In [1]:
```
# Let us import the libraries
#import the libraries
import pandas as pd
import scipy
import numpy as np
from scipy import stats
```

In [2]:
```
# Load the dataset Cutlets
cutlets=pd.read_csv("C:\\Users\\nishi\\Desktop\\Assignments\\Hypothesis_Testing\\
```

In [3]:
```
cutlets.head()
```

Out[3]:

	Unit A	Unit B	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8
0	6.8090	6.7703	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	6.4376	7.5093	NaN	NaN	Anova: Single Factor	NaN	NaN	NaN	NaN
2	6.9157	6.7300	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	7.3012	6.7878	NaN	NaN	SUMMARY	NaN	NaN	NaN	NaN
4	7.4488	7.1522	NaN	NaN	Groups	Count	Sum	Average	Variance

In [4]:
```
unitA=pd.Series(cutlets.iloc[:,0])
print(unitA)

unitB=pd.Series(cutlets.iloc[:,1])
print(unitB)


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

print(p_value[1])     # 2-tail probability

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

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

In [17]:
```
# load the labtat data
labtat=pd.read_csv("C:\\Users\\nishi\\Desktop\\Assignments\\Hypothesis_Testing\\L
```

In [18]:
```
labtat
```

Out[18]:

	Laboratory 1	Laboratory 2	Laboratory 3	Laboratory 4	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
0	185.35	165.53	176.70	166.13	NaN	NaN	NaN	NaN
1	170.49	185.91	198.45	160.79	NaN	NaN	NaN	NaN
2	192.77	194.92	201.23	185.18	NaN	NaN	NaN	NaN
3	177.33	183.00	199.61	176.42	NaN	NaN	NaN	NaN
4	193.41	169.57	204.63	152.60	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
115	178.49	170.66	193.80	172.68	NaN	NaN	NaN	NaN
116	176.08	183.98	215.25	177.64	NaN	NaN	NaN	NaN
117	202.48	174.54	203.99	170.27	NaN	NaN	NaN	NaN
118	182.40	197.18	194.52	150.87	NaN	NaN	NaN	NaN
119	182.09	215.17	221.49	162.21	NaN	NaN	NaN	NaN

120 rows × 12 columns

In [9]:
```
labtat.describe()
```

Out[9]:

	Laboratory 1	Laboratory 2	Laboratory 3	Laboratory 4	Unnamed: 4	Unnamed: 5
count	120.000000	120.000000	120.000000	120.00000	0.0	0.0
mean	178.361583	178.902917	199.913250	163.68275	NaN	NaN
std	13.173594	14.957114	16.539033	15.08508	NaN	NaN
min	138.300000	140.550000	159.690000	124.06000	NaN	NaN
25%	170.335000	168.025000	188.232500	154.05000	NaN	NaN
50%	178.530000	178.870000	199.805000	164.42500	NaN	NaN
75%	186.535000	189.112500	211.332500	172.88250	NaN	NaN
max	216.390000	217.860000	238.700000	205.18000	NaN	NaN

In [24]:
```
labtat1=pd.Series(labtat.iloc[:,0])
print(labtat1)
```

```
0      185.35
1      170.49
2      192.77
3      177.33
4      193.41
        ...
115    178.49
116    176.08
117    202.48
118    182.40
119    182.09
Name: Laboratory 1, Length: 120, dtype: float64
```

In [14]:
```
labtat.iloc[:,1]
```

Out[14]:
```
0      165.53
1      185.91
2      194.92
3      183.00
4      169.57
        ...
115    170.66
116    183.98
117    174.54
118    197.18
119    215.17
Name: Laboratory 2, Length: 120, dtype: float64
```

In [25]:
```
labtat2=pd.Series(labtat.iloc[:,1])
print(labtat2)
```

```
0      165.53
1      185.91
2      194.92
3      183.00
4      169.57
        ...
115    170.66
116    183.98
117    174.54
118    197.18
119    215.17
Name: Laboratory 2, Length: 120, dtype: float64
```

In [20]:
```
labtat3=pd.Series(labtat.iloc[:,2])
print(labtat3)
```

```
0      176.70
1      198.45
2      201.23
3      199.61
4      204.63
        ...
115    193.80
116    215.25
117    203.99
118    194.52
119    221.49
Name: Laboratory 3, Length: 120, dtype: float64
```

In [22]:
```
labtat4=pd.Series(labtat.iloc[:,3])
print(labtat4)
```

```
0      166.13
1      160.79
2      185.18
3      176.42
4      152.60
        ...
115    172.68
116    177.64
117    170.27
118    150.87
119    162.21
Name: Laboratory 4, Length: 120, dtype: float64
```

In [27]:
```
p_value=stats.f_oneway(labtat.iloc[:,0], labtat.iloc[:,1],labtat.iloc[:,2],labtat
print(p_value)

print(p_value[1])
```

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

In [28]:
```
# Importing the BuyerRatio dataset
buyer_ratio=pd.read_csv("C:\\Users\\nishi\\Desktop\\Assignments\\Hypothesis_Testi
```

In [29]:
```
buyer_ratio
```

Out[29]:

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

In [30]:
```
obs=np.array([[50,142,131,70],[435,1523,1356,750]])
obs
```

Out[30]:
```
array([[  50,  142,  131,   70],
       [ 435, 1523, 1356,  750]])
```

In [31]:
```
from scipy import stats
from scipy.stats import norm
from scipy.stats import chi2_contingency
chi2_contingency(obs) # o/p is (Chi2 stats value, p_value, df, expected obsvatior
```

Out[31]:
```
(1.595945538661058,
 0.6603094907091882,
 3,
 array([[  42.76531299,  146.81287862,  131.11756787,   72.30424052],
        [ 442.23468701, 1518.18712138, 1355.88243213,  747.69575948]]))
```

In [32]:
```
# Load the Costomer+Orderorm dataset
data=pd.read_csv("C:\\Users\\nishi\\Desktop\\Assignments\\Hypothesis_Testing\\Cos
data
```

Out[32]:

	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 [33]:
```
data.Phillippines.value_counts()
```

Out[33]:
```
Error Free    271
Defective      29
Name: Phillippines, dtype: int64
```

In [34]:
```
data.Indonesia.value_counts()
```

Out[34]:
```
Error Free    267
Defective      33
Name: Indonesia, dtype: int64
```

In [35]:
```
data.Malta.value_counts()
```

Out[35]:
```
Error Free    269
Defective      31
Name: Malta, dtype: int64
```

In [36]:
```
data.India.value_counts()
```

Out[36]:
```
Error Free    280
Defective      20
Name: India, dtype: int64
```

In [37]:
```
obs=np.array([[271,267,269,280],[29,33,31,20]])
obs
```

Out[37]:
```
array([[271, 267, 269, 280],
       [ 29,  33,  31,  20]])
```

In [38]:
```
chi2_contingency(obs) # o/p is (Chi2 stats value, p_value, df, expected obsvatior
```

Out[38]:
```
(3.858960685820355,
 0.2771020991233135,
 3,
 array([[271.75, 271.75, 271.75, 271.75],
        [ 28.25,  28.25,  28.25,  28.25]]))
```

In [ ]: