standard Deviation | 6.637436 |
| Sample Variance | 44.05556 |
| Kurtosis | -1.47027 |
| Skewness | -0.01282 |
| Range | 18 |
| Minimum | 42 |
| Maximum | 60 |
| Sum | 515 |
| Count | 10 |
| Largest(1) | 60 |
| Smallest(1) | 42 |
| Confidence Level(9 | 4.748136 |

## B. Import data from different data sources (from Excel, csv, mysql, sql server, oracle to R/Python/Excel)

SQLite:

**Code:**

```
import sqlite3 as sq
import pandas as pd
################################################################
Base='C:/VKHCG'
sDatabaseName=Base + '/01-Vermeulen/00-RawData/SQLite/vermeulen.db'
conn = sq.connect(sDatabaseName)
sFileName='C:/VKHCG/01-Vermeulen/01-Retrieve/01-EDS/02-Python/Retrieve_IP_DATA.csv'
print('Loading :',sFileName)
IP_DATA_ALL_FIX=pd.read_csv(sFileName,header=0,low_memory=False)
IP_DATA_ALL_FIX.index.names = ['RowIDCSV']
```

```
sTable='IP_DATA_ALL'
print('Storing :',sDatabaseName,' Table:',sTable)
IP_DATA_ALL_FIX.to_sql(sTable, conn, if_exists="replace")
print('Loading :',sDatabaseName,' Table:',sTable)
TestData=pd.read_sql_query("select * from IP_DATA_ALL;", conn)
print('################')
print('## Data Values')
print('################')
print(TestData)
print('################')
print('## Data Profile')
print('################')
print('Rows :',TestData.shape[0])
print('Columns :',TestData.shape[1])
print('################')
print('### Done!! ############################################')
```

**Output:**

```
Loading : C:/VKHCG/01-Vermeulen/01-Retrieve/01-EDS/02-Python/Retrieve_IP_DATA.csv
Storing : C:/VKHCG/01-Vermeulen/00-RawData/SQLite/vermeulen.db  Table: IP_DATA_ALL
Loading : C:/VKHCG/01-Vermeulen/00-RawData/SQLite/vermeulen.db  Table: IP_DATA_ALL
################
## Data Values
################
         RowIDCSV    RowID       ID  ... Longitude First.IP.Number Last.IP.Number
0               0        0        1  ...   25.9119       692781056      692781567
1               1        1        2  ...   25.9119       692781824      692783103
2               2        2        3  ...   25.9119       692909056      692909311
3               3        3        4  ...   25.9119       692909568      692910079
4               4        4        5  ...   25.9119       693051392      693052415
...           ...      ...      ...  ...       ...             ...            ...
1247497   1247497  1247497  1247498  ...  -79.7611      1068157850     1068157850
1247498   1247498  1247498  1247499  ...    8.7668      1334409600     1334409607
1247499   1247499  1247499  1247500  ...   43.2583      1596886528     1596886783
1247500   1247500  1247500  1247501  ...  -80.4451      1742189568     1742190591
1247501   1247501  1247501  1247502  ...  139.5357      1905782573     1905782573

[1247502 rows x 10 columns]
################
## Data Profile
################
Rows : 1247502
Columns : 10
################
### Done!! ############################################
```

**Microsoft Excel:**

**Code:**

```
##################Retrieve-Country-Currency.py
######################################################################
# -*- coding: utf-8 -*-
######################################################################
import os
import pandas as pd
######################################################################
Base='C:/VKHCG'
######################################################################
```

```
sFileDir=Base + '/01-Vermeulen/01-Retrieve/01-EDS/02-Python'
#if not os.path.exists(sFileDir):
#os.makedirs(sFileDir)
################################################################
CurrencyRawData = pd.read_excel('C:/VKHCG/01-Vermeulen/00-RawData/Country_Currency.xlsx')
sColumns = ['Country or territory', 'Currency', 'ISO-4217']
CurrencyData = CurrencyRawData[sColumns]
CurrencyData.rename(columns={'Country or territory': 'Country', 'ISO-4217':
'CurrencyCode'}, inplace=True)
CurrencyData.dropna(subset=['Currency'],inplace=True)
CurrencyData['Country'] = CurrencyData['Country'].map(lambda x: x.strip())
CurrencyData['Currency'] = CurrencyData['Currency'].map(lambda x:
x.strip())
CurrencyData['CurrencyCode'] = CurrencyData['CurrencyCode'].map(lambda x:
x.strip())
print(CurrencyData)
print('~~~~~~ Data from Excel Sheet Retrived Successfully ~~~~~~~ ')
################################################################
sFileName=sFileDir + '/Retrieve-Country-Currency.csv'
CurrencyData.to_csv(sFileName, index = False)
################################################################
```

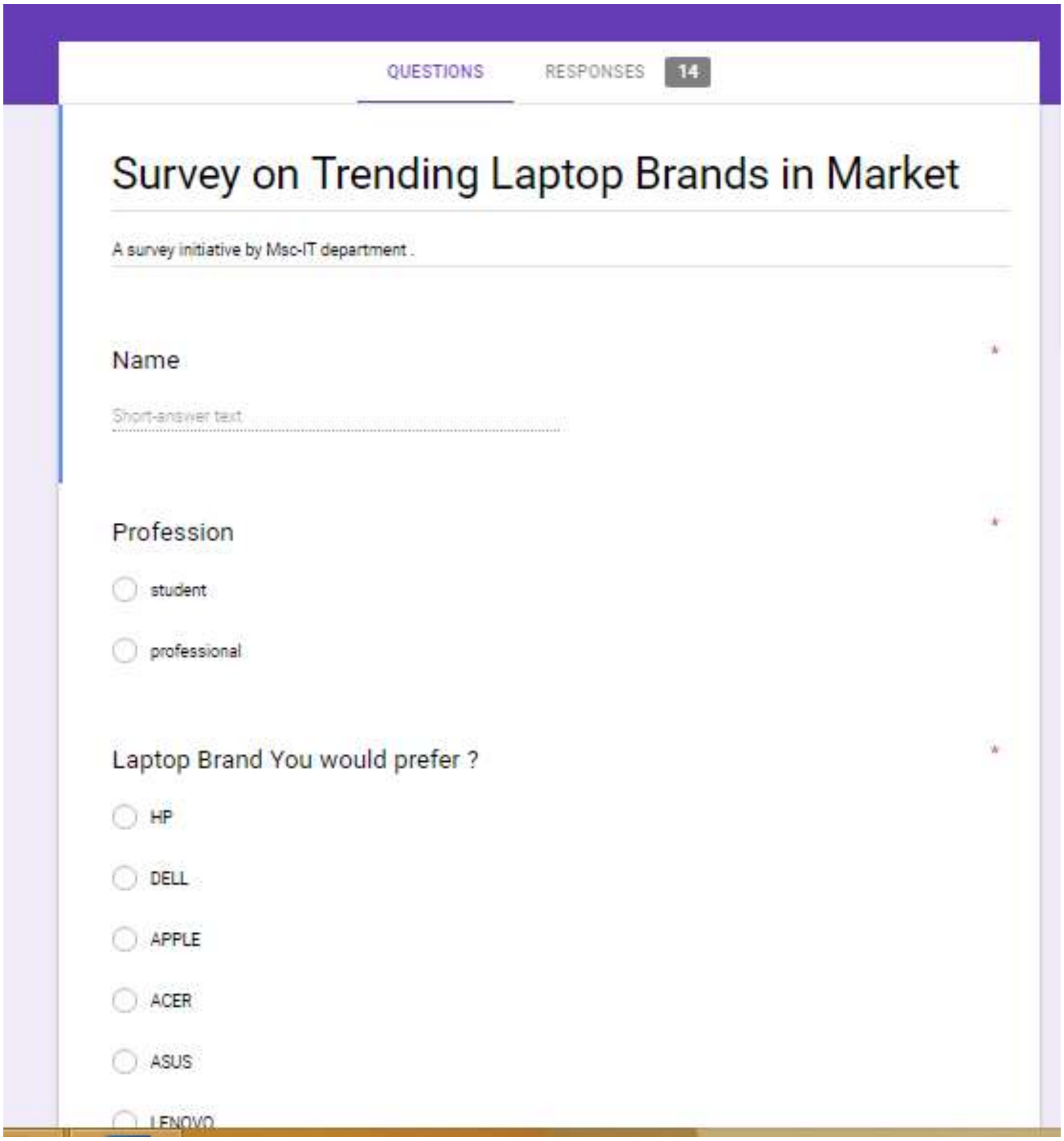
**Output:**

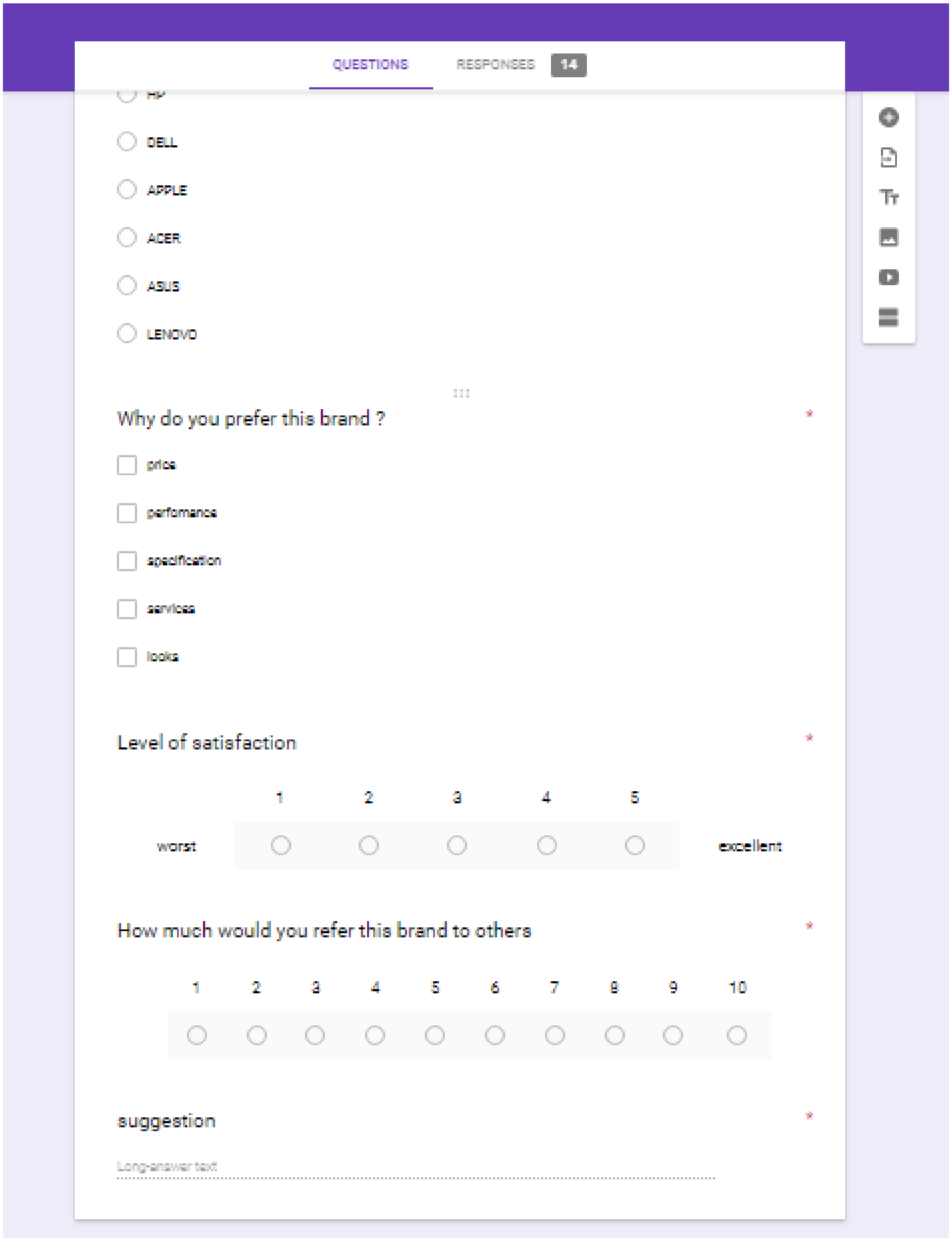
```
                        Country              Currency CurrencyCode
1                   Afghanistan        Afghan afghani          AFN
2    Akrotiri and Dhekelia (UK)         European euro          EUR
3       Aland Islands (Finland)         European euro          EUR
4                       Albania          Albanian lek          ALL
5                       Algeria        Algerian dinar          DZD
..                          ...                   ...          ...
271           Wake Island (USA)  United States dollar          USD
272  Wallis and Futuna (France)             CFP franc          XPF
274                       Yemen           Yemeni rial          YER
276                      Zambia        Zambian kwacha          ZMW
277                    Zimbabwe  United States dollar          USD

[253 rows x 3 columns]
~~~~~~ Data from Excel Sheet Retrived Successfully ~~~~~~~
```

# Practical No 2

A. Design a survey form for a given case study, collect the primary data and analyse it.

➔ Open Google forms →select empty form→edit according to the case study→click on share button:





QUESTIONS RESPONSES 14

HP

DELL

APPLE

ACER

ASUS

LENOVO

Why do you prefer this brand ? *

price

perfomance

specification

services

looks

Level of satisfaction *

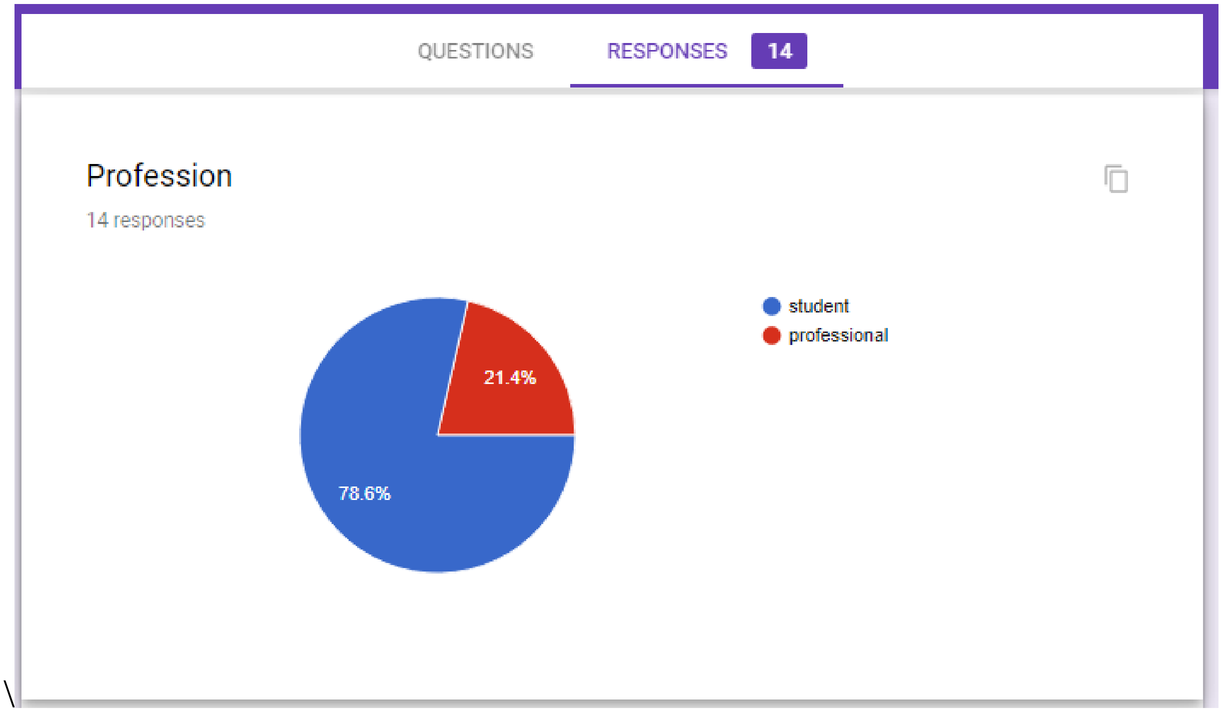
| | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| worst | | | | | | excellent |

How much would you refer this brand to others *

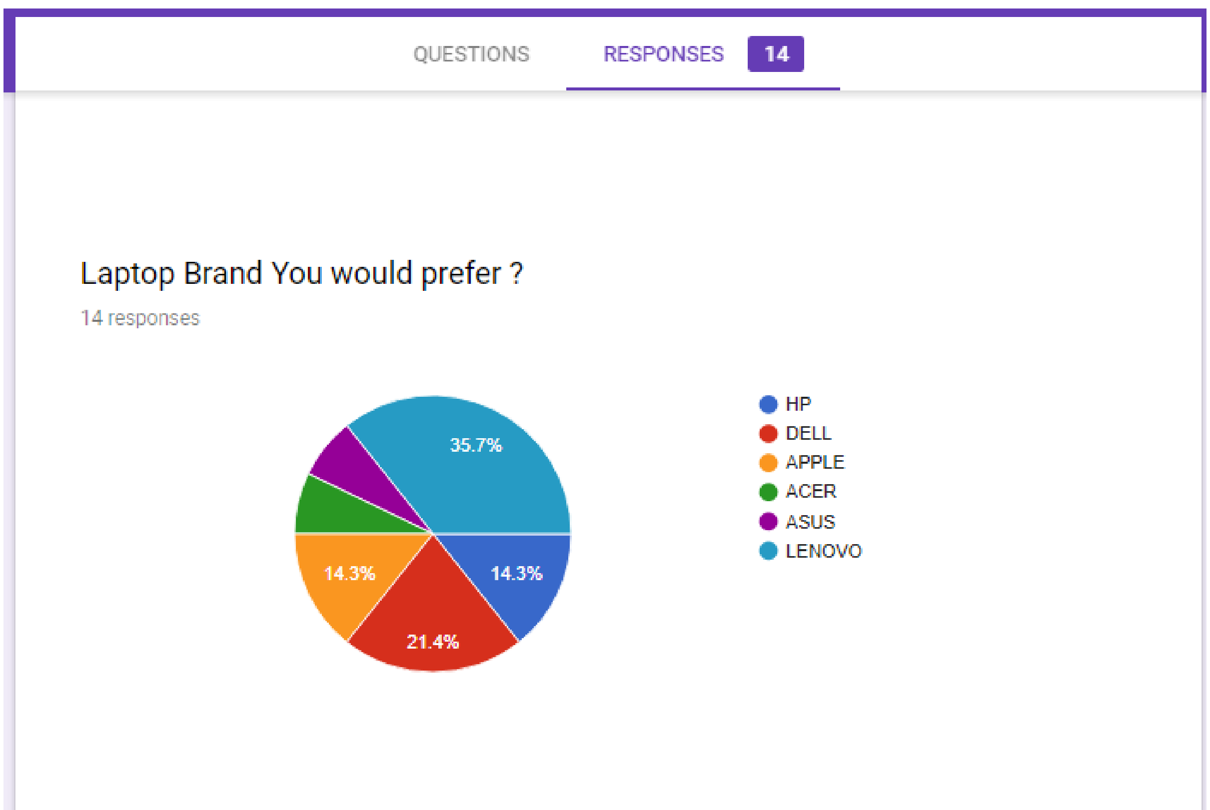
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|

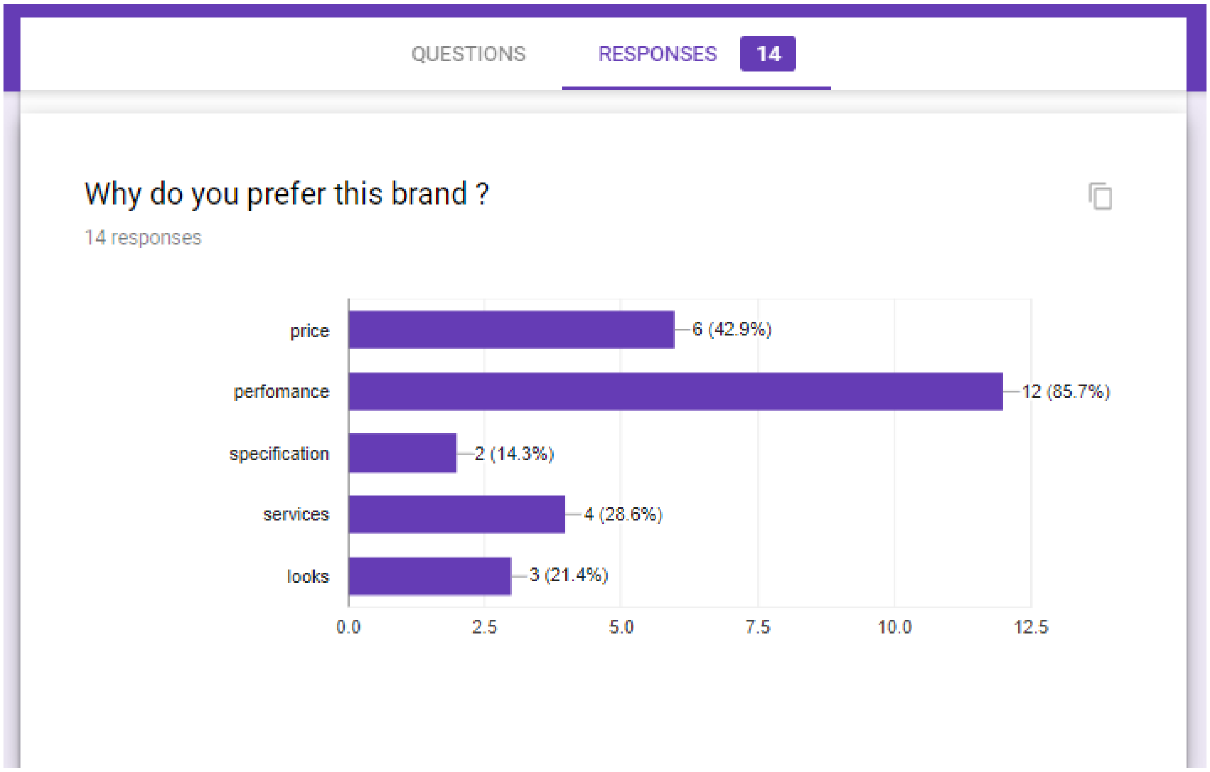
suggestion *

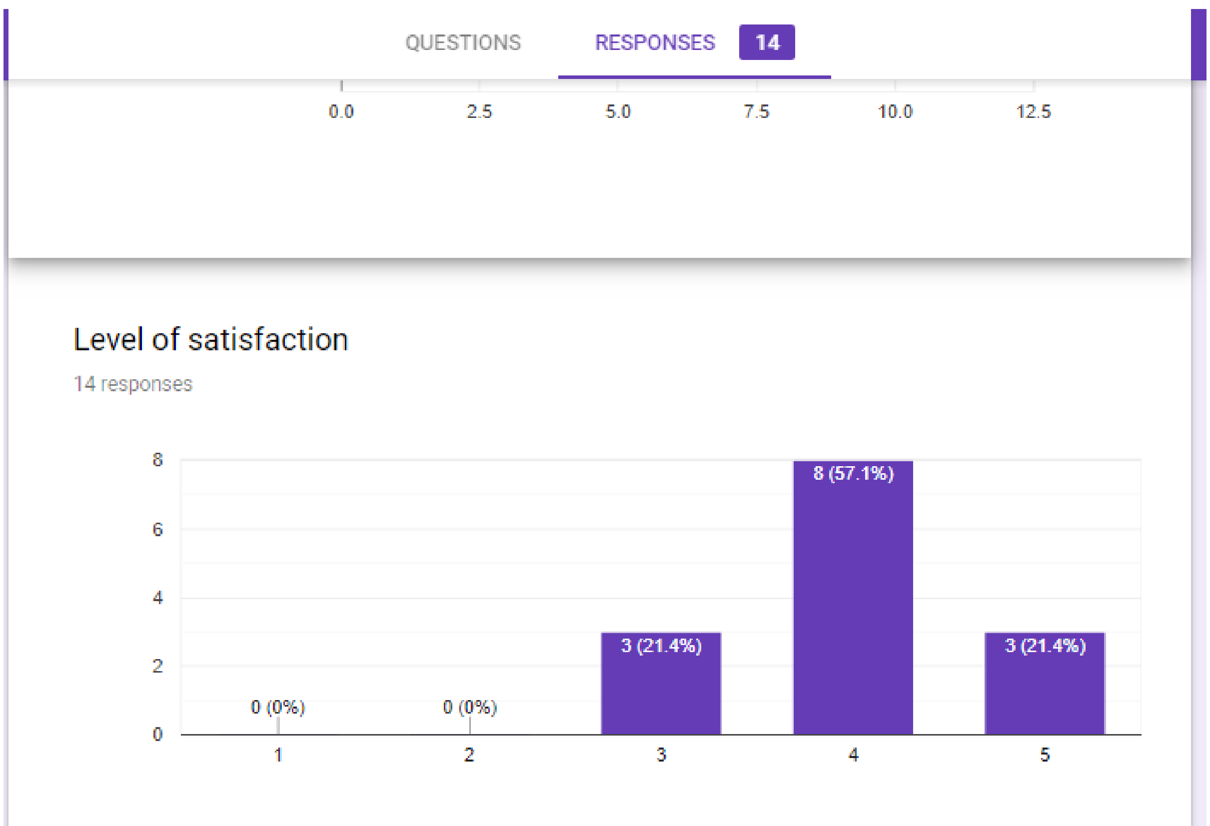
Long-answer text

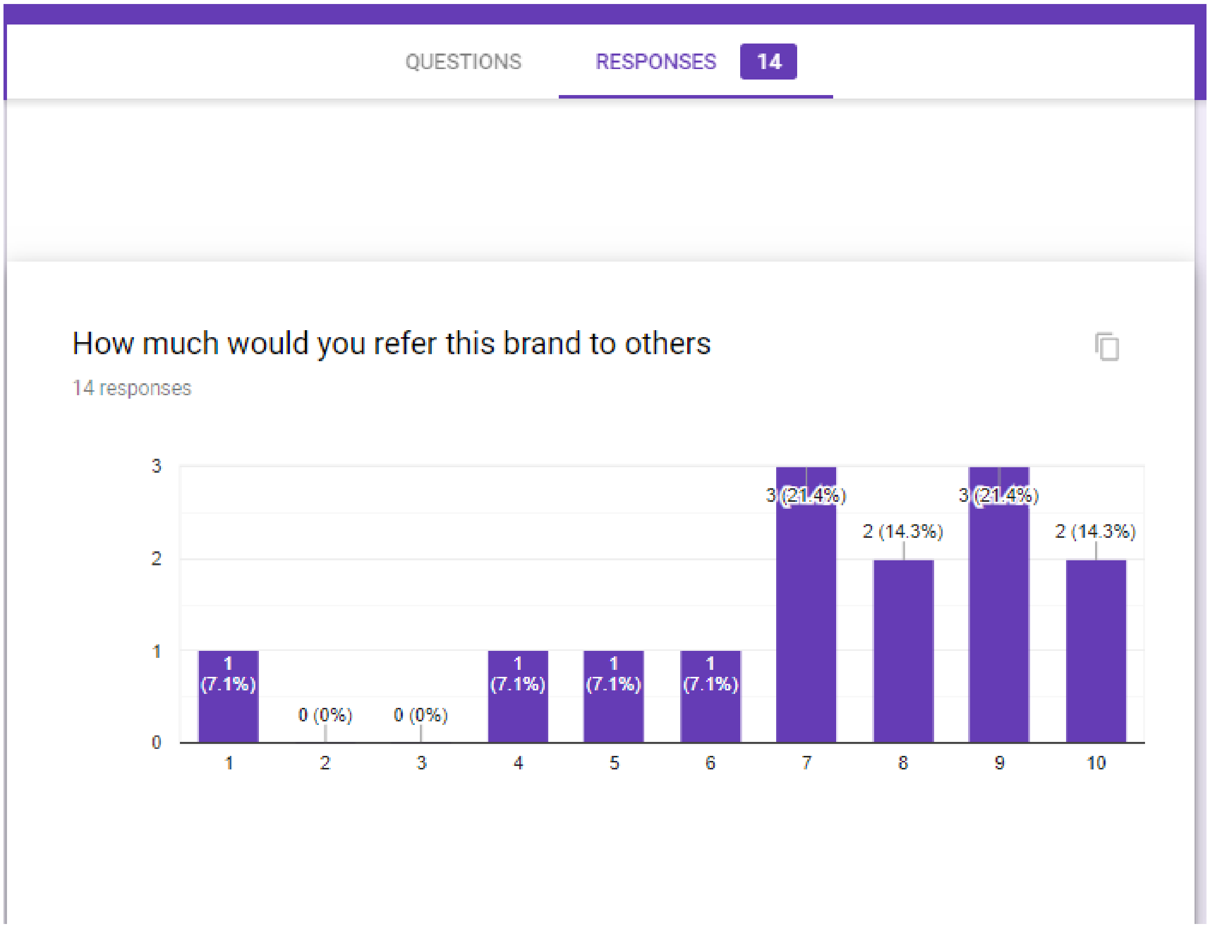

QUESTIONS
RESPONSES 14
Profession
14 responses
21.4%
78.6%
student
professional

\


QUESTIONS
RESPONSES 14
Laptop Brand You would prefer ?
14 responses
35.7%
14.3%
14.3%
21.4%
HP
DELL
APPLE
ACER
ASUS
LENOVO

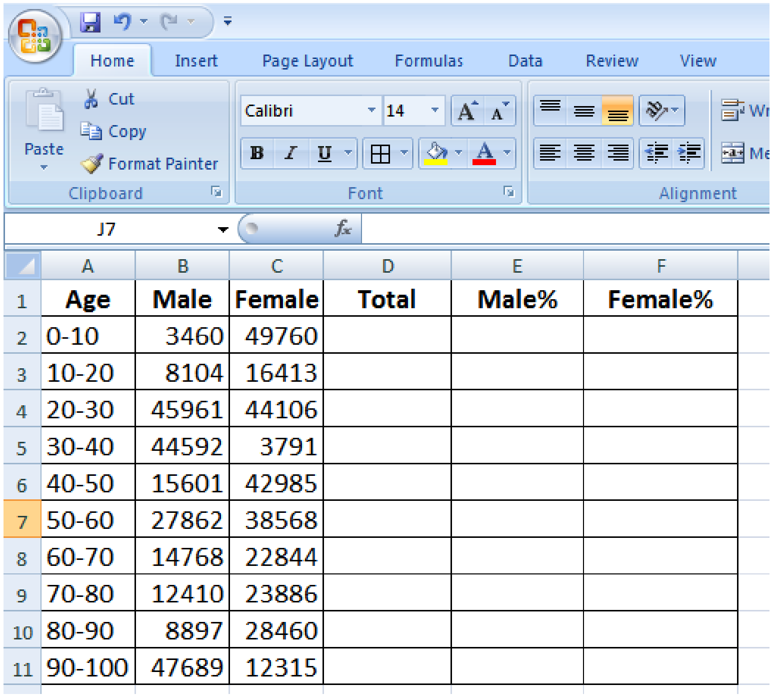

QUESTIONS
RESPONSES 14
Why do you prefer this brand ?
14 responses
price
perfomance
specification
services
looks
6 (42.9%)
12 (85.7%)
2 (14.3%)
4 (28.6%)
3 (21.4%)
0.0
2.5
5.0
7.5
10.0
12.5


QUESTIONS
RESPONSES 14
0.0
2.5
5.0
7.5
10.0
12.5
Level of satisfaction
14 responses
8
6
4
2
0
0 (0%)
0 (0%)
3 (21.4%)
8 (57.1%)
3 (21.4%)
1
2
3
4
5


QUESTIONS
RESPONSES 14
How much would you refer this brand to others
14 responses
3
2
1
0
1 (7.1%)
0 (0%)
0 (0%)
1 (7.1%)
1 (7.1%)
1 (7.1%)
3 (21.4%)
2 (14.3%)
3 (21.4%)
2 (14.3%)
1
2
3
4
5
6
7
8
9
10

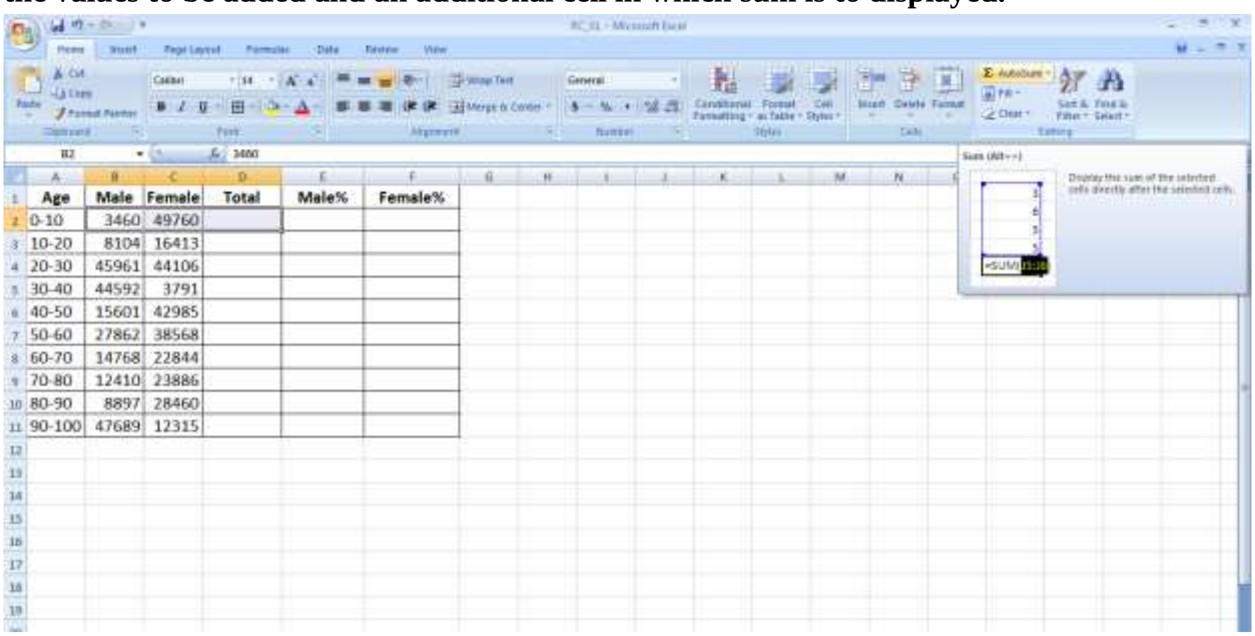
## B. <u>Perform analysis of given secondary data.</u>

**Solution: The data will be given or can be randomly generated.**



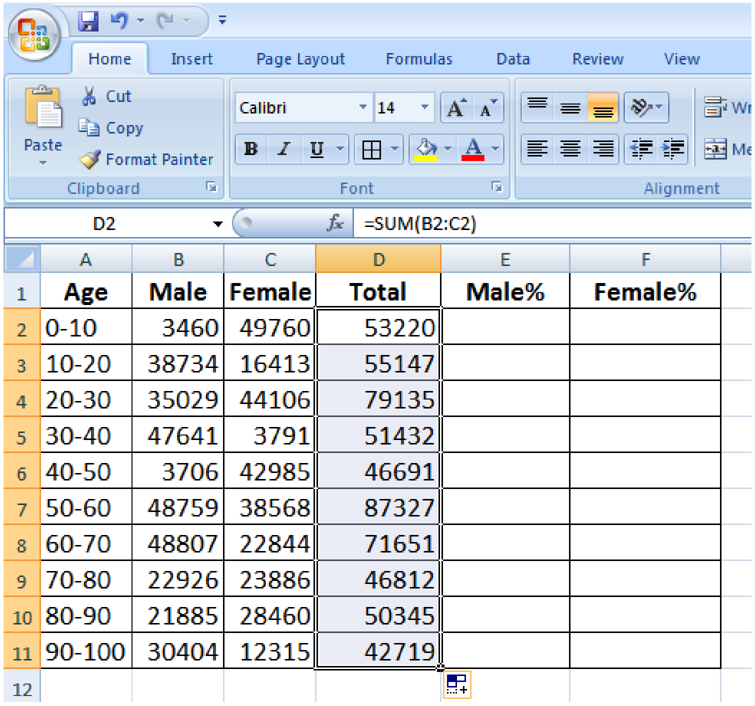

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Age | Male | Female | Total | Male% | Female% |
| 2 | 0-10 | 3460 | 49760 | | | |
| 3 | 10-20 | 8104 | 16413 | | | |
| 4 | 20-30 | 45961 | 44106 | | | |
| 5 | 30-40 | 44592 | 3791 | | | |
| 6 | 40-50 | 15601 | 42985 | | | |
| 7 | 50-60 | 27862 | 38568 | | | |
| 8 | 60-70 | 14768 | 22844 | | | |
| 9 | 70-80 | 12410 | 23886 | | | |
| 10 | 80-90 | 8897 | 28460 | | | |
| 11 | 90-100 | 47689 | 12315 | | | |

**We start by adding the values in the male and female column. To perform addition, select the values to be added and an additional cell in which sum is to displayed.**



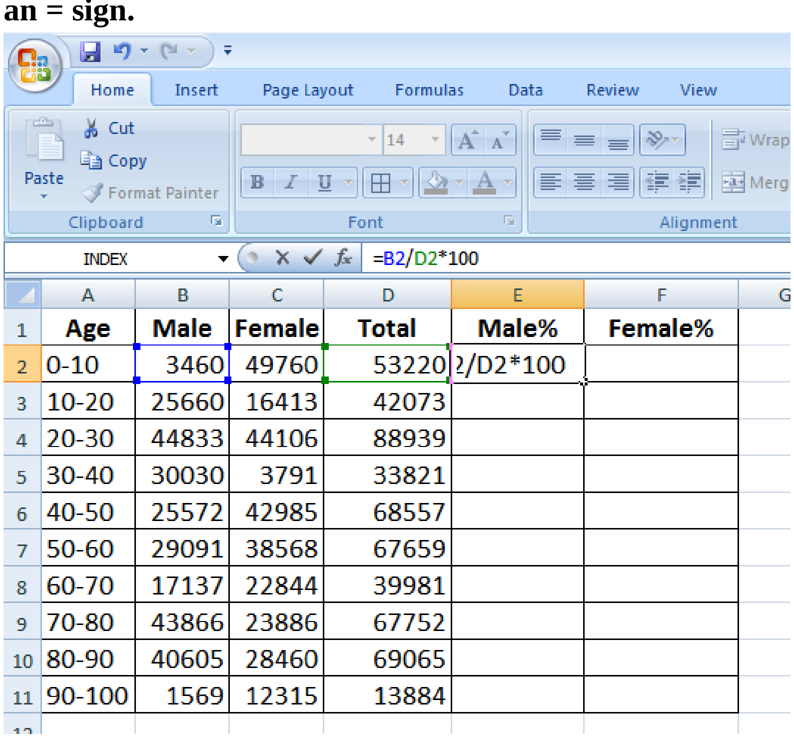
| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Age | Male | Female | Total | Male% | Female% |
| 2 | 0-10 | 3460 | 49760 | | | |
| 3 | 10-20 | 8104 | 16413 | | | |
| 4 | 20-30 | 45961 | 44106 | | | |
| 5 | 30-40 | 44592 | 3791 | | | |
| 6 | 40-50 | 15601 | 42985 | | | |
| 7 | 50-60 | 27862 | 38568 | | | |
| 8 | 60-70 | 14768 | 22844 | | | |
| 9 | 70-80 | 12410 | 23886 | | | |
| 10 | 80-90 | 8897 | 28460 | | | |
| 11 | 90-100 | 47689 | 12315 | | | |

**Next drag the dot at the end of the cell to apply the formula to other columns. Take your mouse near the cell the pointer changes to plus indicating we should drag the mouse in the cells we wish to perform a similar action. This will auto fill the values in the column.**



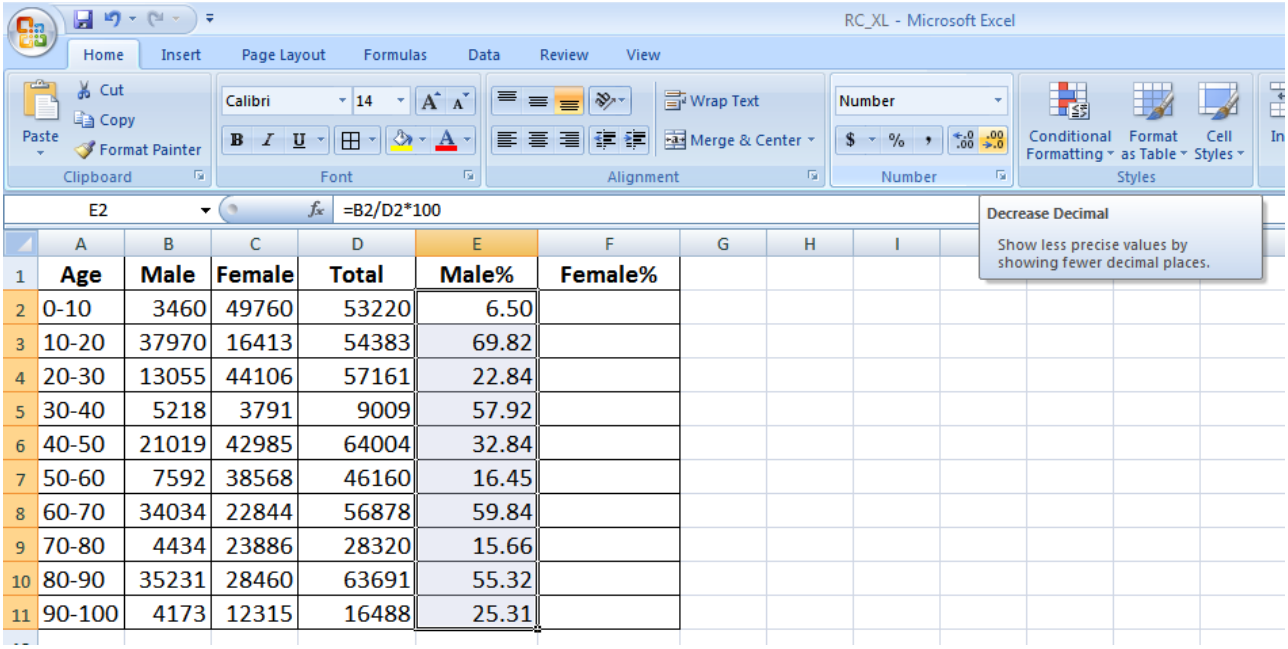

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Age | Male | Female | Total | Male% | Female% |
| 2 | 0-10 | 3460 | 49760 | 53220 | | |
| 3 | 10-20 | 38734 | 16413 | 55147 | | |
| 4 | 20-30 | 35029 | 44106 | 79135 | | |
| 5 | 30-40 | 47641 | 3791 | 51432 | | |
| 6 | 40-50 | 3706 | 42985 | 46691 | | |
| 7 | 50-60 | 48759 | 38568 | 87327 | | |
| 8 | 60-70 | 48807 | 22844 | 71651 | | |
| 9 | 70-80 | 22926 | 23886 | 46812 | | |
| 10 | 80-90 | 21885 | 28460 | 50345 | | |
| 11 | 90-100 | 30404 | 12315 | 42719 | | |
| 12 | | | | | | |

**Next step is to calculate the percentage of males in the data. We need to add formula for this. Click on the textbox near the fx icon to type the formula. Always start a formula with an = sign.**



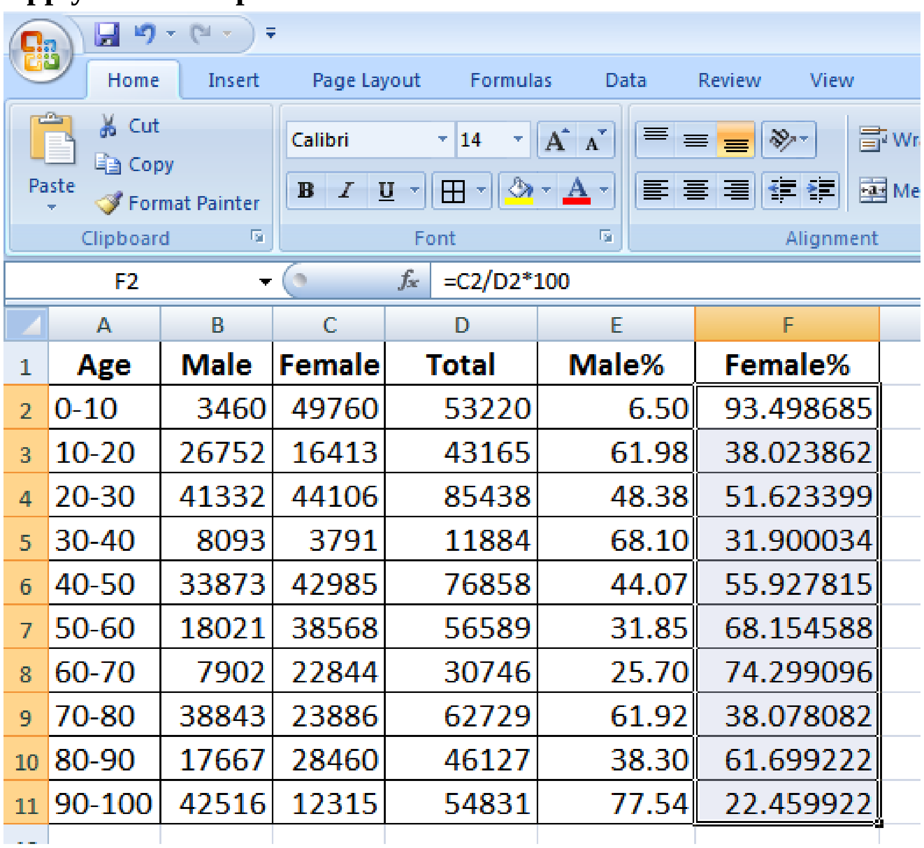

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Age | Male | Female | Total | Male% | Female% |
| 2 | 0-10 | 3460 | 49760 | 53220 | 2/D2*100 | |
| 3 | 10-20 | 25660 | 16413 | 42073 | | |
| 4 | 20-30 | 44833 | 44106 | 88939 | | |
| 5 | 30-40 | 30030 | 3791 | 33821 | | |
| 6 | 40-50 | 25572 | 42985 | 68557 | | |
| 7 | 50-60 | 29091 | 38568 | 67659 | | |
| 8 | 60-70 | 17137 | 22844 | 39981 | | |
| 9 | 70-80 | 43866 | 23886 | 67752 | | |
| 10 | 80-90 | 40605 | 28460 | 69065 | | |
| 11 | 90-100 | 1569 | 12315 | 13884 | | |

**Drag to fill the remaining rows.**



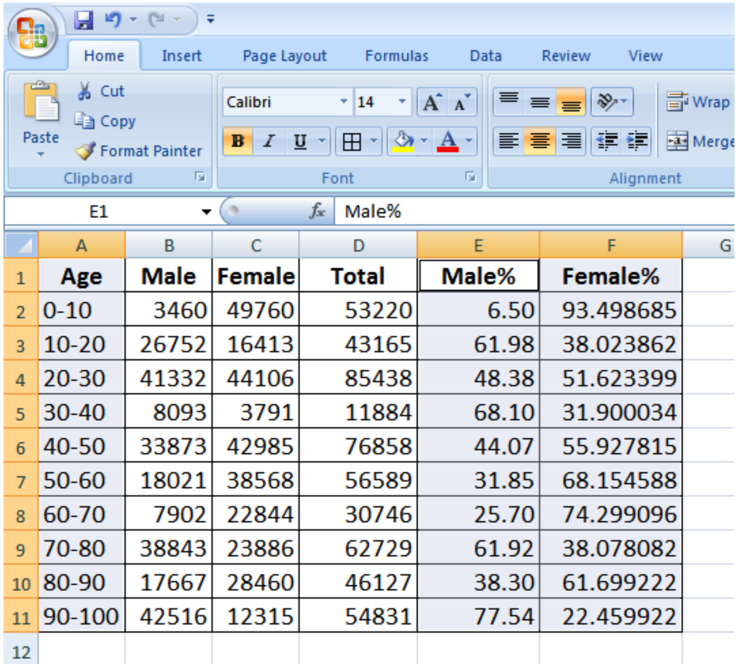

| Age | Male | Female | Total | Male% | Female% |
|---|---|---|---|---|---|
| 0-10 | 3460 | 49760 | 53220 | 6.50 | |
| 10-20 | 37970 | 16413 | 54383 | 69.82 | |
| 20-30 | 13055 | 44106 | 57161 | 22.84 | |
| 30-40 | 5218 | 3791 | 9009 | 57.92 | |
| 40-50 | 21019 | 42985 | 64004 | 32.84 | |
| 50-60 | 7592 | 38568 | 46160 | 16.45 | |
| 60-70 | 34034 | 22844 | 56878 | 59.84 | |
| 70-80 | 4434 | 23886 | 28320 | 15.66 | |
| 80-90 | 35231 | 28460 | 63691 | 55.32 | |
| 90-100 | 4173 | 12315 | 16488 | 25.31 | |

**Apply similar steps to find female %.**



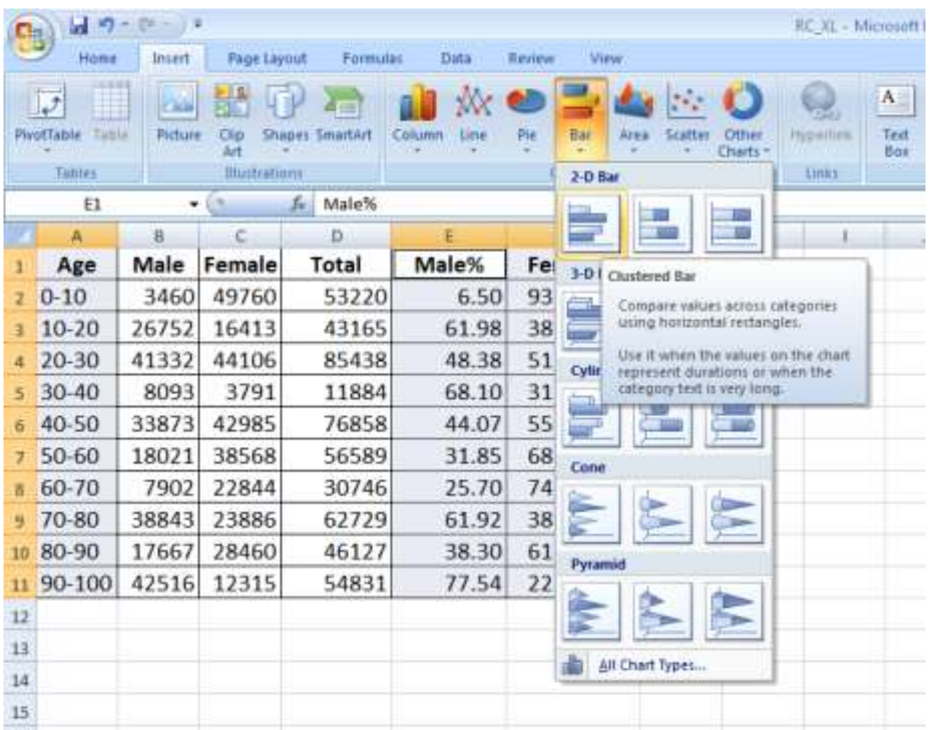

| Age | Male | Female | Total | Male% | Female% |
|---|---|---|---|---|---|
| 0-10 | 3460 | 49760 | 53220 | 6.50 | 93.498685 |
| 10-20 | 26752 | 16413 | 43165 | 61.98 | 38.023862 |
| 20-30 | 41332 | 44106 | 85438 | 48.38 | 51.623399 |
| 30-40 | 8093 | 3791 | 11884 | 68.10 | 31.900034 |
| 40-50 | 33873 | 42985 | 76858 | 44.07 | 55.927815 |
| 50-60 | 18021 | 38568 | 56589 | 31.85 | 68.154588 |
| 60-70 | 7902 | 22844 | 30746 | 25.70 | 74.299096 |
| 70-80 | 38843 | 23886 | 62729 | 61.92 | 38.078082 |
| 80-90 | 17667 | 28460 | 46127 | 38.30 | 61.699222 |
| 90-100 | 42516 | 12315 | 54831 | 77.54 | 22.459922 |

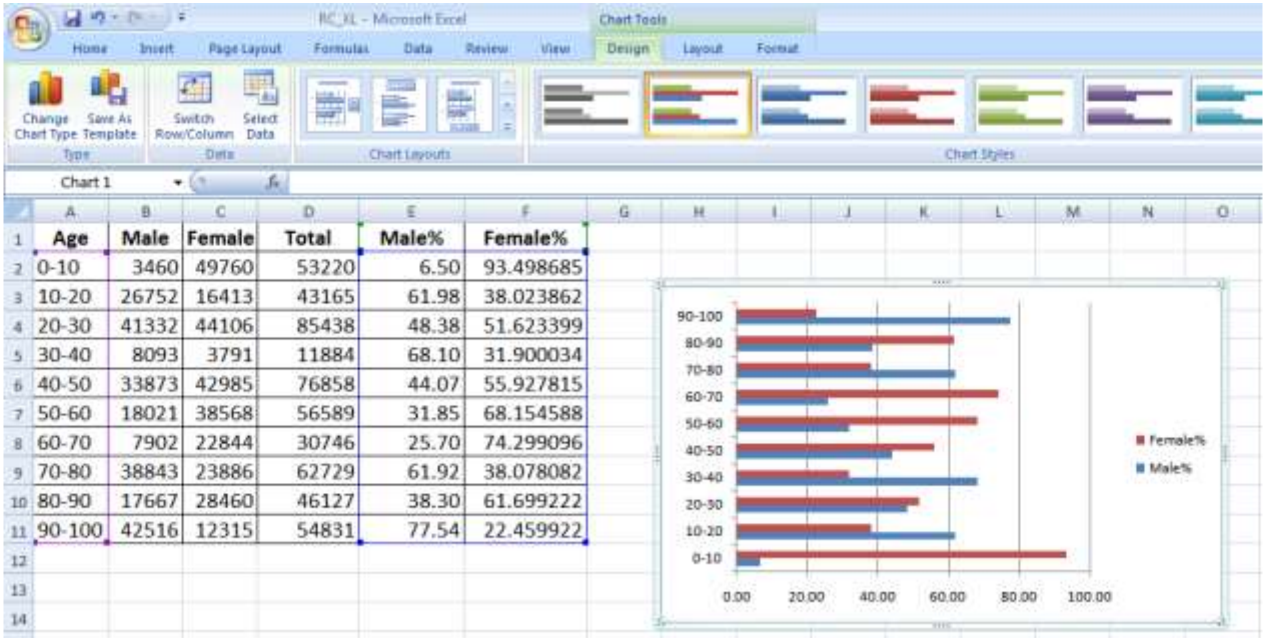
**Select the column to be displayed in the chart including the column name. In our example it is Age, Male% and Female%. Select the age column and press ctrl key. Keep the ctrl key pressed and select the male% and female% column.**




| Age | Male | Female | Total | Male% | Female% |
|---|---|---|---|---|---|
| 0-10 | 3460 | 49760 | 53220 | 6.50 | 93.498685 |
| 10-20 | 26752 | 16413 | 43165 | 61.98 | 38.023862 |
| 20-30 | 41332 | 44106 | 85438 | 48.38 | 51.623399 |
| 30-40 | 8093 | 3791 | 11884 | 68.10 | 31.900034 |
| 40-50 | 33873 | 42985 | 76858 | 44.07 | 55.927815 |
| 50-60 | 18021 | 38568 | 56589 | 31.85 | 68.154588 |
| 60-70 | 7902 | 22844 | 30746 | 25.70 | 74.299096 |
| 70-80 | 38843 | 23886 | 62729 | 61.92 | 38.078082 |
| 80-90 | 17667 | 28460 | 46127 | 38.30 | 61.699222 |
| 90-100 | 42516 | 12315 | 54831 | 77.54 | 22.459922 |

**Go to Insert and select the 2-D Clustered Bar Chart.**

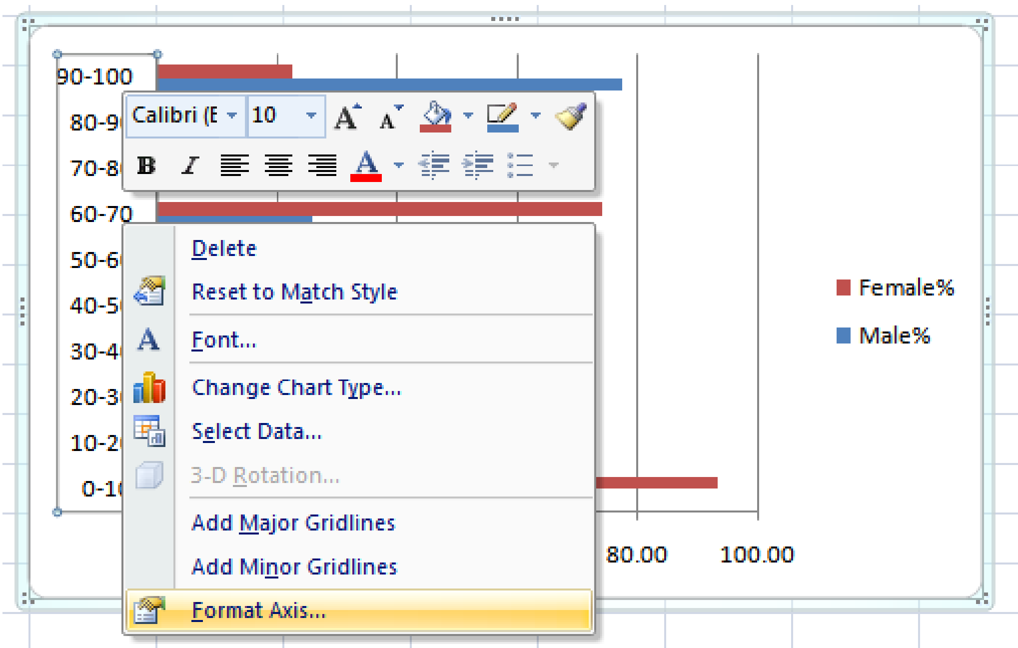




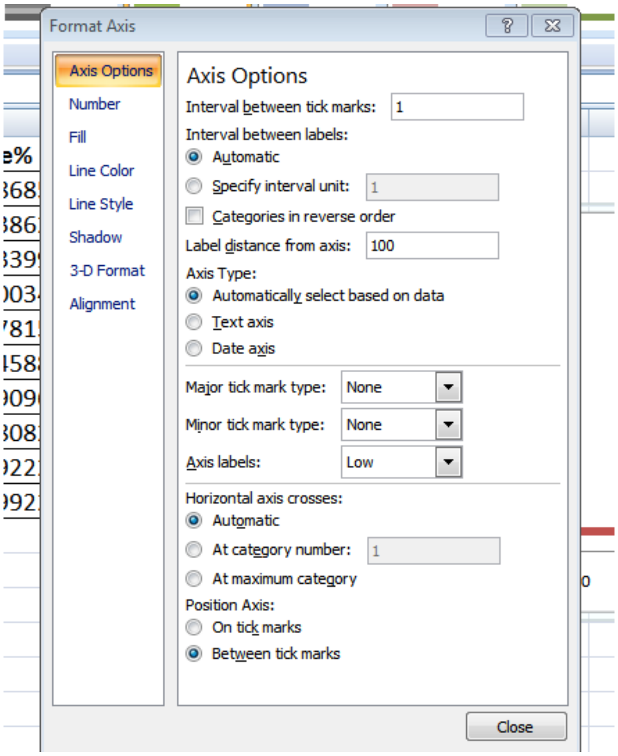
| Age | Male | Female | Total | Male% | Female% |
|---|---|---|---|---|---|
| 0-10 | 3460 | 49760 | 53220 | 6.50 | 93.498685 |
| 10-20 | 26752 | 16413 | 43165 | 61.98 | 38.023862 |
| 20-30 | 41332 | 44106 | 85438 | 48.38 | 51.623399 |
| 30-40 | 8093 | 3791 | 11884 | 68.10 | 31.900034 |
| 40-50 | 33873 | 42985 | 76858 | 44.07 | 55.927815 |
| 50-60 | 18021 | 38568 | 56589 | 31.85 | 68.154588 |
| 60-70 | 7902 | 22844 | 30746 | 25.70 | 74.299096 |
| 70-80 | 38843 | 23886 | 62729 | 61.92 | 38.078082 |
| 80-90 | 17667 | 28460 | 46127 | 38.30 | 61.699222 |
| 90-100 | 42516 | 12315 | 54831 | 77.54 | 22.459922 |



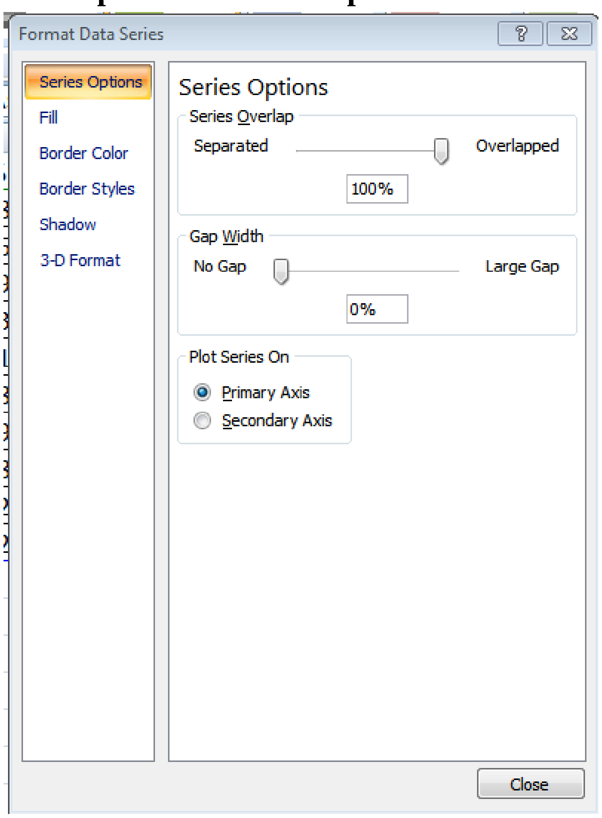
**Click on the Vertical Axis and Select Format Axis.**

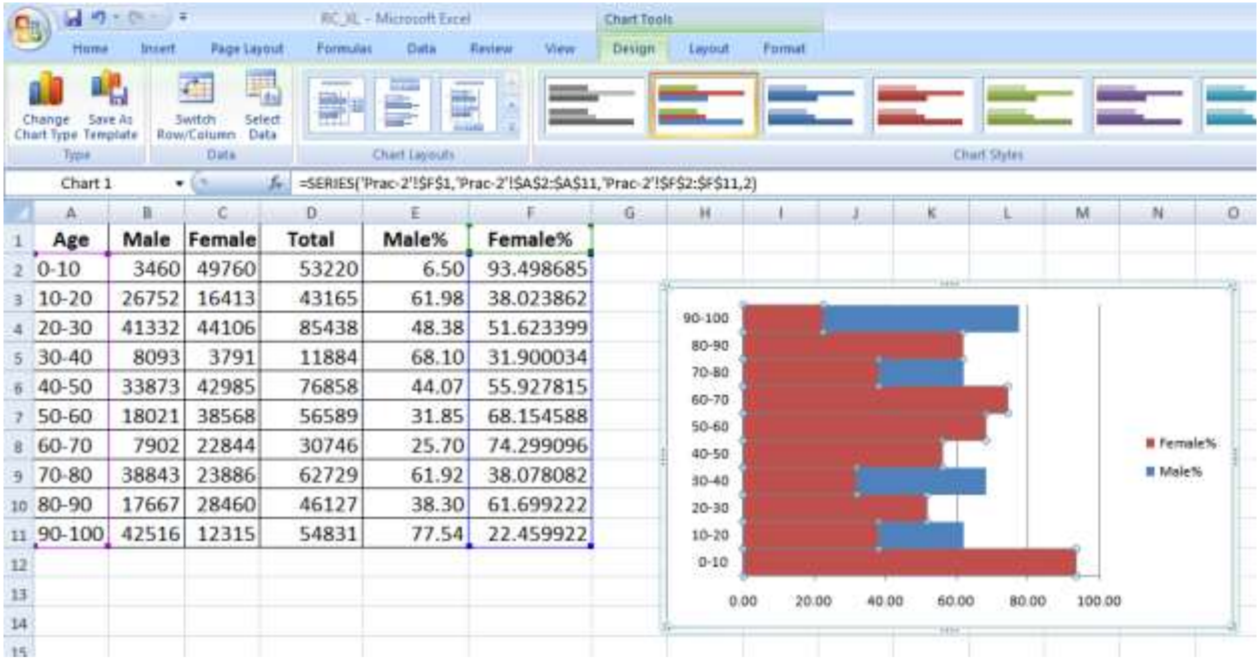


**Set Major Tick mark type and Minor tick mark type as None. Set Axis Labels as Low. Click on Close.**



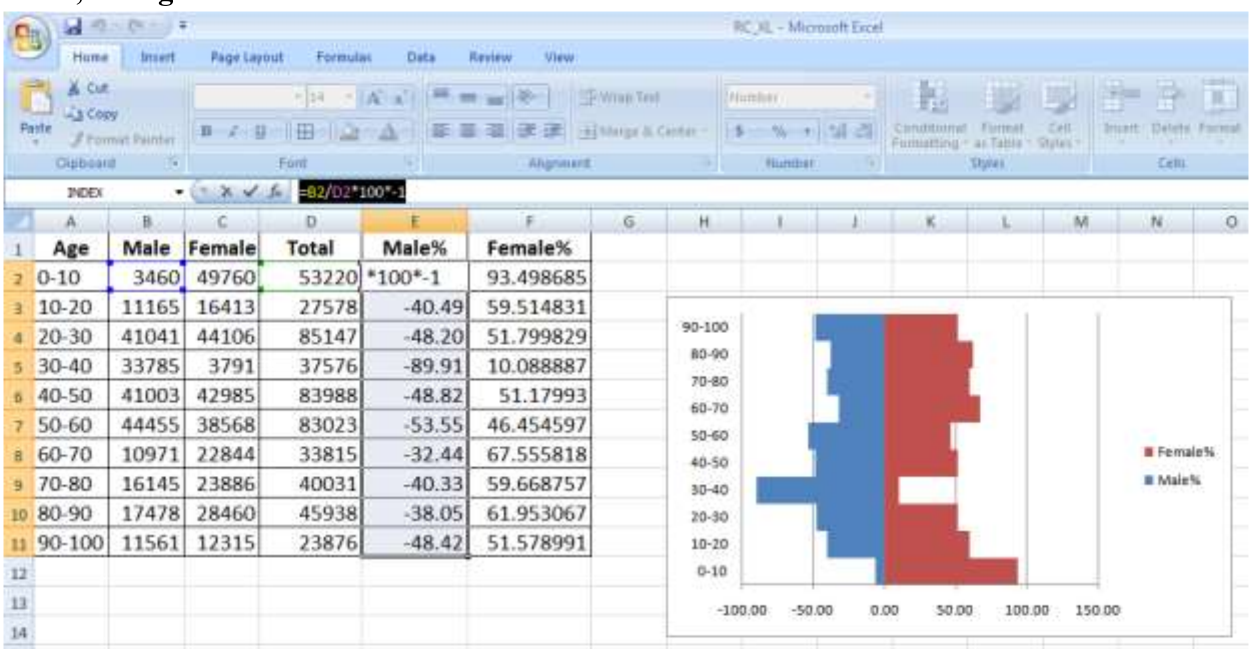

**Right click on any data bar in the chart and click on Format Data Series. Set Series Overlap to 100% and Gap width to 0%. Click on Close.**






| Age | Male | Female | Total | Male% | Female% |
|---|---|---|---|---|---|
| 0-10 | 3460 | 49760 | 53220 | 6.50 | 93.498685 |
| 10-20 | 26752 | 16413 | 43165 | 61.98 | 38.023862 |
| 20-30 | 41332 | 44106 | 85438 | 48.38 | 51.623399 |
| 30-40 | 8093 | 3791 | 11884 | 68.10 | 31.900034 |
| 40-50 | 33873 | 42985 | 76858 | 44.07 | 55.927815 |
| 50-60 | 18021 | 38568 | 56589 | 31.85 | 68.154588 |
| 60-70 | 7902 | 22844 | 30746 | 25.70 | 74.299096 |
| 70-80 | 38843 | 23886 | 62729 | 61.92 | 38.078082 |
| 80-90 | 17667 | 28460 | 46127 | 38.30 | 61.699222 |
| 90-100 | 42516 | 12315 | 54831 | 77.54 | 22.459922 |

**Next , change the formula in male% to "=B2/D2*100*-1". i.e. add *-1 to the formula.**




| Age | Male | Female | Total | Male% | Female% |
|---|---|---|---|---|---|
| 0-10 | 3460 | 49760 | 53220 | *100*-1 | 93.498685 |
| 10-20 | 11165 | 16413 | 27578 | -40.49 | 59.514831 |
| 20-30 | 41041 | 44106 | 85147 | -48.20 | 51.799829 |
| 30-40 | 33785 | 3791 | 37576 | -89.91 | 10.088887 |
| 40-50 | 41003 | 42985 | 83988 | -48.82 | 51.17993 |
| 50-60 | 44455 | 38568 | 83023 | -53.55 | 46.454597 |
| 60-70 | 10971 | 22844 | 33815 | -32.44 | 67.555818 |
| 70-80 | 16145 | 23886 | 40031 | -40.33 | 59.668757 |
| 80-90 | 17478 | 28460 | 45938 | -38.05 | 61.953067 |
| 90-100 | 11561 | 12315 | 23876 | -48.42 | 51.578991 |

# Practical No 3

## A. Perform testing of hypothesis using one sample t-test.

**Code:**

