Assignment-03-Q1 (Hypothesis Testing)

In [ ]:

```
# import pandas as pd
import numpy as np
from scipy import stats
from scipy.stats import norm
```

In [4]:

```
# Load the dataset
data=pd.read_csv('Cutlets.csv')
data.head()
```

Out[4]:

	Unit A	Unit B
0	6.8090	6.7703
1	6.4376	7.5093
2	6.9157	6.7300
3	7.3012	6.7878
4	7.4488	7.1522

In [5]:

```
unitA=pd.Series(data.iloc[:,0])
unitA
```

Out[5]:

```
0     6.8090
1     6.4376
2     6.9157
3     7.3012
4     7.4488
5     7.3871
6     6.8755
7     7.0621
8     6.6840
9     6.8236
10    7.3930
11    7.5169
12    6.9246
13    6.9256
14    6.5797
15    6.8394
16    6.5970
17    7.2705
18    7.2828
19    7.3495
20    6.9438
21    7.1560
22    6.5341
23    7.2854
24    6.9952
25    6.8568
26    7.2163
27    6.6801
28    6.9431
29    7.0852
30    6.7794
31    7.2783
32    7.1561
33    7.3943
34    6.9405
Name: Unit A, dtype: float64
```

In [6]:

```
unitB=pd.Series(data.iloc[:,1])
unitB
```

Out[6]:

```
0     6.7703
1     7.5093
2     6.7300
3     6.7878
4     7.1522
5     6.8110
6     7.2212
7     6.6606
8     7.2402
9     7.0503
10    6.8810
11    7.4059
12    6.7652
13    6.0380
14    7.1581
15    7.0240
16    6.6672
17    7.4314
18    7.3070
19    6.7478
20    6.8889
21    7.4220
22    6.5217
23    7.1688
24    6.7594
25    6.9399
26    7.0133
27    6.9182
28    6.3346
29    7.5459
30    7.0992
31    7.1180
32    6.6965
33    6.5780
34    7.3875
Name: Unit B, dtype: float64
```

In [7]:

```
# 2-sample 2-tail ttest:   stats.ttest_ind(array1,array2)     # ind -> independent sampl
p_value=stats.ttest_ind(unitA,unitB)
p_value
```

Out[7]:

```
Ttest_indResult(statistic=0.7228688704678063, pvalue=0.4722394724599501)
```

In [8]:

```
p_value[1]     # 2-tail probability
```

Out[8]:

```
0.4722394724599501
```

In [9]:

```
# compare p_value with α = 0.05 (At 5% significance level)
```

Assignment-03-Q2 (Hypothesis Testing)

In [10]:

```
import pandas as pd
import numpy as np
from scipy import stats
from scipy.stats import norm
```

In [11]:

```
# load the dataset
data=pd.read_csv('LabTAT.csv')
data.head()
```

Loading [MathJax]/extensions/Safe.js

Out[11]:

	Laboratory 1	Laboratory 2	Laboratory 3	Laboratory 4
0	185.35	165.53	176.70	166.13
1	170.49	185.91	198.45	160.79
2	192.77	194.92	201.23	185.18
3	177.33	183.00	199.61	176.42
4	193.41	169.57	204.63	152.60

In [12]:

```
# Anova ftest statistics: stats.f_oneway(column-1,column-2,column-3,column-4)
p_value=stats.f_oneway(data.iloc[:,0],data.iloc[:,1],data.iloc[:,2],data.iloc[:,3])
p_value
```

Out[12]:

```
F_onewayResult(statistic=118.70421654401437, pvalue=2.1156708949992414e-57)
```

In [13]:

```
p_value[1]  # compare it with α = 0.05
```

Out[13]:

```
2.1156708949992414e-57
```

Assignment-03-Q3 (Hypothesis Testing)

In [14]:

```
import pandas as pd
from scipy import stats as stats
import numpy as np
```

In [15]:

```
df= pd.read_csv('BuyerRatio.csv')
```

In [16]:

```
df.head()
```

Out[16]:

	Observed Values	East	West	North	South
0	Males	50	142	131	70
1	Females	435	1523	1356	750

In [17]:

```
df_table=df.iloc[:,1:6]
df_table
```

Out[17]:

	East	West	North	South
0	50	142	131	70
1	435	1523	1356	750

In [18]:

```
df_table.values
```

Out[18]:

```
array([[  50,  142,  131,   70],
       [ 435, 1523, 1356,  750]], dtype=int64)
```

In [20]:

```
val=stats.chi2_contingency(df_table)
```

In [21]:

```
val
```

Out[21]:

```
(1.595945538661058,
 0.6603094907091882,
 3,
 array([[  42.76531299,   146.81287862,   131.11756787,    72.30424052],
Loading [MathJax]/extensions/Safe.js .23468701, 1518.18712138, 1355.88243213,   747.69575948]]))
```

```
In [22]: type(val)

Out[22]: tuple
```

```
In [23]: no_of_rows=len(df_table.iloc[0:2,0])
         no_of_columns=len(df_table.iloc[0,0:4])
         degree_of_f=(no_of_rows-1)*(no_of_columns-1)
         print('Degree of Freedom=',degree_of_f)

         Degree of Freedom= 3
```

```
In [24]: Expected_value=val[3]
```

```
In [25]: Expected_value

Out[25]: array([[  42.76531299,  146.81287862,  131.11756787,   72.30424052],
                [ 442.23468701, 1518.18712138, 1355.88243213,  747.69575948]])
```

```
In [26]: from scipy.stats import chi2
         chi_square=sum([(o-e)**2/e for o,e in zip(df_table.values,Expected_value)])
         chi_square_statestic=chi_square[0]+chi_square[1]
         chi_square_statestic

Out[26]: 1.5152956451130446
```

```
In [27]: critical_value=chi2.ppf(0.95,3)
         critical_value

Out[27]: 7.814727903251179
```

```
In [28]: if chi_square_statestic >= critical_value:
                 print('Dependent (reject H0)')
         else:
                 print('Independent (fail to reject H0)')

         Independent (fail to reject H0)
```

```
In [29]: pvalue=1-chi2.cdf(chi_square_statestic,3)
         pvalue

Out[29]: 0.6787446296467897
```

```
In [30]: if pvalue <= 0.05:
                 print('Dependent (reject H0)')
         else:
                 print('Independent (fail to reject H0)')

         Independent (fail to reject H0)
```

```
In [31]: no_of_columns

Out[31]: 4
```

```
In [32]: no_of_rows

Out[32]: 2
```

```
In [33]: df_table=pd.crosstab(df['East'],df['Observed Values'])
         df_table
```

Loading [MathJax]/extensions/Safe.js

Out[33]:

Observed Values	Females	Males
East		
50	0	1
435	1	0

In [34]:

```
df_table.values
```

Out[34]:

```
array([[0, 1],
       [1, 0]], dtype=int64)
```

Assignment-03-Q4 (Hypothesis Testing)

In [35]:

```
import pandas as pd
import numpy as np
from scipy import stats
from scipy.stats import norm
from scipy.stats import chi2_contingency
```

In [36]:

```
# load the dataset
data=pd.read_csv('Costomer+OrderForm.csv')
data
```

Out[36]:

	Phillippines	Indonesia	Malta	India
0	Error Free	Error Free	Defective	Error Free
1	Error Free	Error Free	Error Free	Defective
2	Error Free	Defective	Defective	Error Free
3	Error Free	Error Free	Error Free	Error Free
4	Error Free	Error Free	Defective	Error Free
...	...	...	...	...
295	Error Free	Error Free	Error Free	Error Free
296	Error Free	Error Free	Error Free	Error Free
297	Error Free	Error Free	Defective	Error Free
298	Error Free	Error Free	Error Free	Error Free
299	Error Free	Defective	Defective	Error Free

300 rows × 4 columns

In [37]:

```
data.Phillippines.value_counts()
```

Out[37]:

```
Error Free    271
Defective      29
Name: Phillippines, dtype: int64
```

In [38]:

```
data.Indonesia.value_counts()
```

Out[38]:

```
Error Free    267
Defective      33
Name: Indonesia, dtype: int64
```

In [39]:

```
data.Malta.value_counts()
```

Out[39]:

```
Error Free    269
Defective      31
Name: Malta, dtype: int64
```

In [43]:

```python
data.India.value_counts()
```

Out[43]:

```
Error Free    280
Defective      20
Name: India, dtype: int64
```

In [41]:

```python
# Make a contingency table
obs=np.array([[271,267,269,280],[29,33,31,20]])
obs
```

Out[41]:

```
array([[271, 267, 269, 280],
       [ 29,  33,  31,  20]])
```

In [42]:

```python
# Chi2 contengency independence test
chi2_contingency(obs) # o/p is (Chi2 stats value, p_value, df, expected obsvations)
```

Out[42]:

```
(3.858960685820355,
 0.2771020991233135,
 3,
 array([[271.75, 271.75, 271.75, 271.75],
        [ 28.25,  28.25,  28.25,  28.25]]))
```

In [44]:

```python
# Compare p_value with α = 0.05
```

In [45]:

```python
obs
```

Out[45]:

```
array([[271, 267, 269, 280],
       [ 29,  33,  31,  20]])
```

In [47]:

```python
stat, p, dof, expected = chi2_contingency([[271,267,269,280],[29,33,31,20]])
```

In [48]:

```python
stat
```

Out[48]:

```
3.858960685820355
```

In [49]:

```python
p
```

Out[49]:

```
0.2771020991233135
```

In [50]:

```python
print('dof=%d' % dof)
print(expected)
```

```
dof=3
[[271.75 271.75 271.75 271.75]
 [ 28.25  28.25  28.25  28.25]]
```

In [51]:

```python
alpha = 0.05
prob=1-alpha
critical = chi2.ppf(prob, dof)
print('probability=%.3f, critical=%.3f, stat=%.3f' % (prob, critical, stat))
if abs(stat) >= critical:
        print('Dependent (reject H0),variables are related')
else:
        print('Independent (fail to reject H0), variables are not related')
```

```
probability=0.950, critical=7.815, stat=3.859
Independent (fail to reject H0), variables are not related
```

In [52]:

```python
print('significance=%.3f, p=%.3f' % (alpha, p))
if p <= alpha:
```

Loading [MathJax]/extensions/Safe.js

```
        print('Dependent (reject H0)')
else:
        print('Independent (fail to reject H0)')
```

```
significance=0.050, p=0.277
Independent (fail to reject H0)
```

In [ ]:

Loading [MathJax]/extensions/Safe.js