Topic: Basic Probability, Conditional Probability, Hypothesis Testing

source code https://github.com/nitishbuzzpro/Statistics-and-Hypothesis-Tetsing--Titanic-Dataset---Data-Science.git (https://github.com/nitishbuzzpro/Statistics-and-Hypothesis-Tetsing--Titanic-Dataset---Data-Science.git)

In [3]:
```
import pandas as pd
import numpy as np
import piplite
await piplite.install('openpyxl')
from scipy.stats import chi2_contingency
import statsmodels.api as sm
```

Task 1

```
Load the Titanic dataset using Python and analyze thefollowing probabilities:
Probability of surviving or not surviving.
Probability of being male, and female.
Probability of being in first class, second class or the third class. (Pclass column)
```

In [4]:
```
#load the dataset
df = pd.read_excel('train.xlsx')
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

In [5]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 66.2+ KB
```

In [6]:
```
1 df.describe()
```

Out[6]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [7]:
```
1 df.describe(include=object)
```

Out[7]:

	Name	Sex	Ticket	Cabin	Embarked
count	891	891	891	204	889
unique	891	2	681	147	3
top	Braund, Mr. Owen Harris	male	347082	B96 B98	S
freq	1	577	7	4	644

In [8]:
```
# Probability of surviving or not surviving.
total_persons = df['Survived'].count()
count_Surviving = (df['Survived'] == 1).sum()
Probability_surviving = count_Surviving / total_persons
Probability_not_surviving = 1 - Probability_surviving
print('Probabilty of surving:',round(Probability_surviving,2))
print('Probabilty of not surving:',round(Probability_not_surviving,2))
```

```
Probabilty of surving: 0.38
Probabilty of not surving: 0.62
```

In [9]:
```
# Probability of being male, and female
total_persons = df['Sex'].count()
count_male = (df['Sex'] == 'male').sum()
Probability_male = count_male / total_persons
Probability_female = 1 - Probability_male
print('Probabilty of male:',round(Probability_male,2))
print('Probabilty of female:',round(Probability_female,2))
```

```
Probabilty of male: 0.65
Probabilty of female: 0.35
```

In [10]:
```
# Probability of being in first class, second class or the third class.
total_persons = df['Pclass'].count()
count_first_class = (df['Pclass'] == 1).sum()
count_second_class = (df['Pclass'] == 2).sum()
count_third_class = (df['Pclass'] == 3).sum()
Probability_first_class = count_first_class / total_persons
Probability_second_class = count_second_class / total_persons
Probability_third_class = count_third_class / total_persons
print('Probabilty of first_class:',round(Probability_first_class,2))
print('Probabilty of second_class:',round(Probability_second_class,2))
print('Probabilty of third_class:',round(Probability_third_class,2))
```

```
Probabilty of first_class: 0.24
Probabilty of second_class: 0.21
Probabilty of third_class: 0.55
```

In [ ]:
```

```

Task 2

```
Calculate the following conditional probabilities:
Probability of surviving given that the passenger is male or female.
Probability of surviving given that the passenger is in first class,
second class, and third class respectively.
```

In [11]:
```
#Probability of surviving given that the passenger ismale or female.
male_female =  ((df['Sex'] == 'male') | (df['Sex'] == 'female')).sum()/
surviving_male_female  = df[df['Survived']==1 & ((df['Sex'] == 'male')
prob_surv_male_female = surviving_male_female/male_female
print('Probability of surviving given that the passenger is male or fem
```

```
Probability of surviving given that the passenger is male or female is: 0.
38
```

In [12]:

```
#Probability of surviving given that the passenger is in first class, s
number_surv_first = df[(df['Survived'] == 1) & (df['Pclass'] == 1)]['Pa
number_first = df[(df['Pclass'] == 1)]['PassengerId'].count()
prob_surv_first = number_surv_first/number_first
print('Probability of surviving given that the passenger is in first cl

number_surv_second = df[(df['Survived'] == 1) & (df['Pclass'] == 2)]['P
number_second  = df[(df['Pclass'] == 2)]['PassengerId'].count()
prob_surv_second  = number_surv_second /number_second
print('Probability of surviving given that the passenger is in second c

number_surv_third = df[(df['Survived'] == 1) & (df['Pclass'] == 3)]['Pa
number_third  = df[(df['Pclass'] == 3)]['PassengerId'].count()
prob_surv_third  = number_surv_third /number_third
print('Probability of surviving given that the passenger is in third cl
```

```
Probability of surviving given that the passenger is in first class :  0.6
3
Probability of surviving given that the passenger is in second class :  0.
47
Probability of surviving given that the passenger is in third class :  0.2
4
```

In [ ]:

```

```

Task 4

```
Compute the joint probabilities of the following events:
Being a male and surviving
Being a female and surviving
Being in first class and surviving
Being in second class and surviving
Being in third class and surviving
Explain all the Results.
```

In [13]:

```
# Being a male and surviving
count_male_surv = df[(df['Sex'] == 'male') & (df['Survived']==1)]['Pass
count_passengers = df['PassengerId'].count()
jointprob_male_surv = count_male_surv/count_passengers
print('Being a male and surviving : ',round(jointprob_male_surv,2))
print("Explaination : Out of all the passanger if a  passanger is selec
```

```
Being a male and surviving :  0.12
Explaination : Out of all the passanger if a  passanger is selected at ran
dom then the probality that the passanger is a male & who has survived is
0.12
```

In [14]:
```
# Being a female and surviving
count_female_surv = df[(df['Sex'] == 'male') & (df['Survived']==1)]['Pa
count_passengers = df['PassengerId'].count()
jointprob_female_surv = count_female_surv/count_passengers
print('Being a female and surviving : ',round(jointprob_female_surv,2))
print("Explaination : Out of all the passanger if a  passanger is selec
```

```
Being a female and surviving :  0.12
Explaination : Out of all the passanger if a  passanger is selected at ran
dom then the probality that the passanger is a female & who has survived i
s 0.12
```

In [15]:
```
# Being in first class and surviving
count_first_surv = df[(df['Pclass'] == 1) & (df['Survived']==1)]['Passe
count_passengers = df['PassengerId'].count()
jointprob_first_surv = count_first_surv/count_passengers
print('Being in first class and surviving : ',round(jointprob_first_sur
print("Explaination : Out of all the passanger if a  passanger is selec
```

```
Being in first class and surviving :  0.15
Explaination : Out of all the passanger if a  passanger is selected at ran
dom then the probality that the passanger is a first class & who has survi
ved is 0.15
```

In [16]:
```
# Being in second class and surviving
count_second_surv = df[(df['Pclass'] == 2) & (df['Survived']==1)]['Pass
count_passengers = df['PassengerId'].count()
jointprob_second_surv = count_second_surv/count_passengers
print('Being in second class and surviving : ',round(jointprob_second_s
print("Explaination : Out of all the passanger if a  passanger is selec
```

```
Being in second class and surviving :  0.1
Explaination : Out of all the passanger if a  passanger is selected at ran
dom then the probality that the passanger is a second class & who has surv
ived is 0.1
```

In [17]:
```
# Being in third class and surviving
count_third_surv = df[(df['Pclass'] == 3) & (df['Survived']==1)]['Passe
count_passengers = df['PassengerId'].count()
jointprob_third_surv = count_third_surv/count_passengers
print('Being in third class and surviving : ',round(jointprob_third_sur
print("Explaination : Out of all the passanger if a  passanger is selec
```

```
Being in third class and surviving :  0.13
Explaination : Out of all the passanger if a  passanger is selected at ran
dom then the probality that the passanger is a third class & who has survi
ved is 0.13
```

In [ ]:
```
```

Task 5

```
```

```
Calculate the conditional probability of survival given that the passenger is an adult (age greater than or equal to 18).
Calculate the conditional probability of survival given that the passenger is a child (age less than 18).
Determine if survival and passenger class are independent events.
Calculate the probability of surviving in each class and compare it with the overall survival rate.
```

In [18]:

```
# Calculate the conditional probability of survival given that the pass
prob_adult = df[(df['Age']>=18)]['PassengerId'].count()/df['PassengerId
prob_survive_adult = df[(df['Age']>=18) & (df['Survived']==1)]['Passeng
condprob_survive_adult = prob_survive_adult/prob_adult
print("the conditional probability of survival given that the passenger
```

```
the conditional probability of survival given that the passenger is an adu
lt (age greater than or equal to 18) 0.38
```

In [19]:

```
# Calculate the conditional probability of survival given that the pass
prob_child = df[(df['Age']<18)]['PassengerId'].count()/df['PassengerId'
prob_survive_child = df[(df['Age']<18) & (df['Survived']==1)]['Passenge
condprob_survive_child = prob_survive_child/prob_child
print("the conditional probability of survival given that the passenger
```

```
the conditional probability of survival given that the passenger is a chil
d (age less than 18) 0.54
```

In [74]:

```
# Determine if survival and passenger class are independent events. Cal
prob_survival = df[(df['Survived']==1)]['PassengerId'].count()/df['Pass

if (jointprob_first_surv + jointprob_second_surv + jointprob_third_surv
    print("survival and passenger class are independent events")
else:
     print("survival and passenger class are not independent events")
print('\n')
print("probability of surviving in first class: ",round(prob_surv_first
print("probability of surviving in second class: ",round(prob_surv_seco
print("probability of surviving in third class: ", round(prob_surv_thir
print('\n')
if prob_surv_first + prob_surv_second + prob_surv_third == prob_surviva
    print("compare it with the overall survival rate: independent event
else:
    print("compare it with the overall survival rate: not independent e
```

```
survival and passenger class are independent events


probability of surviving in first class:  0.63
probability of surviving in second class:  0.47
probability of surviving in third class:  0.24


compare it with the overall survival rate: not independent events
```

In [ ]:
```
1
```

Task 6 - Hypothesis Testing

Hypothesis: The survival rate of male passengers is equal to the survival rate of female passengers.

Hypothesis: The survival rate of passengers in firstclass is higher than the survival rate of passengers in third class.

Perform a hypothesis test to validate or reject the null hypothesis in each scenario. Use appropriate statistical tests, such as chi-square test or t-test, to analyze the data and calculate the p-values. Provide a conclusion for each hypothesis test, indicating whether the null hypothesis should be accepted or rejected.

1) Hypothesis: The survival rate of male passengers is equal to the survival rate of female passengers.

```
1 Hypothesis formulation
```

Null Hypotheis : The survival rate of male passengers is equal to the survival rate of female passengers

Alternative Hypothesis : The survival rate of male passengers is not equal to the survival rate of female passengers

chi-square test

In [54]:

```python
# Prepare contingency table
contingency_table = pd.crosstab(df['Sex'], df['Survived'])

#choosing significance level
significance_level = 0.05

#Perform chi-square test
chi2, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-square statistic:", chi2)
print("p-value:", p)
print("Degrees of freedom:", dof)
print("Expected frequencies:")
print(expected)
print('\n')
if p_value < significance_level:
    print("Reject : Null Hypothesis")
    print("Accept : Alternative Hypothesis : The survival rate of male
else:
    print("Accept : Null Hypothesis")
    print("Reject : Alternative Hypothesis : The survival rate of male
```

```
Chi-square statistic: 260.71702016732104
p-value: 1.1973570627755645e-58
Degrees of freedom: 1
Expected frequencies:
[[193.47474747 120.52525253]
 [355.52525253 221.47474747]]


Reject : Null Hypothesis
Accept : Alternative Hypothesis : The survival rate of male passengers is
not equal to the survival rate of female passengers
```

Two-proportion z-test

In [49]:

```
# Prepare contingency table
contingency_table = pd.crosstab(df['Sex'], df['Survived'])

#choosing significance level
significance_level = 0.05

# Number of successes (survivors) and number of trials (total passenger
successes = list(contingency_table[1])
trials = list(contingency_table[0] + contingency_table[1])

# Perform two-proportion z-test
z_score,p_value =  sm.stats.proportions_ztest(successes,trials)
print('z_score',z_score)
print('p_value',p_value)

if p_value < significance_level:
    print("Reject : Null Hypothesis")
    print("Accept : Alternative Hypothesis : The survival rate of male
else:
    print("Accept : Null Hypothesis")
    print("Reject : Alternative Hypothesis : The survival rate of male
```

```
z_score 16.218833930670097
p_value 3.7117477701134797e-59
Reject : Null Hypothesis
Accept : Alternative Hypothesis : The survival rate of male passengers is
not equal to the survival rate of female passengers
```

In [ ]:

```

```

2) Hypothesis: The survival rate of passengers in firstclass is higher than the survival rate of passengers in third class.

```
Hypothesis formulation
```

Null Hypotheis : The survival rate of passengers in first class is equal to or lower than the survival rate of passengers in third class.

Alternative Hypothesis : The survival rate of passengers in first class is higher than the survival rate of passengers in third class

chi-square test

In [68]:

```
# Prepare contingency table
contingency_table = pd.crosstab(df[(df['Pclass']==1) | (df['Pclass']==3
#choosing significance level
significance_level = 0.05
#Perform chi-square test
chi2, p_value, dof, expected = chi2_contingency(contingency_table)
print("Chi-square statistic:", chi2)
print("p-value:", p_value)
print("Degrees of freedom:", dof)
print("Expected frequencies:")
print(expected)
print('\n')
if p_value < significance_level:
    print("Reject : Null Hypothesis")
    print("Accept : Alternative Hypothesis : The survival rate of passe
else:
    print("Accept : Null Hypothesis")
    print("Reject : Alternative Hypothesis : The survival rate of passe
```

```
Chi-square statistic: 95.89348388920357
p-value: 1.2123375217498223e-22
Degrees of freedom: 1
Expected frequencies:
[[138.09335219  77.90664781]
 [313.90664781 177.09335219]]


Reject : Null Hypothesis
Accept : Alternative Hypothesis : The survival rate of passengers in first
class is higher than the survival rate of passengers in third class
```

Two-proportion z-test

In [71]:

```
# Prepare contingency table
contingency_table = pd.crosstab(df[(df['Pclass']==1) | (df['Pclass']==3

#choosing significance level
significance_level = 0.05

# Number of successes (survivors) and number of trials (total passenger
successes = list(contingency_table[1])
trials = list(contingency_table[0] + contingency_table[1])

# Perform two-proportion z-test
z_score,p_value =  sm.stats.proportions_ztest(successes,trials)
print('z_score',z_score)
print('p_value',p_value)
print('\n')
if p_value < significance_level:
    print("Reject : Null Hypothesis")
    print("Accept : Alternative Hypothesis : The survival rate of passe
else:
    print("Accept : Null Hypothesis")
    print("Reject : Alternative Hypothesis : The survival rate of passe
```

```
z_score 9.87753617661869
p_value 5.209807780351272e-23


Reject : Null Hypothesis
Accept : Alternative Hypothesis : The survival rate of passengers in first
class is higher than the survival rate of passengers in third class
```