```
# -*- coding: utf-8 -*-
"""
Created on Mon Dec 16 18:01:46 2019
@author: Ahtesham Shaikh
"""
fromscipy.stats import ttest_1samp
importnumpy as np
ages = np.genfromtxt('ages.csv')
print(ages)
ages_mean = np.mean(ages)
print(ages_mean)
tset, pval = ttest_1samp(ages, 30)
print('p-values - ',pval)
if pval< 0.05: # alpha value is 0.05
print(" we are rejecting null hypothesis")
else:
print("we are accepting null hypothesis")
```

**Output:**

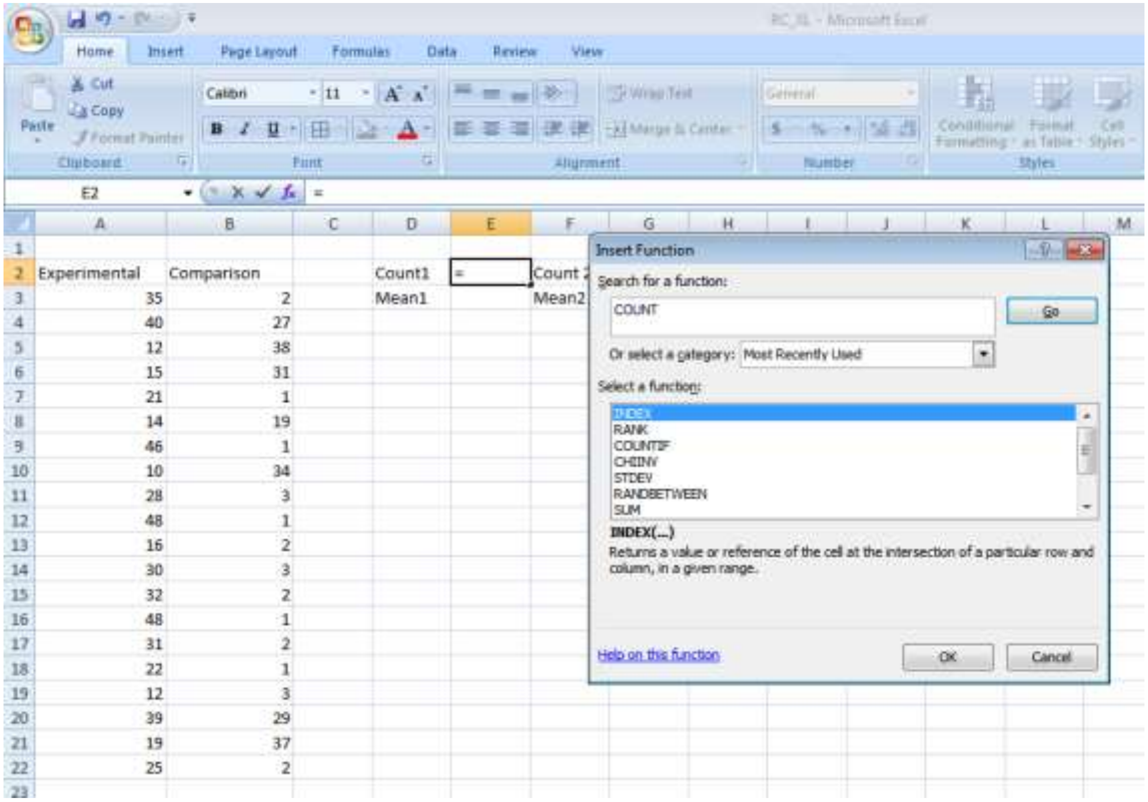
```
In [4]: runfile('K:/Research In Computing/Practical Material/Programs/
Practical_05/Prac_3A.py', wdir='K:/Research In Computing/Practical Material/
Programs/Practical_05')
[20. 30. 25. 13. 16. 17. 34. 35. 38. 42. 43. 45. 48. 49. 50. 51. 54. 55.
 56. 59. 61. 62. 18. 22. 29. 30. 31. 39. 52. 53. 67. 36. 47. 54. 40. 40.
 35. 22. 59. 58. 30. 43. 22. 45. 21. 59. 51. 47. 25. 58. 50. 23. 24. 45.
 37. 59. 28. 28. 48. 42. 54. 36. 36. 24. 26. 24. 50. 48. 34. 44. 56. 55.
 35. 33. 39. 53. 34. 28. 56. 24. 21. 29. 28. 58. 35. 57. 26. 25. 59. 56.
 22. 57. 48. 33. 23. 26. 57. 32. 53. 31. 35. 44. 54. 25. 31. 58. 26. 32.
 26. 50. 41. 49. 26. 33. 34. 24. 43. 42. 51. 36. 38. 38. 40. 38. 56. 39.
 23. 33. 53. 30. 38.]
39.47328244274809
p-values -  5.362905195437013e-14
 we are rejecting null hypothesis
```

## B. Write a program for t-test comparing two means for independent samples.

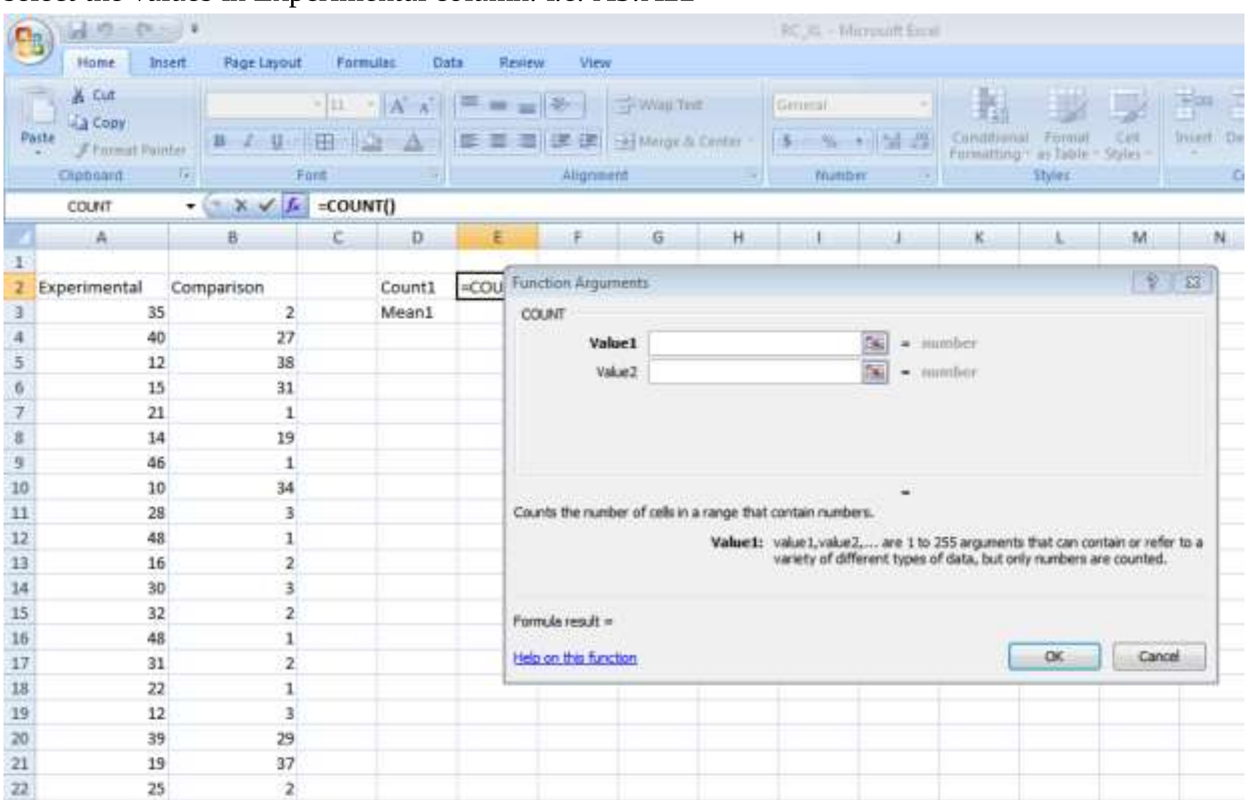
A college Principal informed classroom teachers that some of their students showed unusual potential for intellectual gains. One month later the students identified by the teachers as having potentional for unusual intellectual gains showed significiantly greater gains performance on a test said to measure IQ than did students who were not identified. Below are the data for the students:

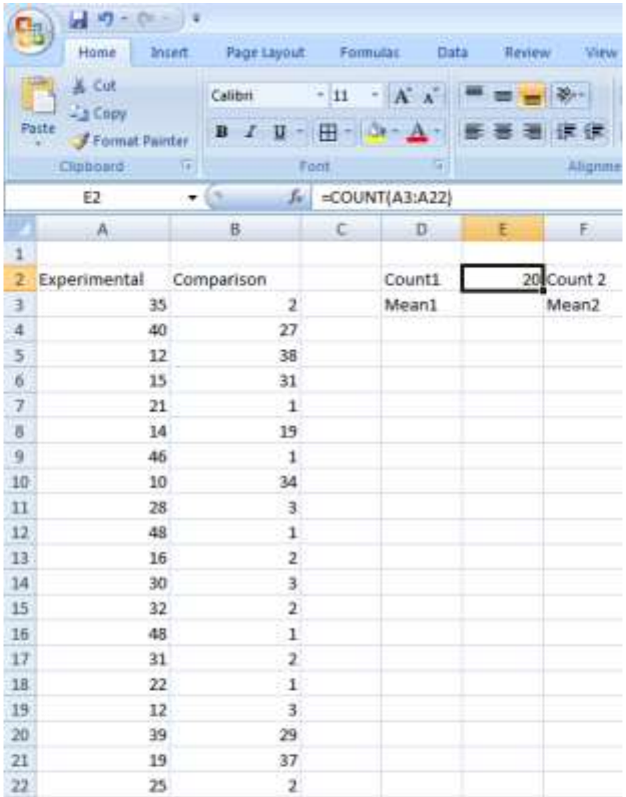
| Experimental | Comparison |
|---|---|
| 35 | 2 |
| 40 | 27 |
| 12 | 38 |
| 15 | 31 |
| 21 | 1 |
| 14 | 19 |
| 46 | 1 |
| 10 | 34 |
| 28 | 3 |
| 48 | 1 |
| 16 | 2 |
| 30 | 3 |
| 32 | 2 |
| 48 | 1 |
| 31 | 2 |
| 22 | 1 |
| 12 | 3 |
| 39 | 29 |
| 19 | 37 |
| 25 | 2 |

Solution:
H0: Difference in score is not likely the result of experimental treatment
H1: Difference in score is likely the result of experimental treatment
We start by counting the size of the sample using count function in Excel.
Click on fx and Type COUNT in the search box. Select the Count function from the list of diaplayed results.



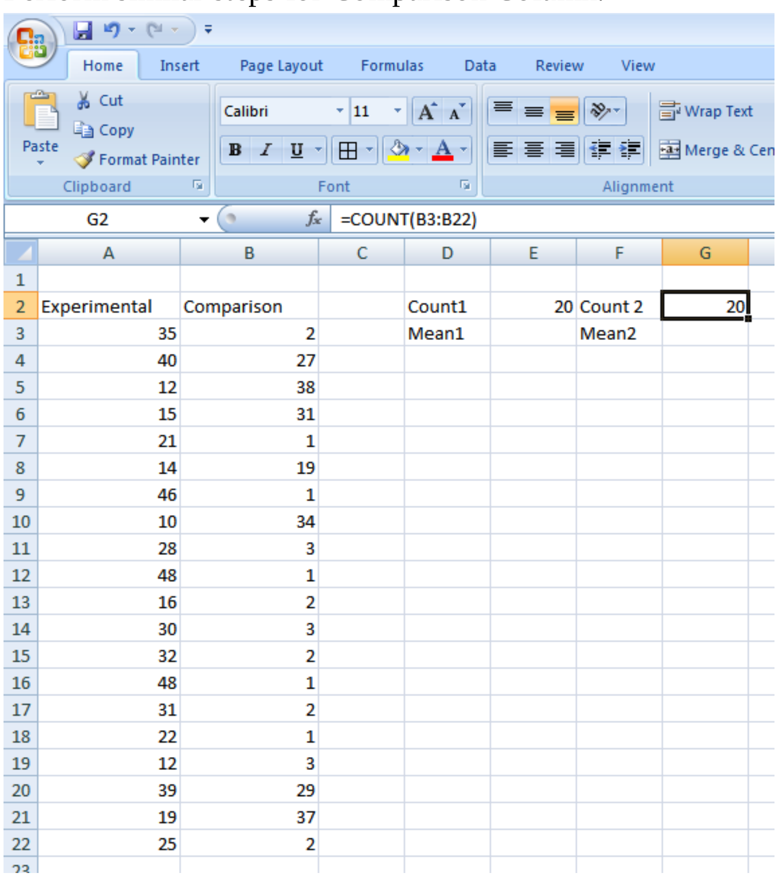
Select the range of cells to be counted by selecting the icon besides value1 and Click OK. We select the values in Experimental column. i.e. A3:A22





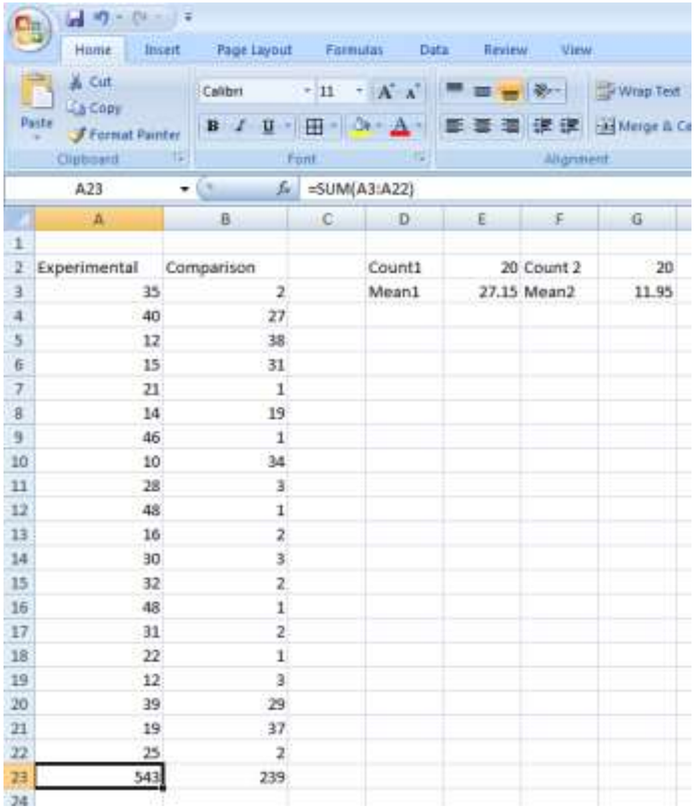

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | Experimental | Comparison | | Count1 | 20 | Count 2 |
| 3 | 35 | 2 | | Mean1 | | Mean2 |
| 4 | 40 | 27 | | | | |
| 5 | 12 | 38 | | | | |
| 6 | 15 | 31 | | | | |
| 7 | 21 | 1 | | | | |
| 8 | 14 | 19 | | | | |
| 9 | 46 | 1 | | | | |
| 10 | 10 | 34 | | | | |
| 11 | 28 | 3 | | | | |
| 12 | 48 | 1 | | | | |
| 13 | 16 | 2 | | | | |
| 14 | 30 | 3 | | | | |
| 15 | 32 | 2 | | | | |
| 16 | 48 | 1 | | | | |
| 17 | 31 | 2 | | | | |
| 18 | 22 | 1 | | | | |
| 19 | 12 | 3 | | | | |
| 20 | 39 | 29 | | | | |
| 21 | 19 | 37 | | | | |
| 22 | 25 | 2 | | | | |

Perform similar steps for Comparison Column.



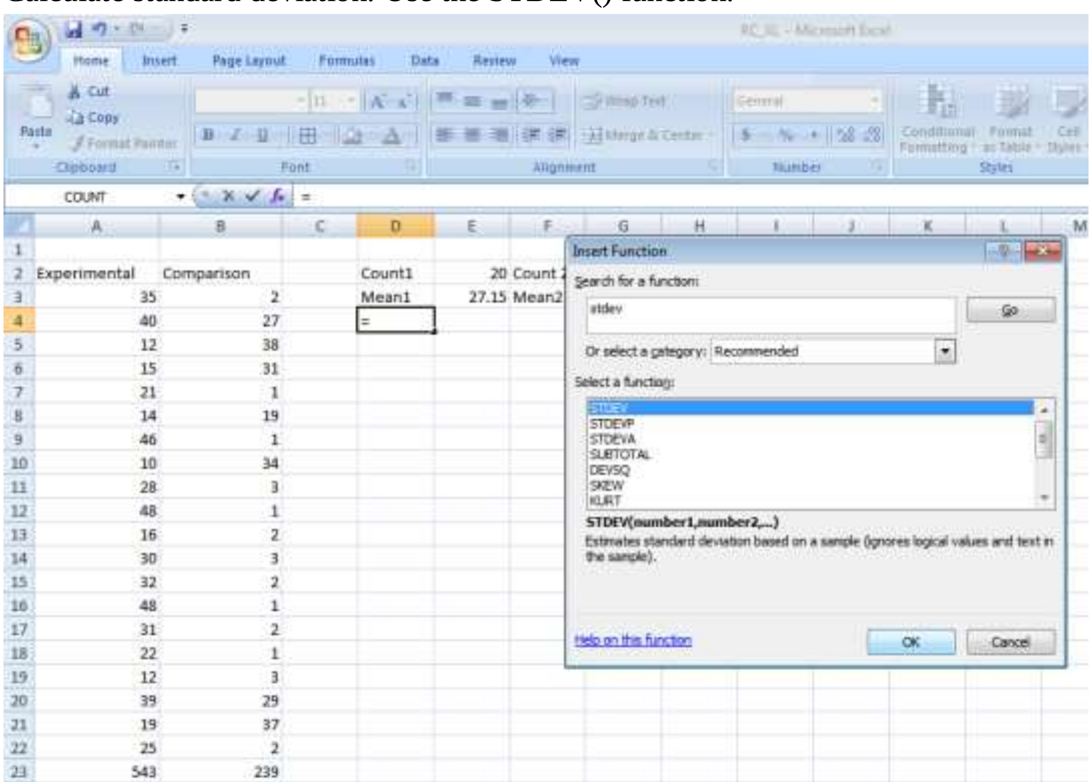

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | | | | | | | |
| 2 | Experimental | Comparison | | Count1 | 20 | Count 2 | 20 |
| 3 | 35 | 2 | | Mean1 | | Mean2 | |
| 4 | 40 | 27 | | | | | |
| 5 | 12 | 38 | | | | | |
| 6 | 15 | 31 | | | | | |
| 7 | 21 | 1 | | | | | |
| 8 | 14 | 19 | | | | | |
| 9 | 46 | 1 | | | | | |
| 10 | 10 | 34 | | | | | |
| 11 | 28 | 3 | | | | | |
| 12 | 48 | 1 | | | | | |
| 13 | 16 | 2 | | | | | |
| 14 | 30 | 3 | | | | | |
| 15 | 32 | 2 | | | | | |
| 16 | 48 | 1 | | | | | |
| 17 | 31 | 2 | | | | | |
| 18 | 22 | 1 | | | | | |
| 19 | 12 | 3 | | | | | |
| 20 | 39 | 29 | | | | | |
| 21 | 19 | 37 | | | | | |
| 22 | 25 | 2 | | | | | |

To calculate Mean Add all the values in a column(A3:A22) in A23 and divide by Count(E2).Similar steps for Comparison column.



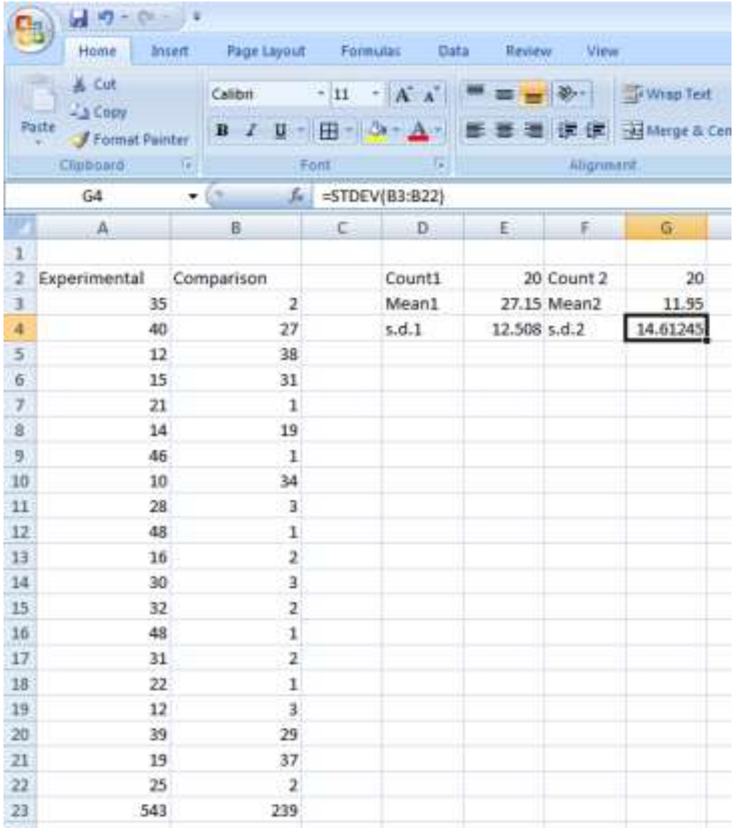

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | | | | | | | |
| 2 | Experimental | Comparison | | Count1 | 20 | Count 2 | 20 |
| 3 | 35 | 2 | | Mean1 | 27.15 | Mean2 | 11.95 |
| 4 | 40 | 27 | | | | | |
| 5 | 12 | 38 | | | | | |
| 6 | 15 | 31 | | | | | |
| 7 | 21 | 1 | | | | | |
| 8 | 14 | 19 | | | | | |
| 9 | 46 | 1 | | | | | |
| 10 | 10 | 34 | | | | | |
| 11 | 28 | 3 | | | | | |
| 12 | 48 | 1 | | | | | |
| 13 | 16 | 2 | | | | | |
| 14 | 30 | 3 | | | | | |
| 15 | 32 | 2 | | | | | |
| 16 | 48 | 1 | | | | | |
| 17 | 31 | 2 | | | | | |
| 18 | 22 | 1 | | | | | |
| 19 | 12 | 3 | | | | | |
| 20 | 39 | 29 | | | | | |
| 21 | 19 | 37 | | | | | |
| 22 | 25 | 2 | | | | | |
| 23 | 543 | 239 | | | | | |

Calculate standard deviation. Use the STDEV() function.

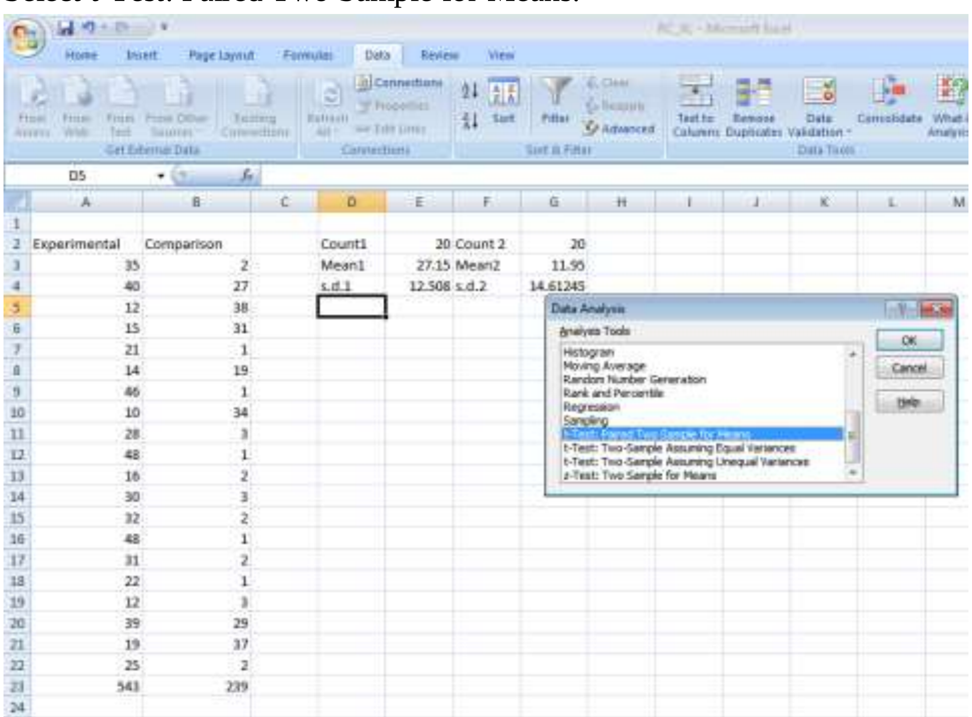



| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | | | | | |
| 2 | Experimental | Comparison | | Count1 | 20 | Count 2 |
| 3 | 35 | 2 | | Mean1 | 27.15 | Mean2 |
| 4 | 40 | 27 | | = | | |
| 5 | 12 | 38 | | | | |
| 6 | 15 | 31 | | | | |
| 7 | 21 | 1 | | | | |
| 8 | 14 | 19 | | | | |
| 9 | 46 | 1 | | | | |
| 10 | 10 | 34 | | | | |
| 11 | 28 | 3 | | | | |
| 12 | 48 | 1 | | | | |
| 13 | 16 | 2 | | | | |
| 14 | 30 | 3 | | | | |
| 15 | 32 | 2 | | | | |
| 16 | 48 | 1 | | | | |
| 17 | 31 | 2 | | | | |
| 18 | 22 | 1 | | | | |
| 19 | 12 | 3 | | | | |
| 20 | 39 | 29 | | | | |
| 21 | 19 | 37 | | | | |
| 22 | 25 | 2 | | | | |
| 23 | 543 | 239 | | | | |

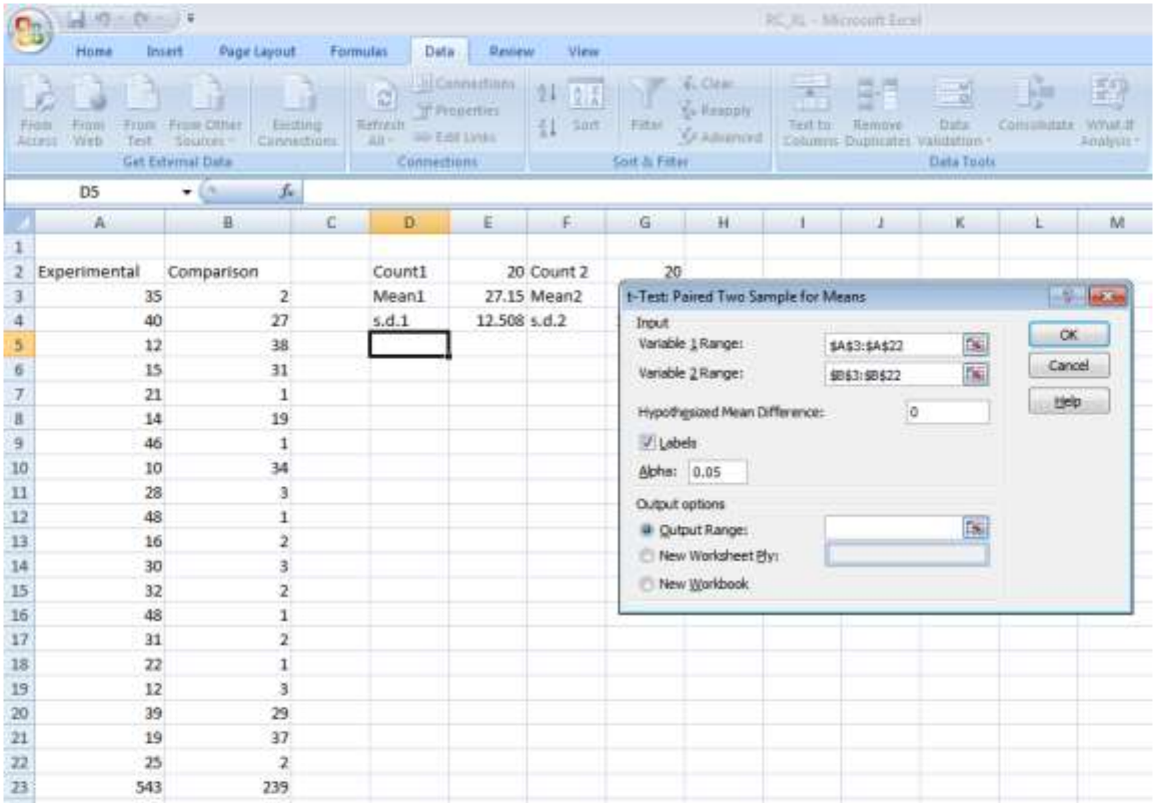
Select the values(A3:A22) for which standard deviation needs to be calculated.Click OK. Similar for 2nd column.



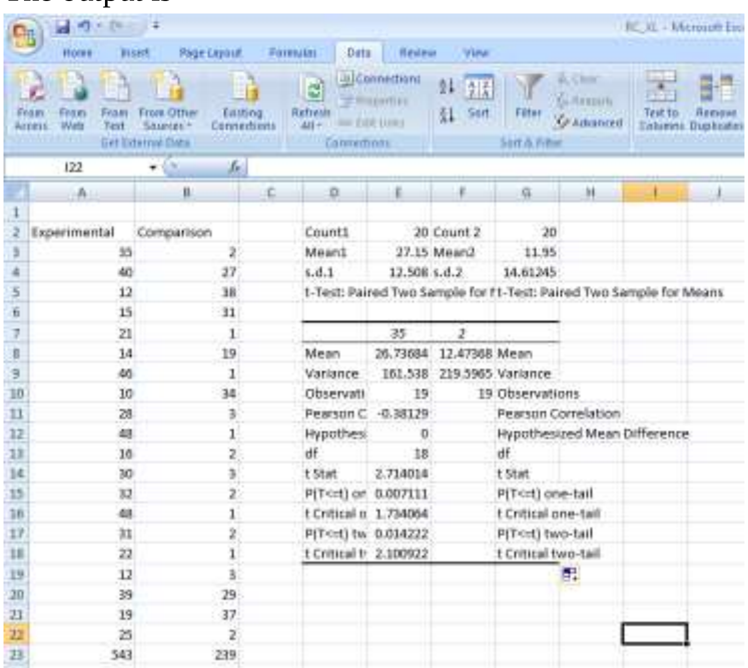

To find t-test statistics, Click on the Data Analysis tool under the Data Lab in the MenuBar. Select t-Test: Paired Two Sample for Means.



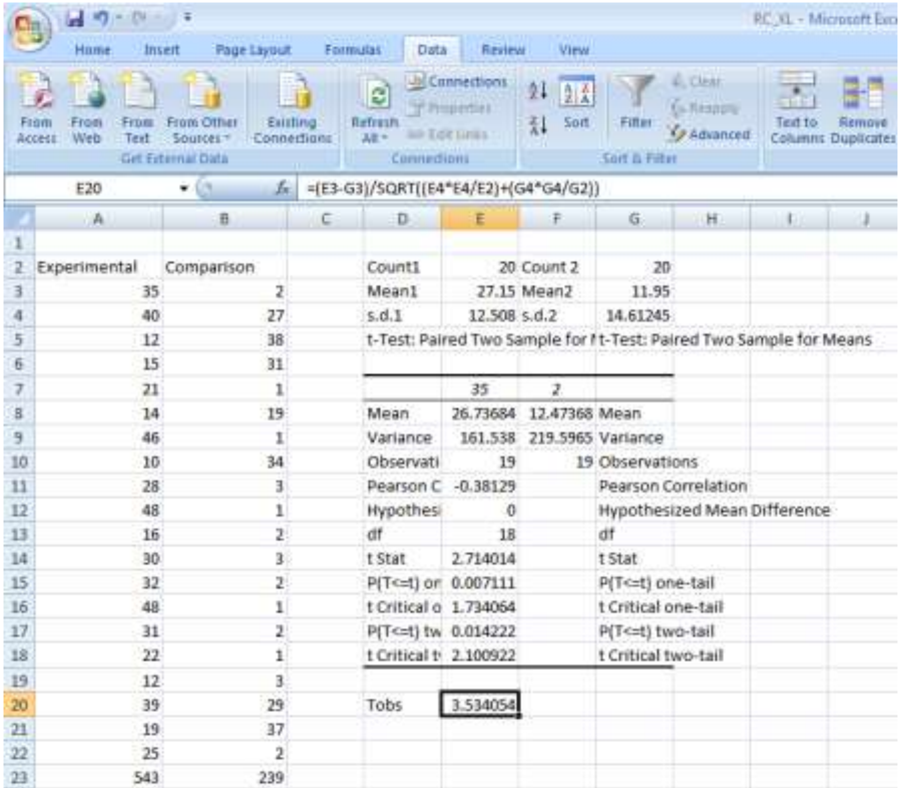
In the variable1 Range select 1st column(A3:A22) . In the variable Range select 2nd column(B3:B22) . In Hypothesized Mean Difference Type 0. Check the Labels Checkbox. And finally select the output range for displaying the output.



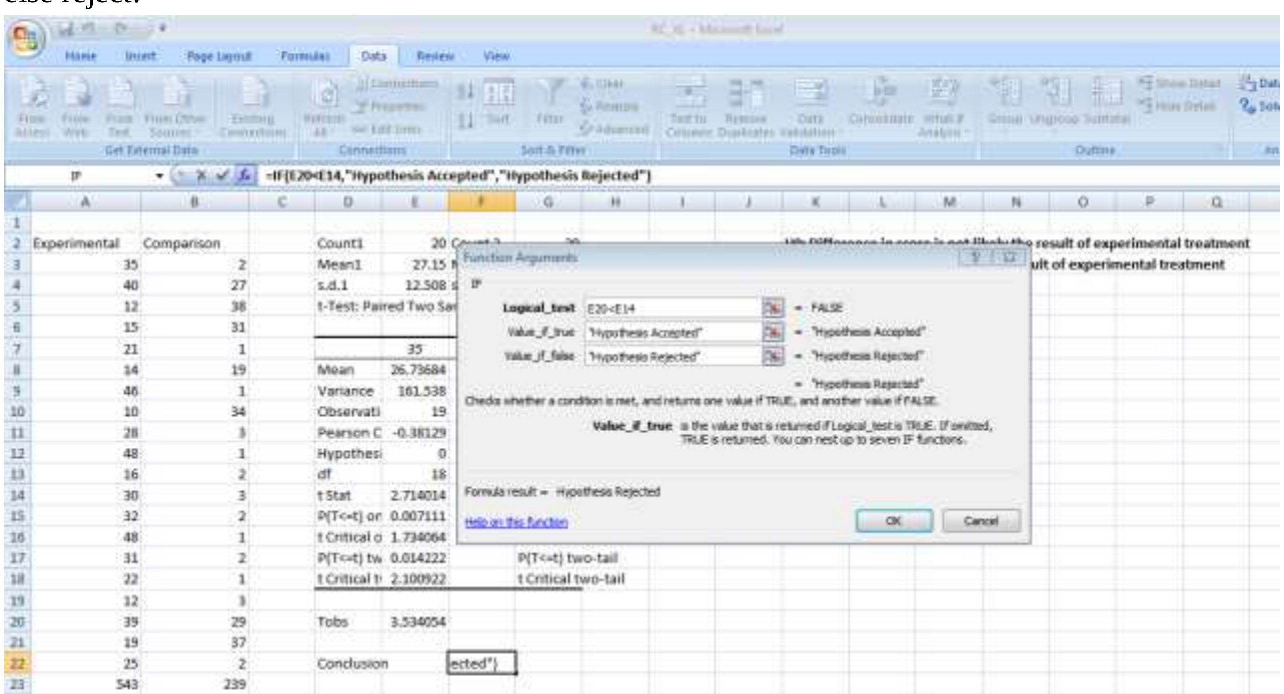

The output is

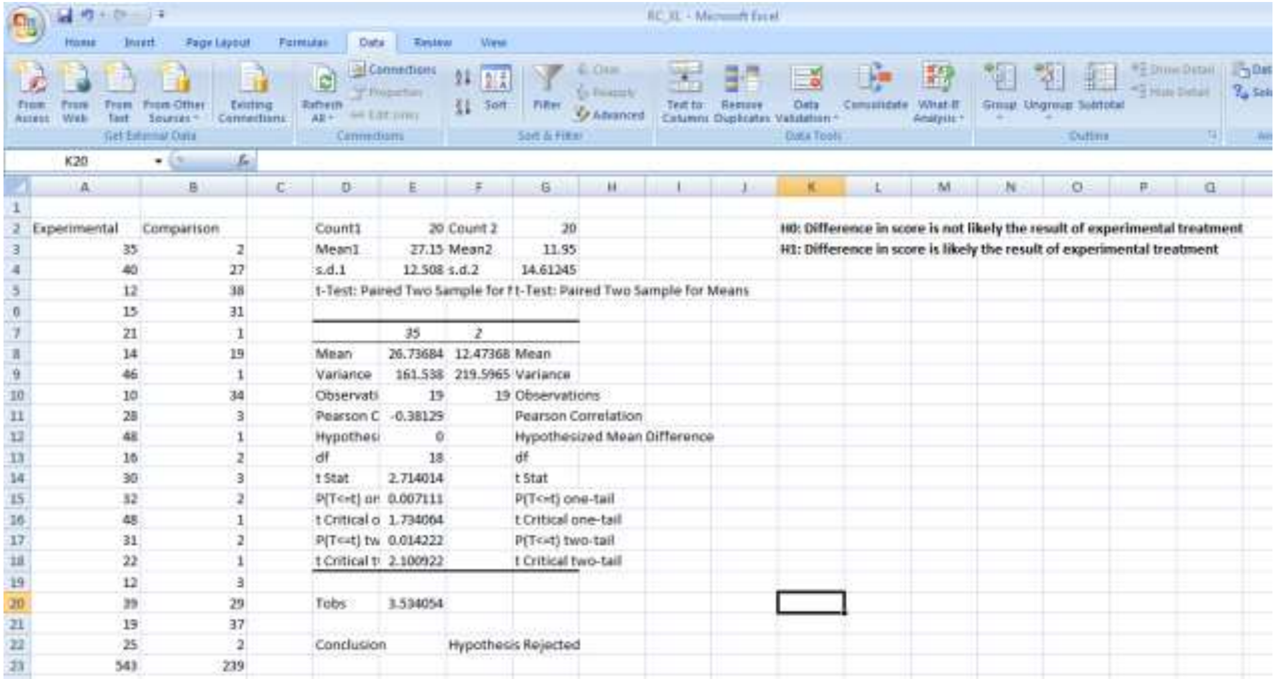


Next, we calculate the t-test value for 2 sample means as $\frac{x1-x2}{\frac{\sqrt{s1^2}}{n1}+\frac{\sqrt{s2^2}}{n2}}$ where x1 and x2 are mean. S1and s2 are standard deviation and n1 and n2 is count or population size.




To draw conclusion about hypothesis we use the if condition. If tobs(E20)<tstat(E14), accept else reject.





| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | | | | | | | |
| 2 | Experimental | Comparison | | Count1 | 20 | Count 2 | 20 |
| 3 | 35 | 2 | | Mean1 | 27.15 | Mean2 | 11.95 |
| 4 | 40 | 27 | | s.d.1 | 12.508 | s.d.2 | 14.61245 |
| 5 | 12 | 38 | | t-Test: Paired Two Sample for I | | t-Test: Paired Two Sample for Means | |
| 6 | 15 | 31 | | | | | |
| 7 | 21 | 1 | | | 35 | 2 | |
| 8 | 14 | 19 | | Mean | 26.73684 | 12.47368 | Mean |
| 9 | 46 | 1 | | Variance | 161.538 | 219.5965 | Variance |
| 10 | 10 | 34 | | Observati | 19 | 19 | Observations |
| 11 | 28 | 3 | | Pearson C | -0.38129 | | Pearson Correlation |
| 12 | 48 | 1 | | Hypothesi | 0 | | Hypothesized Mean Difference |
| 13 | 16 | 2 | | df | 18 | | df |
| 14 | 30 | 3 | | t Stat | 2.714014 | | t Stat |
| 15 | 32 | 2 | | P(T<=t) or | 0.007111 | | P(T<=t) one-tail |
| 16 | 48 | 1 | | t Critical o | 1.734064 | | t Critical one-tail |
| 17 | 31 | 2 | | P(T<=t) tw | 0.014222 | | P(T<=t) two-tail |
| 18 | 22 | 1 | | t Critical t | 2.100922 | | t Critical two-tail |
| 19 | 12 | 3 | | | | | |
| 20 | 39 | 29 | | Tobs | 3.534054 | | |
| 21 | 19 | 37 | | | | | |
| 22 | 25 | 2 | | Conclusion | | Hypothesis Rejected | |
| 23 | 543 | 239 | | | | | |

H0: Difference in score is not likely the result of experimental treatment

H1: Difference in score is likely the result of experimental treatment

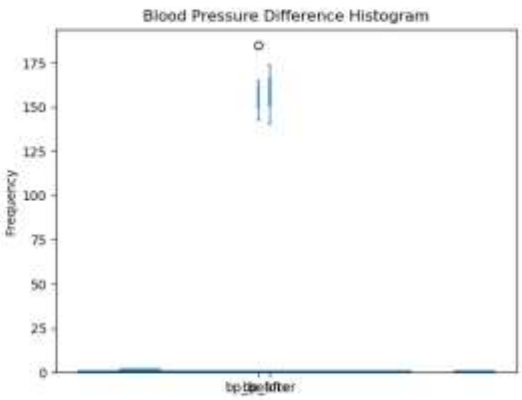
## C.

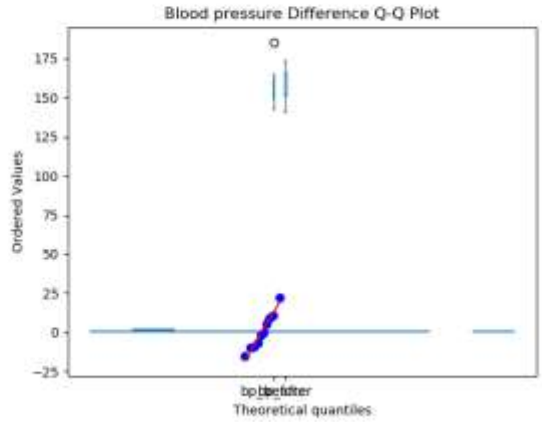
## D. Perform testing of hypothesis using paired t-test

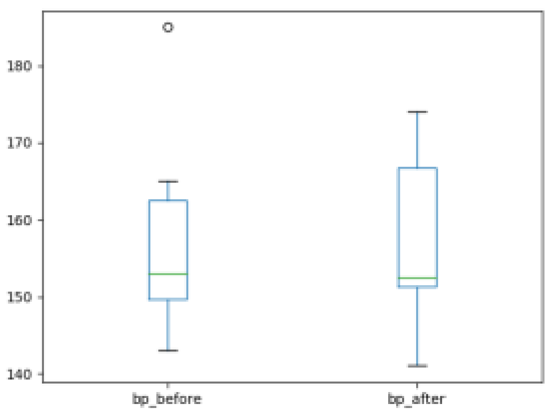
**Code:**

```
# -*- coding: utf-8 -*-
"""
Created on Mon Dec 16 19:49:23 2019
@author: MyHome
"""
from scipy import stats
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv("blood_pressure.csv")
print(df[['bp_before','bp_after']].describe())
#First let's check for any significant outliers in
#each of the variables.
df[['bp_before', 'bp_after']].plot(kind='box')
# This saves the plot as a png file
plt.savefig('boxplot_outliers.png')
# make a histogram to differences between the two scores.
df['bp_difference'] = df['bp_before'] - df['bp_after']
df['bp_difference'].plot(kind='hist', title= 'Blood Pressure Difference Histogram')
#Again, this saves the plot as a png file
plt.savefig('blood pressure difference histogram.png')
stats.probplot(df['bp_difference'], plot= plt)
plt.title('Blood pressure Difference Q-Q Plot')
plt.savefig('blood pressure difference qq plot.png')
stats.shapiro(df['bp_difference'])
stats.ttest_rel(df['bp_before'], df['bp_after'])
```

**Output:**



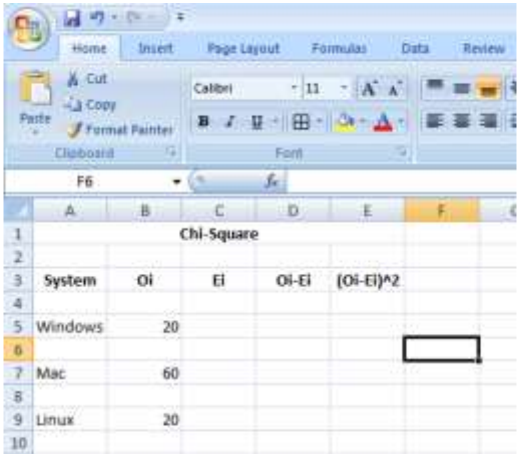




```
       bp_before   bp_after
count  10.000000  10.000000
mean  157.000000 156.600000
std    12.192894  11.529672
min   143.000000 141.000000
25%   149.750000 151.250000
50%   153.000000 152.500000
75%   162.500000 166.750000
max   185.000000 174.000000
>>>
```

# Practical No 4

## A. Perform testing of hypothesis using chi-squared goodness -of -fit test.

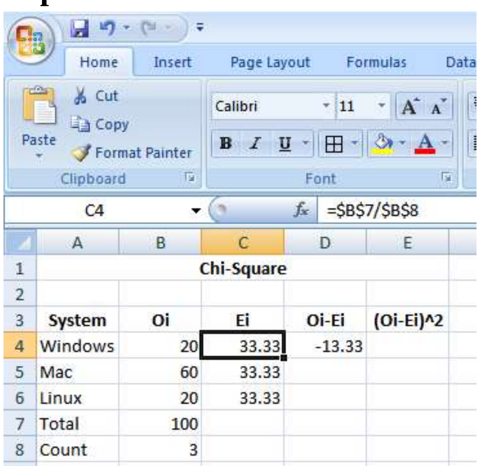
A system administrator needs to upgrade the computers for his division. He wants to know what sort of computer system h is workers prefer. He gives three choices: Windows, Mac, or Linux. Test the hypothesis or theory that an equal percentage of the population prefers each type of computer system.



F6

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | | Chi-Square | | | |
| 2 | | | | | | |
| 3 | System | Oi | Ei | Oi-Ei | (Oi-Ei)^2 | |
| 4 | | | | | | |
| 5 | Windows | 20 | | | | |
| 6 | | | | | | |
| 7 | Mac | 60 | | | | |
| 8 | | | | | | |
| 9 | Linux | 20 | | | | |
| 10 | | | | | | |

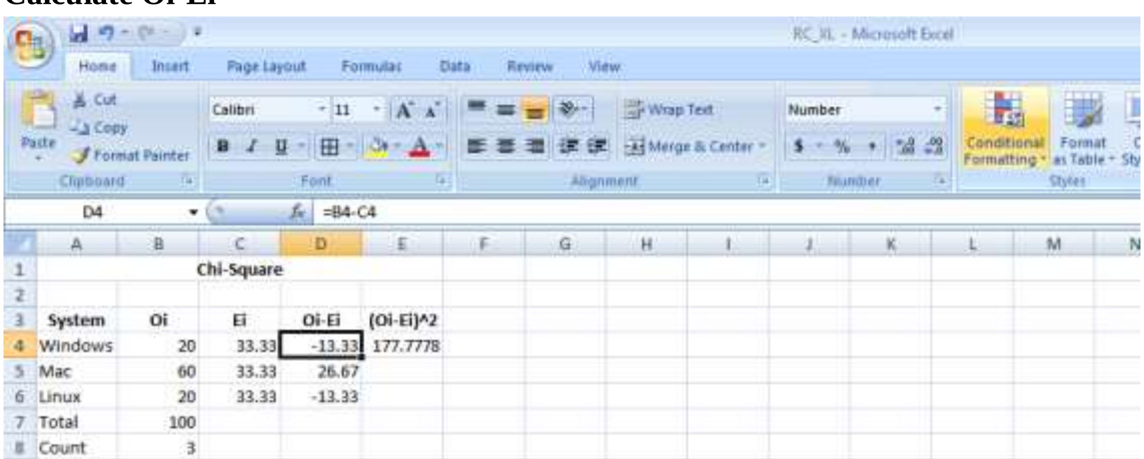
**Expected Value Ei is the total number of values(Sum)/type of values(count)**



C4 =$B$7/$B$8

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | Chi-Square | | |
| 2 | | | | | |
| 3 | System | Oi | Ei | Oi-Ei | (Oi-Ei)^2 |
| 4 | Windows | 20 | 33.33 | -13.33 | |
| 5 | Mac | 60 | 33.33 | | |
| 6 | Linux | 20 | 33.33 | | |
| 7 | Total | 100 | | | |
| 8 | Count | 3 | | | |

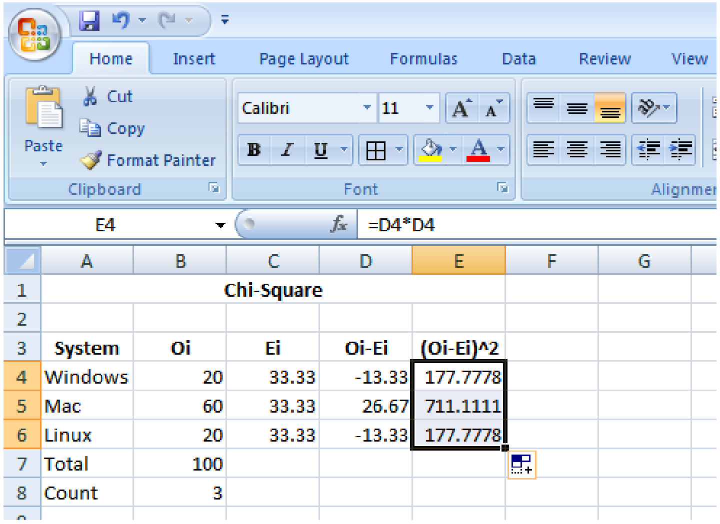
**Calculate Oi-Ei**



D4 =B4-C4

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | Chi-Square | | |
| 2 | | | | | |
| 3 | System | Oi | Ei | Oi-Ei | (Oi-Ei)^2 |
| 4 | Windows | 20 | 33.33 | -13.33 | 177.7778 |
| 5 | Mac | 60 | 33.33 | 26.67 | |
| 6 | Linux | 20 | 33.33 | -13.33 | |
| 7 | Total | 100 | | | |
| 8 | Count | 3 | | | |

## Calculate $(Oi-Ei)^2$



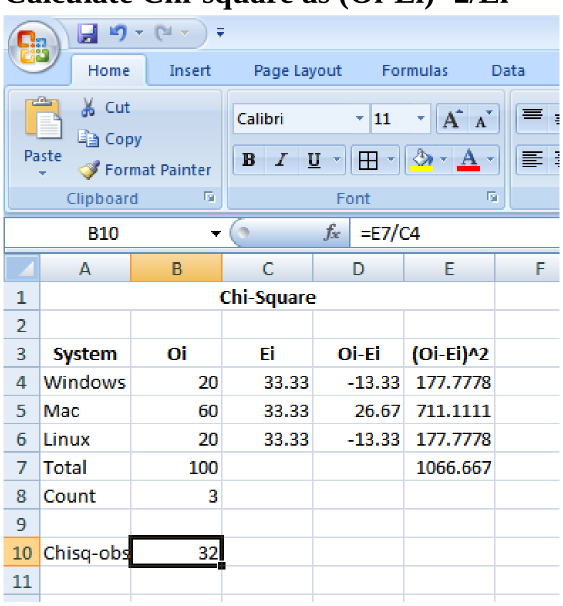

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | Chi-Square | | |
| 2 | | | | | |
| 3 | System | Oi | Ei | Oi-Ei | (Oi-Ei)^2 |
| 4 | Windows | 20 | 33.33 | -13.33 | 177.7778 |
| 5 | Mac | 60 | 33.33 | 26.67 | 711.1111 |
| 6 | Linux | 20 | 33.33 | -13.33 | 177.7778 |
| 7 | Total | 100 | | | |
| 8 | Count | 3 | | | |

## Calculate sum of $(Oi-Ei)^2$

E

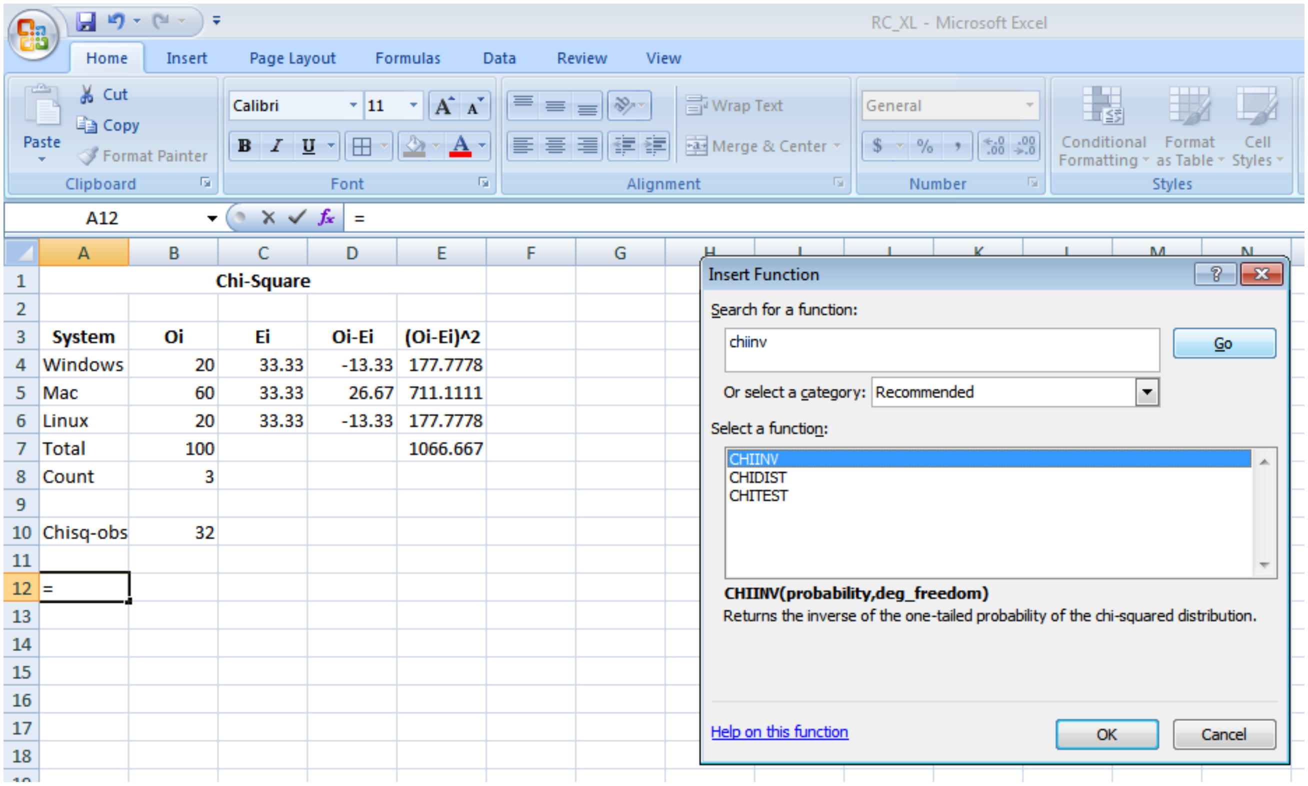

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | Chi-Square | | |
| 2 | | | | | |
| 3 | System | Oi | Ei | Oi-Ei | (Oi-Ei)^2 |
| 4 | Windows | 20 | 33.33 | -13.33 | 177.7778 |
| 5 | Mac | 60 | 33.33 | 26.67 | 711.1111 |
| 6 | Linux | 20 | 33.33 | -13.33 | 177.7778 |
| 7 | Total | 100 | | | 1066.667 |
| 8 | Count | 3 | | | |

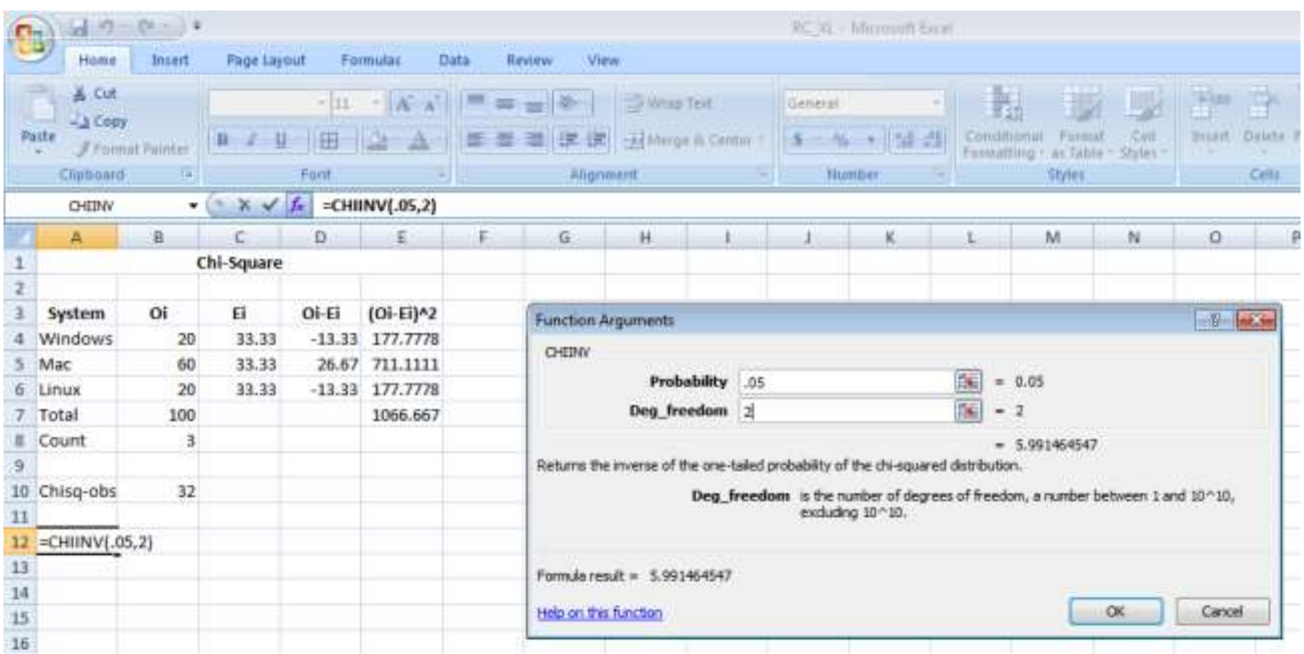
## Calculate Chi-square as (Oi-Ei)^2/Ei



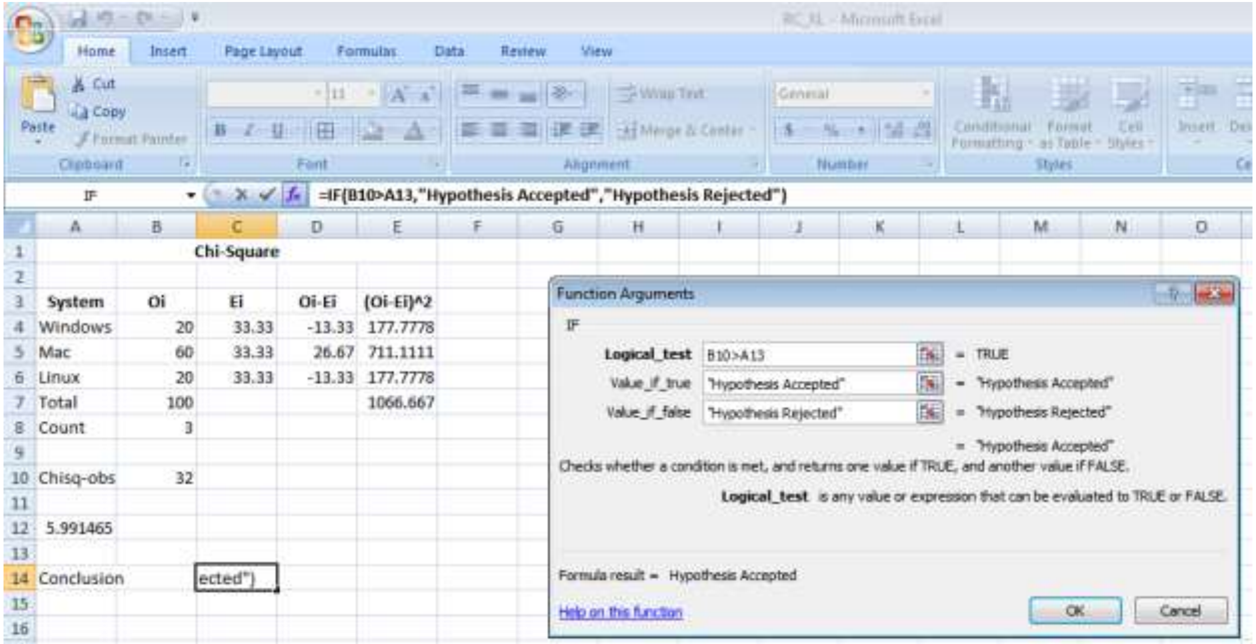

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | | | Chi-Square | | |
| 2 | | | | | |
| 3 | System | Oi | Ei | Oi-Ei | (Oi-Ei)^2 |
| 4 | Windows | 20 | 33.33 | -13.33 | 177.7778 |
| 5 | Mac | 60 | 33.33 | 26.67 | 711.1111 |
| 6 | Linux | 20 | 33.33 | -13.33 | 177.7778 |
| 7 | Total | 100 | | | 1066.667 |
| 8 | Count | 3 | | | |
| 9 | | | | | |
| 10 | Chisq-obs | 32 | | | |
| 11 | | | | | |

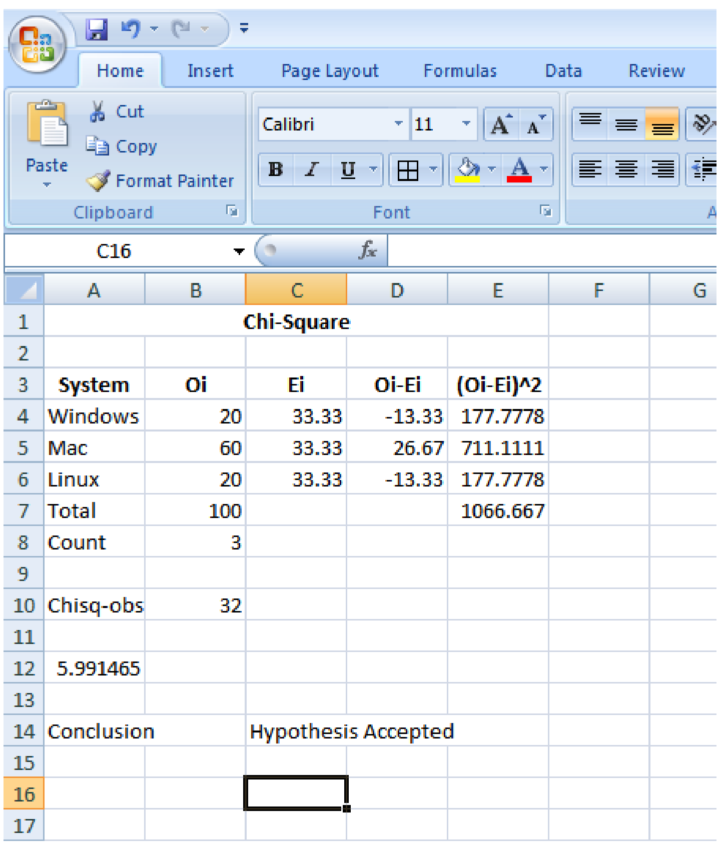
**Calculate chi-squared tabled value using the CHIINV function() where the probability is usually given as α=0.05 and degree of freedom is no of rows-1. In this case, 3-1 =2.**



**Finally, draw the conclusion.**



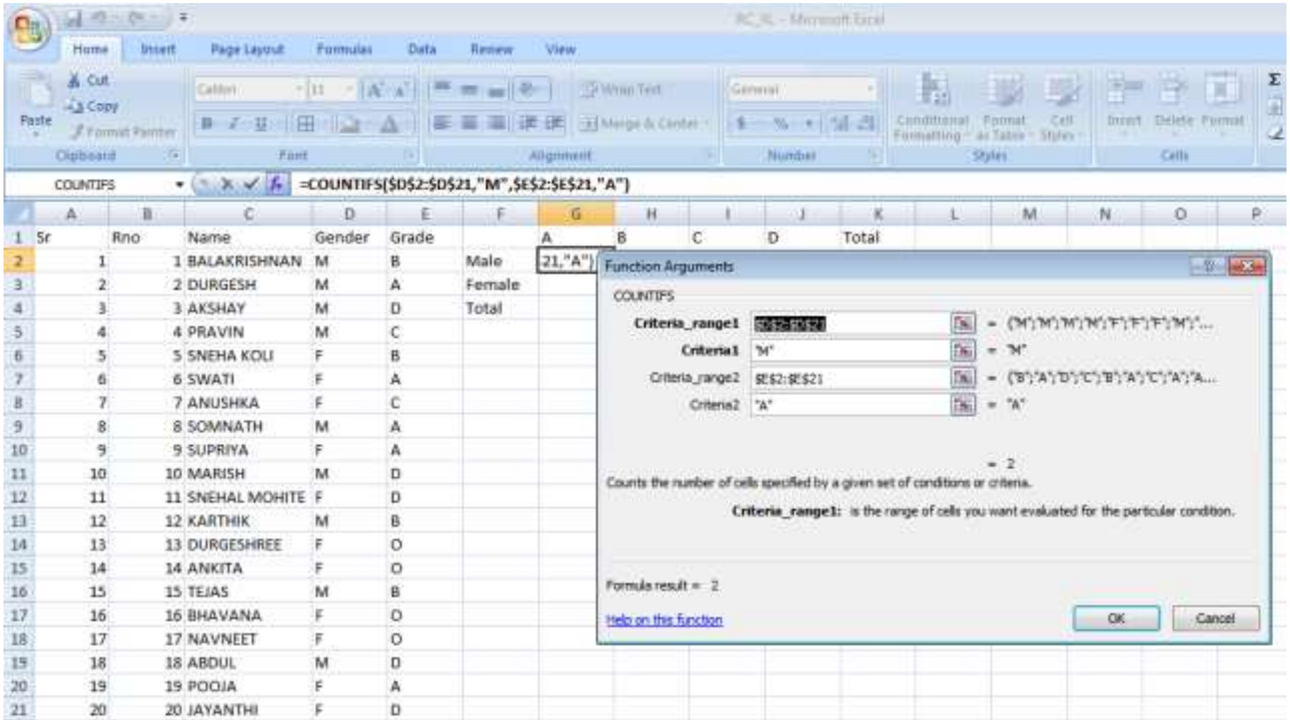



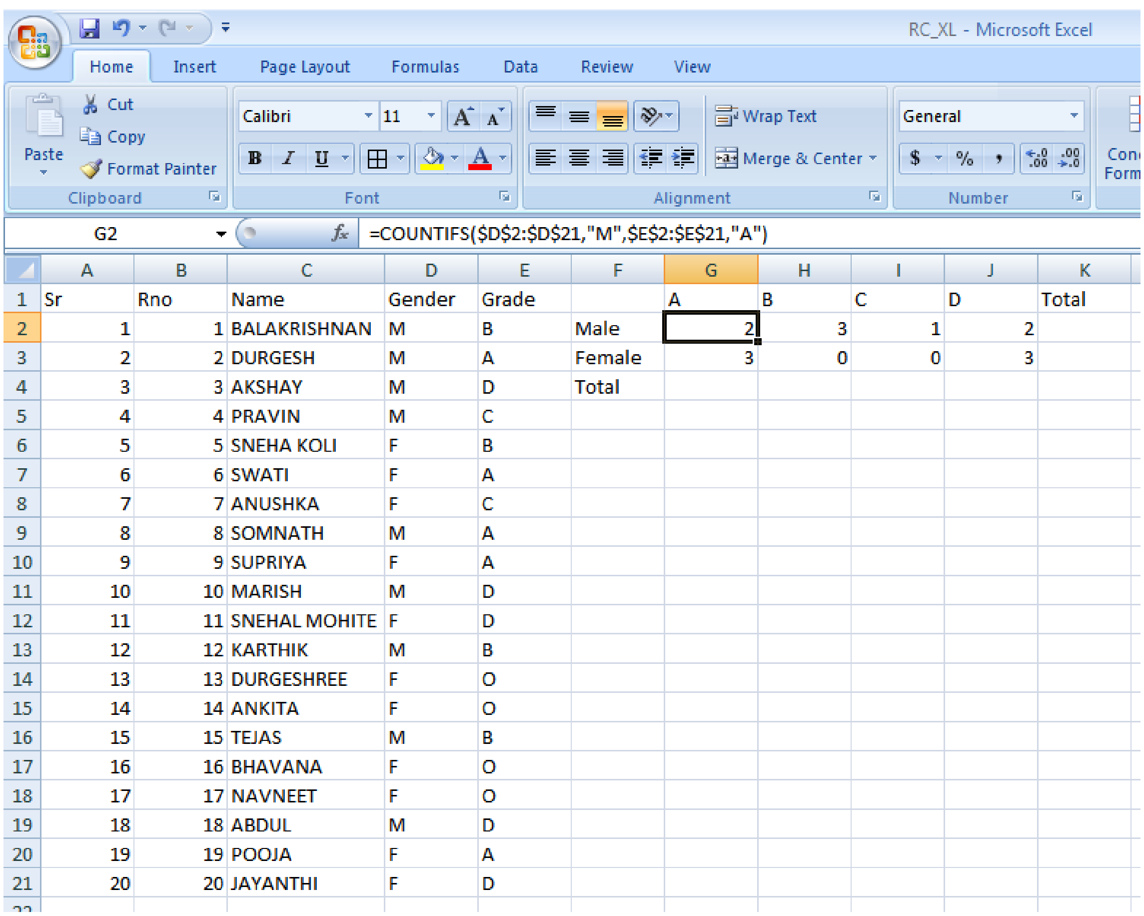
## B. <u>Perform testing of hypothesis using chi-squared test of independence</u>

In a study to understand the performacne of M. Sc. IT Part -1 class, a college selects a random sample of 20 students. Each student was asked his grade obtained in B. Sc.IT . The sample is as given below

| Sr | Rno | Name | Gender | Grade |
|---|---|---|---|---|
| 1 | 1 | BALAKRISHNAN | M | B |
| 2 | 2 | DURGESH | M | A |
| 3 | 3 | AKSHAY | M | D |
| 4 | 4 | PRAVIN | M | C |
| 5 | 5 | SNEHA KOLI | F | B |
| 6 | 6 | SWATI | F | A |
| 7 | 7 | ANUSHKA | F | C |
| 8 | 8 | SOMNATH | M | A |
| 9 | 9 | SUPRIYA | F | A |
| 10 | 10 | MARISH | M | D |
| 11 | 11 | SNEHAL MOHITE | F | D |
| 12 | 12 | KARTHIK | M | B |
| 13 | 13 | DURGESHREE | F | O |
| 14 | 14 | ANKITA | F | O |
| 15 | 15 | TEJAS | M | B |
| 16 | 16 | BHAVANA | F | O |
| 17 | 17 | NAVNEET | F | O |
| 18 | 18 | ABDUL | M | D |
| 19 | 19 | POOJA | F | A |
| 20 | 20 | JAYANTHI | F | D |

**Null Hypothesis - H0 : The performance of girls students is same as boys students.**
**Alternate Hypothesis - H1 : The performance of boys and girls students are different .**
**Solution:**
**We start by calculating the number of male students who got O,A,B,C and D grade respectively. We use the COUNTIFS() function in Excel for the same. COUNTIFS() can work on multiple conditions.**



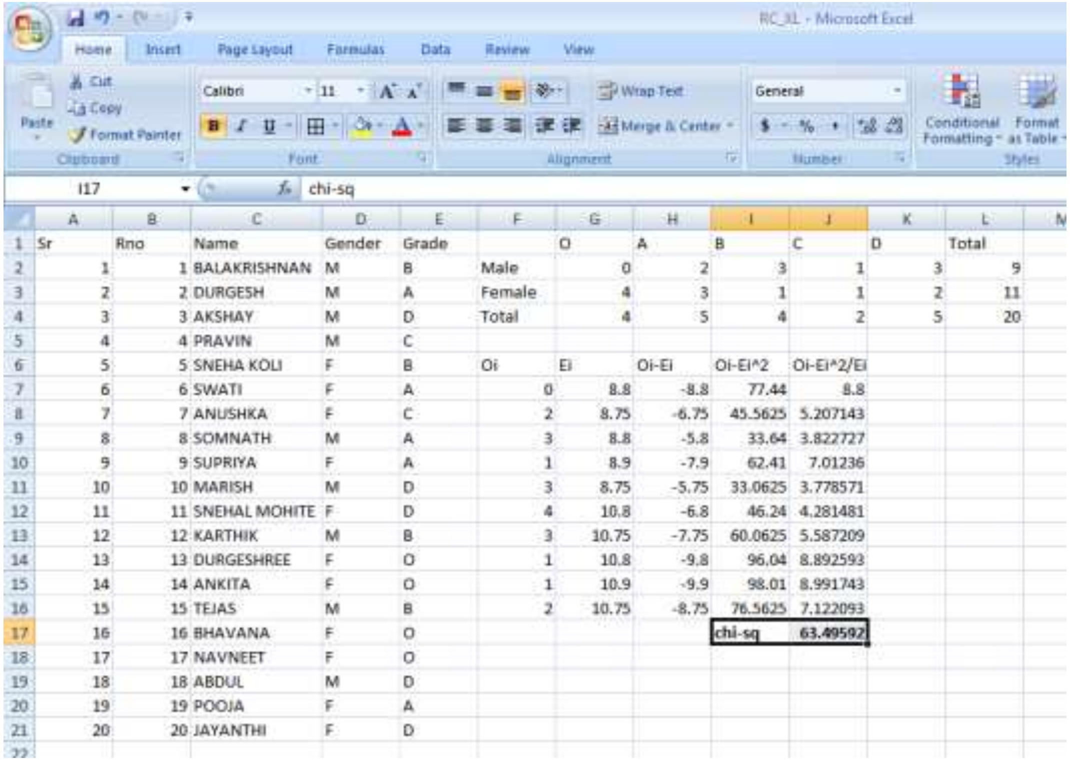

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Sr | Rno | Name | Gender | Grade | | A | B | C | D | Total |
| 2 | 1 | 1 | BALAKRISHNAN | M | B | Male | 2 | 3 | 1 | 2 | |
| 3 | 2 | 2 | DURGESH | M | A | Female | 3 | 0 | 0 | 3 | |
| 4 | 3 | 3 | AKSHAY | M | D | Total | | | | | |
| 5 | 4 | 4 | PRAVIN | M | C | | | | | | |
| 6 | 5 | 5 | SNEHA KOLI | F | B | | | | | | |
| 7 | 6 | 6 | SWATI | F | A | | | | | | |
| 8 | 7 | 7 | ANUSHKA | F | C | | | | | | |
| 9 | 8 | 8 | SOMNATH | M | A | | | | | | |
| 10 | 9 | 9 | SUPRIYA | F | A | | | | | | |
| 11 | 10 | 10 | MARISH | M | D | | | | | | |
| 12 | 11 | 11 | SNEHAL MOHITE | F | D | | | | | | |
| 13 | 12 | 12 | KARTHIK | M | B | | | | | | |
| 14 | 13 | 13 | DURGESHREE | F | O | | | | | | |
| 15 | 14 | 14 | ANKITA | F | O | | | | | | |
| 16 | 15 | 15 | TEJAS | M | B | | | | | | |
| 17 | 16 | 16 | BHAVANA | F | O | | | | | | |
| 18 | 17 | 17 | NAVNEET | F | O | | | | | | |
| 19 | 18 | 18 | ABDUL | M | D | | | | | | |
| 20 | 19 | 19 | POOJA | F | A | | | | | | |
| 21 | 20 | 20 | JAYANTHI | F | D | | | | | | |

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Sr | Rno | Name | Gender | Grade | | O | A | B | C | D | Total |
| 2 | 1 | 1 | BALAKRISHNAN | M | B | Male | 0 | 2 | 3 | 1 | 3 | 9 |
| 3 | 2 | 2 | DURGESH | M | A | Female | 4 | 3 | 1 | 1 | 2 | 11 |
| 4 | 3 | 3 | AKSHAY | M | D | Total | 4 | 5 | 4 | 2 | 5 | 20 |
| 5 | 4 | 4 | PRAVIN | M | C | | | | | | | |
| 6 | 5 | 5 | SNEHA KOLI | F | B | | | | | | | |
| 7 | 6 | 6 | SWATI | F | A | | | | | | | |
| 8 | 7 | 7 | ANUSHKA | F | C | | | | | | | |
| 9 | 8 | 8 | SOMNATH | M | A | | | | | | | |
| 10 | 9 | 9 | SUPRIYA | F | A | | | | | | | |
| 11 | 10 | 10 | MARISH | M | D | | | | | | | |
| 12 | 11 | 11 | SNEHAL MOHITE | F | D | | | | | | | |
| 13 | 12 | 12 | KARTHIK | M | B | | | | | | | |
| 14 | 13 | 13 | DURGESHREE | F | O | | | | | | | |
| 15 | 14 | 14 | ANKITA | F | O | | | | | | | |
| 16 | 15 | 15 | TEJAS | M | B | | | | | | | |
| 17 | 16 | 16 | BHAVANA | F | O | | | | | | | |
| 18 | 17 | 17 | NAVNEET | F | O | | | | | | | |
| 19 | 18 | 18 | ABDUL | M | D | | | | | | | |
| 20 | 19 | 19 | POOJA | F | A | | | | | | | |
| 21 | 20 | 20 | JAYANTHI | F | D | | | | | | | |

**Next we calculate the Ei for all the values. The Ei is calculated as (row total-coltotal)/total data size for individual data sets. Next we perform all the sets similar to chi-sqaure goodness of fit.**

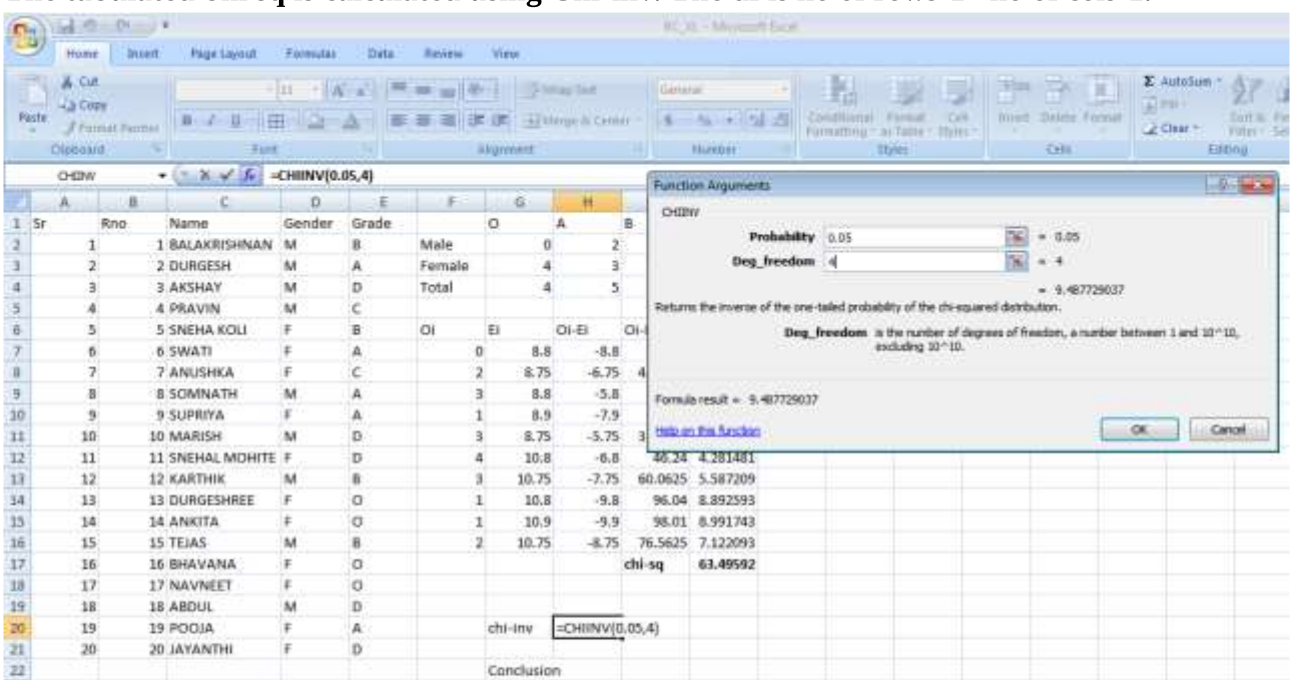
| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Sr | Rno | Name | Gender | Grade | | O | A | B | C | D | Total |
| 2 | 1 | 1 | BALAKRISHNAN | M | B | Male | 0 | 2 | 3 | 1 | 3 | 9 |
| 3 | 2 | 2 | DURGESH | M | A | Female | 4 | 3 | 1 | 1 | 2 | 11 |
| 4 | 3 | 3 | AKSHAY | M | D | Total | 4 | 5 | 4 | 2 | 5 | 20 |
| 5 | 4 | 4 | PRAVIN | M | C | | | | | | | |
| 6 | 5 | 5 | SNEHA KOLI | F | B | Oi | Ei | Oi-Ei | Oi-Ei^2 | Oi-Ei^2/Ei | | |
| 7 | 6 | 6 | SWATI | F | A | 0 | 8.8 | -8.8 | 77.44 | 8.8 | | |
| 8 | 7 | 7 | ANUSHKA | F | C | 2 | 8.75 | -6.75 | 45.5625 | 5.207143 | | |
| 9 | 8 | 8 | SOMNATH | M | A | 3 | 8.8 | -5.8 | 33.64 | 3.822727 | | |
| 10 | 9 | 9 | SUPRIYA | F | A | 1 | 8.9 | -7.9 | 62.41 | 7.01236 | | |
| 11 | 10 | 10 | MARISH | M | D | 3 | 8.75 | -5.75 | 33.0625 | 3.778571 | | |
| 12 | 11 | 11 | SNEHAL MOHITE | F | D | 4 | 10.8 | -6.8 | 46.24 | 4.281481 | | |
| 13 | 12 | 12 | KARTHIK | M | B | 3 | 10.75 | -7.75 | 60.0625 | 5.587209 | | |
| 14 | 13 | 13 | DURGESHREE | F | O | 1 | 10.8 | -9.8 | 96.04 | 8.892593 | | |
| 15 | 14 | 14 | ANKITA | F | O | 1 | 10.9 | -9.9 | 98.01 | 8.991743 | | |
| 16 | 15 | 15 | TEJAS | M | B | 2 | 10.75 | -8.75 | 76.5625 | 7.122093 | | |
| 17 | 16 | 16 | BHAVANA | F | O | | | | | | | |
| 18 | 17 | 17 | NAVNEET | F | O | | | | | | | |
| 19 | 18 | 18 | ABDUL | M | D | | chi-sq | | | | | |
| 20 | 19 | 19 | POOJA | F | A | | | | | | | |
| 21 | 20 | 20 | JAYANTHI | F | D | | | | | | | |

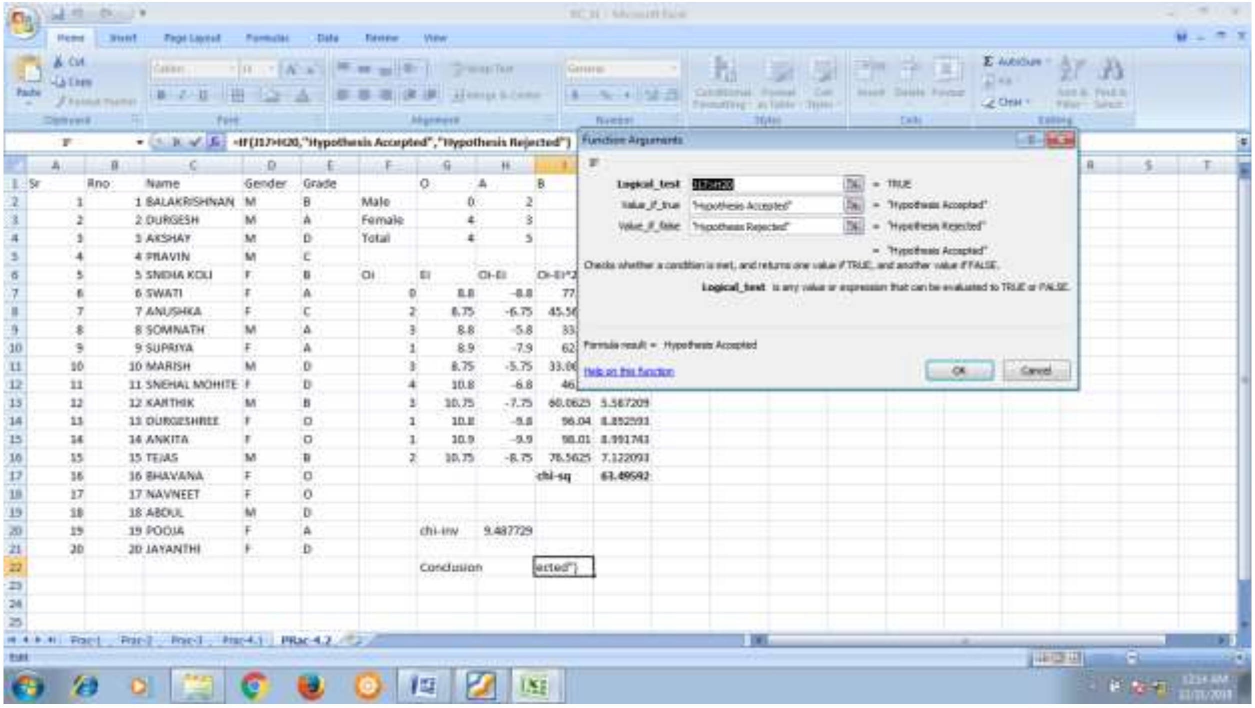
**The observed chi-sq is calculated as the sum of the last column.**

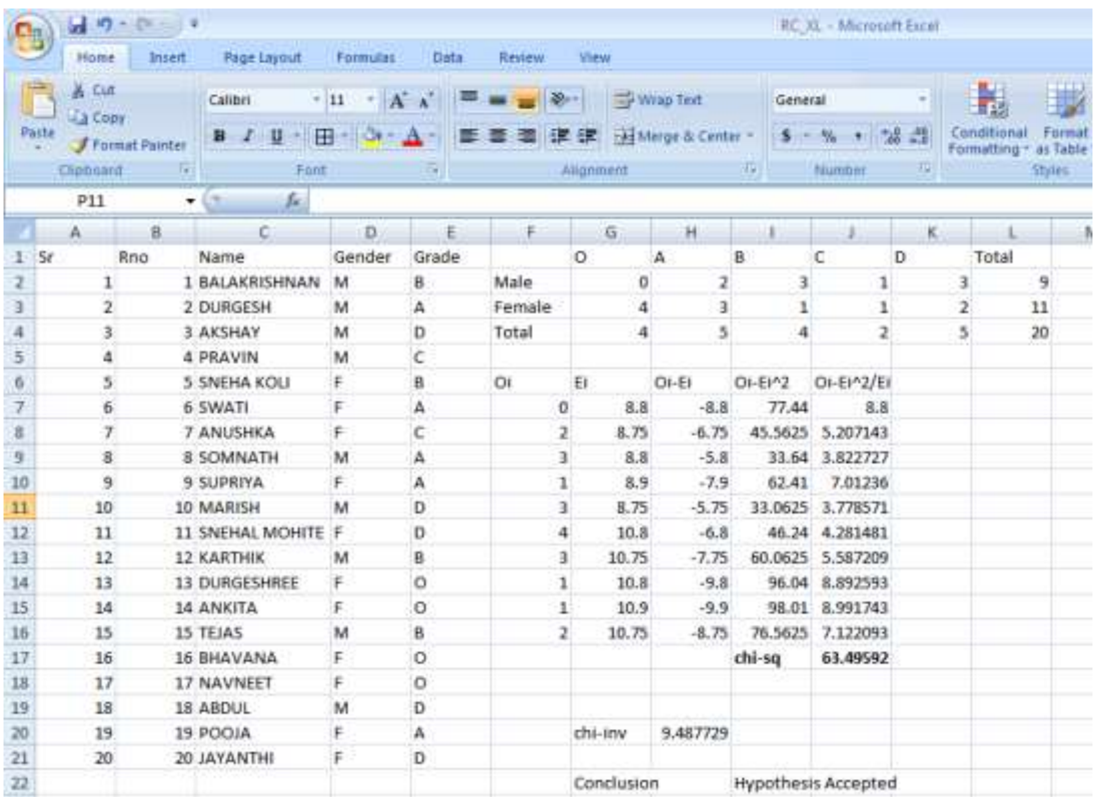


| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Sr | Rno | Name | Gender | Grade | | O | A | B | C | D | Total |
| 2 | 1 | 1 | BALAKRISHNAN | M | B | Male | 0 | 2 | 3 | 1 | 3 | 9 |
| 3 | 2 | 2 | DURGESH | M | A | Female | 4 | 3 | 1 | 1 | 2 | 11 |
| 4 | 3 | 3 | AKSHAY | M | D | Total | 4 | 5 | 4 | 2 | 5 | 20 |
| 5 | 4 | 4 | PRAVIN | M | C | | | | | | | |
| 6 | 5 | 5 | SNEHA KOLI | F | B | Oi | Ei | Oi-Ei | Oi-Ei^2 | Oi-Ei^2/Ei | | |
| 7 | 6 | 6 | SWATI | F | A | 0 | 8.8 | -8.8 | 77.44 | 8.8 | | |
| 8 | 7 | 7 | ANUSHKA | F | C | 2 | 8.75 | -6.75 | 45.5625 | 5.207143 | | |
| 9 | 8 | 8 | SOMNATH | M | A | 3 | 8.8 | -5.8 | 33.64 | 3.822727 | | |
| 10 | 9 | 9 | SUPRIYA | F | A | 1 | 8.9 | -7.9 | 62.41 | 7.01236 | | |
| 11 | 10 | 10 | MARISH | M | D | 3 | 8.75 | -5.75 | 33.0625 | 3.778571 | | |
| 12 | 11 | 11 | SNEHAL MOHITE | F | D | 4 | 10.8 | -6.8 | 46.24 | 4.281481 | | |
| 13 | 12 | 12 | KARTHIK | M | B | 3 | 10.75 | -7.75 | 60.0625 | 5.587209 | | |
| 14 | 13 | 13 | DURGESHREE | F | O | 1 | 10.8 | -9.8 | 96.04 | 8.892593 | | |
| 15 | 14 | 14 | ANKITA | F | O | 1 | 10.9 | -9.9 | 98.01 | 8.991743 | | |
| 16 | 15 | 15 | TEJAS | M | B | 2 | 10.75 | -8.75 | 76.5625 | 7.122093 | | |
| 17 | 16 | 16 | BHAVANA | F | O | | | | chi-sq | 63.49592 | | |
| 18 | 17 | 17 | NAVNEET | F | O | | | | | | | |
| 19 | 18 | 18 | ABDUL | M | D | | | | | | | |
| 20 | 19 | 19 | POOJA | F | A | | | | | | | |
| 21 | 20 | 20 | JAYANTHI | F | D | | | | | | | |

**The tabulated bhi sq is calculated using Chi-inv. The df is no of rows-1* no of cols-1.**





| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Sr | Rno | Name | Gender | Grade | | O | A | B | C | D | Total |
| 2 | 1 | 1 | BALAKRISHNAN | M | B | Male | 0 | 2 | 3 | 1 | 3 | 9 |
| 3 | 2 | 2 | DURGESH | M | A | Female | 4 | 3 | 1 | 1 | 2 | 11 |
| 4 | 3 | 3 | AKSHAY | M | D | Total | 4 | 5 | 4 | 2 | 5 | 20 |
| 5 | 4 | 4 | PRAVIN | M | C | | | | | | | |
| 6 | 5 | 5 | SNEHA KOLI | F | B | Oi | Ei | Oi-Ei | Oi-Ei^2 | Oi-Ei^2/Ei | | |
| 7 | 6 | 6 | SWATI | F | A | 0 | 8.8 | -8.8 | 77.44 | 8.8 | | |
| 8 | 7 | 7 | ANUSHKA | F | C | 2 | 8.75 | -6.75 | 45.5625 | 5.207143 | | |
| 9 | 8 | 8 | SOMNATH | M | A | 3 | 8.8 | -5.8 | 33.64 | 3.822727 | | |
| 10 | 9 | 9 | SUPRIYA | F | A | 1 | 8.9 | -7.9 | 62.41 | 7.01236 | | |
| 11 | 10 | 10 | MARISH | M | D | 3 | 8.75 | -5.75 | 33.0625 | 3.778571 | | |
| 12 | 11 | 11 | SNEHAL MOHITE | F | D | 4 | 10.8 | -6.8 | 46.24 | 4.281481 | | |
| 13 | 12 | 12 | KARTHIK | M | B | 3 | 10.75 | -7.75 | 60.0625 | 5.587209 | | |
| 14 | 13 | 13 | DURGESHREE | F | O | 1 | 10.8 | -9.8 | 96.04 | 8.892593 | | |
| 15 | 14 | 14 | ANKITA | F | O | 1 | 10.9 | -9.9 | 98.01 | 8.991743 | | |
| 16 | 15 | 15 | TEJAS | M | B | 2 | 10.75 | -8.75 | 76.5625 | 7.122093 | | |
| 17 | 16 | 16 | BHAVANA | F | O | | | | chi-sq | 63.49592 | | |
| 18 | 17 | 17 | NAVNEET | F | O | | | | | | | |
| 19 | 18 | 18 | ABDUL | M | D | | | | | | | |
| 20 | 19 | 19 | POOJA | F | A | | chi-inv | 9.487729 | | | | |
| 21 | 20 | 20 | JAYANTHI | F | D | | | | | | | |
| 22 | | | | | | | Conclusion | | Hypothesis Accepted | | | |

# Practical No 5

**Aim:** Perform testing of hypothesis using Z-test.

**Program Code for one-sample Z test.**

**Code:**

```
from statsmodels.stats import weightstats as stests
import pandas as pd
from scipy import stats
df = pd.read_csv("blood_pressure.csv")
df[['bp_before','bp_after']].describe()
print(df)
ztest ,pval = stests.ztest(df['bp_before'], x2=None, value=156)
print(float(pval))
if pval<0.05:
 print("reject null hypothesis")
else:
 print("accept null hypothesis")
```

**Output:**

```
   Patient  gender agegrp  bp_before  bp_after
0        1    Male  30-45        143       153
1        2    Male  30-45        163       170
2        3    Male  45-60        153       168
3        4    Male  45-60        153       142
4        5    Male    60+        146       141
5        6  Female  30-45        152       152
6        7  Female  30-45        161       152
7        8  Female  45-60        165       174
8        9  Female  45-60        149       151
9       10  Female    60+        185       163
0.795362004282681
accept null hypothesis
>>> 
```

**Two-sample Z test**

**Code:**

