Exploratory Data Analysis(EDA)

1. Analysis

   Univariate Analysis

   Multi-Variate Analysis

2. Feature Engineering

   Creating new columns

   Modifying Columns

3. Handling Outliers

   Detect Outliers

   Remove Outliers

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('tested.csv')
```

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

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [4]:
```
df.tail()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

In [5]:
```
df.describe()
```

Out[5]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [6]:
```
df.shape
```

Out[6]: (418, 12)

In [7]:
```
df.columns.values
```

Out[7]:
```
array(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'], dtype=object)
```

Categorical Columns

Survived

Pclass

sibsp

Sex

Parch

Embarked

Numeric Columns

PassengerId

Age

Fare

Mixed Columns

Name

Ticket

Cabin

In [8]:
```
df.dtypes
```

Out[8]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

Find out the total number of Non-values in the dataset.

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

Few conclusions we got from the Non Values

1. Missing vakues in Age,Cabin and Fare.
2. More Have to 80% data are missing into the cabin so,we will have to drop.
3. Few columns have appropriate datatypes.

Dropping Cabin columns.

In [11]:
```
df.drop(columns=['Cabin']).head()
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	S

Imputing missing values for age

strategy -- mean

In [12]:
```
df['Age'].fillna(df['Age'].mean())
```

Out[12]:
```
0      34.50000
1      47.00000
2      62.00000
3      27.00000
4      22.00000
         ...
413    30.27259
414    39.00000
415    38.50000
416    30.27259
417    30.27259
Name: Age, Length: 418, dtype: float64
```

In [13]:
```
df['Parch'].value_counts()
```

Out[13]:
```
Parch
0    324
1     52
2     33
3      3
4      2
9      2
6      1
5      1
Name: count, dtype: int64
```

we found from value counts that 324 passengers have not any parent and so on.

In [14]:
```
df['SibSp'].value_counts()
```

Out[14]:
```
SibSp
0    283
1    110
2     14
3      4
4      4
8      2
5      1
Name: count, dtype: int64
```

Changing datatypes for following cols

```
Survived(categorical)
Pclass(category)
Sex (category)
Age(int)
Embarked(Category)
```

In [15]:
```
df['Survived']=df['Survived'].astype('category')
df['Pclass']=df['Pclass'].astype('category')
df['Sex']=df['Sex'].astype('category')
##df['Age']=df['Age'].astype('int')
df['Embarked']=df['Embarked'].astype('category')
```

In [16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Survived     418 non-null    category
 2   Pclass       418 non-null    category
 3   Name         418 non-null    object
 4   Sex          418 non-null    category
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64
 7   Parch        418 non-null    int64
 8   Ticket       418 non-null    object
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object
 11  Embarked     418 non-null    category
dtypes: category(4), float64(2), int64(3), object(3)
memory usage: 28.4+ KB
```

In [17]:
```
df.describe()
```

Out[17]:

	PassengerId	Age	SibSp	Parch	Fare
count	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	30.272590	0.447368	0.392344	35.627188
std	120.810458	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	76.000000	8.000000	9.000000	512.329200

In [18]:
```
df.shape
```

Out[18]:
```
(418, 12)
```

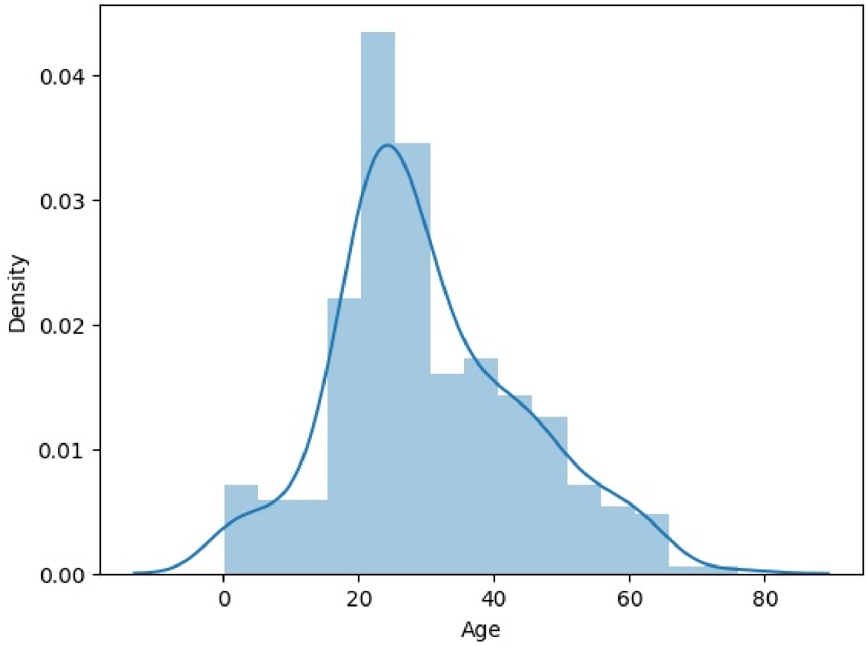
In [19]:
```
print((df['Pclass'].value_counts()/418)*100)
```

```
Pclass
3    52.153110
1    25.598086
2    22.248804
Name: count, dtype: float64
```

In [20]:
```
sns.distplot(df['Age'])
```

```
C:\Users\Public\Documents\iSkysoft\CreatorTemp\ipykernel_3560\3255828239.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Age'])
```

Out[20]: <Axes: xlabel='Age', ylabel='Density'>



In [21]:
```
df['Age'].skew()
```

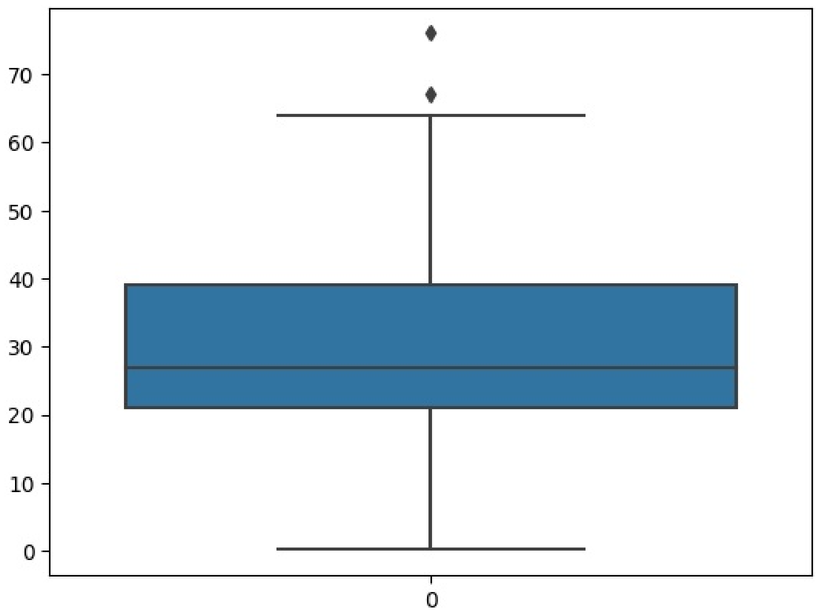
Out[21]: 0.4573612871503845

In [22]:
```
df['Age'].kurt()
```

Out[22]: 0.08378335153796135

In [23]:
```
sns.boxplot(df['Age'])
```

Out[23]: <Axes: >



Conclusion

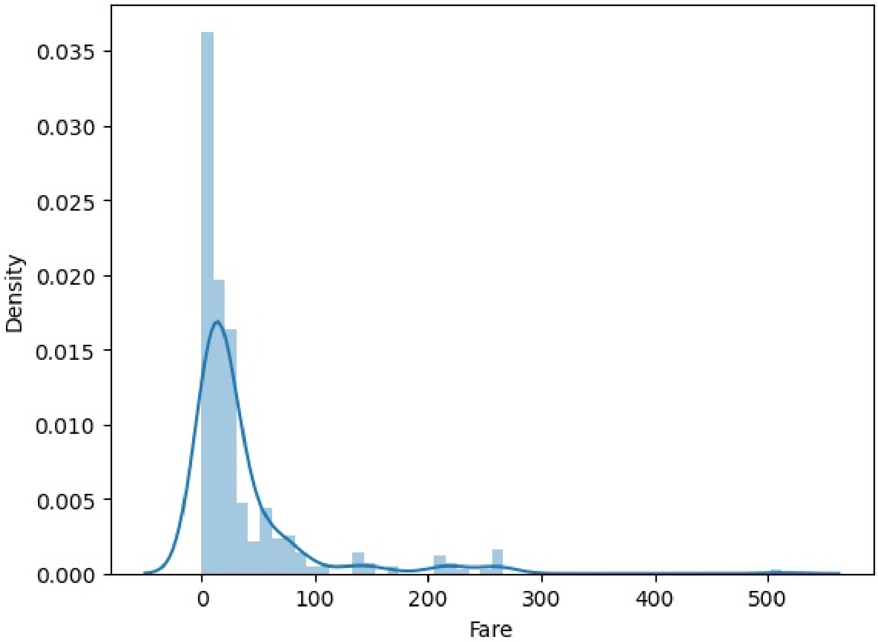
For all practical problems age can be considered as Normal Distribution.

Deeper Analysis is required for the deeper Analysis.

In [24]:
```
sns.distplot(df['Fare'])
```

```
C:\Users\Public\Documents\iSkysoft\CreatorTemp\ipykernel_3560\3425841524.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Fare'])
```

Out[24]: <Axes: xlabel='Fare', ylabel='Density'>



In [25]:
```
df['Fare'].skew()
```

Out[25]: 3.6872133081121405

In [26]:
```
df['Fare'].kurt()
```

Out[26]: 17.92159525773599

Conclusion

1. Highly skewed data means lot of people had cheaper tickets.
2. outliers are in the data

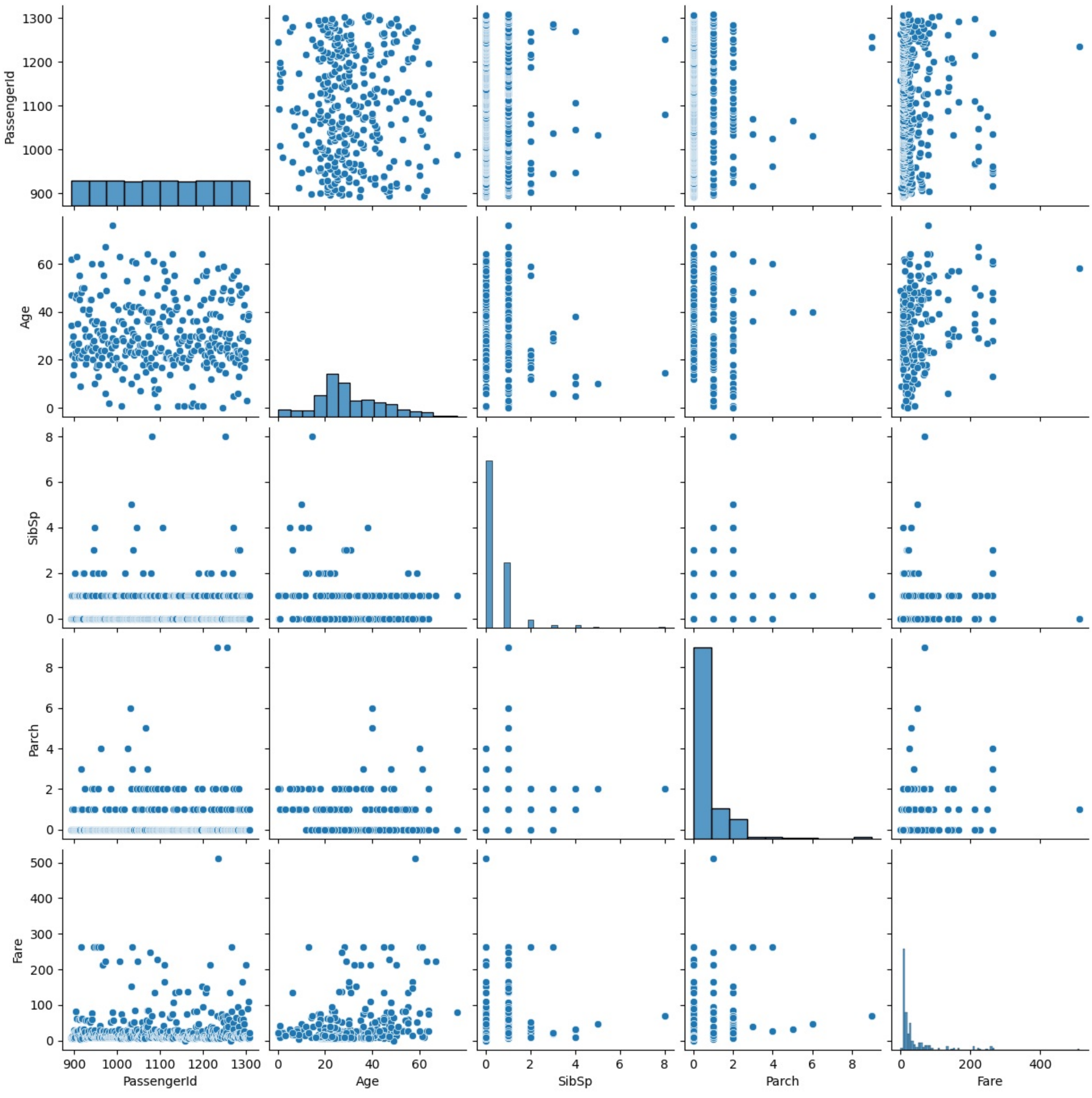
Multivariate Analysis

Survival with Pclass

In [27]:
```
sns.pairplot(df)
```

```
C:\Users\Hp\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The
figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[27]: <seaborn.axisgrid.PairGrid at 0x1e4b5596f10>



Feature Engineering

we will create a new column by the name of family which will be the sum of Sibsp and Parchcols

In [31]:
```
df['family_size'] = df['Parch'] + df['SibSp']
```

In [29]:
```
df.sample(5)
```

Out[29]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	family_size
307	1199	0	3	Aks, Master. Philip Frank	male	0.83	0	1	392091	9.35	NaN	S	1
306	1198	0	1	Allison, Mr. Hudson Joshua Creighton	male	30.00	1	2	113781	151.55	C22 C26	S	3
74	966	1	1	Geiger, Miss. Amalie	female	35.00	0	0	113503	211.50	C130	C	0
50	942	0	1	Smith, Mr. Lucien Philip	male	24.00	1	0	13695	60.00	C31	S	1
351	1243	0	2	Stokes, Mr. Philip Joseph	male	25.00	0	0	F.C.C. 13540	10.50	NaN	S	0

In [32]:
```
df.head()
```

Out[32]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	family_size
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q	0
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S	1
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q	0
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S	0
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S	2

In [33]:
```
df.drop(columns=['Cabin']).head()
```

Out[33]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	family_size
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	Q	0
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	S	1
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	Q	0
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	S	0
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	S	2

Drawing Coclusions

1. Chance of female survival is higher than male survivor.
2. Travelling in Pclass 3 was deadlist.
3. Somehow, people going to C Survived more.
4. people age in the range of 20 to 40 had a higher chance of not surviving.
5. people travelling with smaller families had a higher chance of surviving the accident in comparison to people with large families and travelling alone.

In [ ]: