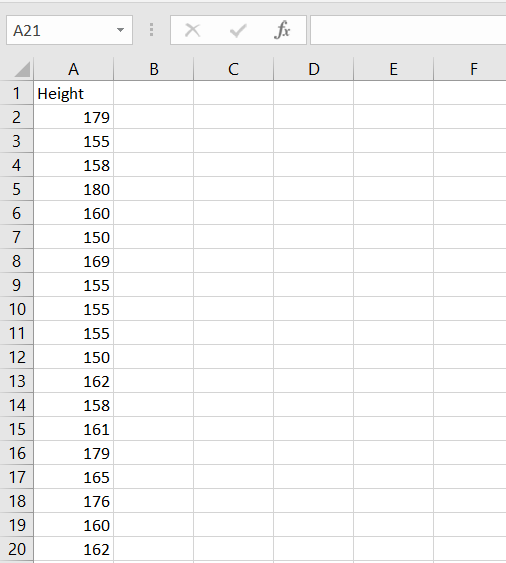
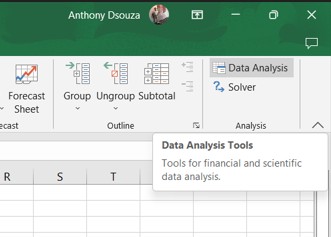
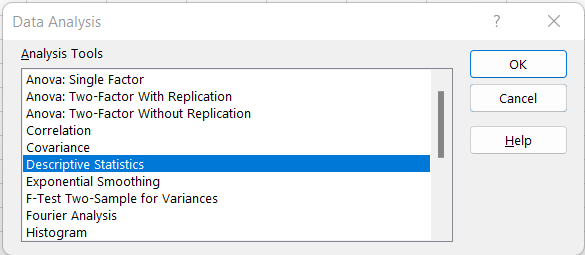
# Practical 1

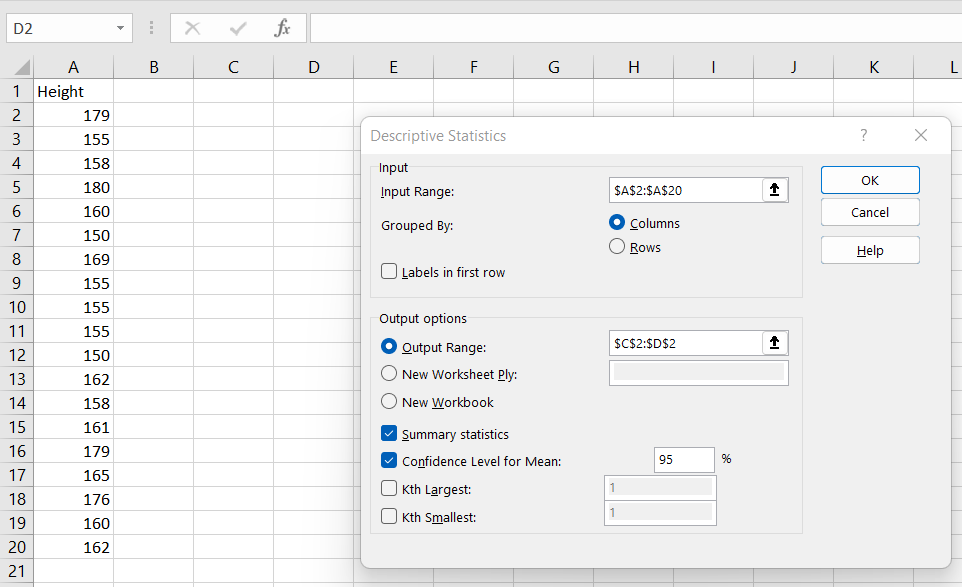
## Write a program for obtaining descriptive statistics of data.

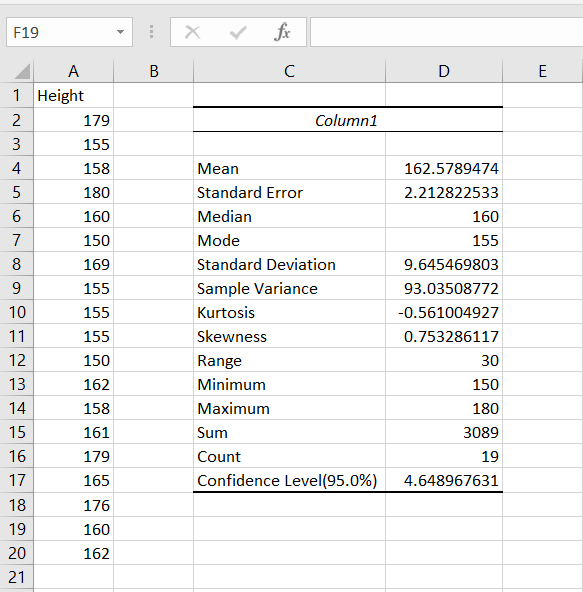
Step 1: Open your data in excel

Step 2: From the Data Tool in the ribbon choose Data Analysis.

Step 3: Select the option of descriptive statistics



Step 4: Select an input range, output range, and check summary statistics and confidence level.

Step 5: The output may appear as follows.

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

### Code:

SQLite3

import sqlite3 as sq import pandas as pd

############################################ Base='C:/VKHCG'

sDatabaseName=Base +

'/01-Vermeulen/04-Transform//SQLite/vermeulen.db' conn = sq.connect(sDatabaseName) ############################################

sFileName='C:/VKHCG/01-Vermeulen/01-Retrieve/01-EDS/02-Python/Retri eve\_IP\_DATA.csv'

print('Loading :',sFileName) IP\_DATA\_ALL\_FIX=pd.read\_csv(sFileName,header=0,low\_memory=False

)

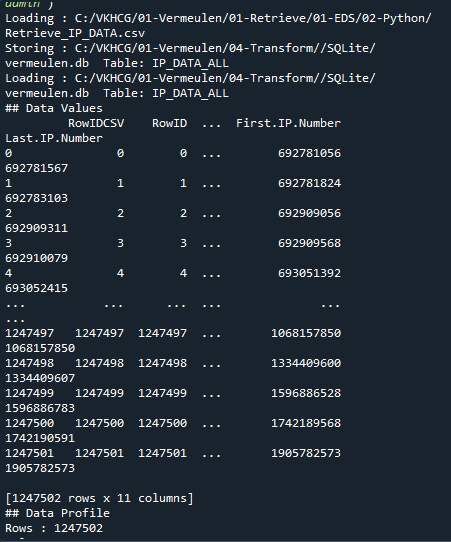
IP\_DATA\_ALL\_FIX.index.names = ['RowIDCSV'] sTable='IP\_DATA\_ALL'

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('## Data Values')

print(TestData) print('## Data Profile')

print('Rows :',TestData.shape[0]) print('Columns :',TestData.shape[1]) print('Successful')

### Output:



Excel import os

import pandas as pd Base='C:/VKHCG'

sFileDir=Base + '/01-Vermeulen/01-Retrieve/01-EDS/02-Python'

CurrencyRawData =

pd.read\_excel('C:/VKHCG/01-Vermeulen/00-RawData/Country\_Currency.xl sx')

sColumns = ['Country or territory', 'Currency', 'ISO-4217'] CurrencyData = CurrencyRawData[sColumns]

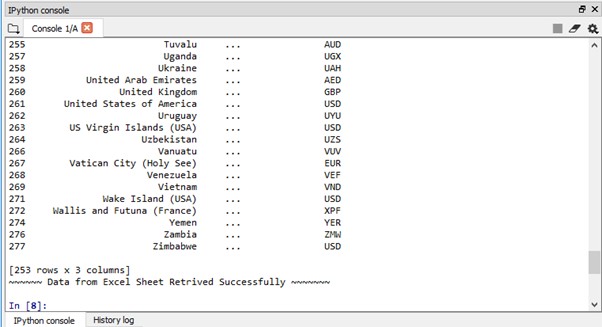
CurrencyData.rename(columns={'Country or territory': 'Country', 'ISO-4217':

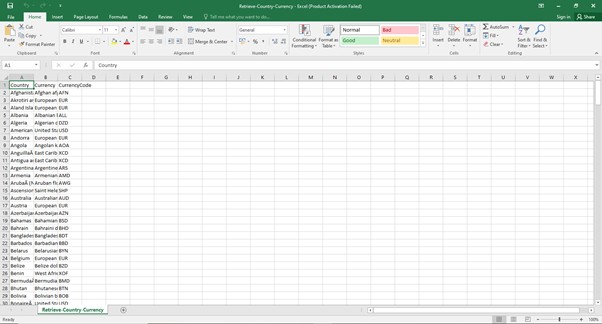
'CurrencyCode'}, inplace=True) CurrencyData.dropna(subset=['Currency'],inplace=True) CurrencyData['Country'] = CurrencyData['Country'].map(lambda x: x.strip()) CurrencyData['Currency'] = CurrencyData['Currency'].map(lambda x: x.strip())

CurrencyData['CurrencyCode'] = CurrencyData['CurrencyCode'].map(lambda x:

x.strip()) print(CurrencyData)

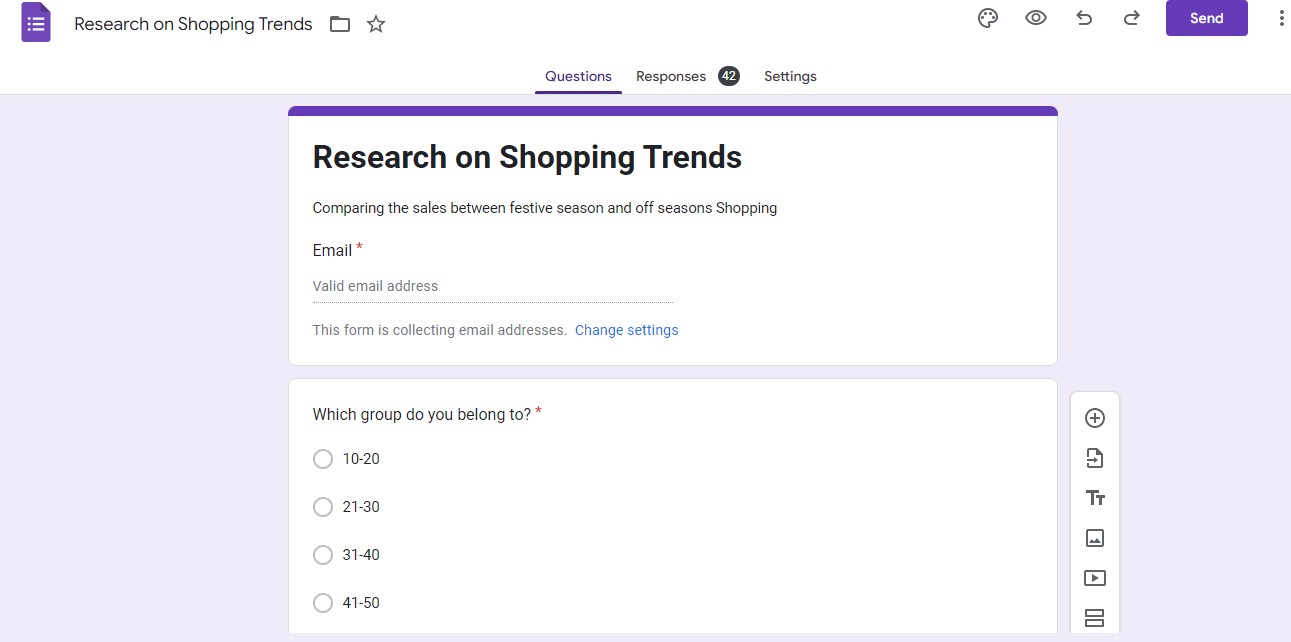
print('~~~~~~ Data from Excel Sheet Retrived Successfully ~~~~~~~') sFileName=sFileDir + '/Retrieve-Country-Currency.csv' CurrencyData.to\_csv(sFileName, index = False)





# Practical 2

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

Step 1: In order to make a survey we used Google forms. Our case study’s aim was to find out which age group did Shopping in festive seasons.

Step 2: The questions and options entered were as follows. Q1.Which group do you belong to?

(a)10-20 (b)21-30 (c)31-40 (d)41-50 (e)51 Above

Q2.What is your Gender? (a)Male (b)Female

Q3.What is your Occupation?

(a)Business (b)Student (c)Employee (d)Retired (e)Self Employed (f)Household

Q4.What's your Monthly Income?

(a)10,000 - 20,000 (b)21,000 - 30,000 (c)31,000 - 40,000 (d)41,000 and

above (e)N/A

Q5.How often do you shop?

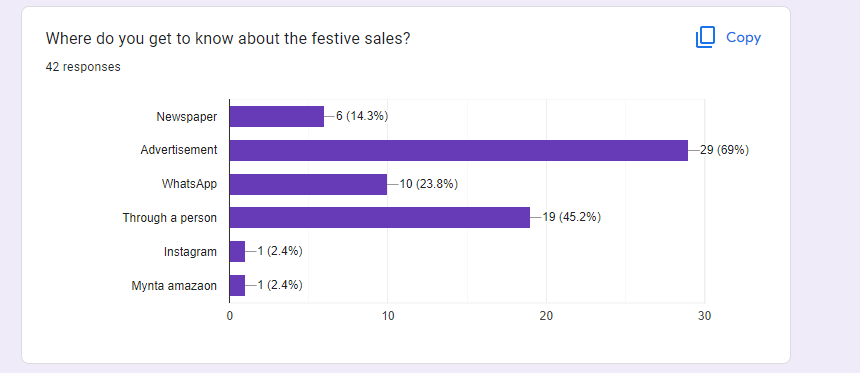
(a)Monthly (b)Occasionaly (c)Festive Seasons (d)Rarely

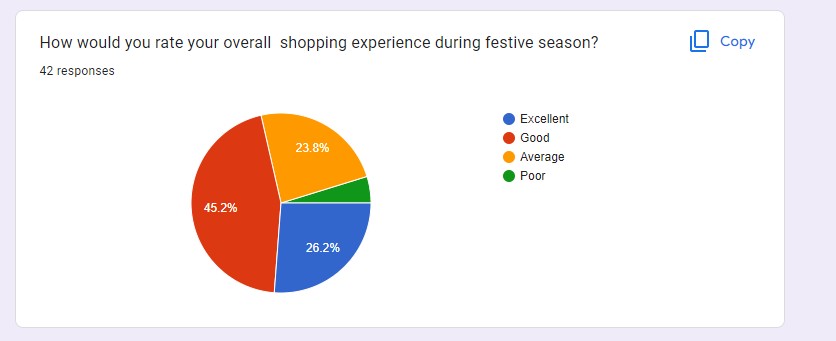
Q6.Purpose of Shopping? (a)Personal Use (b)Gifting

Q7.Where do you get to know about the festive sales? (a)Newspaper (b)Advertisement (c)WhatsApp (d)Through a person

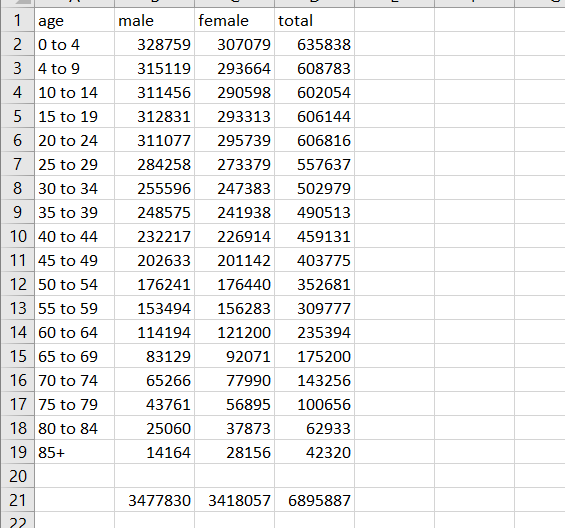
Q8.How would you rate your overall shopping experience during festive season?

(a)Excellent (b)Good (c)Average (d)Poor

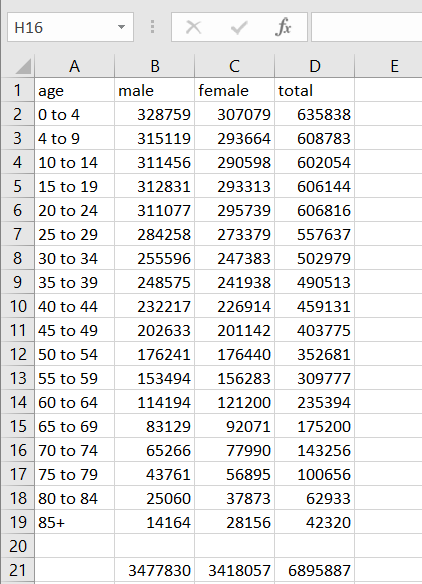
Step 3 :Google forms tool was chosen because analysis becomes easier on Google forms.



## Perform analysis of given secondary data.

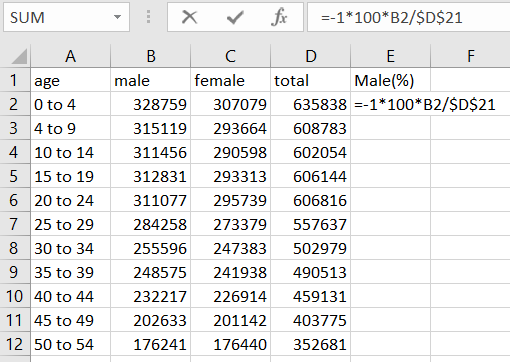
Step 1: Open Data in excel

Step 2: Calculate the total sum of each column. Select the cell for Sum→ add formula SUM in formula bar→ select the range.

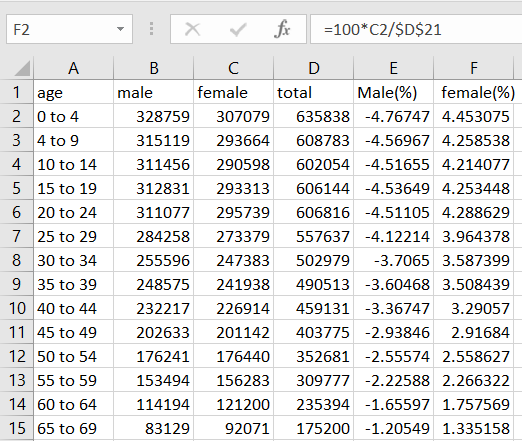


Step 3: Calculate the percentage of male in cell E. Use formula

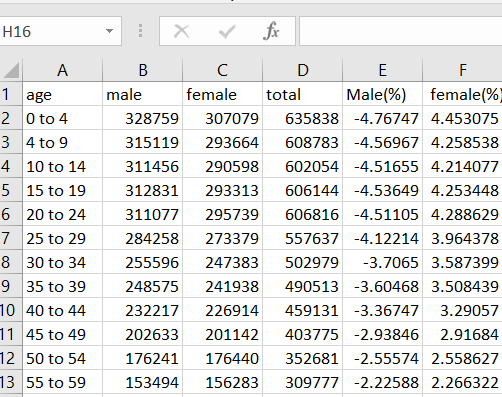
-1\*100\*B2/$D$21



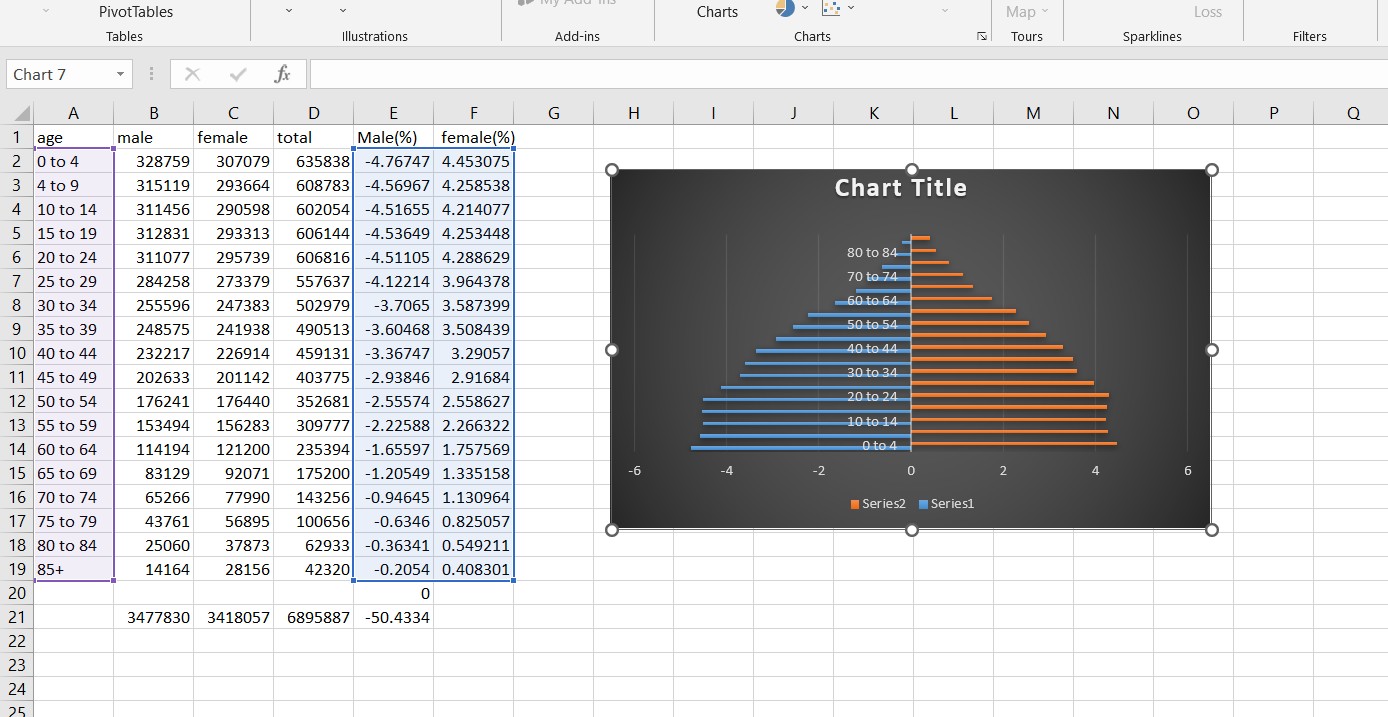
Step 4: Calculate the percentage female in cell F. Use formula 100\*C2/$D$21



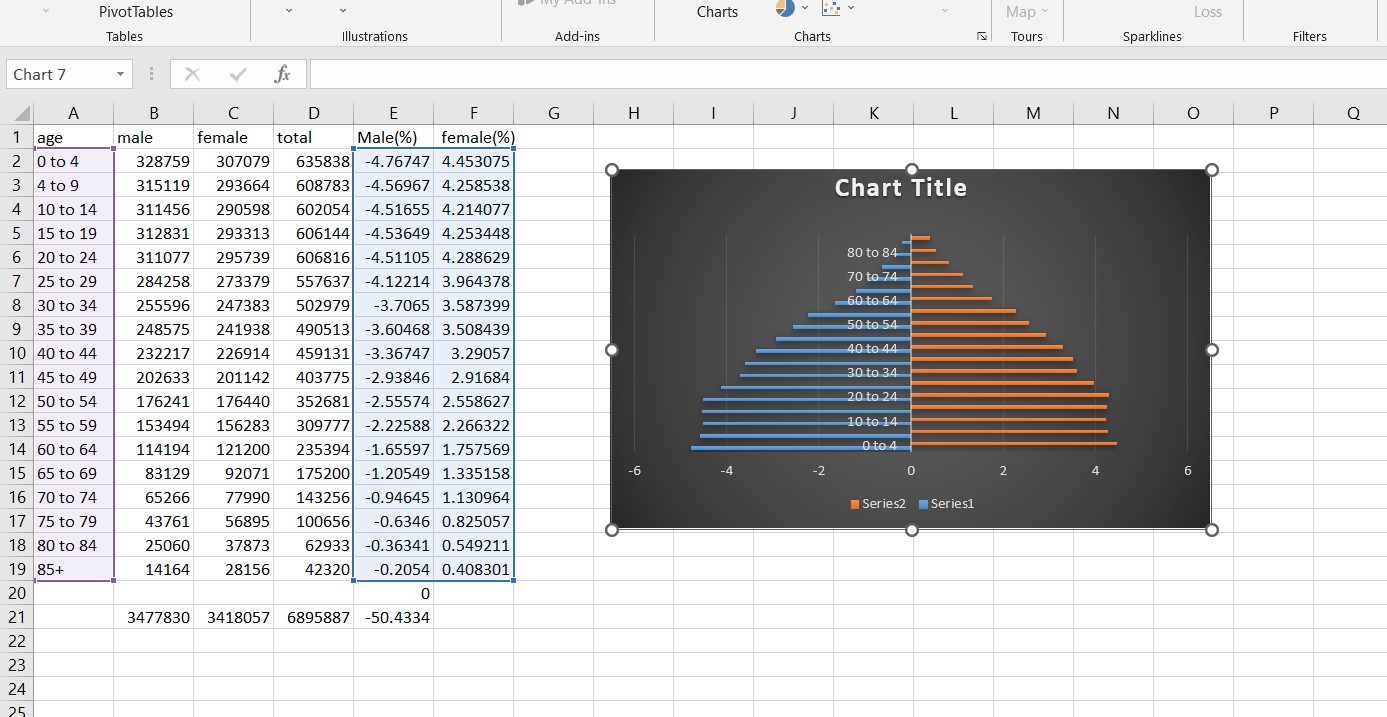
Step 5: The modified data may seem as follows.



Step 6: For analysis go to Insert → Bar → 2D



Step 7 :Drag the data and set the graph for analysis.



# Practical 3

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

### Code:

from scipy.stats import ttest\_1samp import numpy as np ages=np.genfromtxt('H:/ages.csv') print(ages) ages\_mean=np.mean(ages) print(ages\_mean) tset,pval=ttest\_1samp(ages,30) print('p-values-',pval)

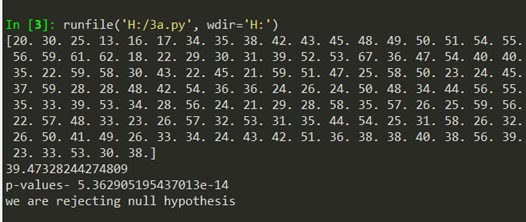
if pval<0.05:#alpha value is 0.05

print("we are rejecting null hypothesis")

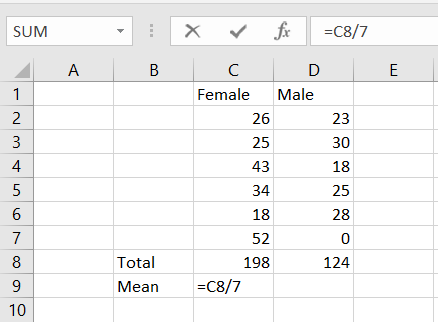
else:

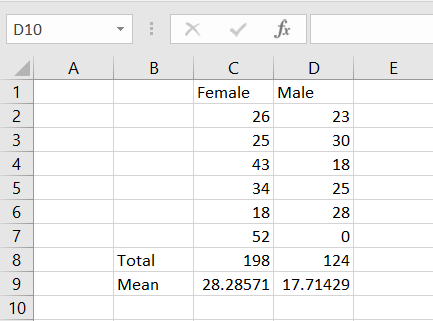
print("we are accepeting null hypothesis")

**Output:**

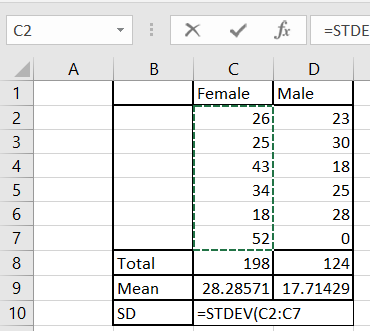


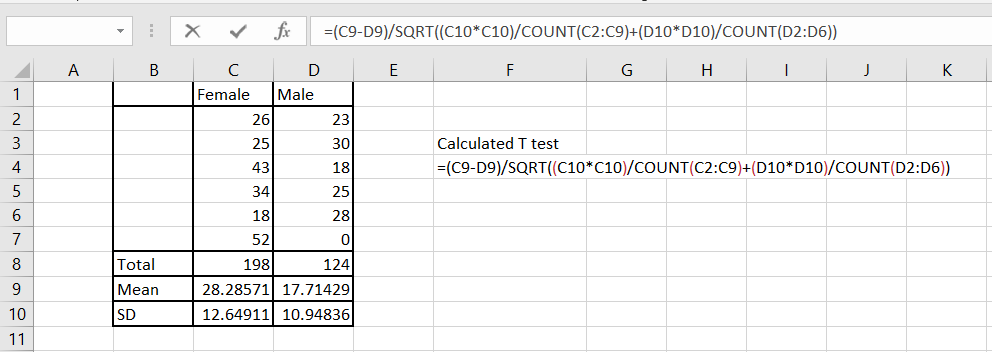
## Write a program for t-test comparing two means for independent samples.

Step 1: Calculate the mean of the samples (Total /Count)

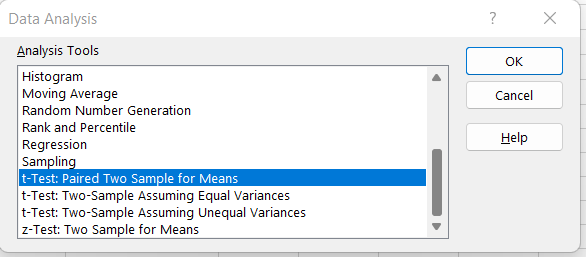


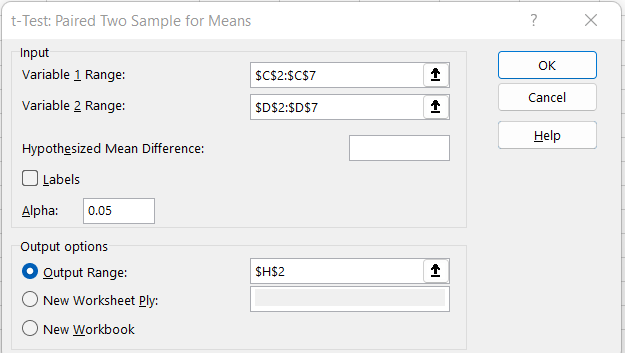
Step 2: Calculate the SD of samples STDEV(range)



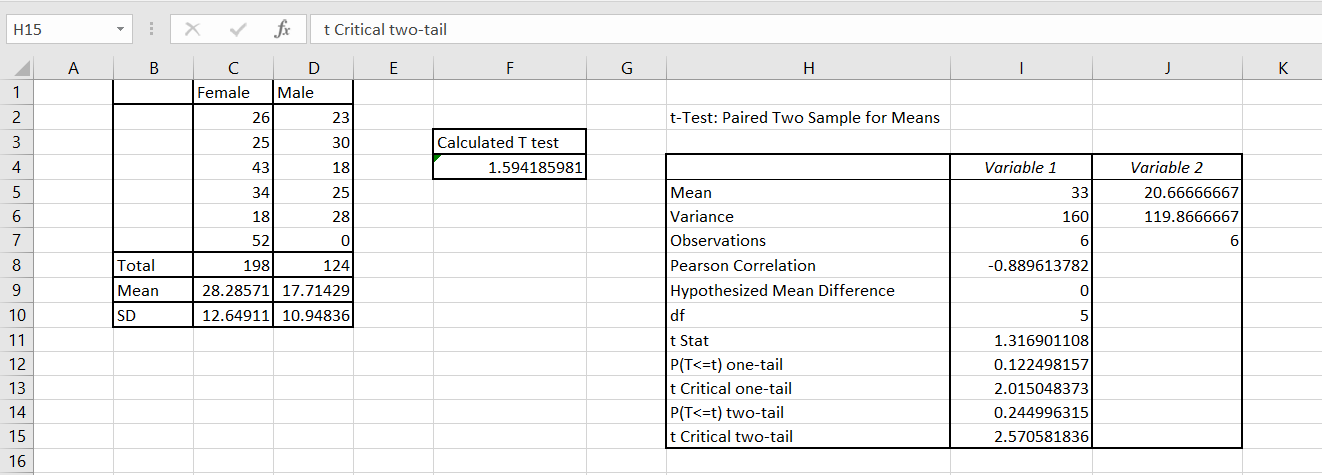
Step 3: Calculate the t-test

Step 4: Apply t-test paired two Samples for Means.

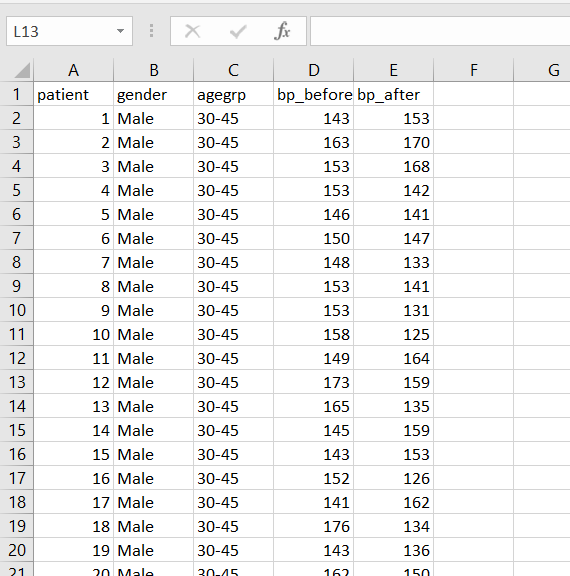


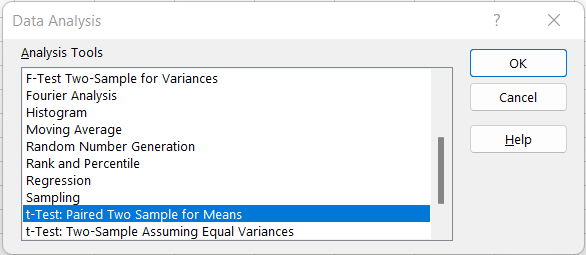
Step 5: Apply input Range and Output range.

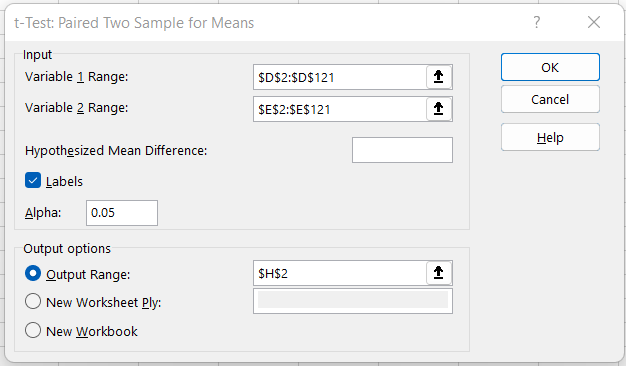
Step 6: Hence we reject null hypothesis.



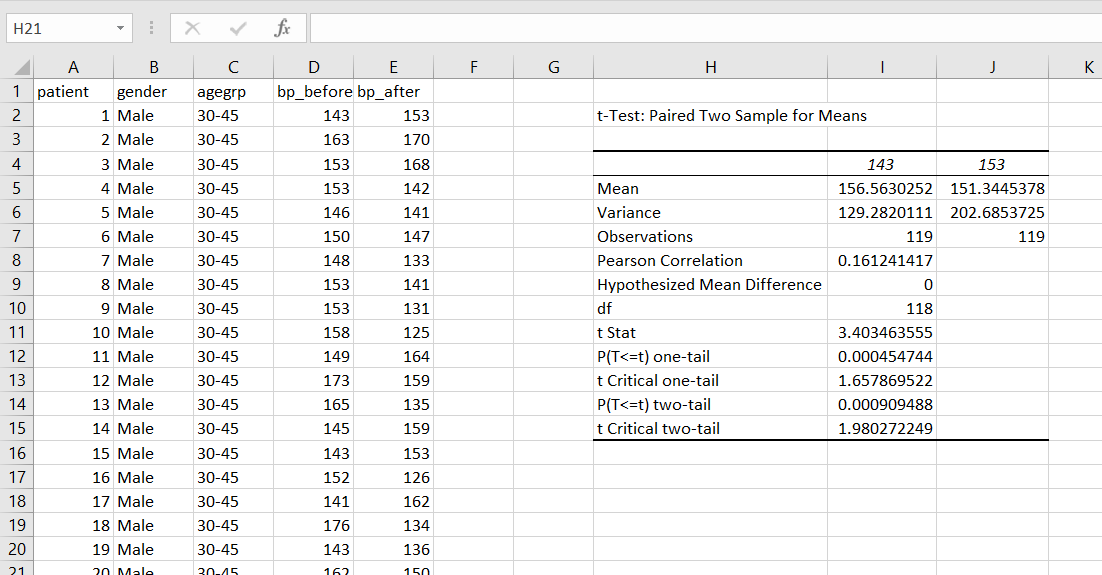
## Perform testing of Hypothesis using paired t-test.

Step 1: Load the Data

Step 2: Data -> Data Analysis -> t-test between samples for means.

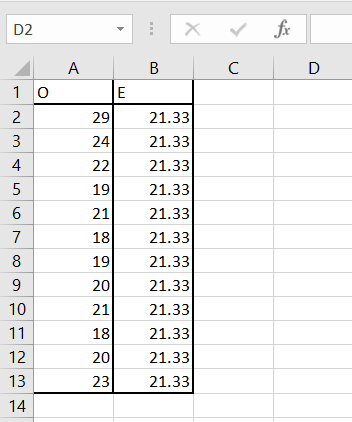
Step 3: Choose the Input and Output range.

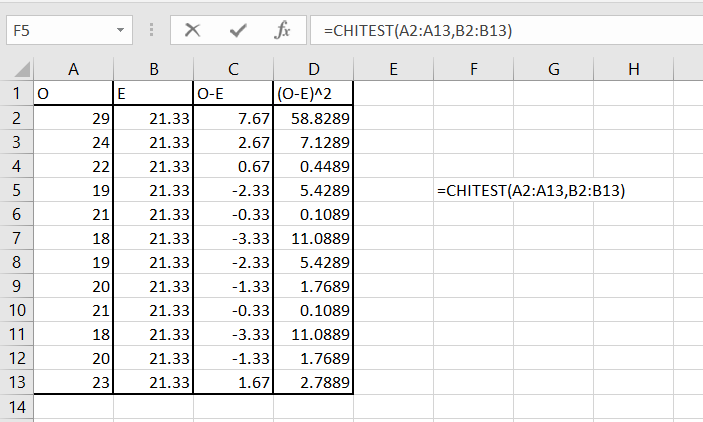
Step 4: Since the samples Means of the Data samples are not equal we reject the null hypothesis that they might be equal.



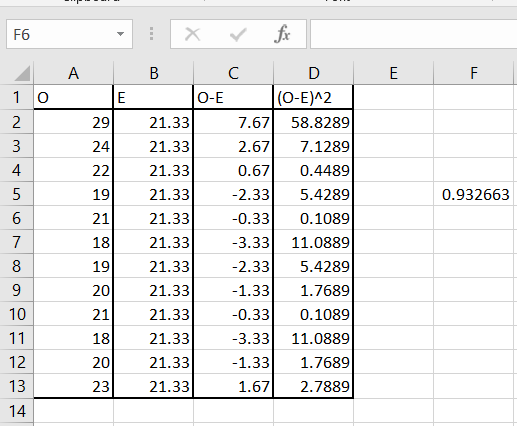
# Practical 4

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

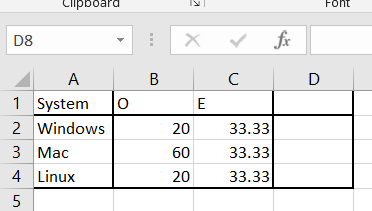
Step 1: Load the data.

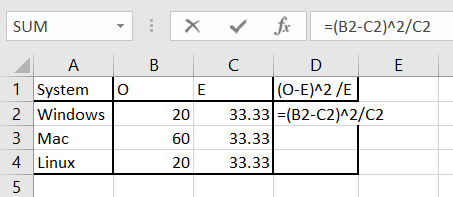
Step 2: Type CHITEST and select Actual(observed value range) and Expected value range.

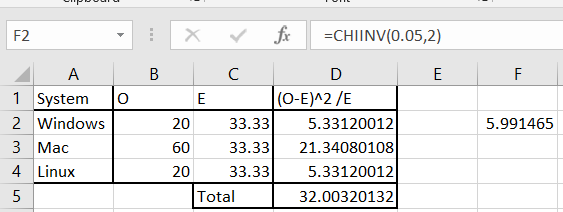
Step 3: This calculated value is less than table value which is 19.68. Hence we accept null hypothesis.

i.e 0.932663 < 19.68. H0 accepted.