```
import pandas as pd
from statsmodels.stats import weightstats as stests
df = pd.read_csv("blood_pressure.csv")
df[['bp_before','bp_after']].describe()
print(df)
ztest ,pval = stests.ztest(df['bp_before'], x2=df['bp_after'], value=0)
print(float(pval))
if pval<0.05:
```

 print("reject null hypothesis")
else:
 print("accept null hypothesis")

**Output:**

Taking value = 156

```
   Patient  gender agegrp  bp_before  bp_after
0        1    Male  30-45        143       153
1        2    Male  30-45        163       170
2        3    Male  45-60        153       168
3        4    Male  45-60        153       142
4        5    Male    60+        146       141
5        6  Female  30-45        152       152
6        7  Female  30-45        161       152
7        8  Female  45-60        165       174
8        9  Female  45-60        149       151
9       10  Female    60+        185       163
5.441689616944356e-189
reject null hypothesis
>>>
```

Taking value = 0

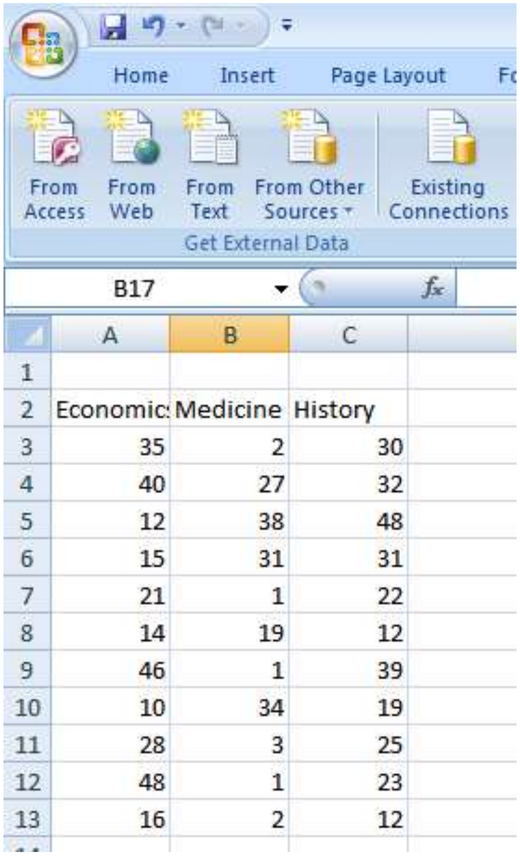
```
 RESTART: D:\D-Drive\SIES\Msc IT\Msc IT (Part-1)\Pra
Material\5_2.py
   Patient  gender agegrp  bp_before  bp_after
0        1    Male  30-45        143       153
1        2    Male  30-45        163       170
2        3    Male  45-60        153       168
3        4    Male  45-60        153       142
4        5    Male    60+        146       141
5        6  Female  30-45        152       152
6        7  Female  30-45        161       152
7        8  Female  45-60        165       174
8        9  Female  45-60        149       151
9       10  Female    60+        185       163
0.9399140932267469
accept null hypothesis
>>>
```

# Practical No 6

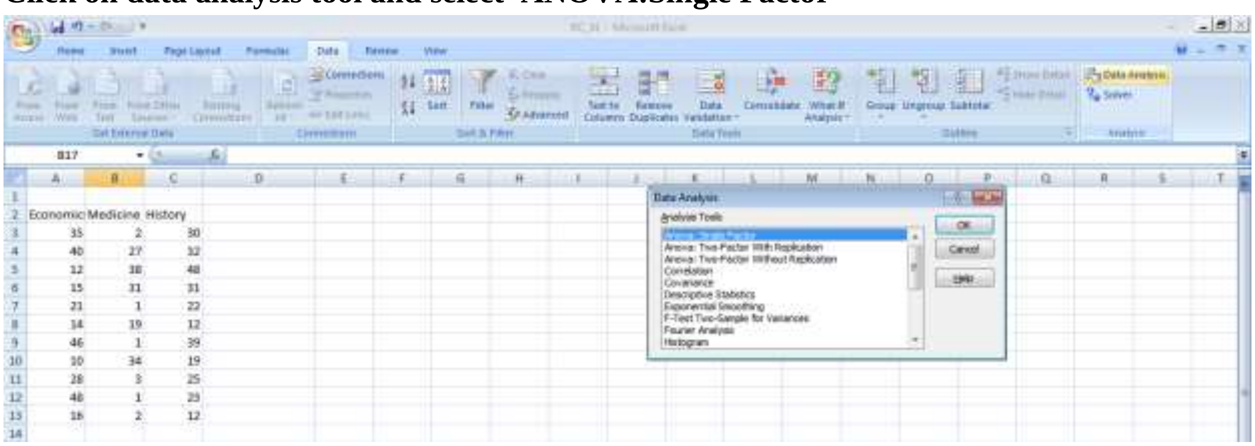
## A. <u>Perform testing of hypothesis using One -way ANOVA.</u>

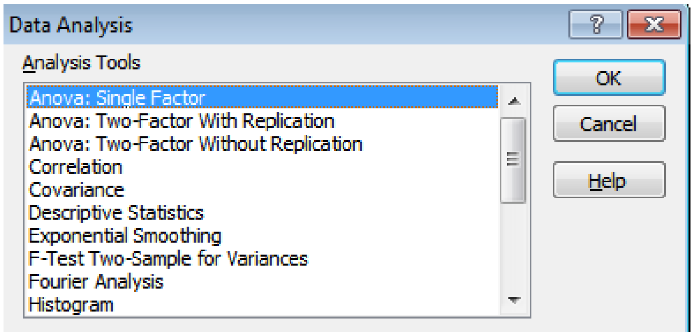
**Solution:**

**Data is given.**

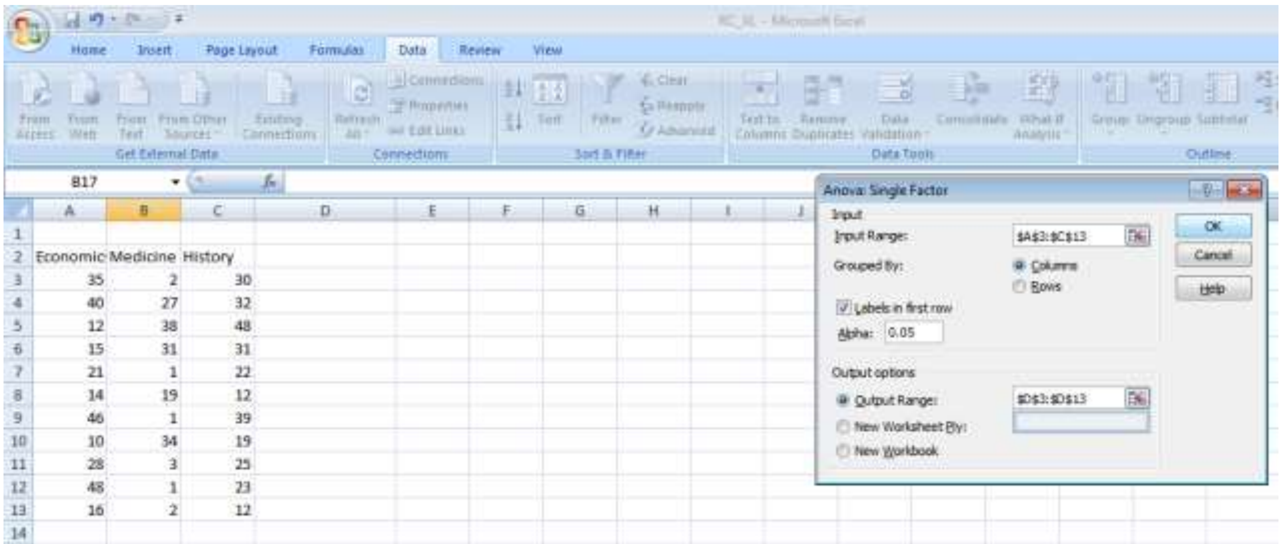


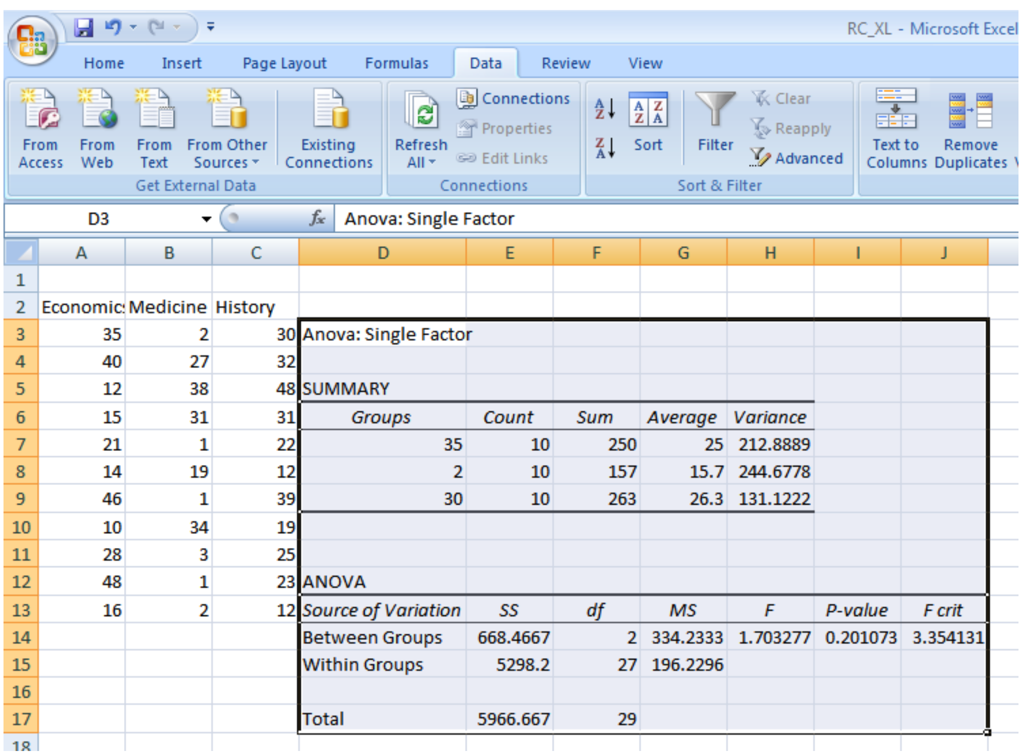

**Click on data analysis tool and select ANOVA:Single Factor**



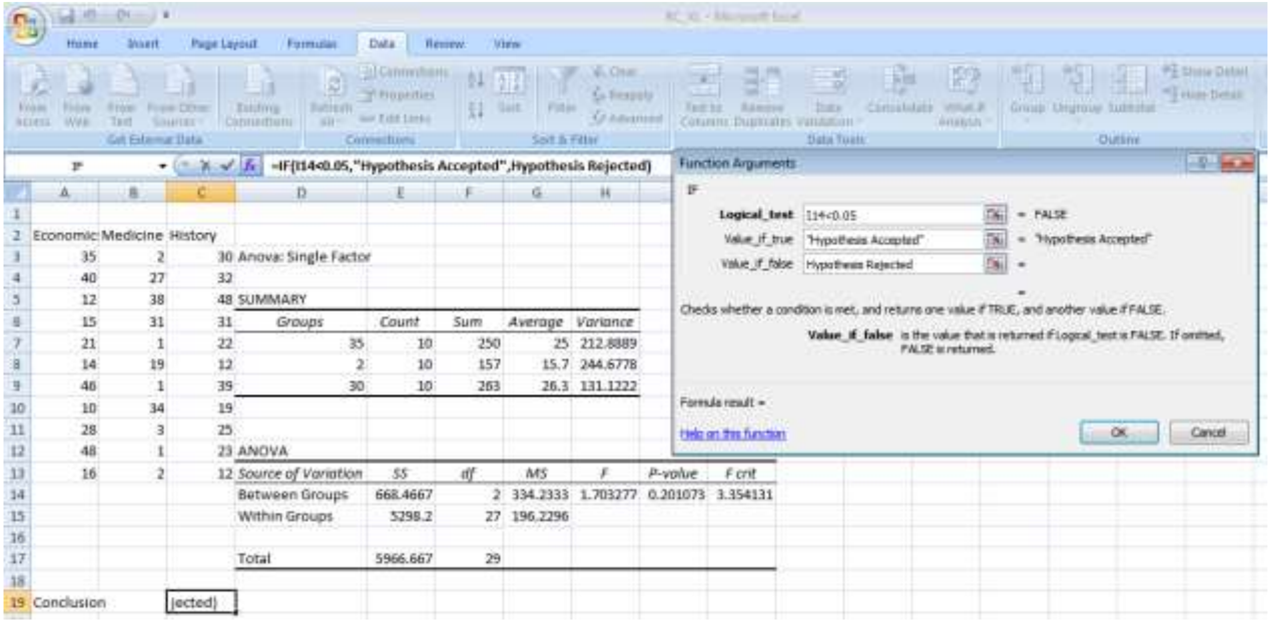


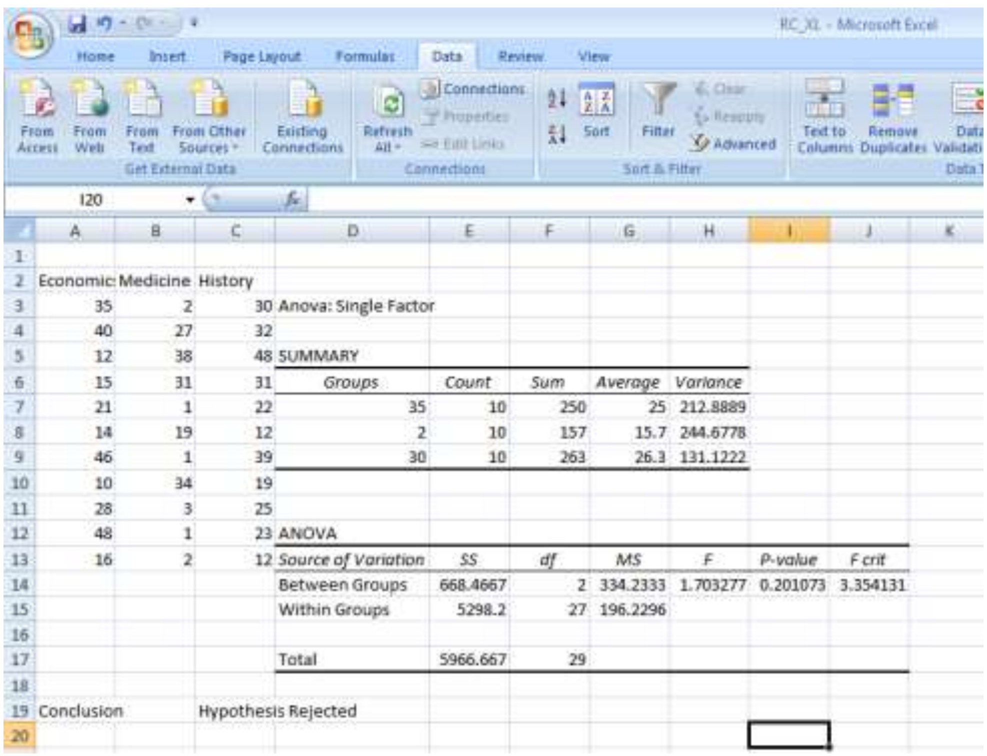

**Select the input range, Output range and define the Alpha value. Click OK.**



**Since the p valueis greater than 0.05. Hypothesis is Rejected.**

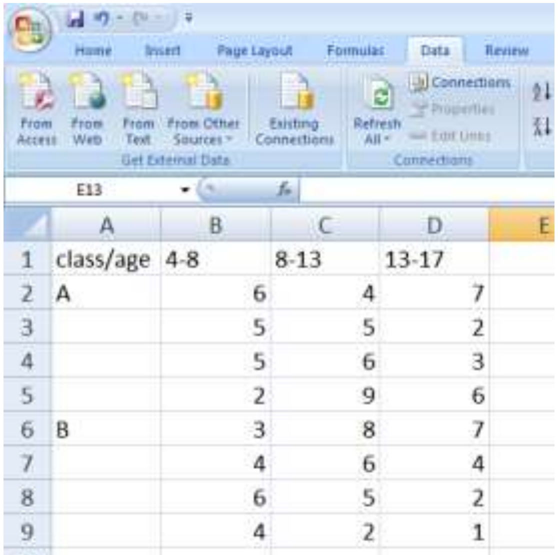


## B. Perform testing of hypothesis using Two -way ANOVA.

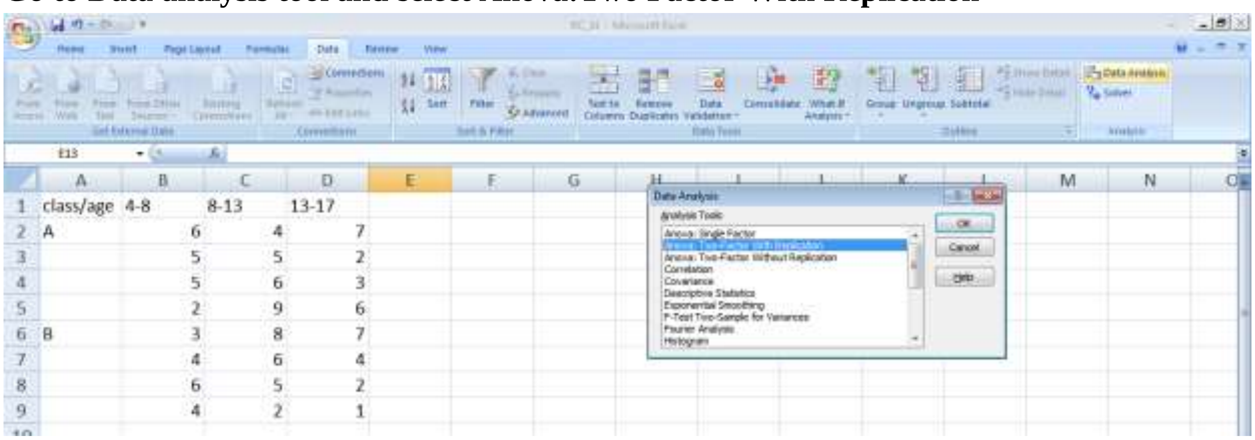
**Solution:**

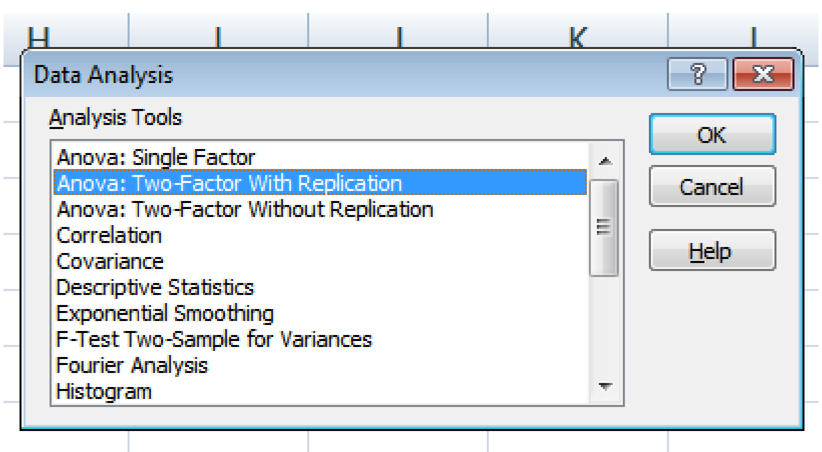
**Given Data Set**



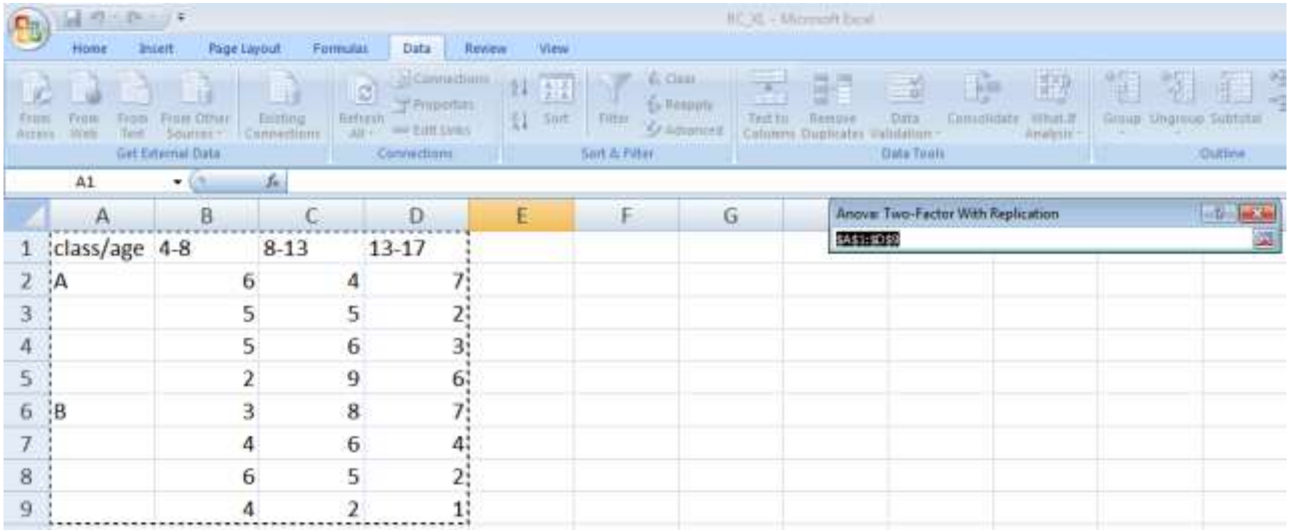
| class/age | 4-8 | 8-13 | 13-17 |
|---|---|---|---|
| A | 6 | 4 | 7 |
| | 5 | 5 | 2 |
| | 5 | 6 | 3 |
| | 2 | 9 | 6 |
| B | 3 | 8 | 7 |
| | 4 | 6 | 4 |
| | 6 | 5 | 2 |
| | 4 | 2 | 1 |

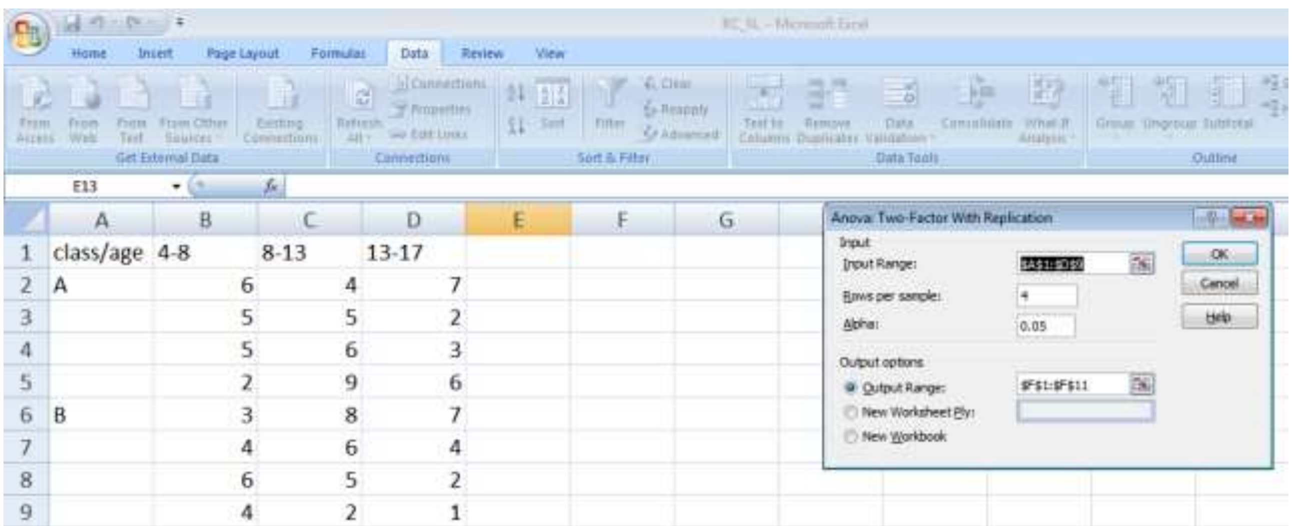
**Go to Data analysis tool and select Anova:Two Factor With Replication**

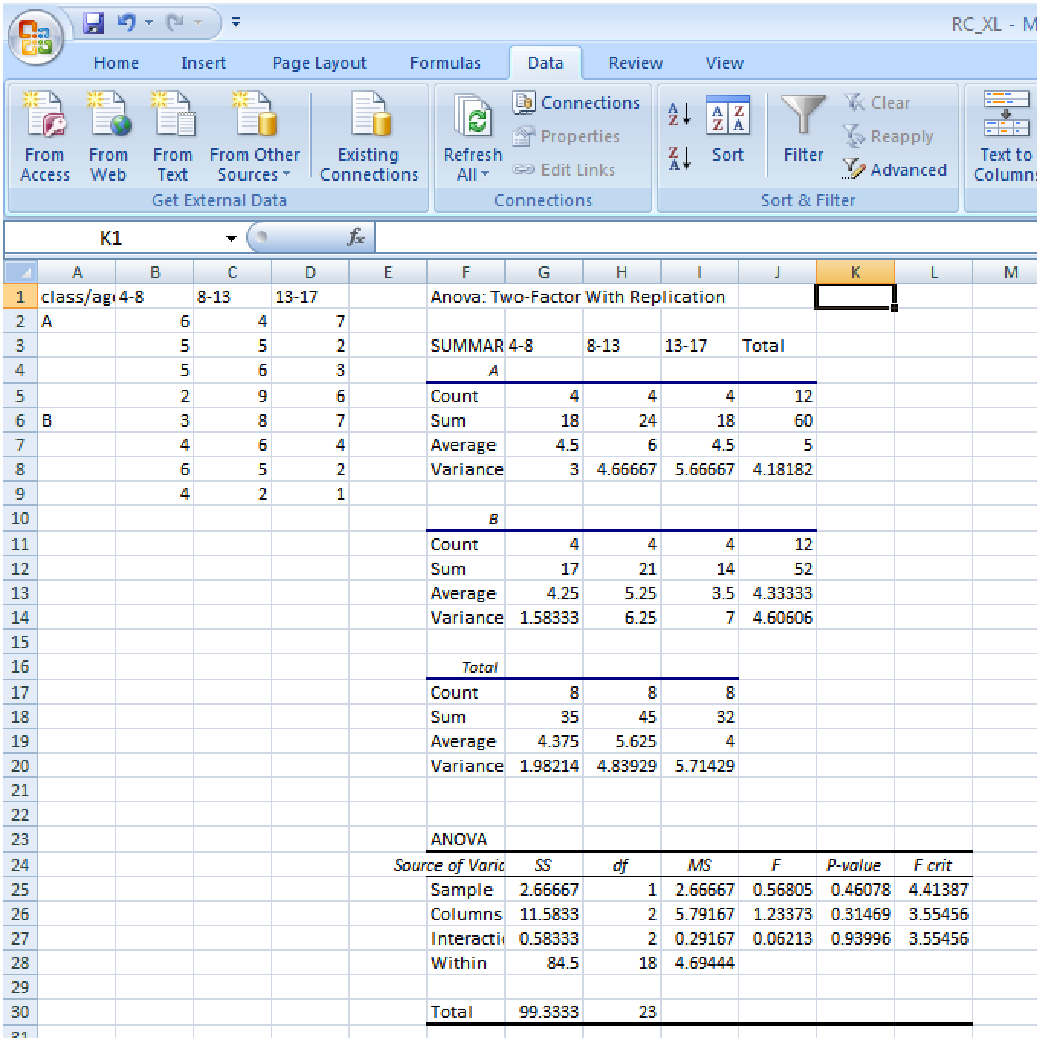




**Select all the values in the table. No of rows should be counted and entered.**





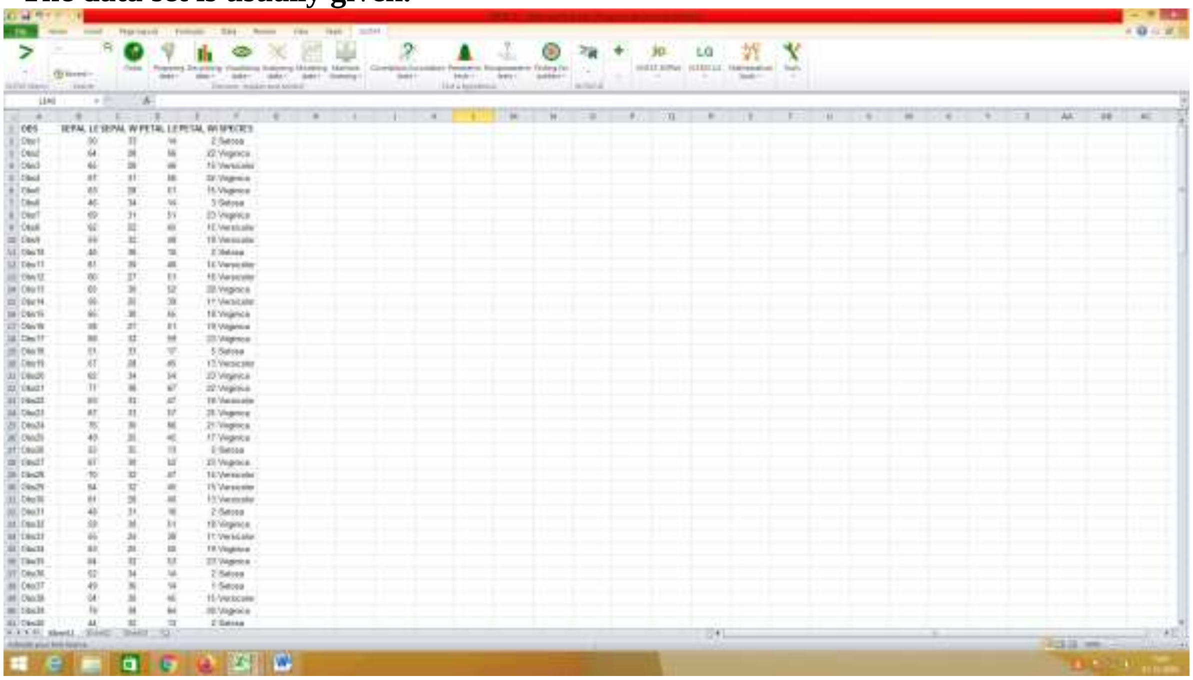

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | class/ag | 4-8 | 8-13 | 13-17 | | Anova: Two-Factor With Replication | | | | | | | |
| 2 | A | 6 | 4 | 7 | | | | | | | | | |
| 3 | | 5 | 5 | 2 | | SUMMAR | 4-8 | 8-13 | 13-17 | Total | | | |
| 4 | | 5 | 6 | 3 | | A | | | | | | | |
| 5 | | 2 | 9 | 6 | | Count | 4 | 4 | 4 | 12 | | | |
| 6 | B | 3 | 8 | 7 | | Sum | 18 | 24 | 18 | 60 | | | |
| 7 | | 4 | 6 | 4 | | Average | 4.5 | 6 | 4.5 | 5 | | | |
| 8 | | 6 | 5 | 2 | | Variance | 3 | 4.66667 | 5.66667 | 4.18182 | | | |
| 9 | | 4 | 2 | 1 | | | | | | | | | |
| 10 | | | | | | B | | | | | | | |
| 11 | | | | | | Count | 4 | 4 | 4 | 12 | | | |
| 12 | | | | | | Sum | 17 | 21 | 14 | 52 | | | |
| 13 | | | | | | Average | 4.25 | 5.25 | 3.5 | 4.33333 | | | |
| 14 | | | | | | Variance | 1.58333 | 6.25 | 7 | 4.60606 | | | |
| 15 | | | | | | | | | | | | | |
| 16 | | | | | | Total | | | | | | | |
| 17 | | | | | | Count | 8 | 8 | 8 | | | | |
| 18 | | | | | | Sum | 35 | 45 | 32 | | | | |
| 19 | | | | | | Average | 4.375 | 5.625 | 4 | | | | |
| 20 | | | | | | Variance | 1.98214 | 4.83929 | 5.71429 | | | | |
| 21 | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | |
| 23 | | | | | | ANOVA | | | | | | | |
| 24 | | | | | | Source of Varic | SS | df | MS | F | P-value | F crit | |
| 25 | | | | | | Sample | 2.66667 | 1 | 2.66667 | 0.56805 | 0.46078 | 4.41387 | |
| 26 | | | | | | Columns | 11.5833 | 2 | 5.79167 | 1.23373 | 0.31469 | 3.55456 | |
| 27 | | | | | | Interacti | 0.58333 | 2 | 0.29167 | 0.06213 | 0.93996 | 3.55456 | |
| 28 | | | | | | Within | 84.5 | 18 | 4.69444 | | | | |
| 29 | | | | | | | | | | | | | |
| 30 | | | | | | Total | 99.3333 | 23 | | | | | |

**Since all the p-values are greater than 0.05, the observation is not statistically significant.**

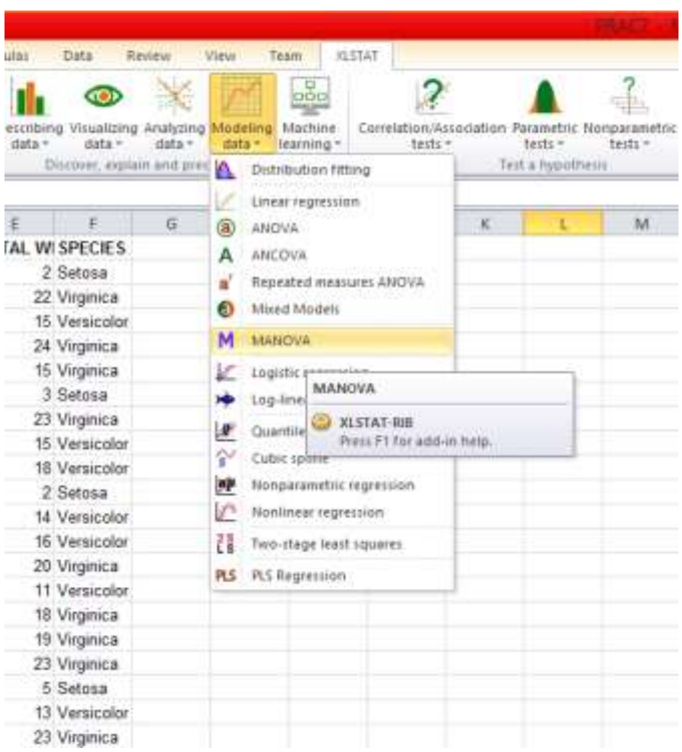
## C. Perform testing of hypothesis using multivariate ANOVA (MANOVA).

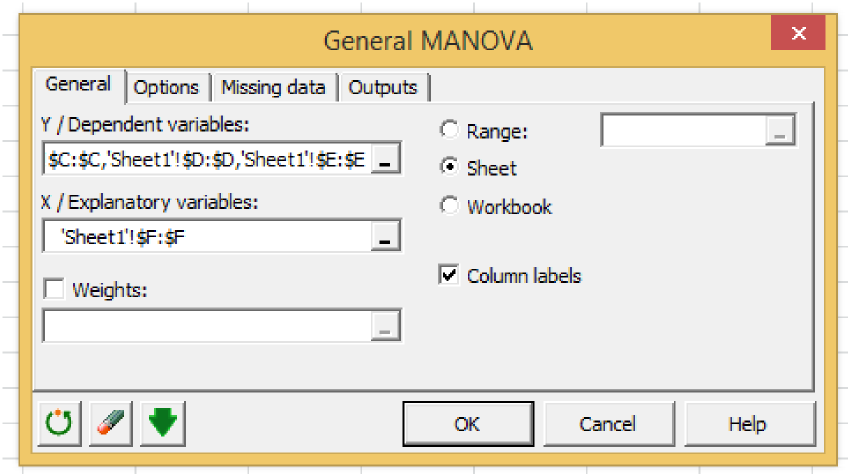
**Solution:**

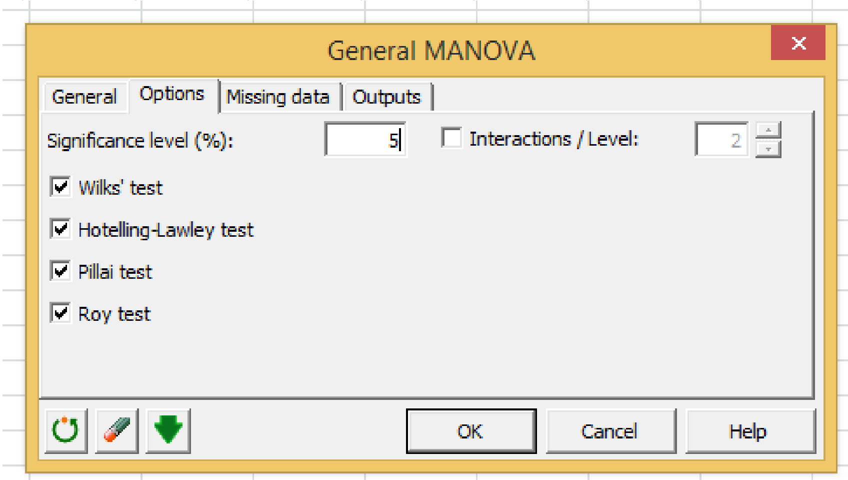
**The data set is usually given.**

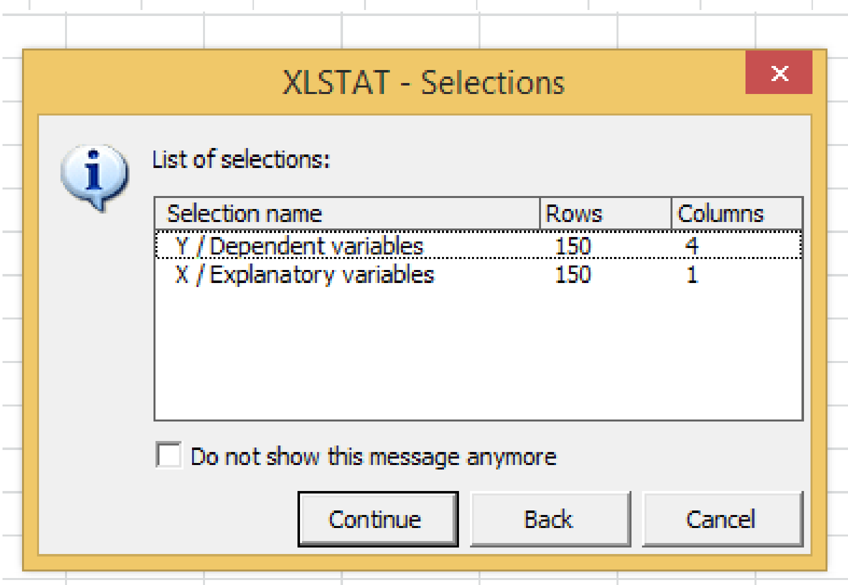


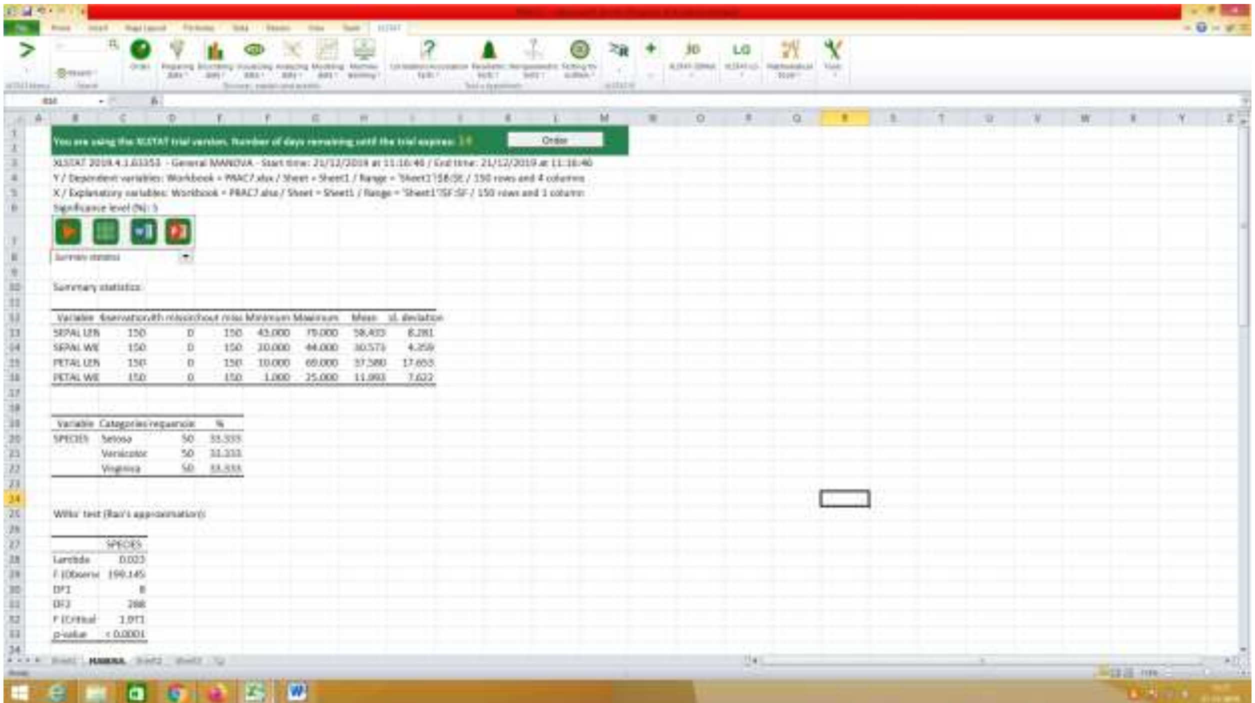
**XLSTAT is an Excel add-in which needs to be downloaded and installed from the XLSTAT Website.**




General MANOVA
General
Options
Missing data
Outputs
Y / Dependent variables:
$C:$C,'Sheet1'!$D:$D,'Sheet1'!$E:$E
X / Explanatory variables:
'Sheet1'!$F:$F
Weights:
Range:
Sheet
Workbook
Column labels
OK
Cancel
Help


General MANOVA
General
Options
Missing data
Outputs
Significance level (%):
5
Interactions / Level:
2
Wilks' test
Hotelling-Lawley test
Pillai test
Roy test
OK
Cancel
Help


XLSTAT - Selections
List of selections:
Selection name
Rows
Columns
Y / Dependent variables
150
4
X / Explanatory variables
150
1
Do not show this message anymore
Continue
Back
Cancel



You are using the XLSTAT trial version. Number of days remaining until the trial expires: 14

XLSTAT 2019.4.1.63353 - General MANOVA - Start time: 21/12/2019 at 11:16:40 / End time: 21/12/2019 at 11:16:40
Y / Dependent variables: Workbook = PRAC7.xlsx / Sheet = Sheet1 / Range = 'Sheet1'!$B:$E / 150 rows and 4 columns
X / Explanatory variables: Workbook = PRAC7.xlsx / Sheet = Sheet1 / Range = 'Sheet1'!$F:$F / 150 rows and 1 column
Significance level (%): 5

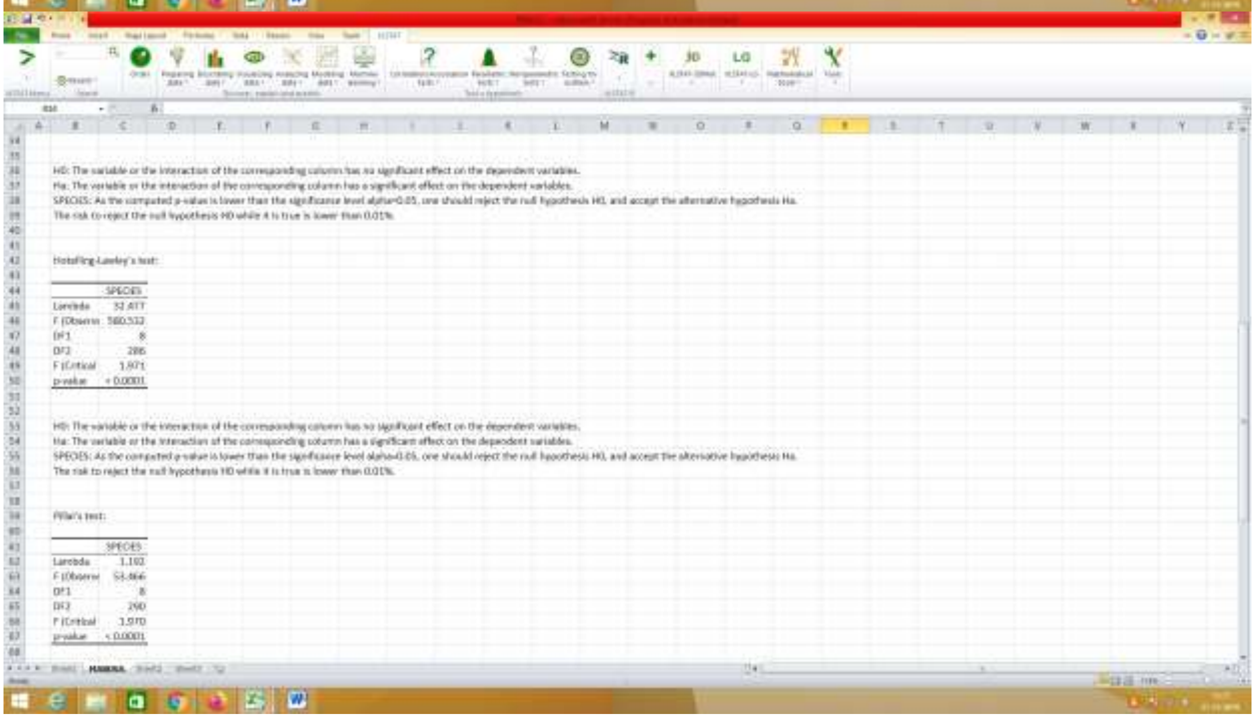
Summary statistics:

| Variable | Observations | Obs. with missing data | Obs. without missing data | Minimum | Maximum | Mean | Std. deviation |
|---|---|---|---|---|---|---|---|
| SEPAL LEN | 150 | 0 | 150 | 43.000 | 79.000 | 58.433 | 8.281 |
| SEPAL WIE | 150 | 0 | 150 | 20.000 | 44.000 | 30.573 | 4.359 |
| PETAL LEN | 150 | 0 | 150 | 10.000 | 69.000 | 37.580 | 17.653 |
| PETAL WIE | 150 | 0 | 150 | 1.000 | 25.000 | 11.993 | 7.622 |

| Variable | Categories | Frequencies | % |
|---|---|---|---|
| SPECIES | Setosa | 50 | 33.333 |
| | Versicolor | 50 | 33.333 |
| | Virginica | 50 | 33.333 |

Wilks' test (Rao's approximation):

| | SPECIES |
|---|---|
| Lambda | 0.023 |
| F (Observed value) | 199.145 |
| DF1 | 8 |
| DF2 | 288 |
| F (Critical value) | 1.971 |
| p-value | < 0.0001 |



H0: The variable or the interaction of the corresponding column has no significant effect on the dependent variables.
Ha: The variable or the interaction of the corresponding column has a significant effect on the dependent variables.
SPECIES: As the computed p-value is lower than the significance level alpha=0.05, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.
The risk to reject the null hypothesis H0 while it is true is lower than 0.01%.

Hotelling-Lawley's test:

| | SPECIES |
|---|---|
| Lambda | 32.477 |
| F (Observed value) | 580.532 |
| DF1 | 8 |
| DF2 | 286 |
| F (Critical value) | 1.971 |
| p-value | < 0.0001 |

H0: The variable or the interaction of the corresponding column has no significant effect on the dependent variables.
Ha: The variable or the interaction of the corresponding column has a significant effect on the dependent variables.
SPECIES: As the computed p-value is lower than the significance level alpha=0.05, one should reject the null hypothesis H0, and accept the alternative hypothesis Ha.
The risk to reject the null hypothesis H0 while it is true is lower than 0.01%.

Pillai's test:

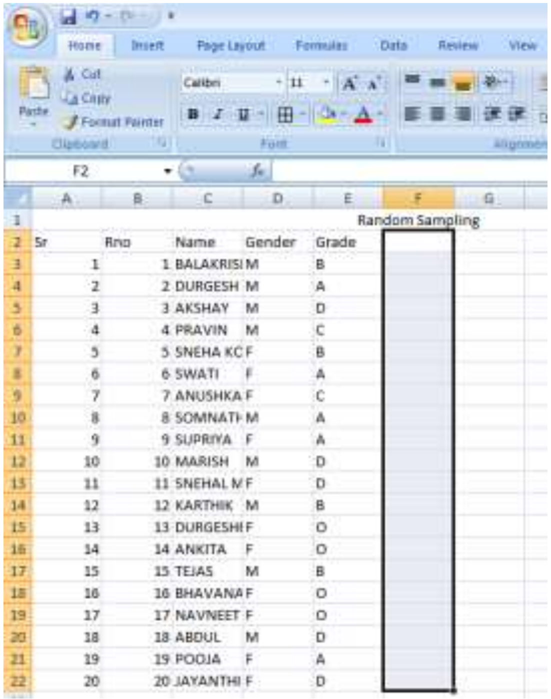
| | SPECIES |
|---|---|
| Lambda | 1.192 |
| F (Observed value) | 53.466 |
| DF1 | 8 |
| DF2 | 290 |
| F (Critical value) | 1.970 |
| p-value | < 0.0001 |

# Practical No 7

## A. Perform the Random sampling for the given data and analyze it.

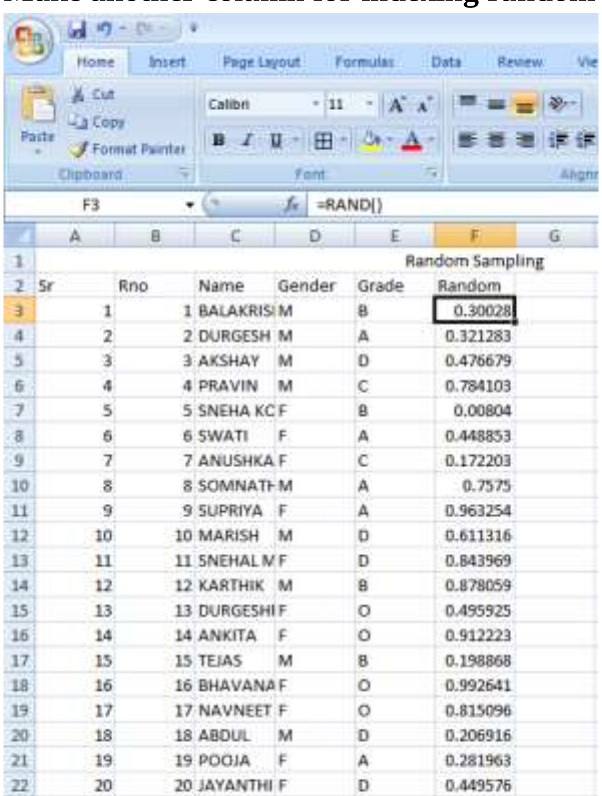
**Solution:**

**Given data set**



| Sr | Rno | Name | Gender | Grade | Random Sampling |
|---|---|---|---|---|---|
| 1 | 1 | BALAKRISI | M | B | |
| 2 | 2 | DURGESH | M | A | |
| 3 | 3 | AKSHAY | M | D | |
| 4 | 4 | PRAVIN | M | C | |
| 5 | 5 | SNEHA KC | F | B | |
| 6 | 6 | SWATI | F | A | |
| 7 | 7 | ANUSHKA | F | C | |
| 8 | 8 | SOMNATI | M | A | |
| 9 | 9 | SUPRIYA | F | A | |
| 10 | 10 | MARISH | M | D | |
| 11 | 11 | SNEHAL M | F | D | |
| 12 | 12 | KARTHIK | M | B | |
| 13 | 13 | DURGESHI | F | O | |
| 14 | 14 | ANKITA | F | O | |
| 15 | 15 | TEJAS | M | B | |
| 16 | 16 | BHAVANA | F | O | |
| 17 | 17 | NAVNEET | F | O | |
| 18 | 18 | ABDUL | M | D | |
| 19 | 19 | POOJA | F | A | |
| 20 | 20 | JAYANTHI | F | D | |

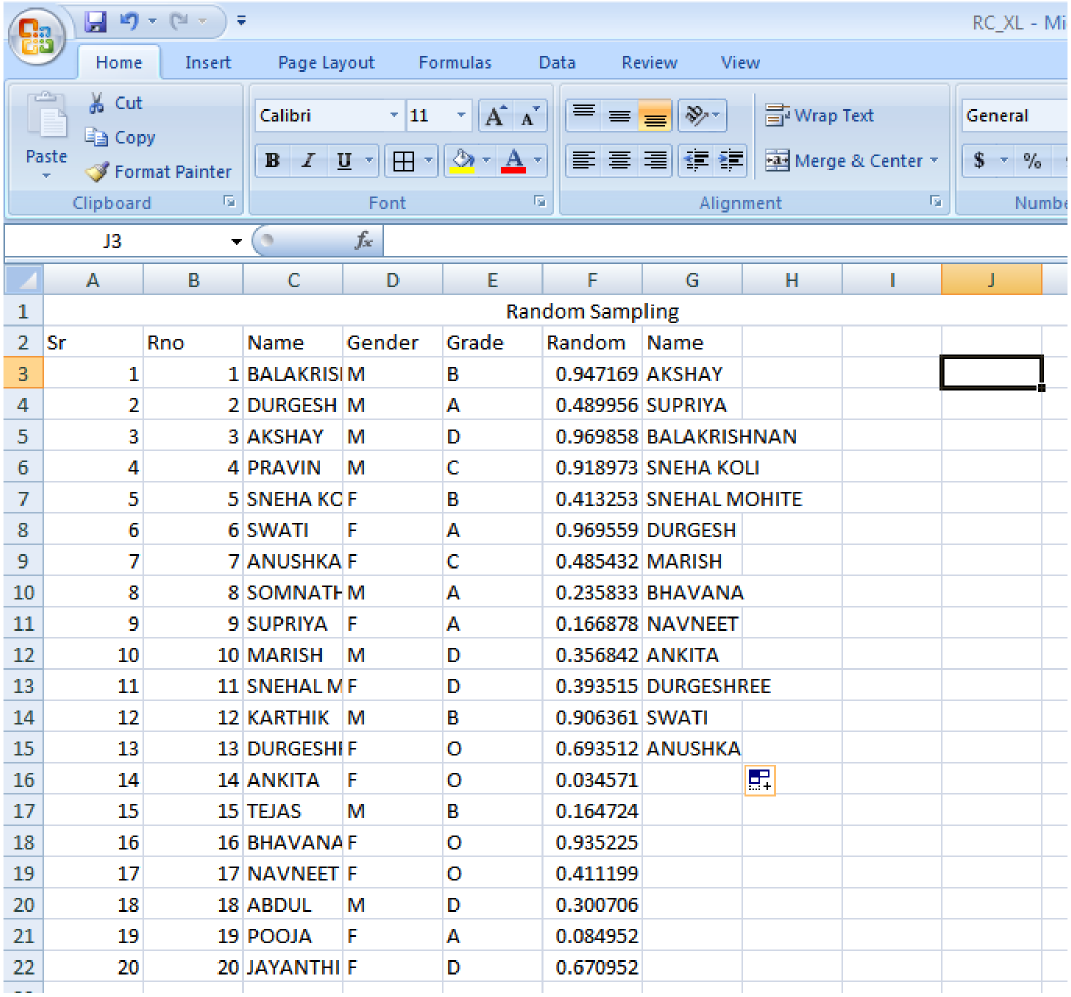
**Make another column for indexing random numbers**



=RAND()

| Sr | Rno | Name | Gender | Grade | Random |
|---|---|---|---|---|---|
| 1 | 1 | BALAKRISI | M | B | 0.30028 |
| 2 | 2 | DURGESH | M | A | 0.321283 |
| 3 | 3 | AKSHAY | M | D | 0.476679 |
| 4 | 4 | PRAVIN | M | C | 0.784103 |
| 5 | 5 | SNEHA KC | F | B | 0.00804 |
| 6 | 6 | SWATI | F | A | 0.448853 |
| 7 | 7 | ANUSHKA | F | C | 0.172203 |
| 8 | 8 | SOMNATI | M | A | 0.7575 |
| 9 | 9 | SUPRIYA | F | A | 0.963254 |
| 10 | 10 | MARISH | M | D | 0.611316 |
| 11 | 11 | SNEHAL M | F | D | 0.843969 |
| 12 | 12 | KARTHIK | M | B | 0.878059 |
| 13 | 13 | DURGESHI | F | O | 0.495925 |
| 14 | 14 | ANKITA | F | O | 0.912223 |
| 15 | 15 | TEJAS | M | B | 0.198868 |
| 16 | 16 | BHAVANA | F | O | 0.992641 |
| 17 | 17 | NAVNEET | F | O | 0.815096 |
| 18 | 18 | ABDUL | M | D | 0.206916 |
| 19 | 19 | POOJA | F | A | 0.281963 |
| 20 | 20 | JAYANTHI | F | D | 0.449576 |

**Select samples randomly using the index function=INDEX($C$3:$C$22,RANK(F3,$F$3:$F$22))**




Random Sampling

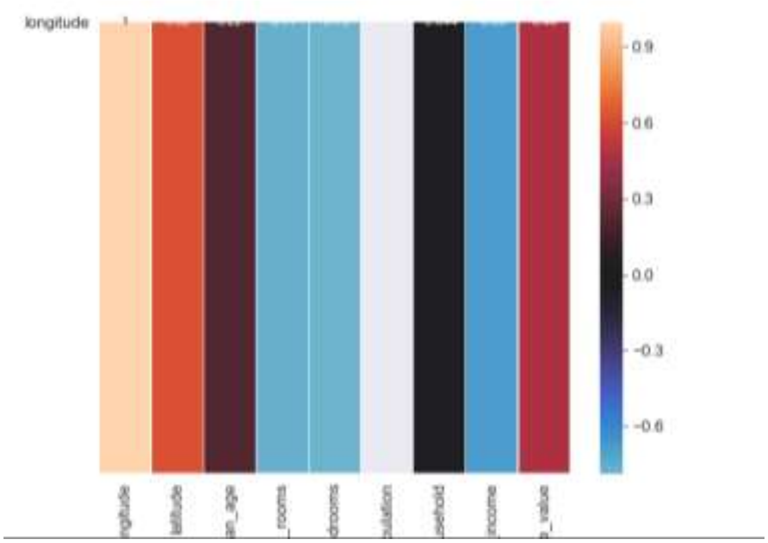
| Sr | Rno | Name | Gender | Grade | Random | Name |
|---|---|---|---|---|---|---|
| 1 | 1 | BALAKRISI | M | B | 0.947169 | AKSHAY |
| 2 | 2 | DURGESH | M | A | 0.489956 | SUPRIYA |
| 3 | 3 | AKSHAY | M | D | 0.969858 | BALAKRISHNAN |
| 4 | 4 | PRAVIN | M | C | 0.918973 | SNEHA KOLI |
| 5 | 5 | SNEHA KC | F | B | 0.413253 | SNEHAL MOHITE |
| 6 | 6 | SWATI | F | A | 0.969559 | DURGESH |
| 7 | 7 | ANUSHKA | F | C | 0.485432 | MARISH |
| 8 | 8 | SOMNATH | M | A | 0.235833 | BHAVANA |
| 9 | 9 | SUPRIYA | F | A | 0.166878 | NAVNEET |
| 10 | 10 | MARISH | M | D | 0.356842 | ANKITA |
| 11 | 11 | SNEHAL M | F | D | 0.393515 | DURGESHREE |
| 12 | 12 | KARTHIK | M | B | 0.906361 | SWATI |
| 13 | 13 | DURGESHI | F | O | 0.693512 | ANUSHKA |
| 14 | 14 | ANKITA | F | O | 0.034571 | |
| 15 | 15 | TEJAS | M | B | 0.164724 | |
| 16 | 16 | BHAVANA | F | O | 0.935225 | |
| 17 | 17 | NAVNEET | F | O | 0.411199 | |
| 18 | 18 | ABDUL | M | D | 0.300706 | |
| 19 | 19 | POOJA | F | A | 0.084952 | |
| 20 | 20 | JAYANTHI | F | D | 0.670952 | |

**B. Perform the Stratified sampling for the given data and analyse it.**

**Code:**

```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
plt.rcParams['axes.labelsize'] = 14
plt.rcParams['xtick.labelsize'] = 12
plt.rcParams['ytick.labelsize'] = 12
import seaborn as sns
color = sns.color_palette()
sns.set_style('darkgrid')
import sklearn
from sklearn.model_selection import train_test_split
housing =pd.read_csv('housing.csv')
print(housing.head())
print(housing.info())
#creating a heatmap of the attributes in the dataset
correlation_matrix = housing.corr()
plt.subplots(figsize=(8,6))
sns.heatmap(correlation_matrix, center=0, annot=True, linewidths=.3)
corr =housing.corr()
print(corr['median_house_value'].sort_values(ascending=False))
sns.distplot(housing.median_income)
plt.show()
```

**Output:**



```
   longitude  latitude  ...  median_house_value  ocean_proximity
0    -122.23     37.88  ...              452600         NEAR BAY
1    -122.22     37.86  ...              358500         NEAR BAY
2    -122.24     37.85  ...              352100         NEAR BAY
3    -122.25     37.85  ...              341300         NEAR BAY
4    -122.25     37.85  ...              342200         NEAR BAY

[5 rows x 10 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 10 columns):
longitude             5 non-null float64
latitude              5 non-null float64
housing_median_age    5 non-null int64
total_rooms           5 non-null int64
total_bedrooms        5 non-null int64
population            0 non-null float64
household             5 non-null int64
median_income         5 non-null int64
median_house_value    5 non-null int64
ocean_proximity       5 non-null object
dtypes: float64(3), int64(6), object(1)
memory usage: 444.0+ bytes
None
median_house_value    1.000000
latitude              0.972922
longitude             0.475427
total_rooms          -0.435764
median_income        -0.450045
total_bedrooms       -0.539064
household            -0.610050
housing_median_age   -0.641518
population                 NaN
Name: median_house_value, dtype: float64
```

# Practical No 8

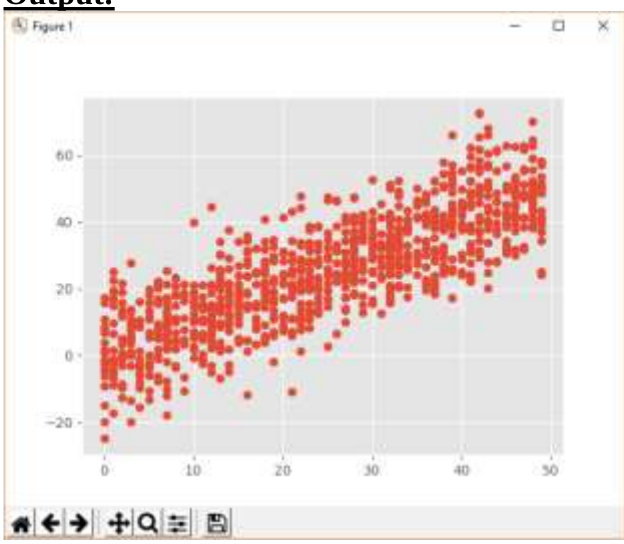
**Aim:** Write a program for computing different correlation.

**Positive Correlation:**

**Code:**

```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(1)
# 1000 random integers between 0 and 50
x = np.random.randint(0, 50, 1000)
# Positive Correlation with some noise
y = x + np.random.normal(0, 10, 1000)
np.corrcoef(x, y)
plt.style.use('ggplot')
plt.scatter(x, y)
plt.show()
```

**Output:**

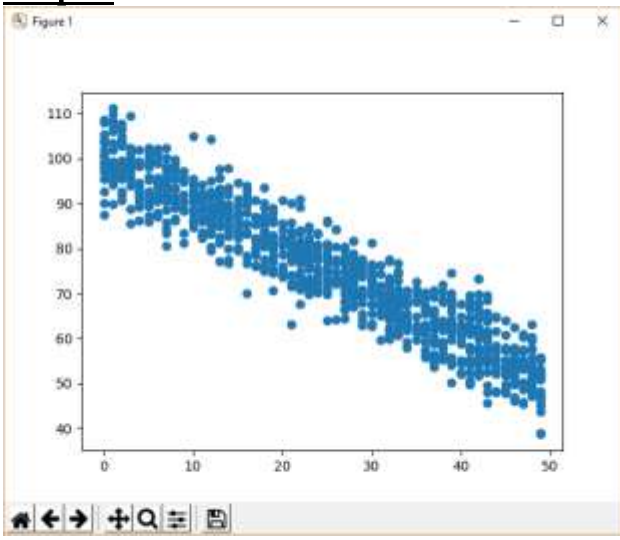



**Negative Correlation:**

**Code:**

```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(1)
# 1000 random integers between 0 and 50
x = np.random.randint(0, 50, 1000)
# Negative Correlation with some noise
y = 100 - x + np.random.normal(0, 5, 1000)
np.corrcoef(x, y)
plt.scatter(x, y)
plt.show()
```

**Output:**

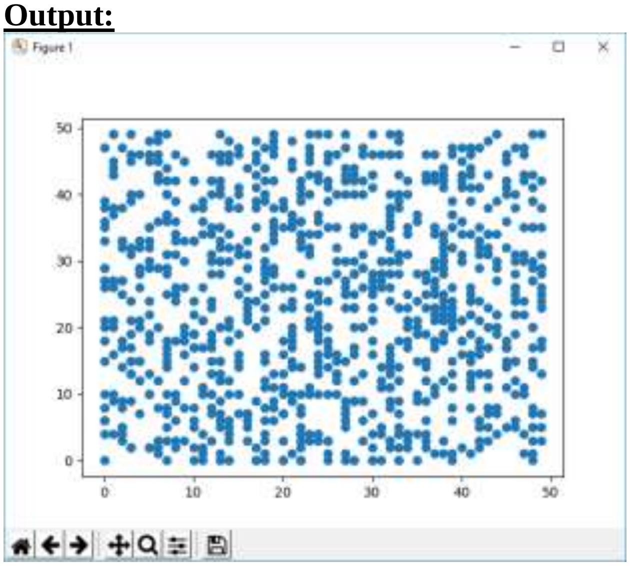



**No/Weak Correlation:**

**Code:**

```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(1)
x = np.random.randint(0, 50, 1000)
y = np.random.randint(0, 50, 1000)
np.corrcoef(x, y)
plt.scatter(x, y)
plt.show()
```

**Output:**



# Practical No 9

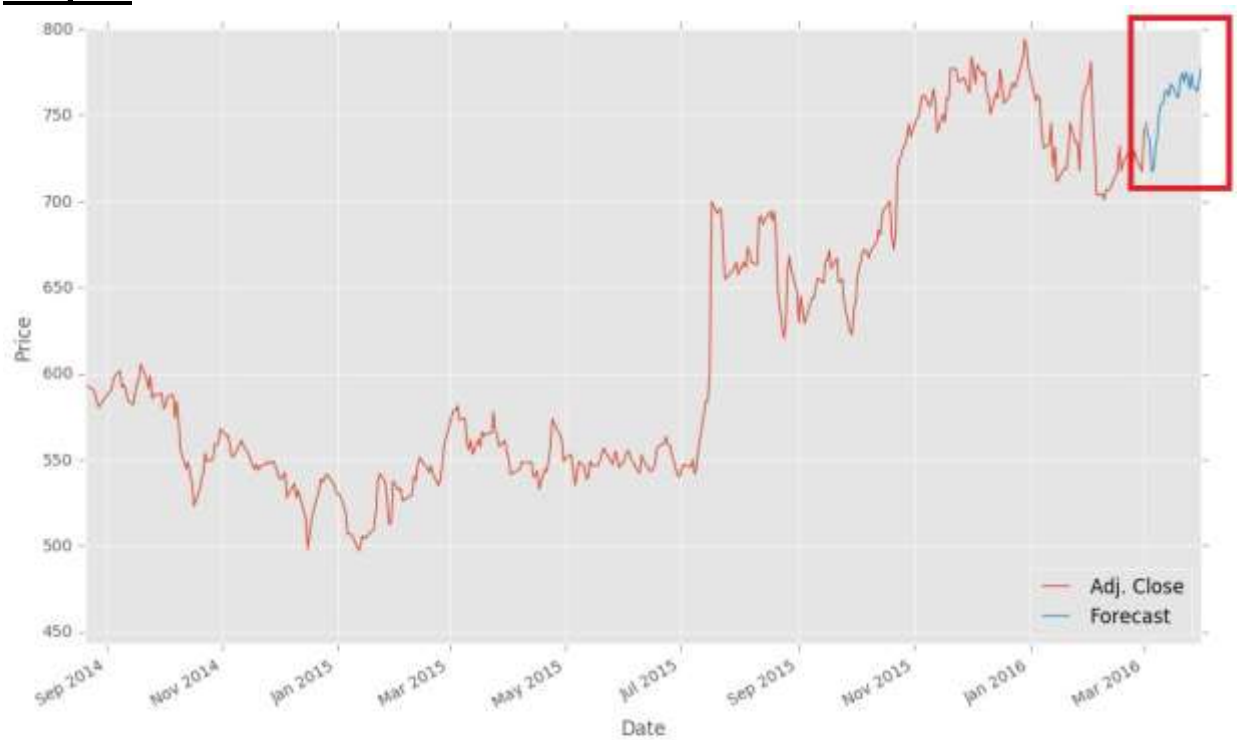
## A. Write a program to Perform linear regression for prediction.

**Code:**

```
# -*- coding: utf-8 -*-
"""
Created on Mon Dec 16 21:56:32 2019
@author: MyHome
"""
import Quandl, math
import numpy as np
import pandas as pd
from sklearn import preprocessing, cross_validation, svm
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
from matplotlib import style
import datetime
style.use('ggplot')
df = Quandl.get("WIKI/GOOGL")
df = df[['Adj. Open', 'Adj. High', 'Adj. Low', 'Adj. Close', 'Adj. Volume']]
df['HL_PCT'] = (df['Adj. High'] - df['Adj. Low']) / df['Adj. Close'] * 100.0
df['PCT_change'] = (df['Adj. Close'] - df['Adj. Open']) / df['Adj. Open'] * 100.0
df = df[['Adj. Close', 'HL_PCT', 'PCT_change', 'Adj. Volume']]
forecast_col = 'Adj. Close'
df.fillna(value=-99999, inplace=True)
forecast_out = int(math.ceil(0.01 * len(df)))
df['label'] = df[forecast_col].shift(-forecast_out)
X = np.array(df.drop(['label'], 1))
X = preprocessing.scale(X)
X_lately = X[-forecast_out:]
X = X[:-forecast_out]
df.dropna(inplace=True)
y = np.array(df['label'])
X_train, X_test, y_train, y_test = cross_validation.train_test_split(X, y,
test_size=0.2)
clf = LinearRegression(n_jobs=-1)
clf.fit(X_train, y_train)
confidence = clf.score(X_test, y_test)
forecast_set = clf.predict(X_lately)
```

```
df['Forecast'] = np.nan
last_date = df.iloc[-1].name
last_unix = last_date.timestamp()
one_day = 86400
next_unix = last_unix + one_day
for i in forecast_set:
next_date = datetime.datetime.fromtimestamp(next_unix)
next_unix += 86400
df.loc[next_date] = [np.nan for _ in range(len(df.columns)-1)]+[i]
df['Adj. Close'].plot()
df['Forecast'].plot()
plt.legend(loc=4)
plt.xlabel('Date')
plt.ylabel('Price')
plt.show()
```

**Output:**

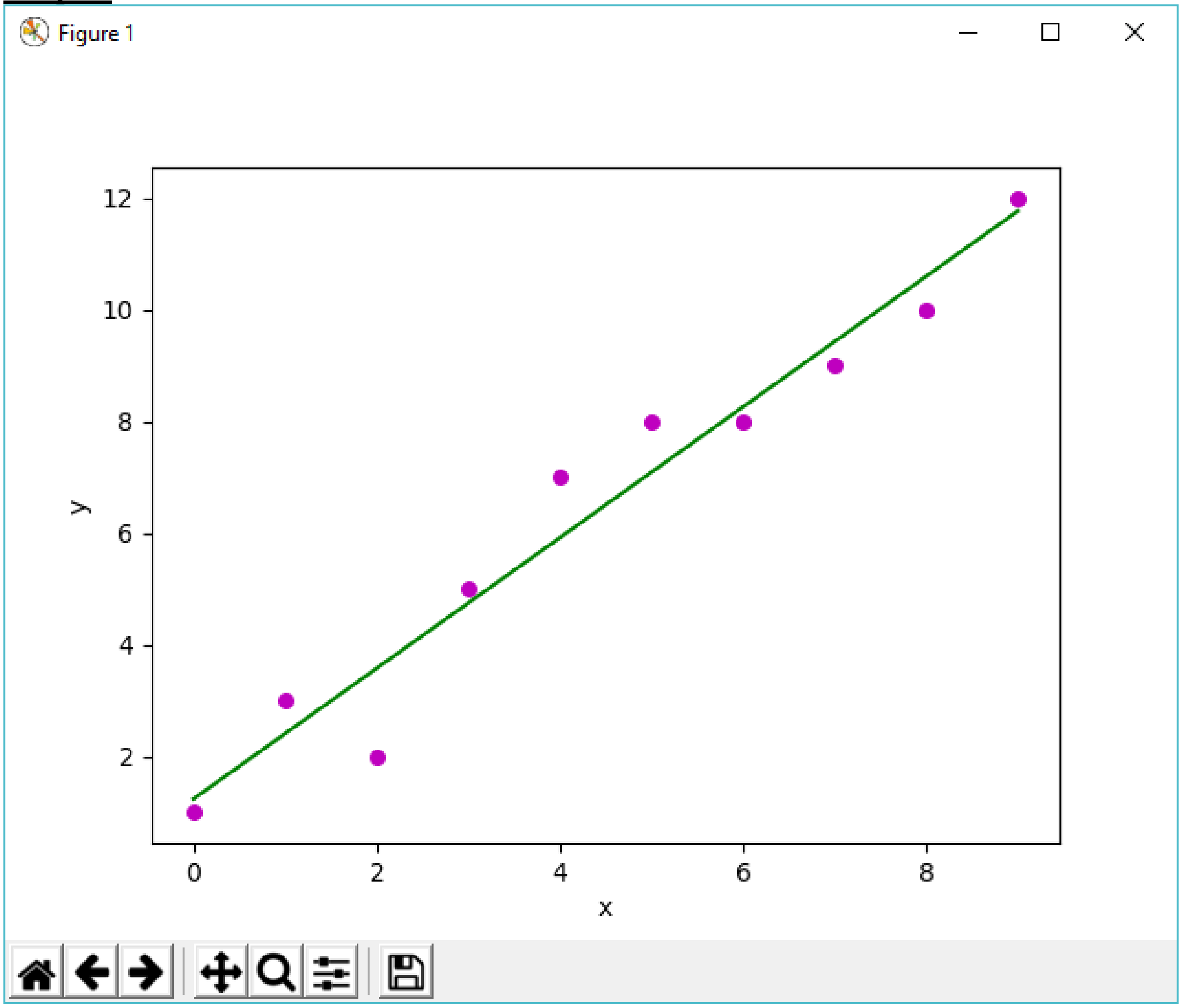


## B. Perform polynomial regression for prediction.

**Code:**

```
import numpy as np
import matplotlib.pyplot as plt
def estimate_coef(x, y):
    #number of observations/points
    n = np.size(x)
    # mean of x and y vector
    m_x, m_y = np.mean(x), np.mean(y)
    # calculating cross-deviation and deviation about x
    SS_xy = np.sum(y*x) - n*m_y*m_x
    SS_xx = np.sum(x*x) - n*m_x*m_x
    # calculating regression coefficients
    b_1 = SS_xy / SS_xx
    b_0 = m_y - b_1*m_x
    return(b_0, b_1)
def plot_regression_line(x, y, b):
    # plotting the actual points as scatter plot
    plt.scatter(x, y, color = "m",
    marker = "o", s = 30)
    # predicted response vector
    y_pred = b[0] + b[1]*x
    # plotting the regression line
    plt.plot(x, y_pred, color = "g")
    # putting labels
    plt.xlabel('x')
    plt.ylabel('y')
    # function to show plot
    plt.show()
def main():
    # observations
    x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
    y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])
    # estimating coefficients
    b = estimate_coef(x, y)
    print("Estimated coefficients:\nb_0 = {} b_1 = {}".format(b[0], b[1]))
    # plotting regression line
    plot_regression_line(x, y, b)
if __name__ == "__main__":
    main()
```

**Output:**




```
 RESTART: D:/D-Drive/SIES/Msc IT/Msc IT (Part-1)/Practicals/Research In Computin
g Material/9_2.py
Estimated coefficients:
b_0 = 1.2363636363636363 b_1 = 1.1696969696969697
>>>
```

# Practical No 10

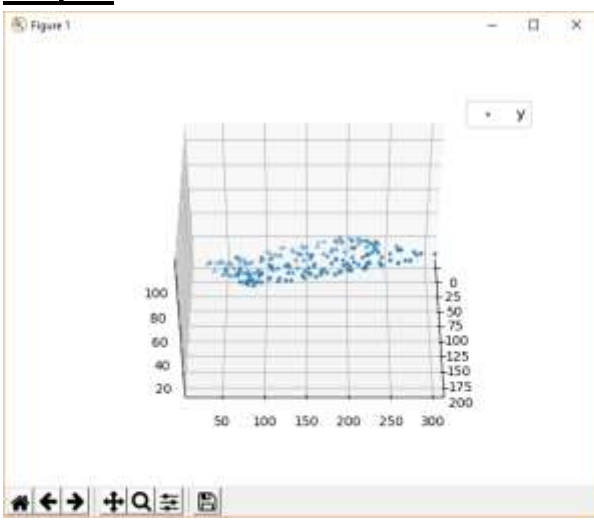
## A. **Write a program for multiple linear regression analysis.**

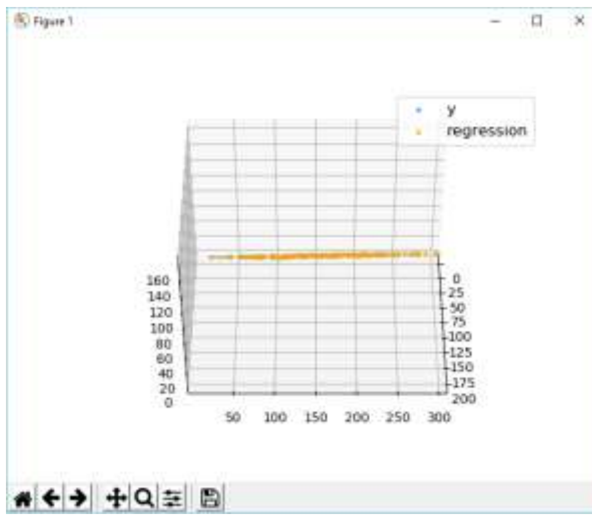
**Code:**

```
import numpy as np
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
def generate_dataset(n):
    x = []
    y = []
    random_x1 = np.random.rand()
    random_x2 = np.random.rand()
    for i in range(n):
        x1 = i
        x2 = i/2 + np.random.rand()*n
        x.append([1, x1, x2])
        y.append(random_x1 * x1 + random_x2 * x2 + 1)
    return np.array(x), np.array(y)
x, y = generate_dataset(200)
mpl.rcParams['legend.fontsize'] = 12
fig = plt.figure()
ax = fig.gca(projection ='3d')
ax.scatter(x[:, 1], x[:, 2], y, label ='y', s = 5)
ax.legend()
ax.view_init(45, 0)
plt.show()
def mse(coef, x, y):
    return np.mean((np.dot(x, coef) - y)**2)/2
def gradients(coef, x, y):
    return np.mean(x.transpose()*(np.dot(x, coef) - y), axis = 1)
def multilinear_regression(coef, x, y, lr, b1 = 0.9, b2 = 0.999, epsilon = 1e-8):
    prev_error = 0
    m_coef = np.zeros(coef.shape)
    v_coef = np.zeros(coef.shape)
    moment_m_coef = np.zeros(coef.shape)
    moment_v_coef = np.zeros(coef.shape)
    t = 0
    while True:
        error = mse(coef, x, y)
        if abs(error - prev_error) <= epsilon:
            break
        prev_error = error
        grad = gradients(coef, x, y)
        t += 1
        m_coef = b1 * m_coef + (1-b1)*grad
        v_coef = b2 * v_coef + (1-b2)*grad**2
```

```
        moment_m_coef = m_coef / (1-b1**t)
        moment_v_coef = v_coef / (1-b2**t)
        delta = ((lr / moment_v_coef**0.5 + 1e-8) *(b1 * moment_m_coef + (1-b1)*grad/(1-
b1**t)))
        coef = np.subtract(coef, delta)
    return coef
coef = np.array([0, 0, 0])
c = multilinear_regression(coef, x, y, 1e-1)
fig = plt.figure()
ax = fig.gca(projection ='3d')
ax.scatter(x[:, 1], x[:, 2], y, label ='y',
s = 5, color ="dodgerblue")
ax.scatter(x[:, 1], x[:, 2], c[0] + c[1]*x[:, 1] + c[2]*x[:, 2],
label ='regression', s = 5, color ="orange")
ax.view_init(45, 0)
ax.legend()
plt.show()
```

**Output:**



## B. Perform logistic regression analysis.

**Code:**

```
import os
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import scipy.stats as stats
from sklearn import linear_model
from sklearn import preprocessing
from sklearn import metrics
matplotlib.style.use('ggplot')
plt.figure(figsize=(9,9))
def sigmoid(t): # Define the sigmoid function
return (1/(1 + np.e**(-t)))
plot_range = np.arange(-6, 6, 0.1)
y_values = sigmoid(plot_range)
# Plot curve
plt.plot(plot_range, # X-axis range
y_values, # Predicted values
color="red")
titanic_train = pd.read_csv("titanic_train.csv") # Read the data
char_cabin = titanic_train["Cabin"].astype(str) # Convert cabin to str
new_Cabin = np.array([cabin[0] for cabin in char_cabin]) # Take first letter
titanic_train["Cabin"] = pd.Categorical(new_Cabin) # Save the new cabin var
# Impute median Age for NA Age values
new_age_var = np.where(titanic_train["Age"].isnull(), # Logical check
28, # Value if check is true

titanic_train["Age"]) # Value if check is false
titanic_train["Age"] = new_age_var
label_encoder = preprocessing.LabelEncoder()
# Convert Sex variable to numeric
encoded_sex = label_encoder.fit_transform(titanic_train["Sex"])
# Initialize logistic regression model
log_model = linear_model.LogisticRegression()
# Train the model
log_model.fit(X = pd.DataFrame(encoded_sex),
y = titanic_train["Survived"])
# Check trained model intercept
print(log_model.intercept_)
# Check trained model coefficients
print(log_model.coef_)
# Make predictions
```

```
preds = log_model.predict_proba(X= pd.DataFrame(encoded_sex))
preds = pd.DataFrame(preds)
preds.columns = ["Death_prob", "Survival_prob"]
# Generate table of predictions vs Sex
pd.crosstab(titanic_train["Sex"], preds.ix[:, "Survival_prob"])
# Convert more variables to numeric
encoded_class = label_encoder.fit_transform(titanic_train["Pclass"])
encoded_cabin = label_encoder.fit_transform(titanic_train["Cabin"])
train_features = pd.DataFrame([encoded_class,
encoded_cabin,
encoded_sex,
titanic_train["Age"]]).T
# Initialize logistic regression model
log_model = linear_model.LogisticRegression()

# Train the model
log_model.fit(X = train_features ,
y = titanic_train["Survived"])
# Check trained model intercept
print(log_model.intercept_)
# Check trained model coefficients
print(log_model.coef_)
# Make predictions
preds = log_model.predict(X= train_features)
# Generate table of predictions vs actual
pd.crosstab(preds,titanic_train["Survived"])
log_model.score(X = train_features ,
y = titanic_train["Survived"])
metrics.confusion_matrix(y_true=titanic_train["Survived"], # True labels
y_pred=preds) # Predicted labels
# View summary of common classification metrics
print(metrics.classification_report(y_true=titanic_train["Survived"],
y_pred=preds) )
# Read and prepare test data
titanic_test = pd.read_csv("titanic_test.csv") # Read the data
char_cabin = titanic_test["Cabin"].astype(str) # Convert cabin to str
new_Cabin = np.array([cabin[0] for cabin in char_cabin]) # Take first letter
titanic_test["Cabin"] = pd.Categorical(new_Cabin) # Save the new cabin var
# Impute median Age for NA Age values
new_age_var = np.where(titanic_test["Age"].isnull(), # Logical check
28, # Value if check is true
titanic_test["Age"]) # Value if check is false

titanic_test["Age"] = new_age_var
# Convert test variables to match model features
encoded_sex = label_encoder.fit_transform(titanic_test["Sex"])
encoded_class = label_encoder.fit_transform(titanic_test["Pclass"])
```

```
encoded_cabin = label_encoder.fit_transform(titanic_test["Cabin"])
test_features = pd.DataFrame([encoded_class,
encoded_cabin,encoded_sex,titanic_test["Age"]]).T
# Make test set predictions
test_preds = log_model.predict(X=test_features)
# Create a submission for Kaggle
submission = pd.DataFrame({"PassengerId":titanic_test["PassengerId"],
"Survived":test_preds})
# Save submission to CSV
submission.to_csv("tutorial_logreg_submission.csv",
index=False) # Do not save index values
print(pd)
```

**Output:**

| Survival_prob | 0.193110906347 | 0.729443792051 |
|---|---|---|
| Sex | | |
| female | 0 | 312 |
| male | 577 | 0 |

```
             precision    recall  f1-score   support

          0       0.82      0.85      0.83       549
          1       0.74      0.70      0.72       340

avg / total       0.79      0.79      0.79       889
```

| Survived | 0 | 1 |
|---|---|---|
| row_0 | | |
| 0 | 467 | 103 |
| 1 | 82 | 237 |

## Example 2:

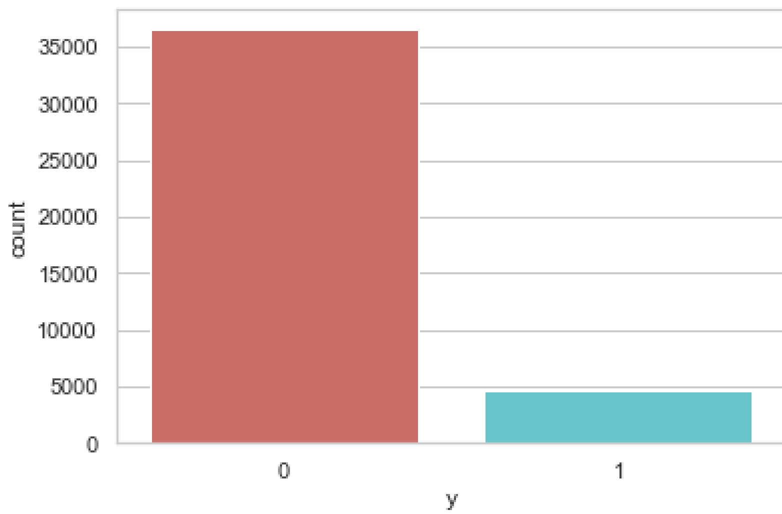
**Code:**

```
# -*- coding: utf-8 -*-
"""
Created on Mon Dec 16 22:24:44 2019
@author: MyHome
"""
import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)
data = pd.read_csv('bank.csv', header=0)
data = data.dropna()
print(data.shape)
print(list(data.columns))
data['education'].unique()
data['education']=np.where(data['education'] =='basic.9y', 'Basic', data['education'])
data['education']=np.where(data['education'] =='basic.6y', 'Basic', data['education'])
data['education']=np.where(data['education'] =='basic.4y', 'Basic', data['education'])
data['education'].unique()
data['y'].value_counts()
sns.countplot(x='y', data=data, palette='hls')
plt.show();
plt.savefig('Practical10B-plot.jpeg')
count_no_sub = len(data[data['y']==0])
count_sub = len(data[data['y']==1])
pct_of_no_sub = count_no_sub/(count_no_sub+count_sub)
print("percentage of no subscription is", pct_of_no_sub*100)
pct_of_sub = count_sub/(count_no_sub+count_sub)
print("percentage of subscription", pct_of_sub*100)
data.groupby('y').mean()
data.groupby('job').mean()
data.groupby('marital').mean()
data.groupby('education').mean()
########### Purchase Frequency for Job Title
pd.crosstab(data.job,data.y).plot(kind='bar')
plt.title('Purchase Frequency for Job Title')
plt.xlabel('Job')
plt.ylabel('Frequency of Purchase')
```

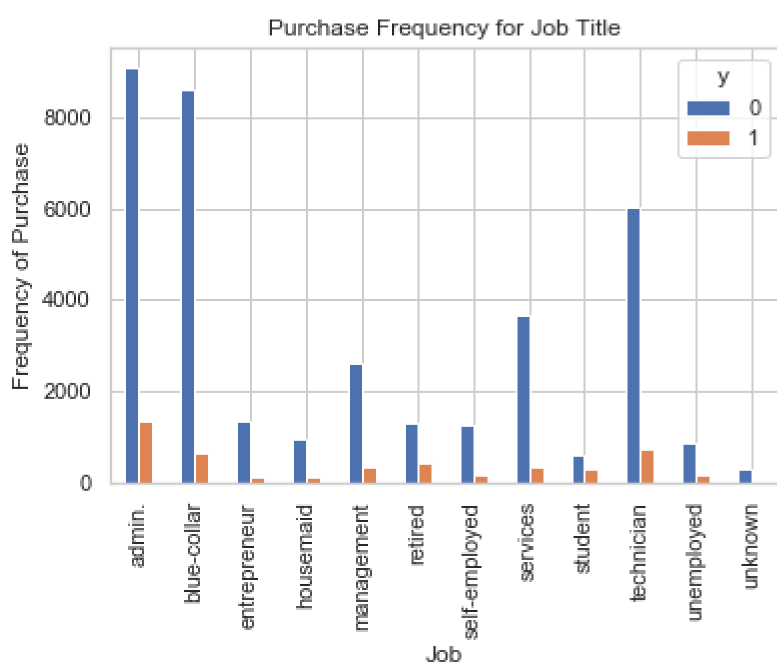
```
plt.savefig('purchase_fre_job')
###################### Marital Status vs Purchase
table=pd.crosstab(data.marital,data.y)
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Stacked Bar Chart of Marital Status vs Purchase')
plt.xlabel('Marital Status')
plt.ylabel('Proportion of Customers')
plt.savefig('mariral_vs_pur_stack')
############ Education vs Purchase
table=pd.crosstab(data.education,data.y)
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True)
plt.title('Stacked Bar Chart of Education vs Purchase')

plt.xlabel('Education')
plt.ylabel('Proportion of Customers')
plt.savefig('edu_vs_pur_stack')
pd.crosstab(data.day_of_week,data.y).plot(kind='bar')
plt.title('Purchase Frequency for Day of Week')
plt.xlabel('Day of Week')
plt.ylabel('Frequency of Purchase')
plt.savefig('pur_dayofweek_bar')
############ Purchase Frequency for Month
pd.crosstab(data.month,data.y).plot(kind='bar')
plt.title('Purchase Frequency for Month')
plt.xlabel('Month')
plt.ylabel('Frequency of Purchase')
plt.savefig('pur_fre_month_bar')
########### Age Purchase frequency pattern
data.age.hist()
plt.title('Histogram of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.savefig('hist_age')
```

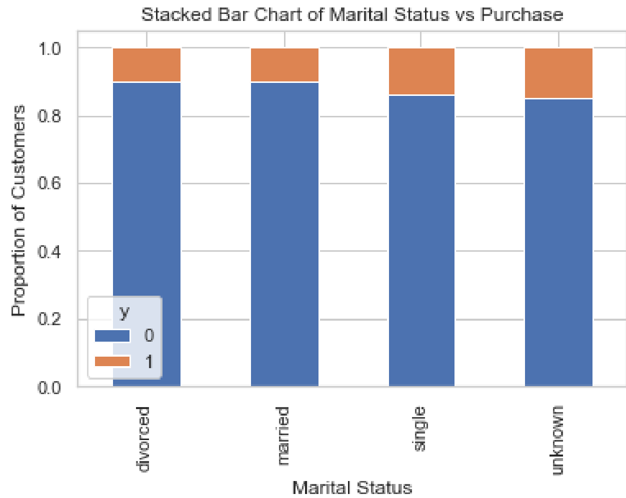
## **Output:**

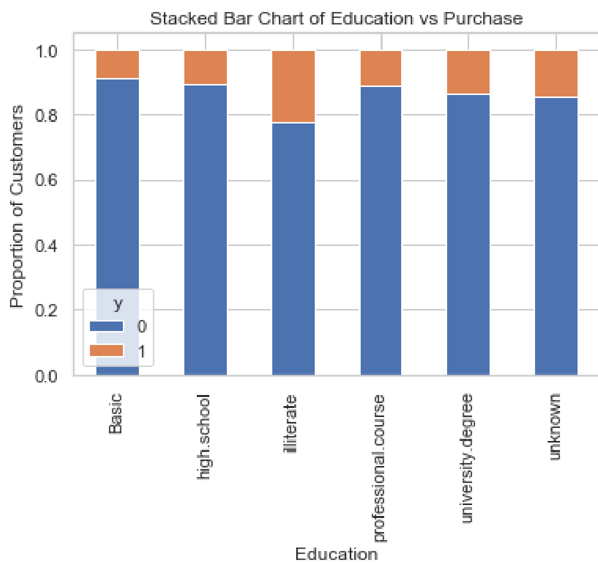
```
In [47]: runfile('K:/Research In Computing/Practical Material/Progra
Practical_10/Practical_10B.py', wdir='K:/Research In Computing/Pract
Material/Programs/Practical_10')
(41188, 21)
['age', 'job', 'marital', 'education', 'default', 'housing', 'loan',
'contact', 'month', 'day_of_week', 'duration', 'campaign', 'pdays',
'previous', 'poutcome', 'emp_var_rate', 'cons_price_idx',
'cons_conf_idx', 'euribor3m', 'nr_employed', 'y']
```

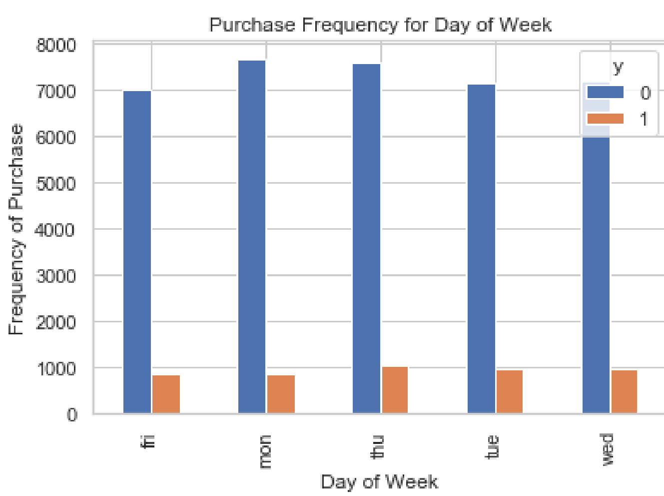


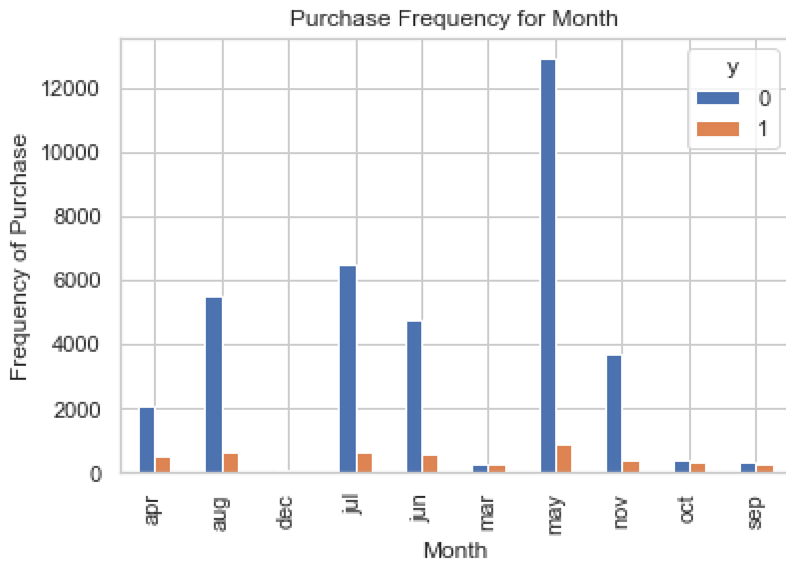

```
percentage of no subscription is 88.73458288821988
percentage of subscription 11.265417111780131
```

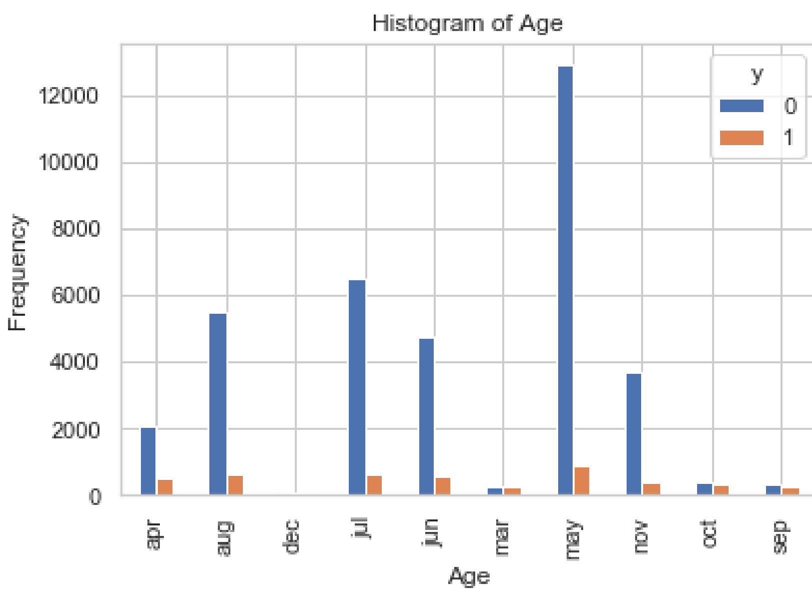



Stacked Bar Chart of Marital Status vs Purchase
Proportion of Customers
1.0
0.8
0.6
0.4
0.2
0.0
y
0
1
divorced
married
single
unknown
Marital Status


Stacked Bar Chart of Education vs Purchase
Proportion of Customers
1.0
0.8
0.6
0.4
0.2
0.0
y
0
1
Basic
high.school
illiterate
professional.course
university.degree
unknown
Education


Purchase Frequency for Day of Week
Frequency of Purchase
8000
7000
6000
5000
4000
3000
2000
1000
0
y
0
1
fri
mon
thu
tue
wed
Day of Week


Purchase Frequency for Month
Frequency of Purchase
Month
y
0
1
12000
10000
8000
6000
4000
2000
0
apr
aug
dec
jul
jun
mar
may
nov
oct
sep


Histogram of Age
Frequency
Age
y
0
1
12000
10000
8000
6000
4000
2000
0
apr
aug
dec
jul
jun
mar
may
nov
oct
sep