In [38]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm,t,binom,expon,chi2,chisquare,chi2_contingency,ttest_1samp,ttest_rel,ttest_ind,f_oneway,kruskal
```

In [3]:
```
!ls
```

```
Gym_Problem_Statement.csv
Hypothesis_Testing_Remedial_Notebook.ipynb
Sachin_ODI.csv
Sound_Experiment.csv
loan.csv
problem_solving.csv
weight-height.csv
```

Simple Imputer

In [4]:
```
df=pd.read_csv("loan.csv")
df
```

Out[4]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
...	...	...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.0	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.0	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semiurban	N

614 rows × 13 columns

In [9]:
```
a=pd.Series([2,2,4,5,np.nan,6,2])
a
```

Out[9]:
```
0    2.0
1    2.0
2    4.0
3    5.0
4    NaN
5    6.0
6    2.0
dtype: float64
```

In [10]:
```
a.mean()
```

Out[10]:
```
3.5
```

In [11]:
```
from sklearn.impute import SimpleImputer
```

In [15]:
```
mean_imputer=SimpleImputer(strategy="mean")
median_imputer=SimpleImputer(strategy="median")
mode_imputer=SimpleImputer(strategy="most_frequent")
constant_imputer=SimpleImputer(strategy="constant",fill_value=100)
```

In [14]:
```
mean_imputer.fit_transform(pd.DataFrame(a))
```

Out[14]:
```
array([[2. ],
       [2. ],
       [4. ],
       [5. ],
       [3.5],
       [6. ],
       [2. ]])
```

In [16]:
```
median_imputer.fit_transform(pd.DataFrame(a))
```

Out[16]:
```
array([[2.],
       [2.],
       [4.],
       [5.],
       [3.],
       [6.],
       [2.]])
```

In [18]:
```
mode_imputer.fit_transform(pd.DataFrame(a))
```

Out[18]:
```
array([[2.],
       [2.],
       [4.],
       [5.],
       [2.],
       [6.],
       [2.]])
```

In [19]:
```
constant_imputer.fit_transform(pd.DataFrame(a))
```

Out[19]:
```
array([[  2.],
       [  2.],
       [  4.],
       [  5.],
       [100.],
       [  6.],
       [  2.]])
```

In [ ]:

Q1

In [20]:
```
binom.pmf(n=4,k=0,p=0.1)
```

Out[20]:
```
0.6561
```

In [21]:
```
1-binom.pmf(n=4,k=0,p=0.1)
```

Out[21]:
```
0.3439
```

In [22]:
```
binom.pmf(n=4,k=4,p=0.9)
```

Out[22]:
```
0.6561
```

In [23]:
```
(1*0.6561)+(5*0.3439)
```

Out[23]:
```
2.3756
```

In [ ]:

Q2

In [24]:
```
ttest_1samp([62,92,75,68,83,95],70,alternative="greater")
```

Out[24]:
```
Ttest_1sampResult(statistic=1.7053136360191492, pvalue=0.07442681355650134)
```

In [25]:
```
(140-100)/(15/np.sqrt(30))
```

Out[25]:
```
14.60593486680443
```

In [26]:
```
1-norm.cdf((140-100)/(15/np.sqrt(30)))
```

Out[26]:
```
0.0
```

In [27]:
```
norm.ppf(0.99)
```

Out[27]:
```
2.3263478740408408
```

In [ ]:

Q3

In [28]:
```
new_machine = np.array([42.1,41,41.3,41.8,42.4,42.8,43.2,42.3,41.8,42.7])
old_machine = np.array([42.7,43.6,43.8,43.3,42.5,43.5,43.1,41.7,44,44.1])
```

In [29]:
```
ttest_rel(new_machine,old_machine,alternative="less")
```

Out[29]:
```
Ttest_relResult(statistic=-3.0614273841115853, pvalue=0.006770167825816246)
```

In [30]:
```
ttest_rel(old_machine,new_machine,alternative="greater")
```

Out[30]:
```
Ttest_relResult(statistic=3.0614273841115853, pvalue=0.006770167825816246)
```

Q4

In [31]:
```
df=pd.read_csv("Gym_Problem_Statement.csv")
```

In [32]:
```
df
```

Out[32]:

	Person	Before	After
0	1	210	197
1	2	205	195
2	3	193	191
3	4	182	174
4	5	259	236
5	6	239	226
6	7	164	157
7	8	197	196
8	9	222	201
9	10	211	196
10	11	187	181
11	12	175	164
12	13	186	181
13	14	243	229
14	15	246	231

In [33]:
```
ttest_rel(df["After"],df["Before"],alternative="less")
```

Out[33]:
```
Ttest_relResult(statistic=-6.6896995348736334, pvalue=5.137828140742704e-06)
```

q5

In [35]:
```
df=pd.read_csv("Sound_Experiment.csv")
df
```

Out[35]:

	Constant_Sound	Unpredictable_Sound	No_Sound
0	7	5	2
1	4	5	4
2	6	3	7
3	8	4	1
4	6	4	2
5	6	7	1
6	2	2	5
7	9	2	5

In [36]:
```
f_oneway(df["Constant_Sound"],df["Unpredictable_Sound"],df["No_Sound"])
```

Out[36]:
```
F_onewayResult(statistic=3.594594594594595, pvalue=0.04543970036605765)
```

In [39]:
```
kruskal(df["Constant_Sound"],df["Unpredictable_Sound"],df["No_Sound"])
```

Out[39]:
```
KruskalResult(statistic=5.695079964460244, pvalue=0.057986794103564006)
```

q6

In [41]:
```
observations = np.array([[138,83,64],[64,67,84]])
```

In [42]:
```
chi2_contingency(observations)