## Perform testing of hypothesis using chi-squared test of independence.

Step 1: Load the data

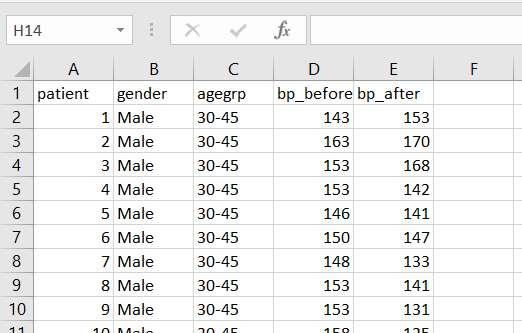
Step 2: Calculate the value of (O-E)^2 /E

Step 3: Calculate Chi square at 5% confidence and degree of freedom n-1=2(in our case)

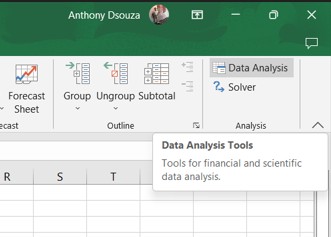
# Practical 5

## Perform testing of hypothesis using Z-test.

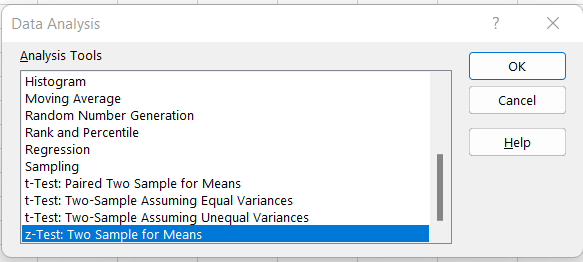
Step 1: Load the data

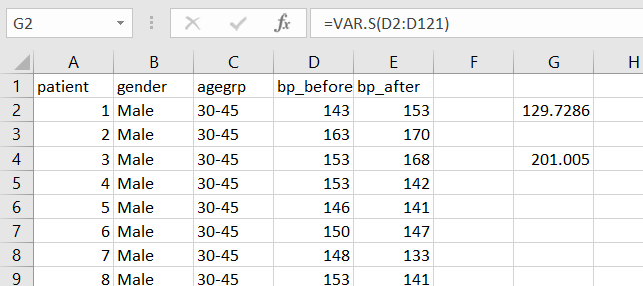


Step 2: To apply Z test we need a sample size over 30. Here our sample size is 120 data points , so to apply Z-test go to Data -> Data analysis.

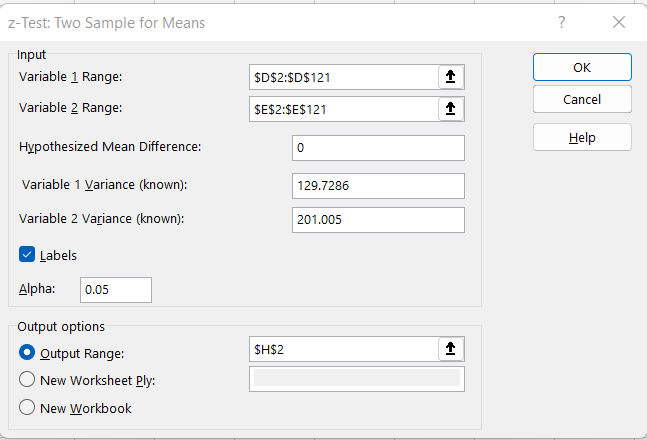


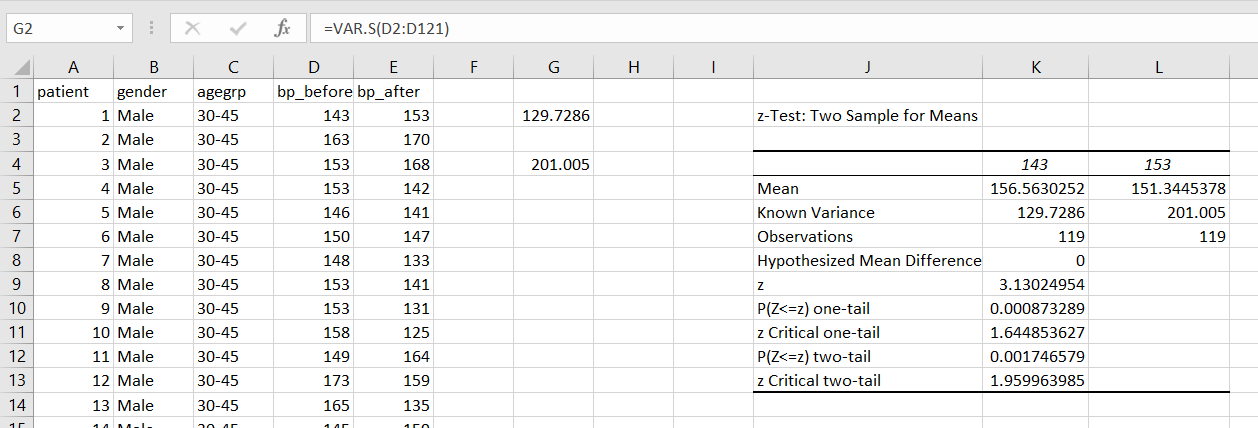
Step 3: Select Z-Test: Two Sample for means.



Step 4: Varience can be calculated using VAR.S (range)

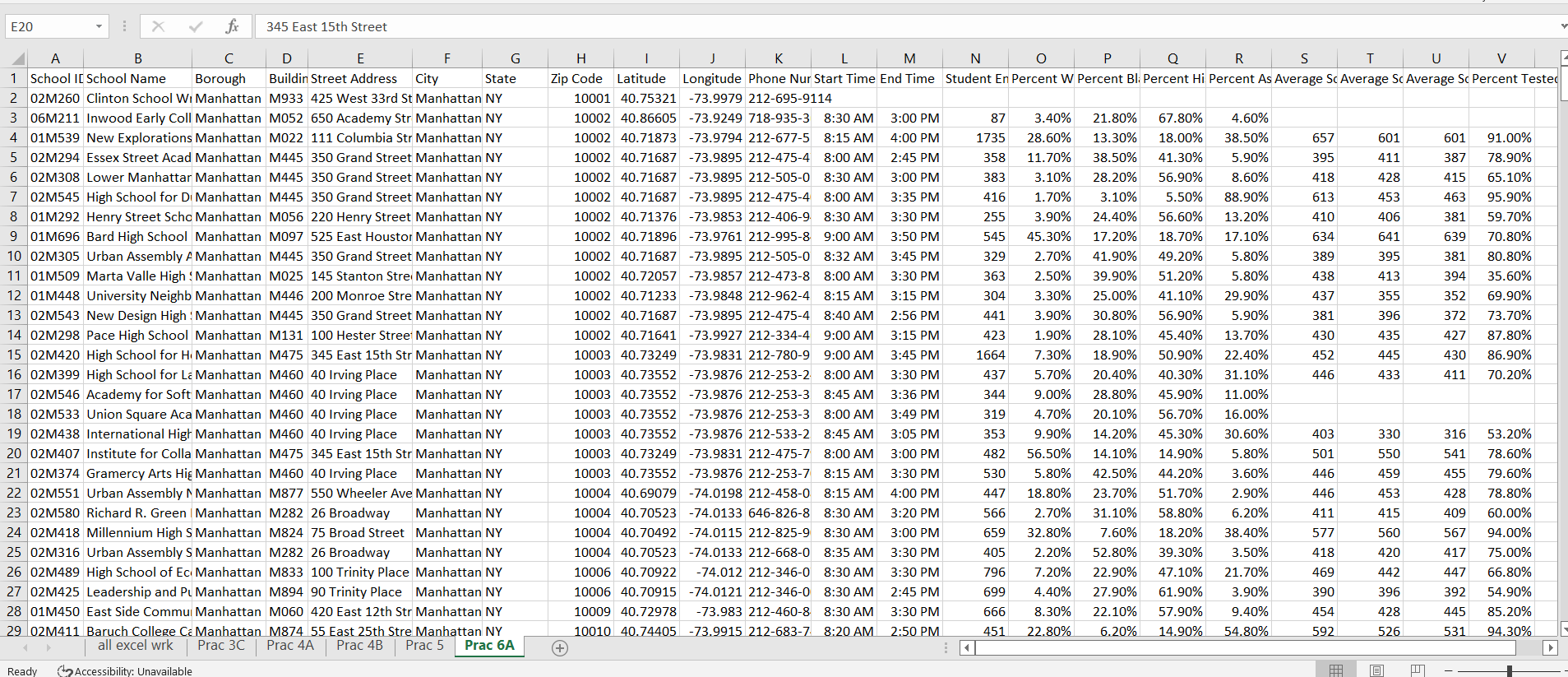
Step 5: Set the Variable 1 and 2 range.



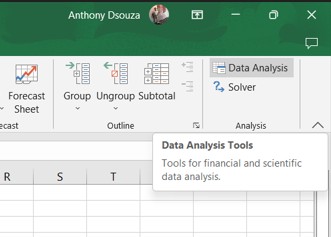
Step 6: Output

# Practical 6

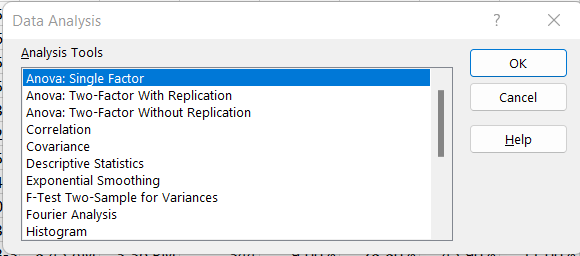
## Perform testing of hypothesis using One-way ANOVA.

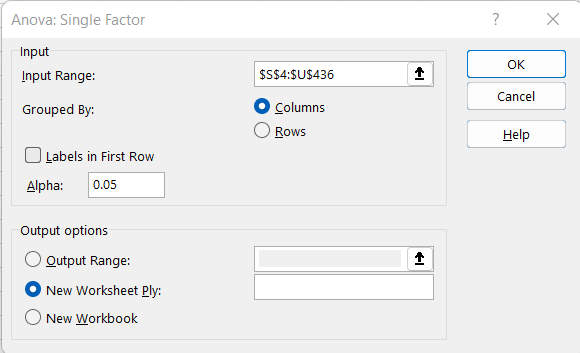
Step 1: Load the data

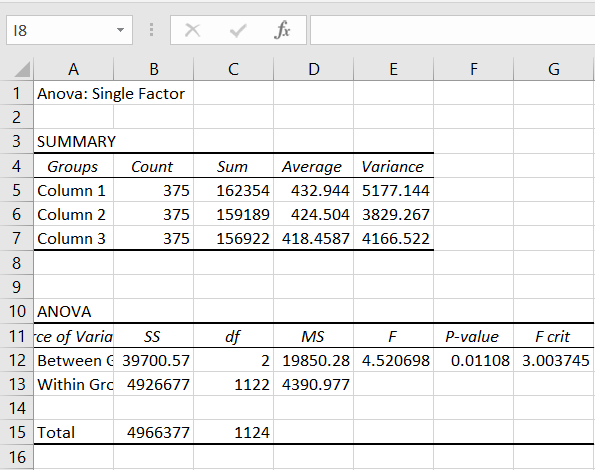
Step 2: To Calculate Single factor(one-way) anova we use cells S,T,U . Data -> Data Analysis.



Step 3: Select Anova: Single Factor

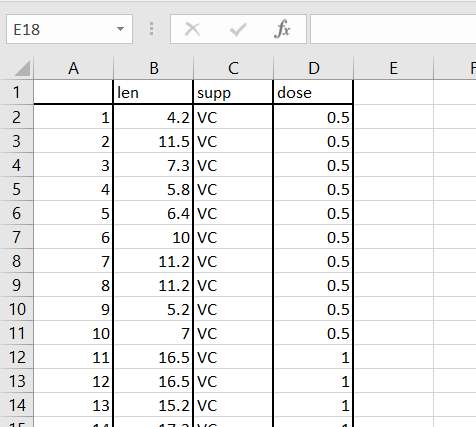


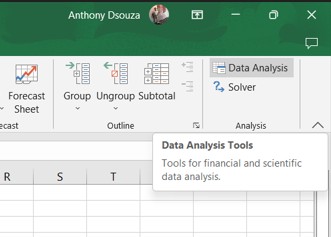
Step 4: Select the cells of S-T-U for input range

Step 5: Output

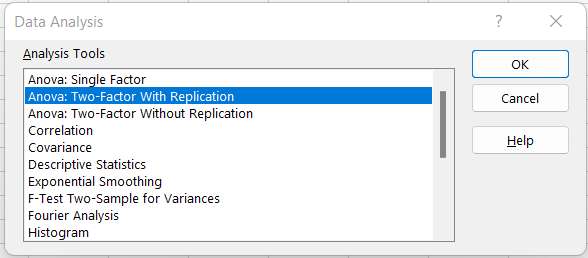
## Perform testing of hypothesis using Two-way ANOVA.

Step 1: Load the data.

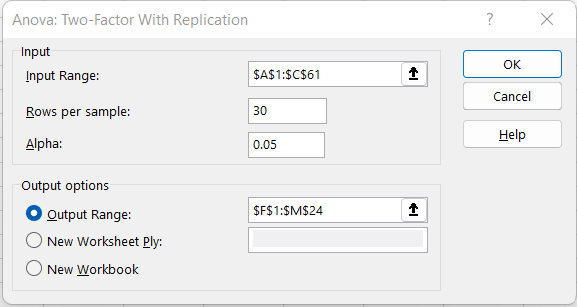


Step 2: Data -> Data Analysis

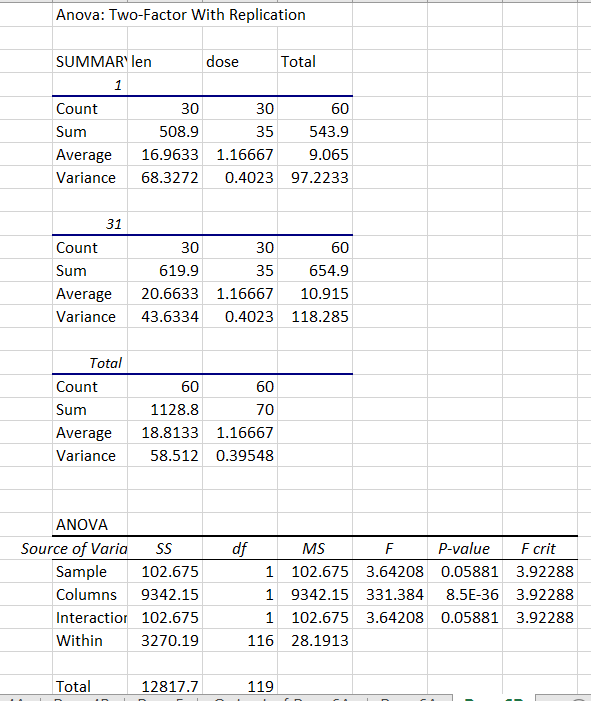
Step 3: Select Anova: Two Factor with Replication



Step 4: Select input and output range

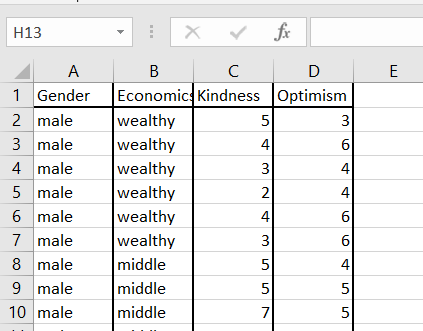


Step 5: Output

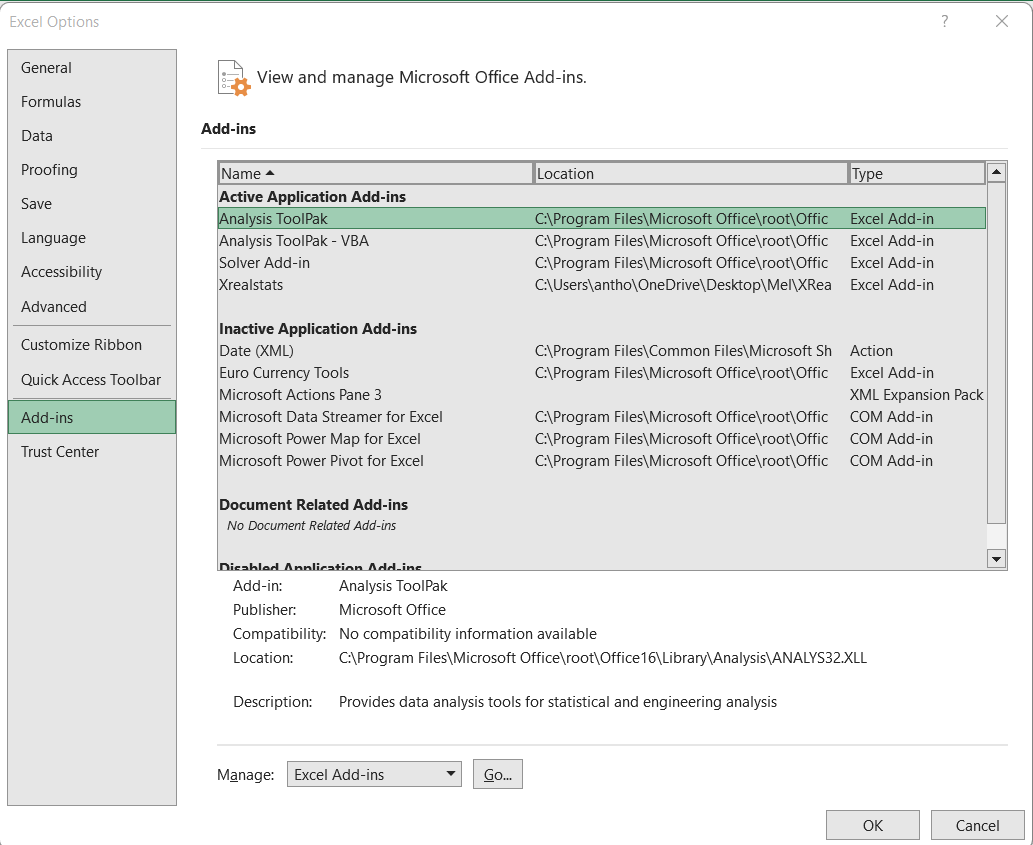


## Perform testing of hypothesis using MANOVA.

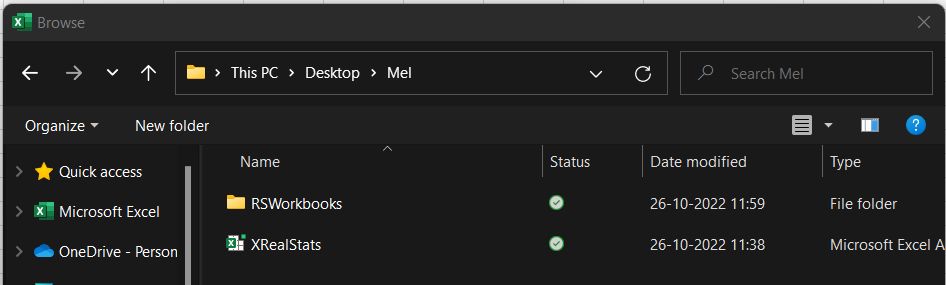
Step 1: Load the Data.

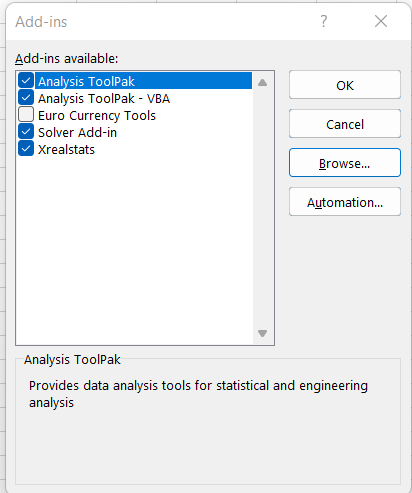


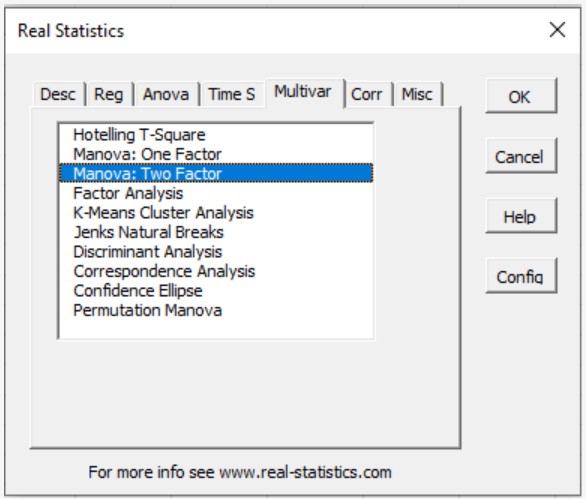
Step 2: Install Add-in in excel.

Select **File -> Help | Options -> Add-Ins** and click on the **Go** button at the bottom of the window.

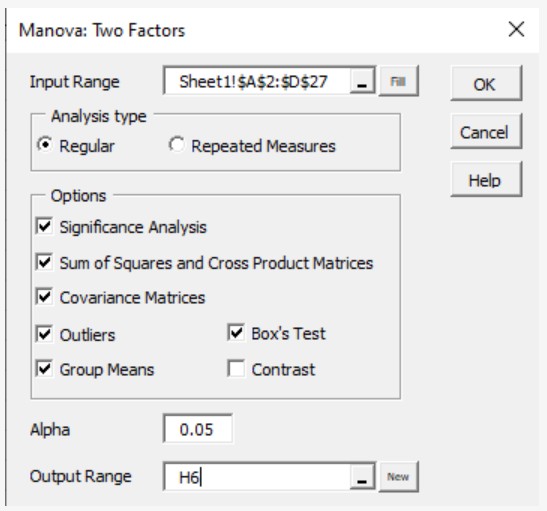
Step 3: Click on browse and select XrealStats file. Check the following check boxes.

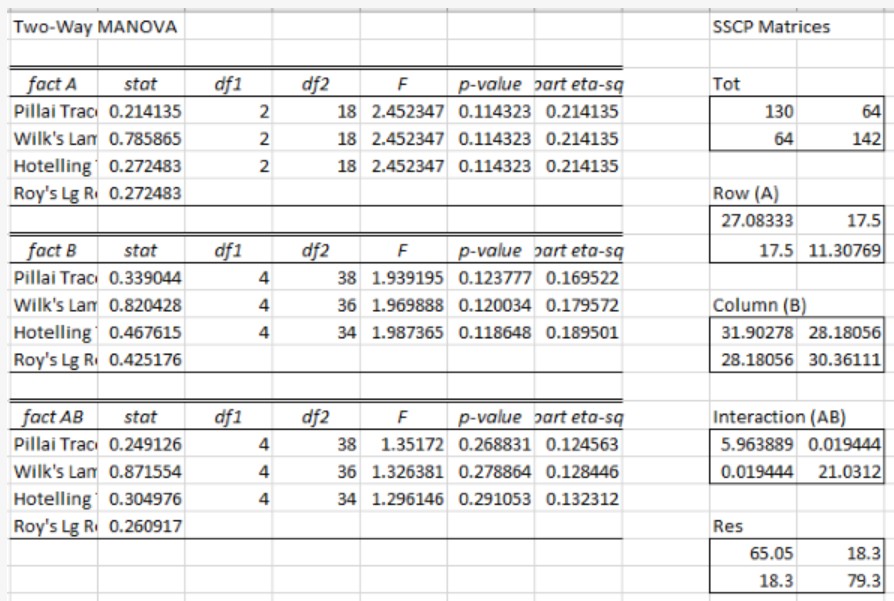




Step 4: Once loaded , CTRL+M to perform MANOVA. Click Multivar -> Manova: two factor -> OK.

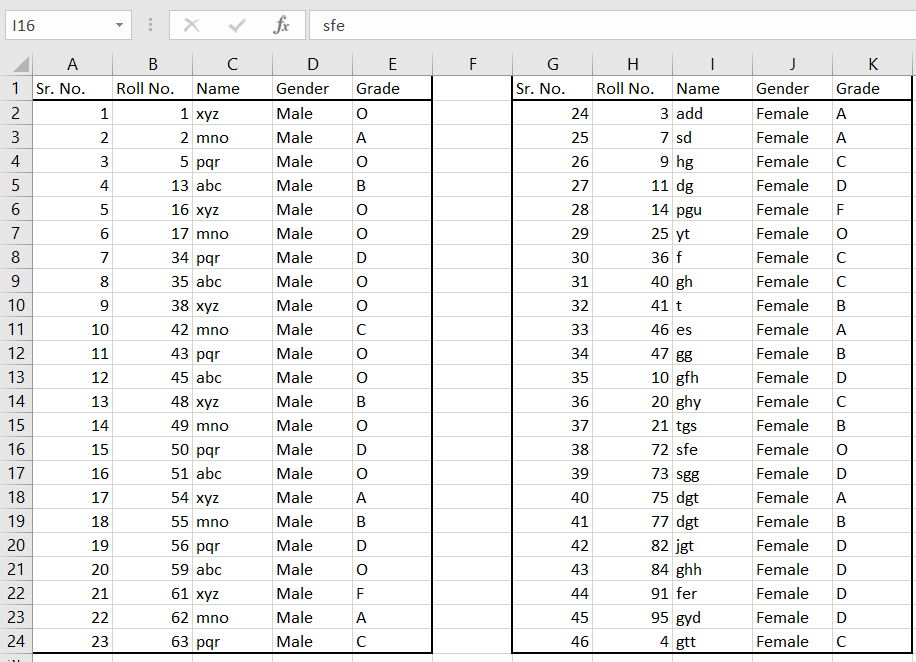
Step 5: Select the data excluding column names. Select a cell for output



Step 6: Output

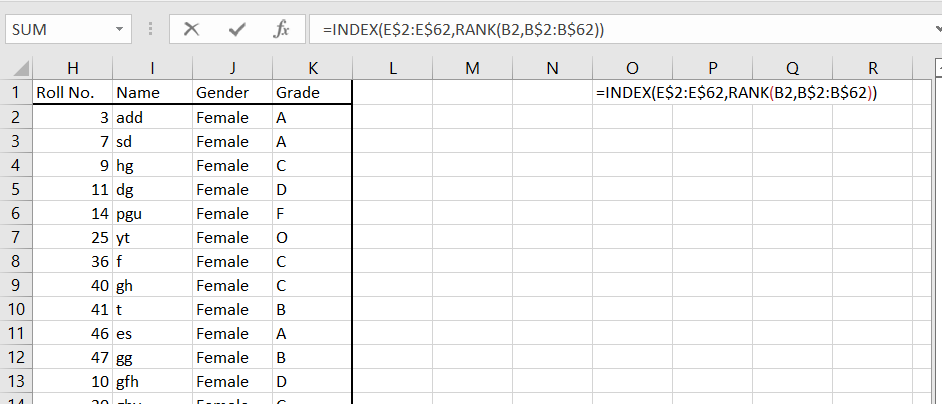
# Practical 7

## Perform the Random sampling for the given data and analyse it.

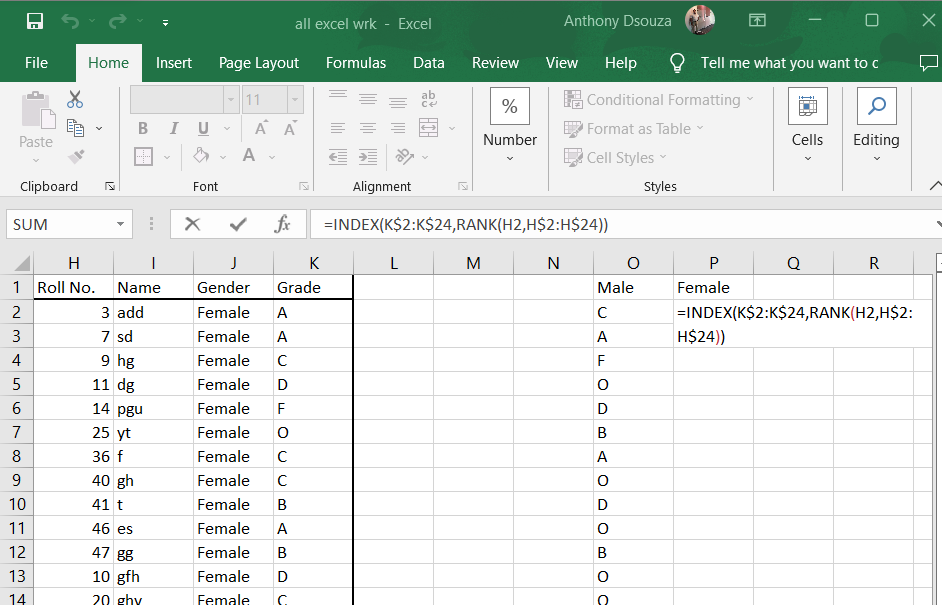
Step 1: Load the data.

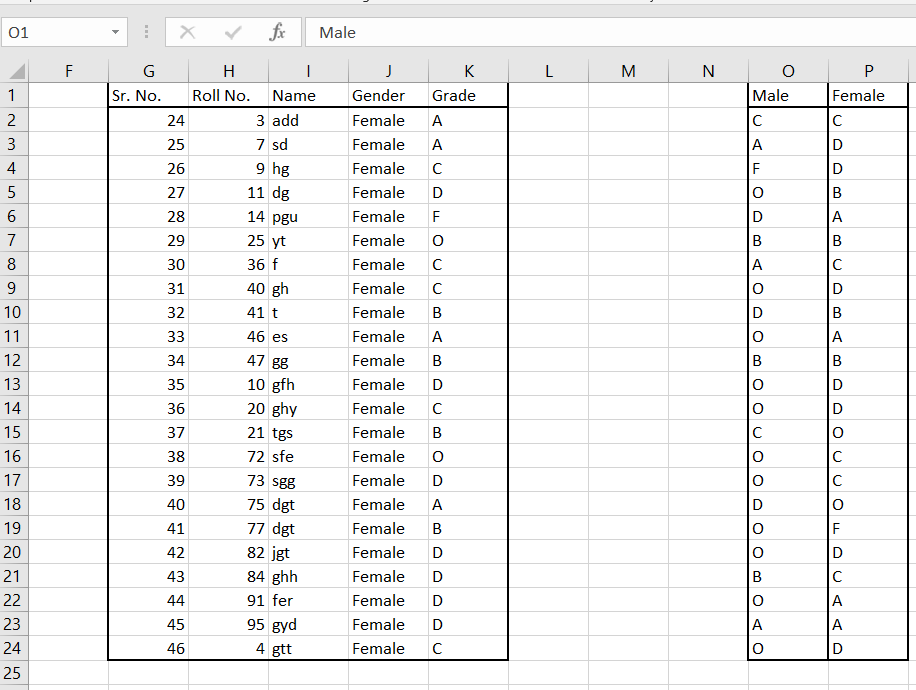
Step 2: Set Cell O1= Male and Cell P2= Female

To generate a random sample for male students from given population go to Cell O1 and type =INDEX(E$2:E$62,RANK(B2,B$2:B$62))

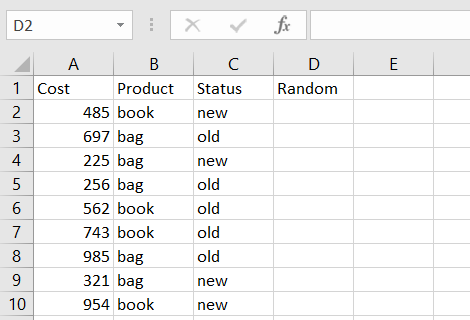


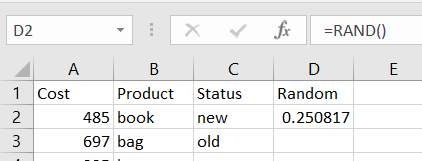
Step 3: Repeat the step to perform random sampling on Female data.



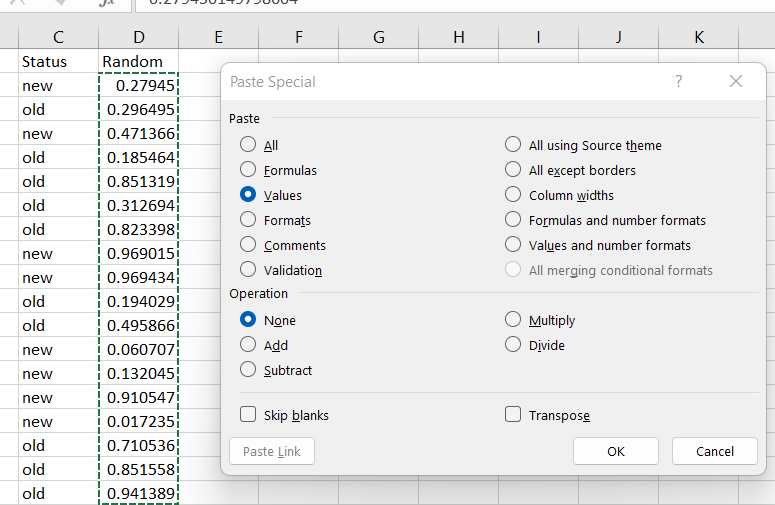
Step 4: Output

## Perform the Stratified sampling for the given data and analyse it

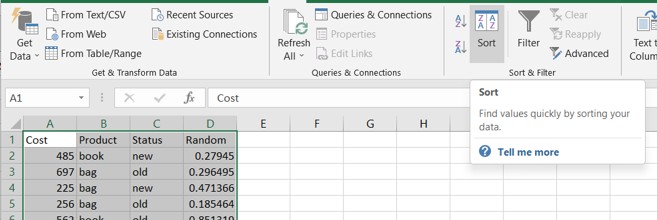
Step 1: Load the data.

Step 2: Assign Random values using RAND() function.

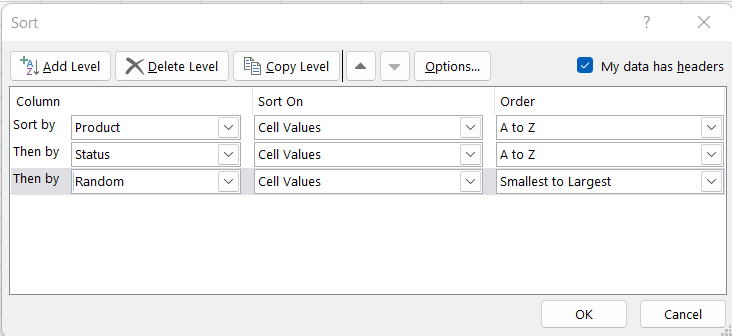
Step 3: Copy the entire column D and paste only values



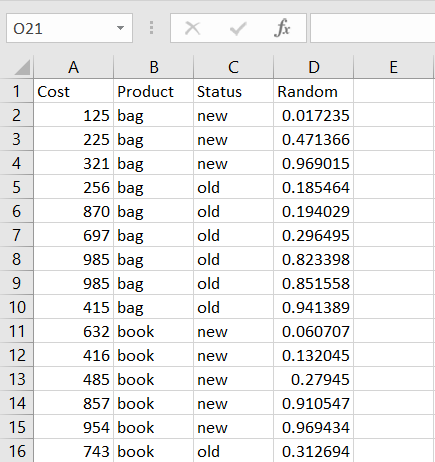
Step 4: To perform stratified sampling and obtain data where we get 2 new books cost, 2 old books cost, 2 new bags cost, 2 old bags cost. Select the data and click on sort.



Step 5: Sort the data by product , status and random.



Step 6: the output obtained will be as follows.



# Practical 8

## Write a program for computing different correlation

1. Positive Correlation.

### Code:

import matplotlib import numpy as np

import matplotlib.pyplot as plt np.random.seed(1)

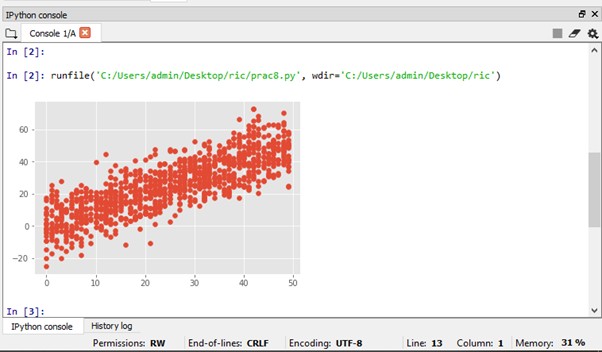
# 1000 random integers between 0 and 50

x = np.random.randint(0, 50, 1000)

# Positive Correlation with some noise y = x + np.random.normal(0, 10, 1000) np.corrcoef(x, y) matplotlib.style.use('ggplot') plt.scatter(x, y)

plt.show()

**Output:**



## Negative Correlation.

### Code:

import matplotlib import numpy as np

import matplotlib.pyplot as plt np.random.seed(1)

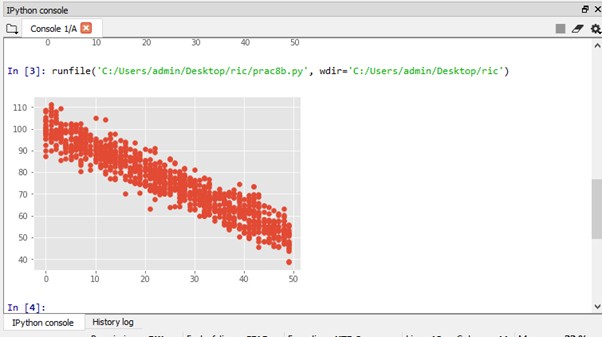
# 1000 random integers between 0 and 50

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 9

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

### Code:

* #Perform linear regression

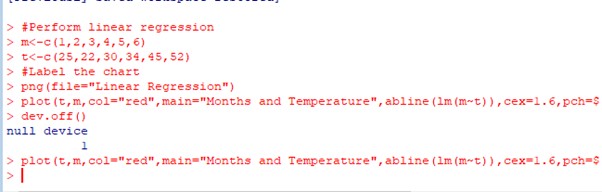
> m<-c(1,2,3,4,5,6)

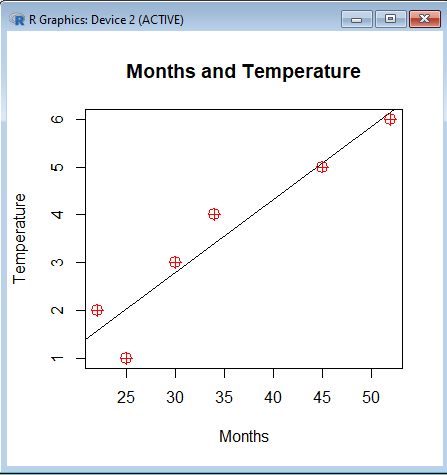
> t<-c(25,22,30,34,45,52)

* #Label the chart
* png(file="Linear Regression")
* plot(t,m,col="red",main="Months and Temperature",abline(lm(m~t)),cex=1.6,pch=10,xlab="Months",ylab="Tempe rature")
* dev.off() null device

1

* plot(t,m,col="red",main="Months and Temperature",abline(lm(m~t)),cex=1.6,pch=10,xlab="Months",ylab="Tempe rature")

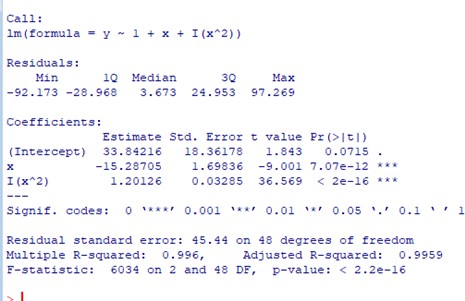




## Polynomial Regression.

### Code:

* #Polynomial Regression
* set.seed(16)
* x<-0:50
* y<-2.3-15.1\*x+1.2\*x^2+rnorm(length(x),20,50)
* plot(x,y)
* fit <- lm(y ~ 1 + x + I(x^2))
* points(x, predict(fit), type="l")
* summary(fit)

**Output:**

# Practical 10

## Multiple linear regression.

### Code:

* #Multiple linear regression

>ip<-mtcars[,c("mpg","disp","hp")]

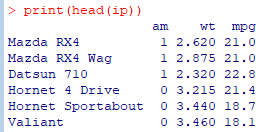
* print(head(ip))
* m<-lm(mpg~disp+hp,data=ip)
* print(m)
* a<-coef(m)

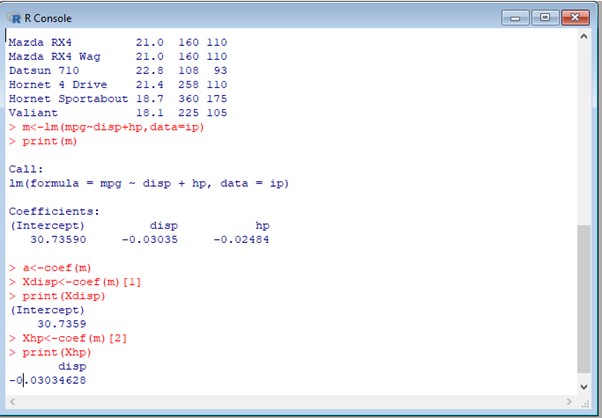
>Xdisp<-coef(m)[1]

* print(Xdisp)

>Xhp<-coef(m)[2]

* print(Xhp)

 **Output:**



## Logistic Regression.

### Code:

* #Logistic regression

>ip<-mtcars[,c("am","wt","mpg")]

* print(head(ip))

>am.data<-glm(formula=am~wt+mpg,data=ip,family=binomial)

* summary(am.data)

### Output: