MAHARASHTRA COLLEGE OF ARTS,SCIENCE & COMMERCE
MUMBAI
400-008

DEPARTMENT OF INFORMATION TECHNOLOGY

MSC I.T PART-1

CERTIFICATE

This is to certify that the practical done at MAHARASHTRA COLLEGE By Ms/Mr._____

(Seat No._____) in partial fulfillment for M.Sc.Degree Examination has been found satisfactory. This practical journal had not been submitted for any other examination and does not form part of any other course undergone by candidate.

Signature	Signature	Signature
Lecture-In-Charge	External Examiner	Course Coordinator
Guided by	Examined By	Certified By

College Stamp

Research
In Computing

INDEX

SR NO.	PRACTICAL AIM	DATE	SIGNATURE
1	**A:** Write a program for obtaining descriptive statistics of data. **B:** Import data from different data sources (from Excel, csv, mysql, sql server, oracle to R/Python/Excel)		
2	**:** Perform suitable analysis of given secondary data.		
3	**A:** Perform testing of hypothesis using one sample t-test **B:** Perform testing of hypothesis using two sample t-test **C:** Perform testing of hypothesis using paired t-test.		
4	**A:**Perform testing of hypothesis using chi-squared goodness-of-fit test. **B:** Perform testing of hypothesis using chi-squared Test of Independence		
5	Perform testing of hypothesis using Z-test. **A:** One Sample Z test **B:** Two Sample Z test		
6	**A:** Perform testing of hypothesis using one-way ANOVA**.** **B:** Perform testing of hypothesis using two-way ANOVA. **C:** Perform testing of hypothesis using multivariate ANOVA (MANOVA).		
7	**A:** Perform the Random sampling for the given data and analyse it. **B:** Perform the Stratified sampling for the given data and analyse it.		
8	Compute different types of correlation. **A:** Positive Correlation **B:** Negative Correlation **C:** No/Weak Correlation		

Practical No. 1

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

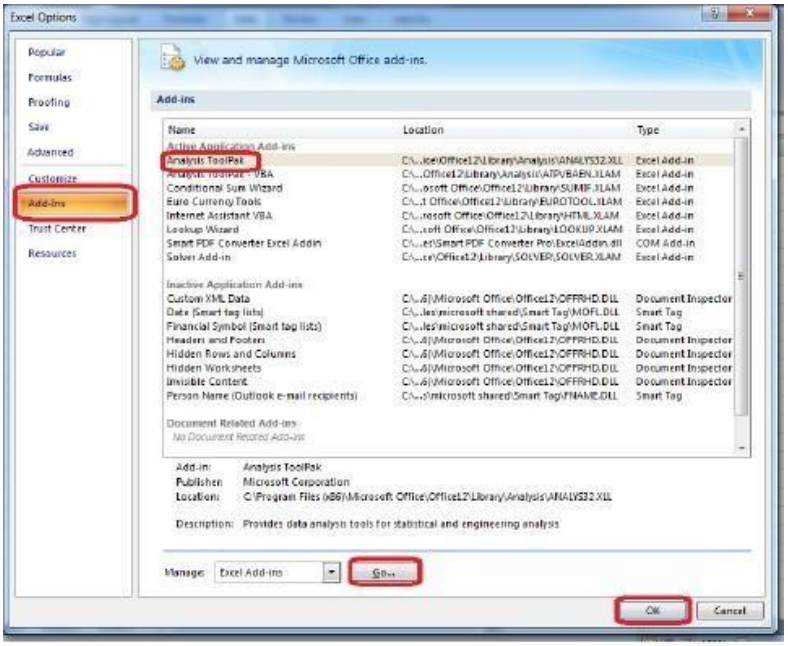
```
import pandas as pd
#Create a Dictionary of series
d = {'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])}
#Create a DataFrame
df = pd.DataFrame(d)
print(df)
print('############# Sum ########## ')
print (df.sum())
print('############# Mean ########## ')
print (df.mean())
print('############# Standard Deviation ########## ')
print (df.std())
print('############# Descriptive Statistics ########## ')
print (df.describe())
```

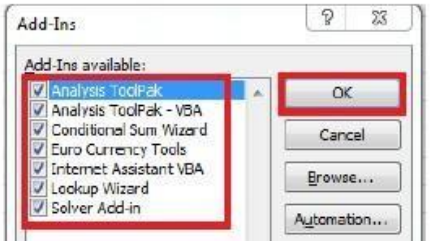
Ouput :

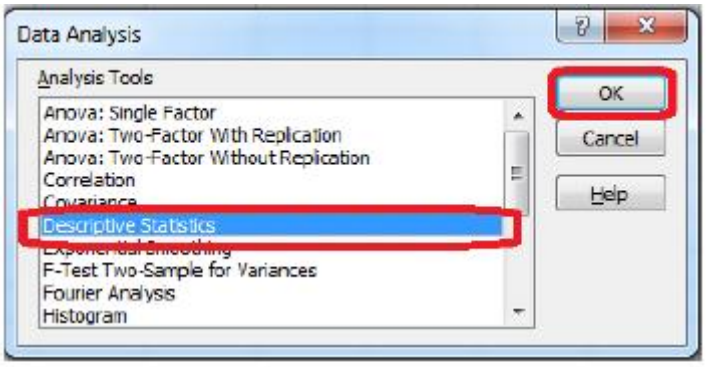
```
    Age  Rating
0    25    4.23
1    26    3.24
2    25    3.98
3    23    2.56
4    30    3.20
5    29    4.60
6    23    3.80
7    34    3.78
8    40    2.98
9    30    4.80
10   51    4.10
11   46    3.65
###########  Sum ##########
Age       382.00
Rating     44.92
dtype: float64
###########  Mean ##########
Age       31.833333
Rating     3.743333
dtype: float64
###########  Standard Deviation ##########
Age       9.232682
Rating    0.661628
dtype: float64
###########  Descriptive Statistics ##########
             Age     Rating
count  12.000000  12.000000
mean   31.833333   3.743333
std     9.232682   0.661628
min    23.000000   2.560000
25%    25.000000   3.230000
50%    29.500000   3.790000
75%    35.500000   4.132500
max    51.000000   4.800000
```

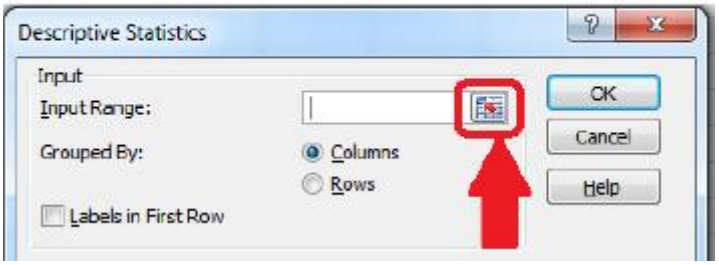
Using Excel :

Go to File Menu → Options → Add-Ins→Select Analysis ToolPak →Press OK

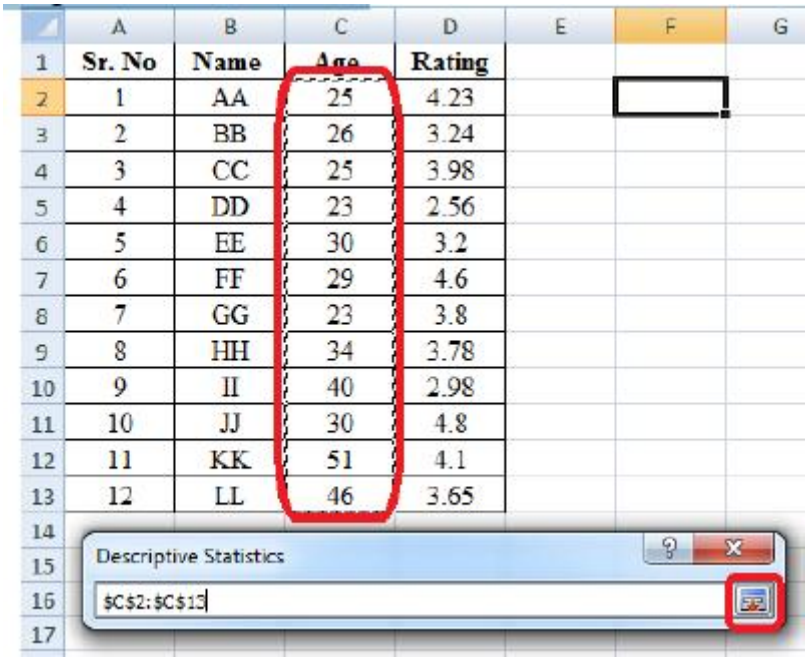


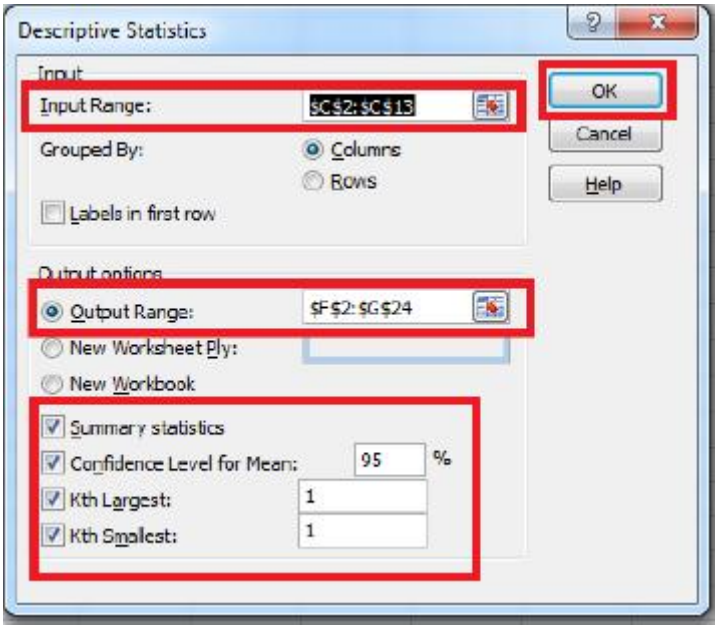






Select the data range from the excel worksheet.





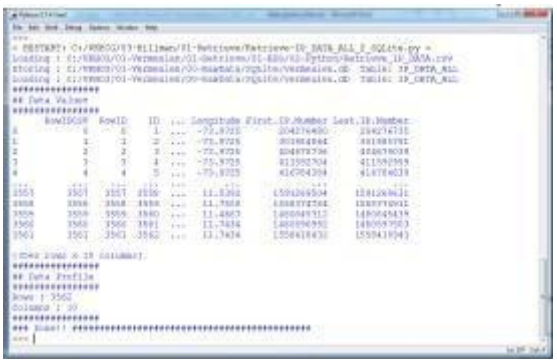
Output :

	A	B	C	D	E	F	G
1	Sr. No	Name	Age	Rating			
2	1	AA	25	4.23		*Column1*	
3	2	BB	26	3.24			
4	3	CC	25	3.98		Mean	31.83333
5	4	DD	23	2.56		Standard Error	2.665246
6	5	EE	30	3.2		Median	29.5
7	6	FF	29	4.6		Mode	25
8	7	GG	23	3.8		Standard Deviation	9.232682
9	8	HH	34	3.78		Sample Variance	85.24242
10	9	II	40	2.98		Kurtosis	0.24931
11	10	JJ	30	4.8		Skewness	1.135089
12	11	KK	51	4.1		Range	28
13	12	LL	46	3.65		Minimum	23
14						Maximum	51
15						Sum	382
16						Count	12
17						Largest(1)	51
18						Smallest(1)	23
19						Confidence Level(95.0%)	5.866167

1B: Import data from different data sources (from Excel, csv, mysql, sql server, oracle to R/Python/Excel)

```
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:



MySQL:

Open MySql

Create a database “DataScience”

Create a python file and add the following code:

```
################ Connection With MySQL ######################
Import mysql.connector
conn = mysql.connector.connect(host='localhost',
database='DataScience',
user='root',
```

```
password='root')
conn.connect
if(conn.is_connected):
print('###### Connection With MySql Established Successfullly ##### ')
else:
print('Not Connected -- Check Connection Properites')
```

Output:

```
>>>
 RESTART: C:/Users/User/AppData/Local/Programs/Python/Python37-32/mysqlconnection.py
###### Connection With MySql Established Successfullly #####
>>>
```

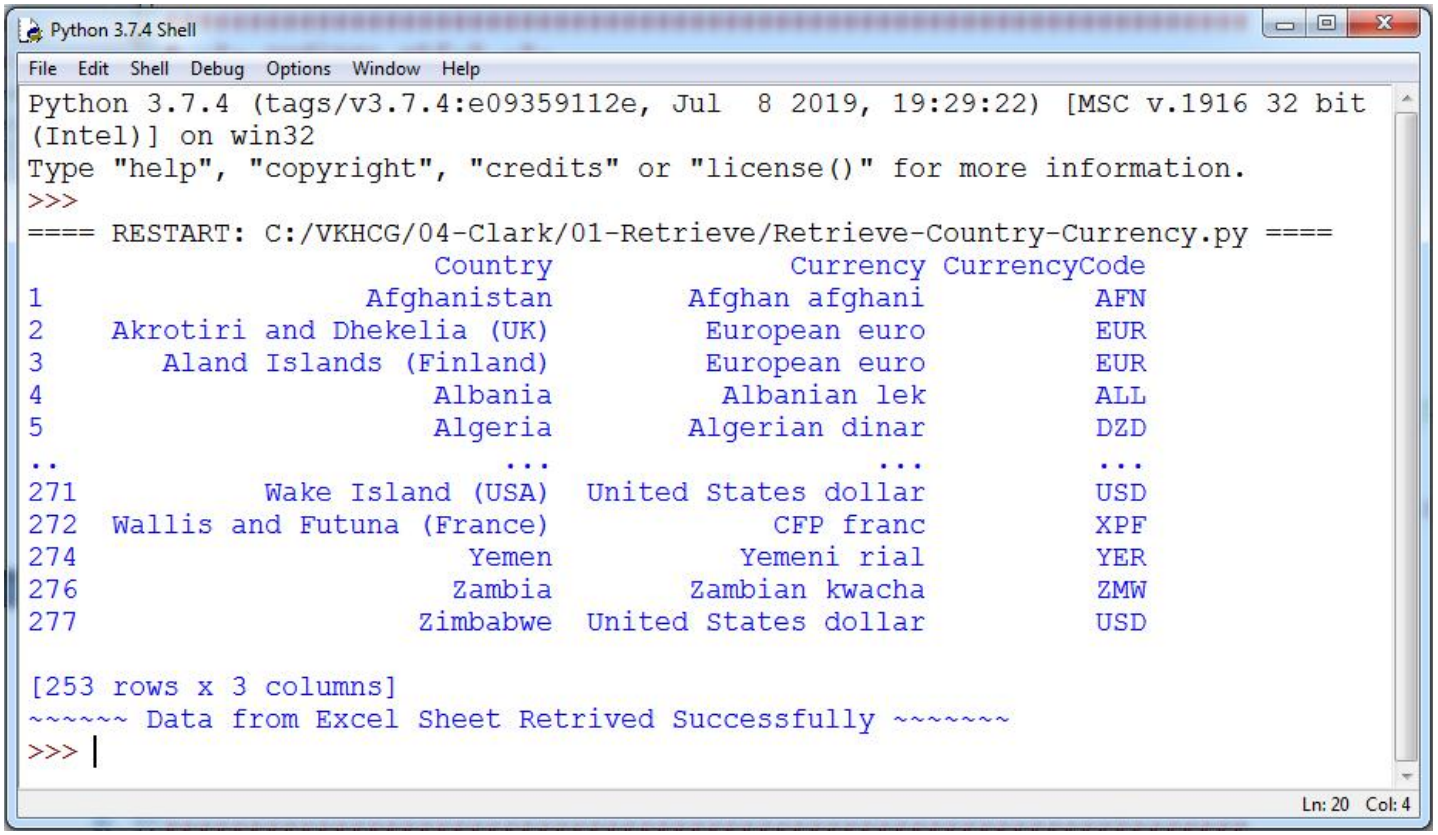
Microsoft Excel

```
###################Retrieve-Country-Currency.py
#######################################################################
# -*- coding: utf-8 -*-
#######################################################################
importos
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:



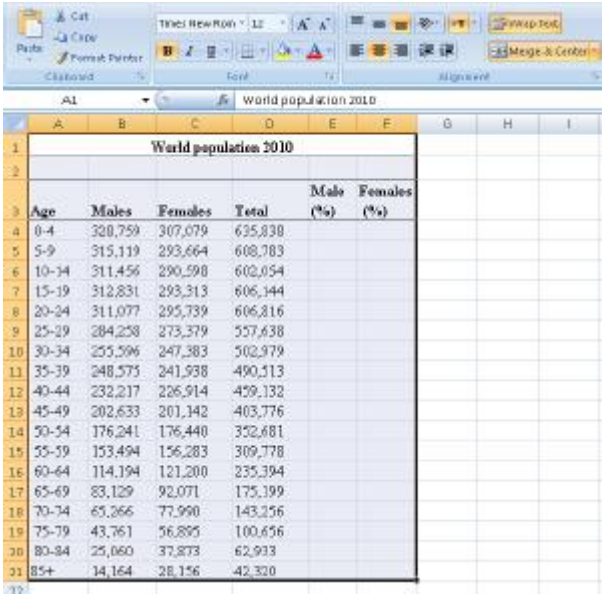
Practical No. 2

2A: Perform analysis of given secondary data.

Steps in Secondary Data Analysis

1. **Determine your research question** – Knowing exactly what you are looking for.
2. **Locating data**– Knowing what is out there and whether you can gain access to it. A quick Internet search, possibly with the help of a librarian, will reveal a wealth of options.
3. **Evaluating relevance of the data** – Considering things like the data's original purpose, when it was collected, population, sampling strategy/sample, data collection protocols, operationalization of concepts, questions asked, and form/shape of the data.
4. **Assessing credibility of the data** – Establishing the credentials of the original researchers, searching for full explication of methods including any problems encountered, determining how consistent the data is with data from other sources, and discovering whether the data has been used in any credible published research.
5. **Analysis –** This will generally involve a range of statistical processes.

Example: Analyze the given Population Census Data for Planning and Decision Making by using the size and composition of populations.



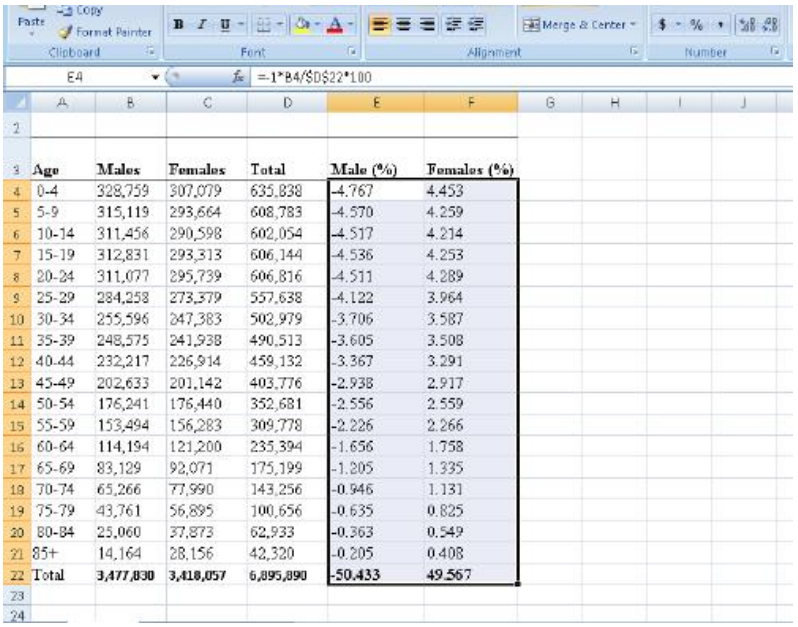
World population 2010

Age	Males	Females	Total	Male (%)	Females (%)
0-4	328,759	307,079	635,838		
5-9	315,119	293,664	608,783		
10-14	311,456	290,598	602,054		
15-19	312,831	293,313	606,144		
20-24	311,077	295,739	606,816		
25-29	284,258	273,379	557,638		
30-34	255,596	247,383	502,979		
35-39	248,575	241,938	490,513		
40-44	232,217	226,914	459,132		
45-49	202,633	201,142	403,776		
50-54	176,241	176,440	352,681		
55-59	153,494	156,283	309,778		
60-64	114,194	121,200	235,394		
65-69	83,129	92,071	175,199		
70-74	65,266	77,990	143,256		
75-79	43,761	56,895	100,656		
80-84	25,060	37,873	62,933		
85+	14,164	28,156	42,320		

Put the cursor in cell **B22** and click on the **AutoSum** and then click **Enter**. This will calculate the total population. Then copy the formula in cell **D22** across the row **22.**

To calculate the percent of males in cell **E4**, enter the formula **=-1*100*B4/$D$22** . And copy the formula in cell **E4** down to cell **E21.**

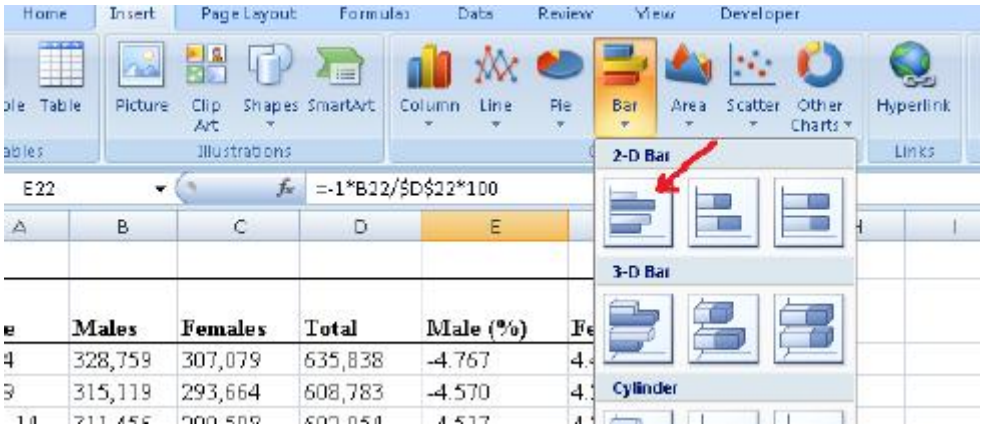
To calculate the percent of females in cell **F4**, enter the formula =**100*C4/$D$22**. Copy the formula in cell **F4** down to cell **F21.**



	A	B	C	D	E	F
3	Age	Males	Females	Total	Male (%)	Females (%)
4	0-4	328,759	307,079	635,838	-4.767	4.453
5	5-9	315,119	293,664	608,783	-4.570	4.259
6	10-14	311,456	290,598	602,054	-4.517	4.214
7	15-19	312,831	293,313	606,144	-4.536	4.253
8	20-24	311,077	295,739	606,816	-4.511	4.289
9	25-29	284,258	273,379	557,638	-4.122	3.964
10	30-34	255,596	247,383	502,979	-3.706	3.587
11	35-39	248,575	241,938	490,513	-3.605	3.508
12	40-44	232,217	226,914	459,132	-3.367	3.291
13	45-49	202,633	201,142	403,776	-2.938	2.917
14	50-54	176,241	176,440	352,681	-2.556	2.559
15	55-59	153,494	156,283	309,778	-2.226	2.266
16	60-64	114,194	121,200	235,394	-1.656	1.758
17	65-69	83,129	92,071	175,199	-1.205	1.335
18	70-74	65,266	77,990	143,256	-0.946	1.131
19	75-79	43,761	56,895	100,656	-0.635	0.825
20	80-84	25,060	37,873	62,933	-0.363	0.549
21	85+	14,164	28,156	42,320	-0.205	0.408
22	Total	3,477,830	3,418,057	6,895,890	-50.433	49.567

To build the population pyramid, we need to choose a horizontal bar chart with two series of data (% male and % female) and the age labels in column A as the **Category X-axis** labels. Highlight the range **A3:A21**, hold down the CTRL key and highlight the range **E3:F21**

Under **inset** tab, under horizontal bar charts select **clustered bar chart**

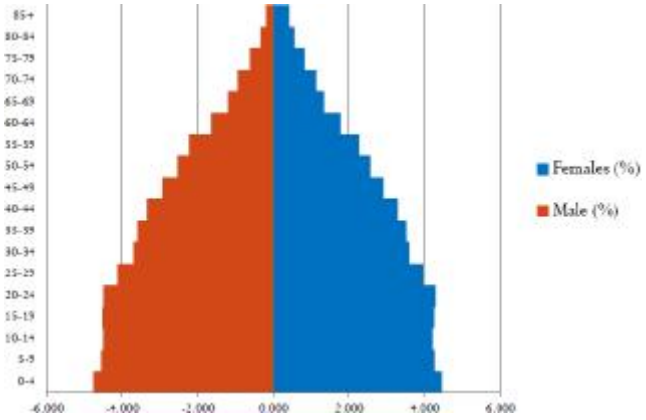


Put the tip of your mouse arrow on the **Y-axis** (vertical axis) so it says "Category Axis", right click and chose **Format Axis**

Choose **Axis options** tab and set the major and minor tick mark type to **None**, Axis labels to **Low**, and click **OK**.

Click on any of the bars in your pyramid, click right and select "format data series". Set the **Overlap** to **100** and **Gap Width** to **0**. Click **OK**.

Output :


85+
80-84
75-79
70-74
65-69
60-64
55-59
50-54
45-49
40-44
35-39
30-34
25-29
20-24
15-19
10-14
5-9
0-4
-6.000
-4.000
-2.000
0.000
2.000
4.000
6.000
Females (%)
Male (%)

Practical No. 3

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

One sample t-test : The One Sample *t* Test determines whether the sample mean is statistically different from a known or hypothesised population mean. The One Sample *t* Test is a parametric test.

Program 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.

The *t* distribution provides a good way to perform one sample tests on the mean when the population variance is not known provided the population is normal or the sample is sufficiently large so that the Central Limit Theorem applies

Two Sample t Test

Example: A college Princiapal informed classroom teachers that some of their students showedunusual potential for intellectual gains. One months later the students identified to teachers ashaving potentional for unusual intellectual gains showed significiantly greater gains performanceon a test said to measure IQ than did students who were not so identified. Below are the data forthe 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	
27.15	11.95	Mean
12.51	14.61	Sd

Experimental Data

To calculate Standard Mean go to cell A22 and type =SUM(A2:A21)/20

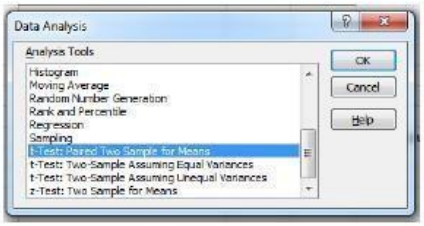
To calculate Standard Deviation go to cell A23 and type =STDEV(A2:A21)

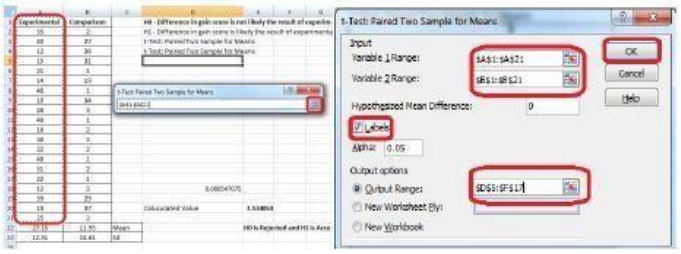
Comparison Data

To calculate Standard Mean go to cell B22 and type =SUM(B2:B21)/20

To calculate Standard Deviation go to cell B23 and type =STDEV(B2:B21)

To find T-Test Statistics go to data Data Analysis





To calculate the T-Test square value go to cell E20 and type
=(A22-B22)/SQRT((A23*A23)/COUNT(A2:A21)+(B23*B23)/COUNT(A2:A21))
Now go to cell E20 and type

=IF(E20<E12,"H0 is Accepted", "H0 is Rejected and H1 is Accepted")

Our calculated value is larger than the tabled value at alpha = .01, so we reject the null hypothesisand accept the alternative hypothesis, namely, that the difference in gain scores is likely the resultof the experimental treatment and not the result of chance variation.

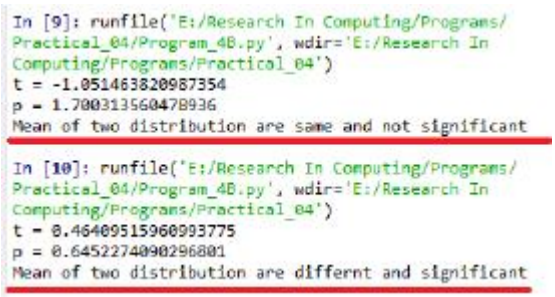
Output:

	A	B	C	D	E	F
1	**Experimental**	**Comparison**		**H0 - Difference in gain score is not likely the result of experimental treatment.**		
2	35	2		H1 - Difference in gain score is likely the result of experimental treatment and not the result of change variation.		
3	40	27		t-Test: Paired Two Sample for Means		
4	12	38		t-Test: Paired Two Sample for Means		
5	15	31		t-Test: Paired Two Sample for Means		
6	21	1				
7	14	19			*Experimental*	*Comparison*
8	46	1		Mean	27.15	11.95
9	10	34		Variance	156.45	213.5236842
10	28	3		Observations	20	20
11	48	1		Pearson Correlation	-0.395904927	
12	16	2		Hypothesized Mean Difference	0	
13	30	3		df	19	
14	32	2		t Stat	2.996289153	
15	48	1		P(T<=t) one-tail	0.003711226	
16	31	2		t Critical one-tail	1.729132792	
17	22	1		P(T<=t) two-tail	0.007422452	
18	12	3		t Critical two-tail	2.09302405	
19	39	29				
20	19	37		Calculated Value	3.534053898	
21	25	2				
22	27.15	11.95	Mean		**H0 is Rejected and H1 is Accepted**	
23	12.51	14.61	Sd			

Using python :

```
import numpy as np
from  scipy import stats
from numpy.random import randn
N = 20
#a = [35,40,12,15,21,14,46,10,28,48,16,30, 32,48,31,22,12,39,19,25]
#b = [2,27,31,38,1,19,1,34,3,1,2,1,3,1,2,1,3,29,37,2]
a = 5 * randn(100) + 50
b = 5 * randn(100) + 51
var_a = a.var(ddof=1)
var_b = b.var(ddof=1)
s = np.sqrt((var_a + var_b)/2)
t = (a.mean() - b.mean())/(s*np.sqrt(2/N))
df = 2*N - 2
#p-value after comparison with the t
p = 1 - stats.t.cdf(t,df=df)
print("t = " + str(t))
print("p = " + str(2*p))
if t> p :
print('Mean of two distribution are differnt and significant')
else:
print('Mean of two distribution are same and not significant')
```

Output :



C. Perform testing of hypothesis using paired t-test.

The paired sample t-test is also called dependent sample t-test. It's an univariate test that tests for a significant difference between 2 related variables. An example of this is if you where to collect the blood pressure for an individual before and after some treatment, condition, or time point. The data set contains blood pressure readings before and after an intervention. These are variables "bp_before" and "bp_after".

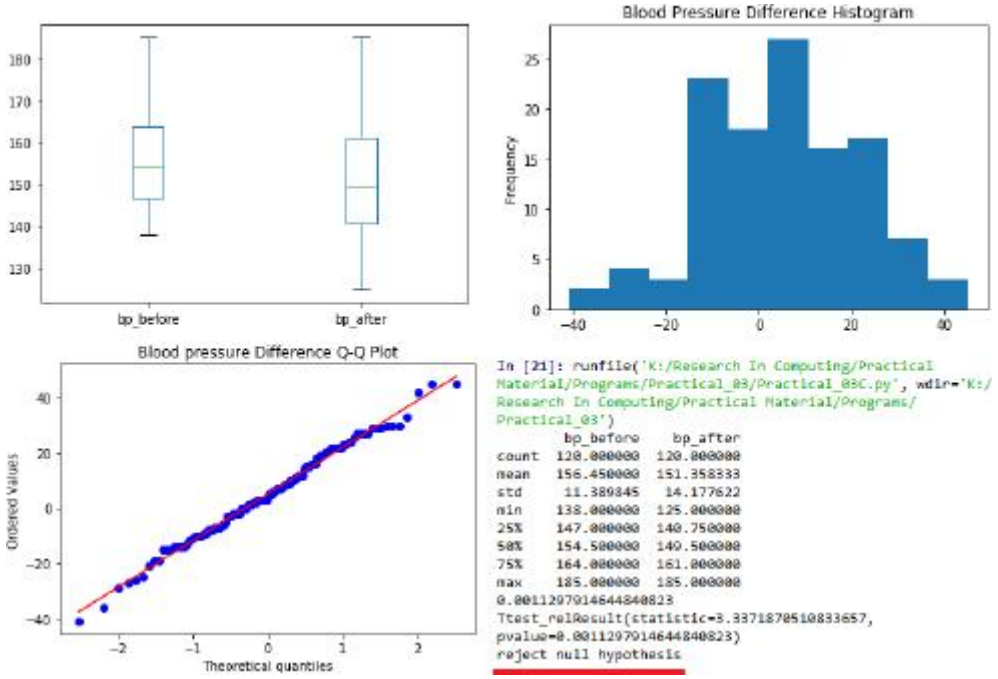
The hypothesis being test is:

- **H0** - The mean difference between sample 1 and sample 2 is equal to 0.
- **H0** - The mean difference between sample 1 and sample 2 is not equal to 0

Program 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:




```
In [21]: runfile('K:/Research In Computing/Practical
Material/Programs/Practical_03/Practical_03C.py', wdir='K:/
Research In Computing/Practical Material/Programs/
Practical_03')
          bp_before    bp_after
count   120.000000  120.000000
mean    156.450000  151.358333
std      11.389345   14.177622
min     138.000000  125.000000
25%     147.000000  140.750000
50%     154.500000  149.500000
75%     164.000000  161.000000
max     185.000000  185.000000
0.0011297914644840823
Ttest_relResult(statistic=3.3371870510833657,
pvalue=0.0011297914644840823)
reject null hypothesis
```

A paired sample t-test was used to analyze the blood pressure before and after the intervention to test if the intervention had a significant affect on the blood pressure. The blood pressure before the intervention was higher (156.45 ± 11.39 units) compared to the blood pressure post intervention (151.36 ± 14.18 units); there was a statistically significant decrease in blood pressure ($t(119)=3.34$, $p= 0.0011$) of 5.09 units.

Practical No. 4

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

Problem

An system administrator needs to upgrade the computers for his division. He wants to know what sort of computer system his 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

System	O	Ei	$\sum \frac{(O_i - E_i)^2}{Ei}$
Windows	20	33.33%	
Mac	60	33.33%	
Linux	20	33.33%	

H0 : The population distribution of the variable is the same as the proposed distribution

HA : The distributions are different

To calculate the Chi –Squred value for Windows go to cell D2 and type **=((B2-C2)*(B2-C2))/C2**

To calculate the Chi –Squred value for Mac go to cell D3 and type **=((B3-C3)*(B3-C3))/C3**

To calculate the Chi –Squred value for Mac go to cell D3 and type **=((B4-C4)*(B4-C4))/C4**

Go to Cell D5 for $\sum \frac{(O_i - E_i)^2}{Ei}$ and type=**SUM(D2:D4)**

To get the table value for Chi-Square for α = 0.05 and dof = 2, go to cell D7 and type =**CHIINV(0.05,2)**

At cell D8 type =**IF(D5>D7, "H0 Accepted","H0 Rejected")**

Output:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	System	O	Ei	$\sum \frac{(O_i - E_i)^2}{Ei}$										
2	Windows	20	33.33	5.333333		H₀ : The population distribution of the variable is the same as the proposed distribution								
3	Mac	60	33.33	21.33333		H1 - : The distributions are different								
4	Linux	20	33.33	5.333333										
5	Total	100	100	32										
6														
7			Table Value	5.991465										
8			H0 Accepted											

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

Sr. No	Roll No	Student's Name	Gen	Grade
1	1	Gaborone	m	O
2	2	Francistown	m	O
3	5	Niamey	m	O
4	13	Maxixe	m	O
5	16	Tema	m	O
6	17	Kumasi	m	O
7	34	Bida	m	O
8	35	Oran	m	O
9	38	Seefda	m	O
10	42	Constantine	m	O
11	43	Annaba	m	O
12	45	Bejaeia	m	O
13	48	Medea	m	O
14	49	Djelfa	m	O
15	50	Tipaza	m	O
16	51	Becher	m	O
17	54	Mostaganem	m	O
18	55	Tiaret	m	O
19	56	Bouira	m	O
20	59	Tebessa	m	O
21	61	El Harrach	m	O
22	62	Mila	m	O
23	65	Fouka	m	O
24	66	El Eulma	m	O
25	68	SidiBel Abbes	m	O
26	69	Jijel	m	O
27	70	Guelma	m	O
28	85	Khamis El Khechna	m	O
29	87	Bordj El Kiffan	m	O
30	88	Lakhdaria	m	O
31	6	Maputo	m	D
32	12	Lichinga	m	D
33	15	Resseno Garcia	m	D
34	19	Accra	m	D
35	27	Wa	m	D
36	28	Nevrongo	m	D
37	37	Mascara	m	D
38	44	Batna	m	D
39	57	El Biar	m	D
40	60	Boufarik	m	D
41	63	OuedRhiou	m	D
42	64	Souk Ahras	m	D
43	71	Dar El Befda	m	D
44	86	Birtouta	m	D
45	18	Takoradi	m	C
46	22	Cape Coast	m	C
47	29	Kwaberg	m	C
48	30	Algiers	m	C
49	31	Leghouat	m	C
50	39	Relizane	m	C
51	52	Setif	m	C
52	53	Biskra	m	C
53	67	Kolea	m	C
54	100	AefnFakroun	m	C
55	26	Nima	m	B
56	32	TiziOuzou	m	B
57	33	Chief	m	B

Sr. No	Roll No	Student's Name	Gen	Grade
62	3	Maun	f	O
63	7	Tete	f	O
64	9	Chimoio	f	O
65	11	Pemba	f	O
66	14	Chibuto	f	O
67	25	Mampong	f	O
68	36	Tlemcen	f	O
69	40	Adrar	f	O
70	41	Tindouf	f	O
71	46	Skikda	f	O
72	47	Ouargla	f	O
73	10	Matola	f	D
74	20	Lagos	f	D
75	21	Sunyani	f	D
76	72	Teenes	f	D
77	73	Koubs	f	D
78	75	HussenDey	f	D
79	77	Khenchela	f	D
80	82	HassiBahbah	f	D
81	84	Beraki	f	D
82	91	Boudouaou	f	D
83	95	Tadjenanet	f	D
84	4	Molepolole	f	C
85	8	Quelimane	f	C
86	23	Bolgatanga	f	C
87	58	Mohammadia	f	C
88	83	Mercuana	f	C
89	24	Ashaiman	f	B
90	76	N'gaous	f	B
91	90	Beb El Oued	f	B
92	92	BordjMenael	f	B
93	93	Kaar El Boukhari	f	B
94	74	Reghaia	f	A
95	78	Cheria	f	A
96	79	Mouzaia	f	A
97	80	Meskiana	f	A
98	81	Miliana	f	A
99	94	Sig	f	A
100	99	Kadiria	f	A

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.

Open Excel Workbook

	O	A	B	C	D	Total	$\sum \frac{(O_i - E_i)^2}{Ei}$
Girls	11	7	5	5	11	**39**	6.075
Boys	30	4	3	10	14	**61**	6.075
Total	41	11	8	15	25	**100**	**12.150**
Ei	**20.5**	**5.5**	**4**	**7.5**	**12.5**	**50**	

Prepare a contingency table as shown above.

To calculate Girls Students with ‘O’ Grade

Go to Cell N6 and type =COUNTIF($J$2:$K$40,"O")

To calculate Girls Students with ‘A’ Grade

Go to Cell O6 and type =COUNTIF($J$2:$K$40,"A")

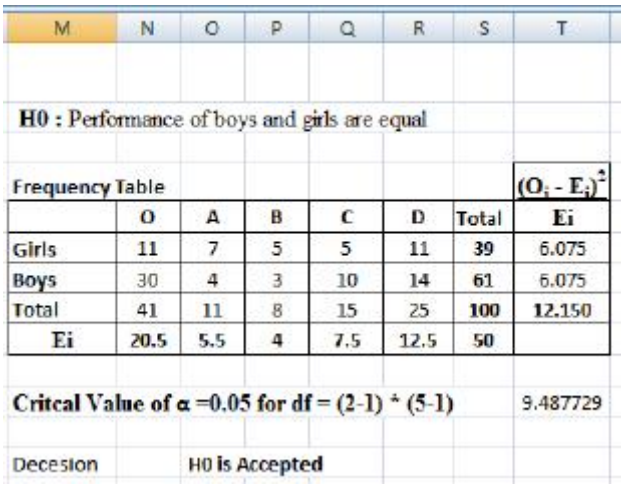
To calculate Girls Students with ‘B’ Grade
Go to Cell P6 and type =COUNTIF($J$2:$K$40,"B")
To calculate Girls Students with ‘C’ Grade
Go to Cell Q6 and type =COUNTIF($J$2:$K$40,"C")
To calculate Girls Students with ‘D’ Grade
Go to Cell R6 and type =COUNTIF($J$2:$K$40,"D")
To calculate Boys Students with ‘O’ Grade
Go to Cell N7 and type =COUNTIF($D$2:$E$62,"O")
To calculate Boys Students with ‘A’ Grade
Go to Cell O7 and type =COUNTIF($D$2:$E$62,"A")
To calculate Boys Students with ‘B’ Grade
Go to Cell P7 and type =COUNTIF($D$2:$E$62,"B")
To calculate Boys Students with ‘C’ Grade
Go to Cell Q7 and type =COUNTIF($D$2:$E$62,"C")

To calculate Boys Students with ‘D’ Grade
Go to Cell R7 and type =COUNTIF($D$2:$E$62,"D")

To calculated the expected value Ei
Go to Cell N9 and type =N8/2
Go to Cell O9 and type =O8/2
Go to Cell P9 and type =P8/2
Go to Cell Q9 and type =Q8/2
Go to Cell R9 and type =R8/2
Go to Cell S6 and calculate total girl students = **SUM(N6:R6)**
Go to Cell S7 and calculate total girl students = **SUM(N7:R7)**

Now Calculate
Go to cell **T6** and type
=SUM((N6-$N$9)^2/$N$9,(O6-$O$9)^2/$O$9,(P6-$P$9)^2/$P$9,(Q6-Q$9)^2/$Q$9, (R6-$R$9)^2/$R$9)
Go to cell **T7** and type
=SUM((N7-$N$9)^2/$N$9,(O7-$O$9)^2/$O$9,(P7-$P$9)^2/$P$9,(Q7-Q$9)^2/$Q$9, (R7-$R$9)^2/$R$9)
To get the table value go to cell T11 and type **=CHIINV(0.05,4)**
Go to cell O13 and type =IF(T8>=T11," H0 is Accepted", "H0 is Rejected")

Output :



H0 : Performance of boys and girls are equal

Frequency Table

	O	A	B	C	D	Total	$(O_i - E_i)^2$ / Ei
Girls	11	7	5	5	11	39	6.075
Boys	30	4	3	10	14	61	6.075
Total	41	11	8	15	25	100	12.150
Ei	20.5	5.5	4	7.5	12.5	50	

Critcal Value of α =0.05 for df = (2-1) * (5-1) 9.487729

Decesion **H0 is Accepted**

Using Python :

```
Import numpy as np
import pandas as pd
importscipy.stats as stats
np.random.seed(10)
stud_grade = np.random.choice(a=["O","A","B","C","D"],
p=[0.20, 0.20 ,0.20, 0.20, 0.20], size=100)
stud_gen = np.random.choice(a=["Male","Female"], p=[0.5, 0.5], size=100)
mscpart1 = pd.DataFrame({"Grades":stud_grade, "Gender":stud_gen})
print(mscpart1)
stud_tab = pd.crosstab(mscpart1.Grades, mscpart1.Gender, margins=True)
stud_tab.columns = ["Male", "Female", "row_totals"]
stud_tab.index = ["O", "A", "B", "C", "D", "col_totals"]
observed = stud_tab.iloc[0:5, 0:2 ]
print(observed)
expected = np.outer(stud_tab["row_totals"][0:5],
stud_tab.loc["col_totals"][0:2]) / 100
print(expected)
chi_squared_stat = (((observed-expected)**2)/expected).sum().sum()
print('Calculated : ',chi_squared_stat)
crit = stats.chi2.ppf(q=0.95, df=4)
print('Table Value : ',crit)
ifchi_squared_stat>= crit:
print('H0 is Accepted ')
else:
print('H0 is Rejected ')
```

Output :

```
In [1]: runfile('E:/Research In Computing/Programs/
Practical_03/ChiSquaer.py', wdir='E:/Research In
Computing/Programs/Practical_03')
   Grades  Gender
0       C  Female
1       O  Female
2       C    Male
3       C    Male
4       B  Female
..    ...     ...
95      B    Male
96      D  Female
97      B  Female
98      A    Male
99      B    Male

[100 rows x 2 columns]
   Male  Female
O    11      12
A     9      13
B     7      11
C    10       8
D    12       7
[[11.27 11.73]
 [10.78 11.22]
 [ 8.82  9.18]
 [ 8.82  9.18]
 [ 9.31  9.69]]
Calculated :  3.158915138993211
Table Value :  9.487729036781154
H0 is Rejected
```

Practical5

Perform testing of hypothesis using Z-test.

Use a Z test if:

• Your sample size is greater than 30. Otherwise, use a t test.

• Data points should be independent from each other. In other words, one data point isn't related or doesn't affect another data point.

• Your data should be normally distributed. However, for large sample sizes (over 30) this doesn't always matter.

• Your data should be randomly selected from a population, where each item has an equal chance of being selected.

• Sample sizes should be equal if at all possible.

Ho - Blood pressure has a mean of 156 units

Program Code for one-sample Z test.

```
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:

```
In [26]: runfile('K:/Research In Computing/Practical
Material/Programs/Practical_05/Z_Test_One_Sample.py',
wdir='K:/Research In Computing/Practical Material/Programs/
Practical_05')
     patient  gender agegrp  bp_before  bp_after
0          1    Male  30-45        143       153
1          2    Male  30-45        163       170
2          3    Male  30-45        153       168
3          4    Male  30-45        153       142
4          5    Male  30-45        146       141
..       ...     ...    ...        ...       ...
115      116  Female    60+        152       152
116      117  Female    60+        161       152
117      118  Female    60+        165       174
118      119  Female    60+        149       151
119      120  Female    60+        185       163

[120 rows x 5 columns]
0.6651614730255063
accept null hypothesis
```

Two-sample Z test- In two sample z-test , similar to t-test here we are checking two independent data groups and deciding whether sample mean of two group is equal or not.

H0 : mean of two group is 0

H1 : mean of two group is not 0

```
# -*- coding: utf-8 -*-
"""
Created on Mon Dec 16 20:42:17 2019
@author: MyHome
"""
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,alternative='two-sided')
print(float(pval))
if pval<0.05:
print("reject null hypothesis")
else:
print("accept null hypothesis")
```

Output:

```
In [29]: runfile('K:/Research In Computing/Practical
Material/Programs/Practical_05/Z_Test_Two_Sample.py',
wdir='K:/Research In Computing/Practical Material/Programs/
Practical_05')
     patient  gender agegrp  bp_before  bp_after
0          1    Male  30-45        143       153
1          2    Male  30-45        163       170
2          3    Male  30-45        153       168
3          4    Male  30-45        153       142
4          5    Male  30-45        146       141
..       ...     ...    ...        ...       ...
115      116  Female    60+        152       152
116      117  Female    60+        161       152
117      118  Female    60+        165       174
118      119  Female    60+        149       151
119      120  Female    60+        185       163

[120 rows x 5 columns]
0.002162306611369422
reject null hypothesis
```

Practical No. 6

A. Perform testing of hypothesis using One-way ANOVA.
ANOVA Assumptions

The dependent variable (SAT scores in our example) should be continuous.
• The independent variables (districts in our example) should be two or more categorical groups.
• There must be different participants in each group with no participant being in more than one group. In our case, each school cannot be in more than one district.
• The dependent variable should be approximately normally distributed for each category.
• Variances of each group are approximately equal.

From our data exploration, we can see that the average SAT scores are quite different for each district. Since we have five different groups, we cannot use the t-test, use the 1-way ANOVA test anyway just to understand the concepts.

H0 - There are no significant differences between the groups' mean SAT scores.
$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$

H1 - There is a significant difference between the groups' mean SAT scores.

If there is at least one group with a significant difference with another group, the null hypothesis will be rejected.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy
import stats
data = pd.read_csv("scores.csv")
data.head()
data['Borough'].value_counts()
```

################ There is no total score column, have to create it.
########In addition, find the mean score of the each district across all schools.

```
data['total_score'] = data['Average Score (SAT Reading)'] + \
data['Average Score (SAT Math)'] + \
data['Average Score (SAT Writing)']

data = data[['Borough', 'total_score']].dropna()
x = ['Brooklyn', 'Bronx', 'Manhattan', 'Queens', 'Staten Island']
district_dict = {}
#Assigns each test score series to a dictionary key
for district in x:
district_dict[district] = data[data['Borough'] == district]['total_score']
y = []
yerror = []
#Assigns the mean score and 95% confidence limit to each district
for district in x:
```

```
y.append(district_dict[district].mean())
yerror.append(1.96*district_dict[district].std()/np.sqrt(district_dict[district].shape[0]))
print(district + '_std : {}'.format(district_dict[district].std()))
sns.set(font_scale=1.8)
fig = plt.figure(figsize=(10,5))
ax = sns.barplot(x, y, yerr=yerror)
ax.set_ylabel('Average Total SAT Score')
plt.show()
```

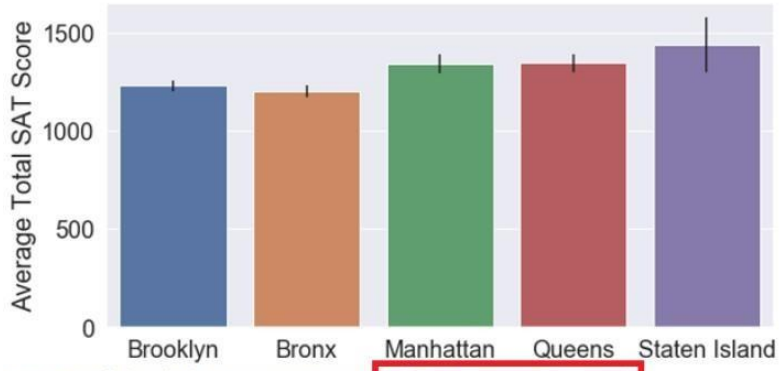
########################## **Perform 1-way ANOVA**

```
print(stats.f_oneway(
district_dict['Brooklyn'], district_dict['Bronx'], \
district_dict['Manhattan'], district_dict['Queens'], \
district_dict['Staten Island']
))
districts = ['Brooklyn', 'Bronx', 'Manhattan', 'Queens', 'Staten Island']
ss_b = 0
for d in districts:
ss_b += district_dict[d].shape[0] * \
np.sum((district_dict[d].mean() - data['total_score'].mean())**2)
ss_w = 0
for d in districts:
ss_w += np.sum((district_dict[d] - district_dict[d].mean())**2)

msw = ss_w/(len(data)-5)
f=msb/msw
print('F_statistic: {}'.format(f))
ss_t = np.sum((data['total_score']-data['total_score'].mean())**2)
eta_squared = ss_b/ss_t
print('eta_squared: {}'.format(eta_squared))
```

Output :

```
In [37]: runfile('E:/Research In Computing/Programs/Practical_05/Annova.py', wdir='E:/Research In
Computing/Programs/Practical_05')
Brooklyn_std : 154.8684270520867
Bronx_std : 150.39390071890668
Manhattan_std : 230.2941395363782
Queens_std : 195.25289850192115
Staten Island_std : 222.30359621222706
```




```
F_onewayResult(statistic=12.733085029201668, pvalue=1.0161974965566023e-09)
F_statistic: 12.733085029201687
eta_squared: 0.12099887621529214
```

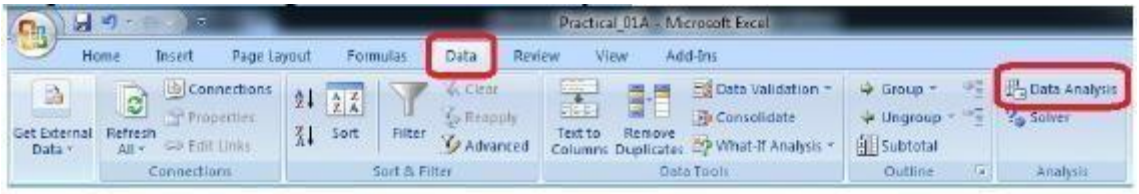
Using Excel

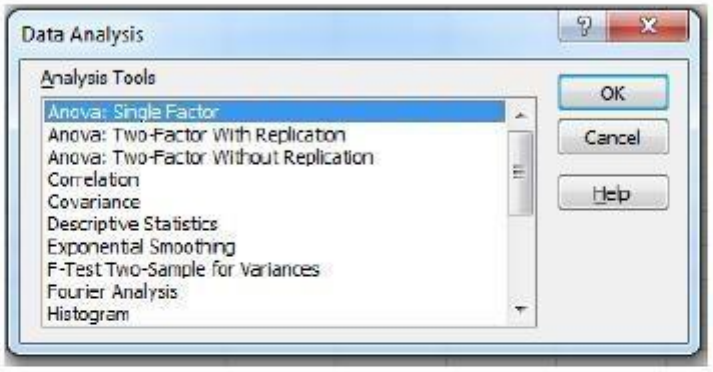
H0 - There are no significant differences between the Subject's mean SAT scores.

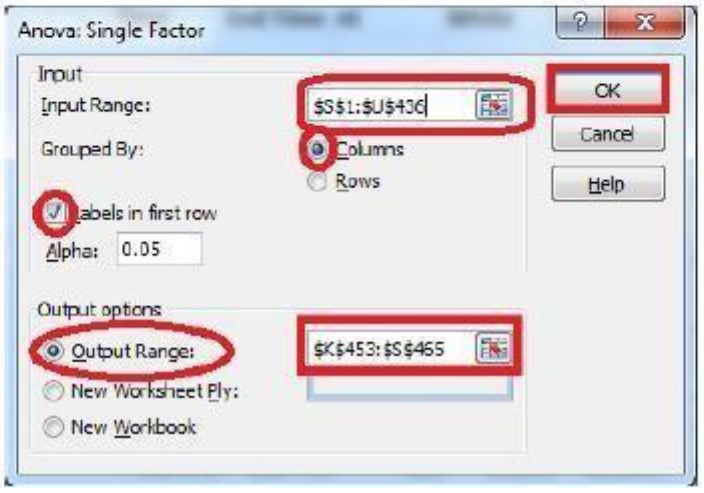
μ1 = μ2 = μ3 = μ4 = μ5

H1 - There is a significant difference between the Subject's mean SAT scores.

To perform ANOVA go to data Data Analysis







Input Range : $S$1:$U$436*(Select columns to be analyzed in group)*

Output Range :$K$453:$S$465*(Can be any Range)*

Anova: Single Factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Average Score (SAT Math)	375	162354	432.944	5177.144		
Average Score (SAT Reading)	375	159189	424.504	3829.267		
Average Score (SAT Writing)	375	156922	418.4587	4166.522		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	39700.57	2	19850.28	4.520698	0.01108	3.003745
Within Groups	4926677	1122	4390.977			
Total	4966377	1124				

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

Program Code:

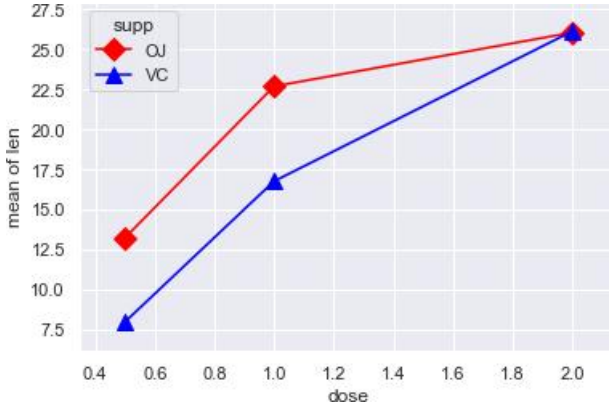
```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
from statsmodels.graphics.factorplots import interaction_plot
import matplotlib.pyplot as plt
from scipy import stats
def eta_squared(aov):
aov['eta_sq'] = 'NaN'
aov['eta_sq'] = aov[:-1]['sum_sq']/sum(aov['sum_sq'])
return aov
def omega_squared(aov):
mse = aov['sum_sq'][-1]/aov['df'][-1]
aov['omega_sq'] = 'NaN'
aov['omega_sq'] = (aov[:-1]['sum_sq']-(aov[:-1]['df']*mse))/(sum(aov['sum_sq'])+mse)
return aov
datafile = "ToothGrowth.csv"
data = pd.read_csv(datafile)
fig = interaction_plot(data.dose, data.supp, data.len,
colors=['red','blue'], markers=['D','^'], ms=10)
N = len(data.len)
df_a = len(data.supp.unique()) - 1
df_b = len(data.dose.unique()) - 1
df_axb = df_a*df_b
df_w = N - (len(data.supp.unique())*len(data.dose.unique()))
grand_mean = data['len'].mean()
#Sum of Squares A – supp
ssq_a = sum([(data[data.supp ==l].len.mean()-grand_mean)**2 for l in data.supp])
#Sum of Squares B – supp
ssq_b = sum([(data[data.dose ==l].len.mean()-grand_mean)**2 for l in data.dose])
#Sum of Squares Total
ssq_t = sum((data.len - grand_mean)**2)
vc = data[data.supp == 'VC']
oj = data[data.supp == 'OJ']
vc_dose_means = [vc[vc.dose == d].len.mean() for d in vc.dose]
oj_dose_means = [oj[oj.dose == d].len.mean() for d in oj.dose]
ssq_w = sum((oj.len - oj_dose_means)**2) +sum((vc.len - vc_dose_means)**2)
ssq_axb = ssq_t-ssq_a-ssq_b-ssq_w
ms_a = ssq_a/df_a #Mean Square A
ms_b = ssq_b/df_b #Mean Square B
ms_axb = ssq_axb/df_axb #Mean Square AXB
ms_w = ssq_w/df_w
f_a = ms_a/ms_w
f_b = ms_b/ms_w
```

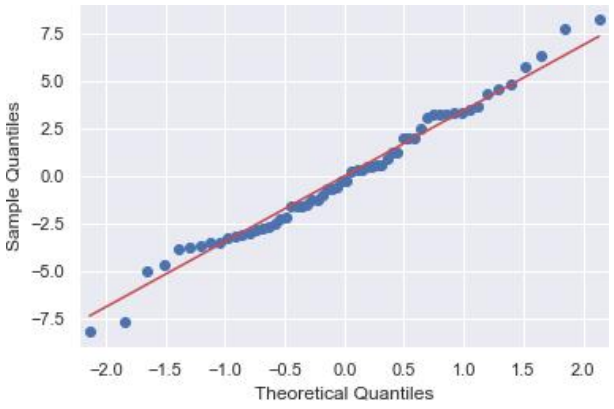
```
f_axb = ms_axb/ms_w
p_a = stats.f.sf(f_a, df_a, df_w)
p_b = stats.f.sf(f_b, df_b, df_w)
p_axb = stats.f.sf(f_axb, df_axb, df_w)
results = {'sum_sq':[ssq_a, ssq_b, ssq_axb, ssq_w],
'df':[df_a, df_b, df_axb, df_w],
'F':[f_a, f_b, f_axb, 'NaN'],
'PR(>F)':[p_a, p_b, p_axb, 'NaN']}
columns=['sum_sq', 'df', 'F', 'PR(>F)']
aov_table1 = pd.DataFrame(results, columns=columns,
index=['supp', 'dose',
'supp:dose', 'Residual'])
formula = 'len ~ C(supp) + C(dose) + C(supp):C(dose)'
model = ols(formula, data).fit()
aov_table = anova_lm(model, typ=2)
eta_squared(aov_table)
omega_squared(aov_table)
print(aov_table.round(4))
res = model.resid
fig = sm.qqplot(res, line='s')
plt.show()
```

Output:

```
In [40]: runfile('K:/Research In Computing/Practical Material/Programs/
Practical_06/Annova_2_Way.py', wdir='K:/Research In Computing/Practical
Material/Programs/Practical_06')
                   sum_sq    df       F  PR(>F)  eta_sq  omega_sq
C(supp)           205.3500   1.0  15.572  0.0002  0.0595    0.0555
C(dose)          2426.4343   2.0  92.000  0.0000  0.7029    0.6926
C(supp):C(dose)   108.3190   2.0   4.107  0.0219  0.0314    0.0236
Residual          712.1060  54.0     NaN     NaN     NaN       NaN
```

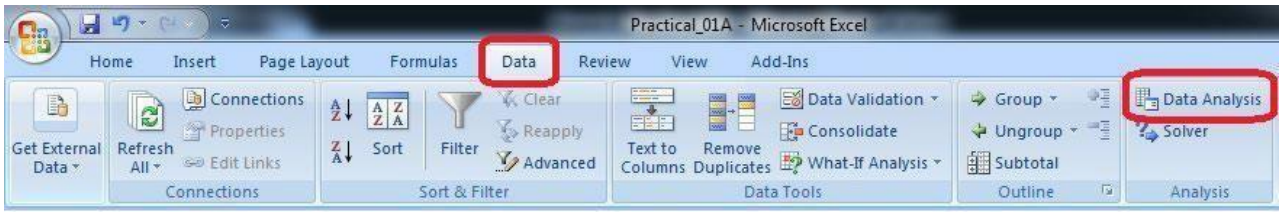
0

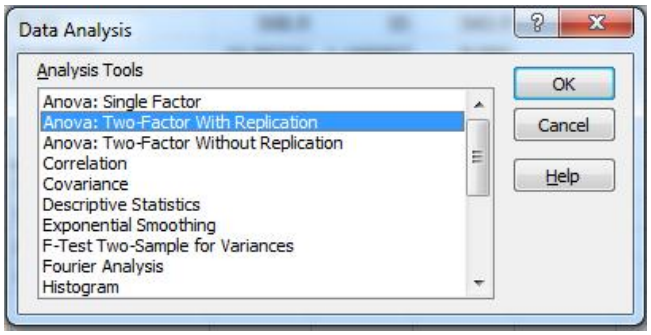


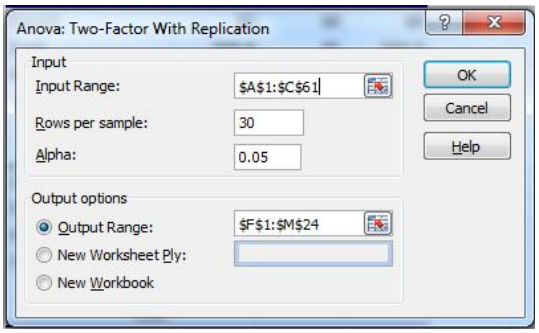


Using Excel:

Go to Data tab → Data Analysis







Rows Per Sample – 30 (Beacause 30 Patients are given each dose)

Alpha – 0.05

Output Range - $F$1:$M$24

Output:

Anova: Two-Factor With Replication						
SUMMARY	len	dose	Total			
1						
Count	30	30	60			
Sum	508.9	35	543.9			
Average	16.96333	1.166667	9.065			
Variance	68.32723	0.402299	97.22333			
31						
Count	30	30	60			
Sum	619.9	35	654.9			
Average	20.66333	1.166667	10.915			
Variance	43.63344	0.402299	118.2854			
Total						
Count	60	60				
Sum	1128.8	70				
Average	18.81333	1.166667				
Variance	58.51202	0.39548				
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Sample	102.675	1	102.675	3.642079	0.058808	3.922879
Columns	9342.145	1	9342.145	331.3838	8.55E-36	3.922879
Interaction	102.675	1	102.675	3.642079	0.058808	3.922879
Within	3270.193	116	28.19132			
Total	12817.69	119				

P-value = 0.0588079 column in the ANOVA Source of Variation table at the bottom of the output. Because the p-values for both medicin dose and interaction are less than our significance level, these factors are statistically significant. On the other hand, the interaction effect is not significant because its p-value (0.0588) is greater than our significance level. Because the interaction effect is not significant, we can focus on only the main effects and not consider the interaction effect of the dose.

C. Perform testing of hypothesis using MANOVA.

MANOVA is the acronym for Multivariate Analysis of Variance. When analyzing data, we may encounter situations where we have there multiple response variables (dependent variables). In MANOVA there also some assumptions, like ANOVA. Before performing MANOVA we have to check the following assumptions are satisfied or not.

- The samples, while drawing, should be independent of each other.
- The dependent variables are continuous in nature and the independent variables are categorical.
- The dependent variables should follow a multivariate normal distribution.
- The population variance-covariance matrices of each group are same, i.e. groups are homogeneous.

Code:

```
import pandas as pd
from statsmodels.multivariate.manova
import MANOVA
df = pd.read_csv('iris.csv', index_col=0)
df.columns = df.columns.str.replace(".", "_")
df.head()
print('~~~~~~~~~ Data Set ~~~~~~~~~')
print(df)
maov = MANOVA.from_formula('Sepal_Length + Sepal_Width + \
Petal_Length + Petal_Width ~ Species', data=df)
print('~~~~~~~~~ MANOVA Test Result ~~~~~~~~~')
print(maov.mv_test())
```

Output:

```
In [42]: runfile('E:/Research In Computing/Programs/Practical_10/Manova_Test.py', wdir='E:/Research
In Computing/Programs/Practical_10')
~~~~~~~~~ Data Set ~~~~~~~~~
     Sepal_Length  Sepal_Width  Petal_Length  Petal_Width    Species
1             5.1          3.5           1.4          0.2     setosa
2             4.9          3.0           1.4          0.2     setosa
3             4.7          3.2           1.3          0.2     setosa
4             4.6          3.1           1.5          0.2     setosa
5             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
146           6.7          3.0           5.2          2.3  virginica
147           6.3          2.5           5.0          1.9  virginica
148           6.5          3.0           5.2          2.0  virginica
149           6.2          3.4           5.4          2.3  virginica
150           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
~~~~~~~~~ MANOVA Test Result ~~~~~~~~~
                  Multivariate linear model
================================================================

----------------------------------------------------------------
       Intercept         Value  Num DF  Den DF   F Value  Pr > F
----------------------------------------------------------------
          Wilks' lambda  0.0170 4.0000 144.0000 2086.7720 0.0000
         Pillai's trace  0.9830 4.0000 144.0000 2086.7720 0.0000
 Hotelling-Lawley trace 57.9659 4.0000 144.0000 2086.7720 0.0000
    Roy's greatest root 57.9659 4.0000 144.0000 2086.7720 0.0000
----------------------------------------------------------------

----------------------------------------------------------------
        Species          Value  Num DF  Den DF   F Value  Pr > F
----------------------------------------------------------------
          Wilks' lambda  0.0234 8.0000 288.0000  199.1453 0.0000
         Pillai's trace  1.1919 8.0000 290.0000   53.4665 0.0000
 Hotelling-Lawley trace 32.4773 8.0000 203.4024  582.1970 0.0000
    Roy's greatest root 32.1919 4.0000 145.0000 1166.9574 0.0000
================================================================
```

Practical No. 7

A: Perform the Random sampling for the given data and analyse it.

Example 1: From a population of 10 women and 10 men as given in the table in Figure 1 on the left below, create a random sample of 6 people for Group 1 and a periodic sample consisting of every 3rd woman for Group 2.

You need to run the sampling data analysis tool twice, once to create Group 1 and again to create Group 2. For Group 1 you select all 20 population cells as the Input Range and Random as the Sampling Method with 6 for the Random Number of Samples. For Group 2 you select the 10 cells in the Women column as Input Range and Periodic with Period 3.

Open existing excel sheet with population data

Sample Sheet looks as given below:

	A	B	C	D	E	F	G	H	I	J	K
1	Sr. No	Roll No	Student's Name	Gender	Grade		Sr. No	Roll No	Student's Name	Gender	Grade
2	1	1	Gaborone	m	O		62	3	Maun	f	O
3	2	2	Francistown	m	O		63	7	Tete	f	O
4	3	5	Niamey	m	O		64	9	Chimoio	f	O
5	4	13	Maxixe	m	O		65	11	Pemba	f	O
6	5	16	Tema	m	O		66	14	Chibuto	f	O
7	6	17	Kumasi	m	O		67	25	Mampong	f	O
8	7	34	Blida	m	O		68	36	Tlemcen	f	O
9	8	35	Oran	m	O		69	40	Adrar	f	O
10	9	38	Saefda	m	O		70	41	Tindouf	f	O
11	10	42	Constantine	m	O		71	46	Skikda	f	O
12	11	43	Annaba	m	O		72	47	Ouargla	f	O
13	12	45	Bejaefa	m	O		73	10	Matola	f	D
14	13	48	Medea	m	O		74	20	Legon	f	D
15	14	49	Djelfa	m	O		75	21	Sunyani	f	D
16	15	50	Tipaza	m	O		76	72	Teenas	f	D
17	16	51	Bechar	m	O		77	73	Kouba	f	D
18	17	54	Mostaganem	m	O		78	75	Hussen Dey	f	D
19	18	55	Tiaret	m	O		79	77	Khenchela	f	D
20	19	56	Boulra	m	O		80	82	Hassi Bahbah	f	D
21	20	59	Tebessa	m	O		81	84	Baraki	f	D
22	21	61	El Harrach	m	O		82	91	Boudouaou	f	D
23	22	62	Mila	m	O		83	95	Tadjenanet	f	D
24	23	65	Fouka	m	O		84	4	Molepolole	f	C

Set Cell O1 = Male and Cell O2 = 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))**

Drag the formula to the desired no of cell to select random sample

Now, to generate a random sample for female students go to cell P1 and type =**INDEX(K$2:K$40,RANK(H2,H$2:H$40))**

Drag the formula to the desired no of cell to select random sample.

Output:

O	P
Male	Female
A	A
A	A
A	A
B	A
C	B
C	C
D	C
D	C
D	C
D	C
D	D
D	A
D	B
D	B
O	D
O	D
O	D
O	D
O	O
O	O
O	O
O	O
O	A

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

we are to carry out a **hypothetical** housing quality survey across Lagos state, Nigeria. And we looking at a total of 5000 houses (hypothetically). We don't just go to one local government and select 5000 houses, rather we ensure that the 5000 houses are a representative of the whole 20 local government areas Lagos state is comprised of. This is called stratified sampling. The population is divided into homogenous strata and the right number of instances is sampled from each stratum to guarantee that the test-set (which in this case is the 5000 houses) is a representative of the overall population. If we used random sampling, there would be a significant chance of having bias in the survey results

Program Code:

```
import pandas as pd
importnumpy as np
importmatplotlib
importmatplotlib.pyplot as plt
plt.rcParams['axes.labelsize'] = 14
plt.rcParams['xtick.labelsize'] = 12
plt.rcParams['ytick.labelsize'] = 12
importseaborn as sns
color = sns.color_palette()
sns.set_style('darkgrid'
```

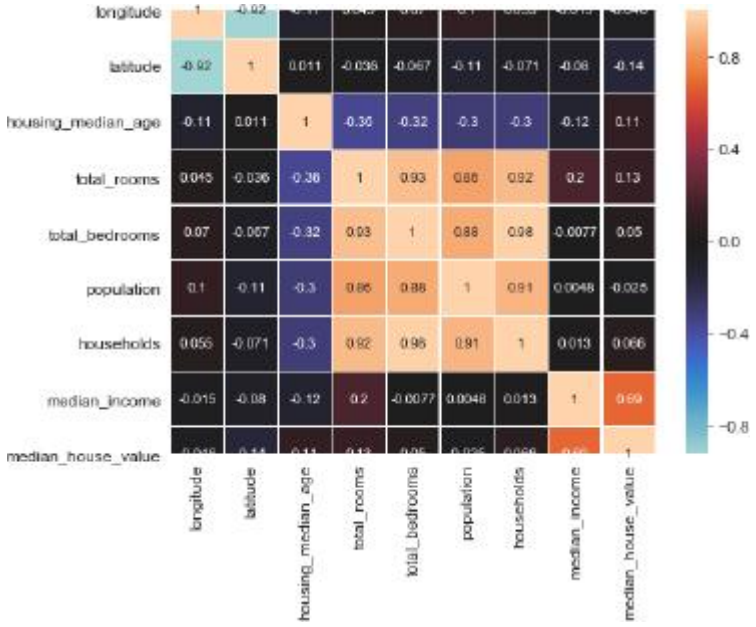
```
importsklearn
fromsklearn.model_selection import train_test_split
housing =pd.read_csv('housing.csv')
print(housing.head())
print(housing.info())
#creating a heatmap of the attributes in the dataset

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:

```
In [28]: runfile('J:/Research In Computing/Practical Material/Programs/Practical_05/
Stratified_Sample.py', wdir='J:/Research In Computing/Practical Material/Programs/Practical_05')
   longitude  latitude  ...  median_house_value  ocean_proximity
0    -122.23     37.88  ...            452600.0         NEAR BAY
1    -122.22     37.86  ...            358500.0         NEAR BAY
2    -122.24     37.85  ...            352100.0         NEAR BAY
3    -122.25     37.85  ...            341300.0         NEAR BAY
4    -122.25     37.85  ...            342200.0         NEAR BAY

[5 rows x 10 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
longitude             20640 non-null float64
latitude              20640 non-null float64
housing_median_age    20640 non-null float64
total_rooms           20640 non-null float64
total_bedrooms        20433 non-null float64
population            20640 non-null float64
households            20640 non-null float64
median_income         20640 non-null float64
median_house_value    20640 non-null float64
ocean_proximity       20640 non-null object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
None
median_house_value    1.000000
median_income         0.688075
total_rooms           0.134153
housing_median_age    0.105623
households            0.065843
total_bedrooms        0.049686
population           -0.024650
longitude            -0.045967
latitude             -0.144160
Name: median_house_value, dtype: float64
```



Practical No. 8

A: Write a program for computing different correlation.

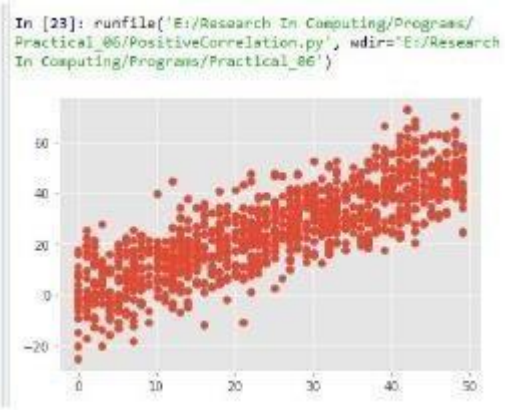
Positive Correlation:

Let's take a look at a positive correlation. Numpy implements a corrcoef() function that returns a matrix of correlations of x with x, x with y, y with x and y with y. We're interested in the values of correlation of x with y (so position (1, 0) or (0, 1)).

Code:

```
importnumpy as np
importmatplotlib.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:



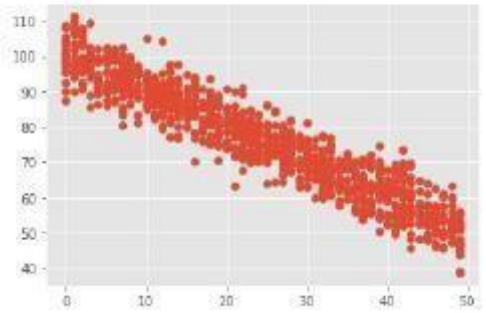
B: Negative Correlation:

```
importnumpy as np
importmatplotlib.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 :

```
In [24]: runfile('E:/Research In Computing/Programs/
Practical_06/NegativeCorrelation.py', wdir='E:/Research
In Computing/Programs/Practical_06')
```



C: No/Weak Correlation:

```
importnumpy as np
importmatplotlib.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:

```
In [25]: runfile('E:/Research In Computing/Programs/
Practical_06/No_or_Weak_Correlation.py', wdir='E:/
Research In Computing/Programs/Practical_06')
```