```

Out[42]:
```
(22.152468645918482,
 1.547578021398957e-05,
 2,
 array([[115.14,  85.5 ,  84.36],
        [ 86.86,  64.5 ,  63.64]]))
```

Q7

In [43]:
```
non_smokers= [130, 122, 128, 129, 118, 122, 116, 127, 135, 120, 122, 120, 115, 123]
smokers= [124, 134, 136, 125, 133, 127, 135, 131, 133, 125, 118]
```

In [44]:
```
ttest_ind(non_smokers,smokers,alternative="two-sided")
```

Out[44]:
```
Ttest_indResult(statistic=-2.5234545079093134, pvalue=0.018984295301644454)
```

In [ ]:

q8

In [45]:
```
low_temp=[42,41,37,29,35,40,32]
mid_temp=[36,35,32,38,39,42,34]
high_temp=[33,44,40,36,44,37,45]
```

In [46]:
```
f_oneway(low_temp,mid_temp,high_temp)
```

Out[46]:
```
F_onewayResult(statistic=1.3324937027707806, pvalue=0.2886265256259175)
```

In [ ]:

q9 chi square test of independance

In [47]:
```
observations = np.array([[22,278],[26,374]])
observations
```

Out[47]:
```
array([[ 22, 278],
       [ 26, 374]])
```

In [49]:
```
chi2_contingency(observations)
```

Out[49]:
```
(0.07875074556918876,
 0.7789975538960225,
 1,
 array([[ 20.57142857, 279.42857143],
        [ 27.42857143, 372.57142857]]))
```

In [ ]:

In [50]:
```
df=pd.read_csv("loan.csv")
```

In [51]:
```
df
```

Out[51]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
...	...	...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.0	Rural	Y
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.0	Urban	Y
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.0	Urban	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semiurban	N

614 rows × 13 columns

In [54]:
```
df_graduate_unmarried_men=df.loc[(df["Gender"]=="Male")&(df["Education"]=="Graduate")&(df["Married"]=="No")]
df_graduate_unmarried_men
```

Out[54]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
13	LP001029	Male	No	0	Graduate	No	1853	2840.0	114.0	360.0	1.0	Rural	N
15	LP001032	Male	No	0	Graduate	No	4950	0.0	125.0	360.0	1.0	Urban	Y
31	LP001095	Male	No	0	Graduate	No	3167	0.0	74.0	360.0	1.0	Urban	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...
577	LP002874	Male	No	0	Graduate	No	3229	2739.0	110.0	360.0	1.0	Urban	Y
579	LP002888	Male	No	0	Graduate	NaN	3182	2917.0	161.0	360.0	1.0	Urban	Y
581	LP002893	Male	No	0	Graduate	No	1836	33837.0	90.0	360.0	1.0	Urban	N
597	LP002943	Male	No	NaN	Graduate	No	2987	0.0	88.0	360.0	0.0	Semiurban	N
603	LP002958	Male	No	0	Graduate	No	3676	4301.0	172.0	360.0	1.0	Rural	Y

99 rows × 13 columns

In [55]:
```
df_graduate_women=df.loc[(df["Gender"]=="Female")&(df["Education"]=="Graduate")]
df_graduate_women
```

Out[55]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
17	LP001036	Female	No	0	Graduate	No	3510	0.0	76.0	360.0	0.0	Urban	N
29	LP001087	Female	No	2	Graduate	NaN	3750	2083.0	120.0	360.0	1.0	Semiurban	Y
37	LP001112	Female	Yes	0	Graduate	No	3667	1459.0	144.0	360.0	1.0	Semiurban	Y
45	LP001137	Female	No	0	Graduate	No	3410	0.0	88.0	NaN	1.0	Urban	Y
48	LP001146	Female	Yes	0	Graduate	No	2645	3440.0	120.0	360.0	0.0	Urban	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...
582	LP002894	Female	Yes	0	Graduate	No	3166	0.0	36.0	360.0	1.0	Semiurban	Y
600	LP002949	Female	No	3+	Graduate	NaN	416	41667.0	350.0	180.0	NaN	Urban	N
604	LP002959	Female	Yes	1	Graduate	No	12000	0.0	496.0	360.0	1.0	Semiurban	Y
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semiurban	N

92 rows × 13 columns

In [60]:
```
pd.crosstab(index=df_graduate_women["Loan_Status"],columns=df_graduate_unmarried_men["Loan_Status"])
```

Out[60]:

Loan_Status
Loan_Status

In [67]:
```
df_graduate_women["Loan_Status"].value_counts()['Y']
```

Out[67]:
```
61
```

In [63]:
```
df_graduate_unmarried_men["Loan_Status"].value_counts()
```

Out[63]:
```
Y    62
N    37
Name: Loan_Status, dtype: int64
```

In [68]:
```
observations= [[df_graduate_women["Loan_Status"].value_counts()['Y'],df_graduate_women["Loan_Status"].value_counts()['N']],[df_graduate_unmarried_men["Loan_Status"].value_counts()["Y"],df_graduate_unmarried_men["Loan_Status"].value_counts()["N"]]]
observations
```

Out[68]:
```
[[61, 31], [62, 37]]
```

In [69]:
```
chi2_contingency(observations)
```

Out[69]:
```
(0.14381605008456969,
 0.7045164323229686,
 1,
```

```
array([[59.2468733, 32.7539267],
       [63.7539267, 36.2468733]])
```

84 [ ]: