

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: titanic = pd.read_csv(r"C:\Users\ANITHA\Downloads\13th - ML - Data preprocessing\13
```

```
In [3]: titanic.tail()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75

```
In [4]: titanic.describe()
```

	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 [6]: del titanic["Name"]
titanic.head()
```

Out[6]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	male	35.0	0	0	373450	8.0500	NaN	



```
In [7]: del titanic["Ticket"]
titanic.head()
```

Out[7]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	male	22.0	1	0	7.2500	NaN	S
1	2	1	1	female	38.0	1	0	71.2833	C85	C
2	3	1	3	female	26.0	0	0	7.9250	NaN	S
3	4	1	1	female	35.0	1	0	53.1000	C123	S
4	5	0	3	male	35.0	0	0	8.0500	NaN	S

```
In [8]: del titanic["Fare"]
titanic.head()
```

Out[8]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Cabin	Embarked
0	1	0	3	male	22.0	1	0	NaN	S
1	2	1	1	female	38.0	1	0	C85	C
2	3	1	3	female	26.0	0	0	NaN	S
3	4	1	1	female	35.0	1	0	C123	S
4	5	0	3	male	35.0	0	0	NaN	S

```
In [9]: del titanic['Cabin']
titanic.head()
```

Out[9]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Embarked
0	1	0	3	male	22.0	1	0	S
1	2	1	1	female	38.0	1	0	C
2	3	1	3	female	26.0	0	0	S
3	4	1	1	female	35.0	1	0	S
4	5	0	3	male	35.0	0	0	S

In [10]: # Changing Value for "Male, Female" string values to numeric values , male=1 and fe
def getNumber(str):
 if str=="male":
 return 1
 else:
 return 2
titanic["Gender"] = titanic["Sex"].apply(getNumber)
#We have created a new column called "Gender" and
#filling it with values 1,2 based on the values of sex column
titanic.head()

Out[10]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Embarked	Gender
0	1	0	3	male	22.0	1	0	S	1
1	2	1	1	female	38.0	1	0	C	2
2	3	1	3	female	26.0	0	0	S	2
3	4	1	1	female	35.0	1	0	S	2
4	5	0	3	male	35.0	0	0	S	1

In [11]:

#Deleting Sex column, since no use of it now
del titanic["Sex"]
titanic.head()

Out[11]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Embarked	Gender
0	1	0	3	22.0	1	0	S	1
1	2	1	1	38.0	1	0	C	2
2	3	1	3	26.0	0	0	S	2
3	4	1	1	35.0	1	0	S	2
4	5	0	3	35.0	0	0	S	1

In [12]:

titanic.isnull().sum()

```
Out[12]: PassengerId      0
          Survived        0
          Pclass          0
          Age           177
          SibSp          0
          Parch          0
          Embarked       2
          Gender          0
          dtype: int64
```

Fill the null values of the Age column. Fill mean Survived age(mean age of the survived people) in the column where the person has survived and mean not Survived age (mean age of the people who have not survived) in the column where person has not survived

```
In [13]: means= titanic[titanic.Survived==1].Age.mean()
means
```

```
Out[13]: np.float64(28.343689655172415)
```

Creating a new "Age" column , filling values in it with a condition if goes True then given values (here meanS) is put in place of last values else nothing happens, simply the values are copied from the "Age" column of the dataset

```
In [15]: titanic["age"]=np.where(pd.isnull(titanic.Age) & titanic["Survived"]==1 ,means, titanic.head())
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Embarked	Gender	age
0	1	0	3	22.0	1	0	S	1	22.0
1	2	1	1	38.0	1	0	C	2	38.0
2	3	1	3	26.0	0	0	S	2	26.0
3	4	1	1	35.0	1	0	S	2	35.0
4	5	0	3	35.0	0	0	S	1	35.0

```
In [16]: titanic.isnull().sum()
```

```
Out[16]: PassengerId      0
          Survived        0
          Pclass          0
          Age           177
          SibSp          0
          Parch          0
          Embarked       2
          Gender          0
          age            125
          dtype: int64
```

```
In [17]: #Finding the mean age of "Not Survived" people
means=titanic[titanic.Survived==0].Age.mean()
means
```

```
Out[17]: np.float64(30.62617924528302)
```

```
In [19]: titanic.age.fillna(means,inplace=True)
titanic.head()
```

C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\698378622.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.

The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
titanic.age.fillna(means,inplace=True)
```

```
Out[19]:   PassengerId  Survived  Pclass  Age  SibSp  Parch  Embarked  Gender  age
          0            1        0    3  22.0     1      0        S      1  22.0
          1            2        1    1  38.0     1      0        C      2  38.0
          2            3        1    3  26.0     0      0        S      2  26.0
          3            4        1    1  35.0     1      0        S      2  35.0
          4            5        0    3  35.0     0      0        S      1  35.0
```

```
In [20]: titanic.isnull().sum()
```

```
Out[20]: PassengerId      0
          Survived        0
          Pclass          0
          Age           177
          SibSp          0
          Parch          0
          Embarked       2
          Gender          0
          age            0
          dtype: int64
```

```
In [21]: del titanic['Age']
titanic.head()
```

```
Out[21]:   PassengerId  Survived  Pclass  SibSp  Parch  Embarked  Gender  age
          0             1        0     3      1      0          S        1  22.0
          1             2        1     1      1      0          C        2  38.0
          2             3        1     3      0      0          S        2  26.0
          3             4        1     1      1      0          S        2  35.0
          4             5        0     3      0      0          S        1  35.0
```

```
In [22]: # Finding the number of people who have survived
# given that they have embarked or boarded from a particular port

survivedQ = titanic[titanic.Embarked == 'Q'][titanic.Survived == 1].shape[0]
survivedC = titanic[titanic.Embarked == 'C'][titanic.Survived == 1].shape[0]
survivedS = titanic[titanic.Embarked == 'S'][titanic.Survived == 1].shape[0]
print(survivedQ)
print(survivedC)
print(survivedS)
```

30
93
217

```
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3300902897.py:4: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  survivedQ = titanic[titanic.Embarked == 'Q'][titanic.Survived == 1].shape[0]
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3300902897.py:5: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  survivedC = titanic[titanic.Embarked == 'C'][titanic.Survived == 1].shape[0]
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3300902897.py:6: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  survivedS = titanic[titanic.Embarked == 'S'][titanic.Survived == 1].shape[0]
```

```
In [23]: survivedQ = titanic[titanic.Embarked == 'Q'][titanic.Survived == 0].shape[0]
survivedC = titanic[titanic.Embarked == 'C'][titanic.Survived == 0].shape[0]
survivedS = titanic[titanic.Embarked == 'S'][titanic.Survived == 0].shape[0]
print(survivedQ)
print(survivedC)
print(survivedS)
```

30

75

427

```
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3296936882.py:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    survivedQ = titanic[titanic.Embarked == 'Q'][titanic.Survived == 0].shape[0]
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3296936882.py:2: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    survivedC = titanic[titanic.Embarked == 'C'][titanic.Survived == 0].shape[0]
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3296936882.py:3: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
    survivedS = titanic[titanic.Embarked == 'S'][titanic.Survived == 0].shape[0]
```

In [24]:

```
titanic.dropna(inplace=True)
titanic.head()
```

Out[24]:

	PassengerId	Survived	Pclass	SibSp	Parch	Embarked	Gender	age
0	1	0	3	1	0	S	1	22.0
1	2	1	1	1	0	C	2	38.0
2	3	1	3	0	0	S	2	26.0
3	4	1	1	1	0	S	2	35.0
4	5	0	3	0	0	S	1	35.0

In [25]:

```
titanic.isnull().sum()
```

Out[25]:

PassengerId	0
Survived	0
Pclass	0
SibSp	0
Parch	0
Embarked	0
Gender	0
age	0
dtype:	int64

In [26]:

```
#Renaming "age" and "gender" columns
titanic.rename(columns={'age':'Age'}, inplace=True)
titanic.head()
```

Out[26]:

	PassengerId	Survived	Pclass	SibSp	Parch	Embarked	Gender	Age
0	1	0	3	1	0	S	1	22.0
1	2	1	1	1	0	C	2	38.0
2	3	1	3	0	0	S	2	26.0
3	4	1	1	1	0	S	2	35.0
4	5	0	3	0	0	S	1	35.0

```
In [27]: titanic.rename(columns={'Gender': 'Sex'}, inplace=True)
titanic.head()
```

Out[27]:

	PassengerId	Survived	Pclass	SibSp	Parch	Embarked	Sex	Age
0	1	0	3	1	0	S	1	22.0
1	2	1	1	1	0	C	2	38.0
2	3	1	3	0	0	S	2	26.0
3	4	1	1	1	0	S	2	35.0
4	5	0	3	0	0	S	1	35.0

```
In [28]: def getEmb(str):
    if str=="S":
        return 1
    elif str=='Q':
        return 2
    else:
        return 3
titanic["Embark"] = titanic["Embarked"].apply(getEmb)
titanic.head()
```

Out[28]:

	PassengerId	Survived	Pclass	SibSp	Parch	Embarked	Sex	Age	Embark
0	1	0	3	1	0	S	1	22.0	1
1	2	1	1	1	0	C	2	38.0	3
2	3	1	3	0	0	S	2	26.0	1
3	4	1	1	1	0	S	2	35.0	1
4	5	0	3	0	0	S	1	35.0	1

```
In [29]: del titanic['Embarked']
titanic.rename(columns={'Embark': 'Embarked'}, inplace=True)
titanic.head()
```

Out[29]:

	PassengerId	Survived	Pclass	SibSp	Parch	Sex	Age	Embarked
0	1	0	3	1	0	1	22.0	1
1	2	1	1	1	0	2	38.0	3
2	3	1	3	0	0	2	26.0	1
3	4	1	1	1	0	2	35.0	1
4	5	0	3	0	0	1	35.0	1

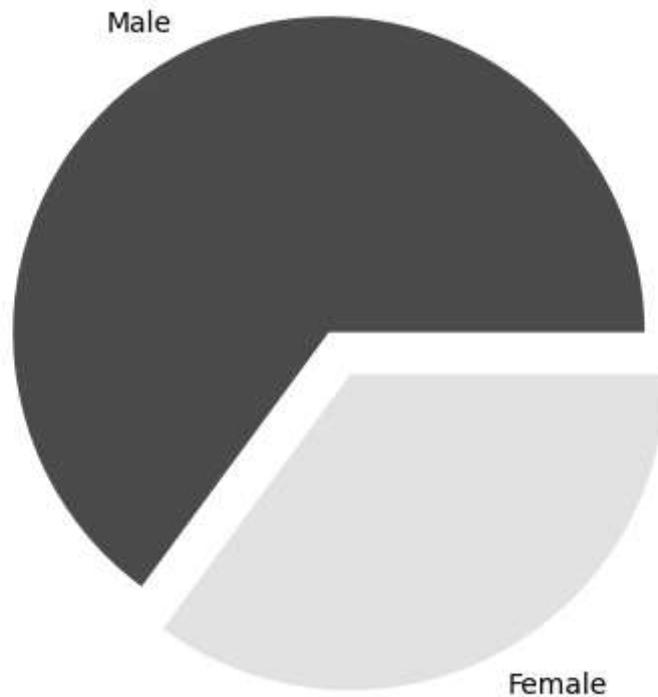
```
In [30]: #Drawing a pie chart for number of males and females aboard
import matplotlib.pyplot as plt
```

```
from matplotlib import style

males = (titanic['Sex'] == 1).sum()
#Summing up all the values of column gender with a
#condition for male and similarly for females
females = (titanic['Sex'] == 2).sum()
print(males)
print(females)
p = [males, females]
plt.pie(p, #giving array
        labels = ['Male', 'Female'], #Correspondingly giving Labels
        colors = ['green', 'yellow'], # Corresponding colors
        explode = (0.15, 0), #How much the gap should be there between the pies
        startangle = 0) #what start angle should be given
plt.axis('equal')
plt.show()
```

577

312



```
In [31]: # More Precise Pie Chart
MaleS=titanic[titanic.Sex==1][titanic.Survived==1].shape[0]
print(MaleS)
MaleN=titanic[titanic.Sex==1][titanic.Survived==0].shape[0]
print(MaleN)
FemaleS=titanic[titanic.Sex==2][titanic.Survived==1].shape[0]
print(FemaleS)
FemaleN=titanic[titanic.Sex==2][titanic.Survived==0].shape[0]
print(FemaleN)
```

109

468

231

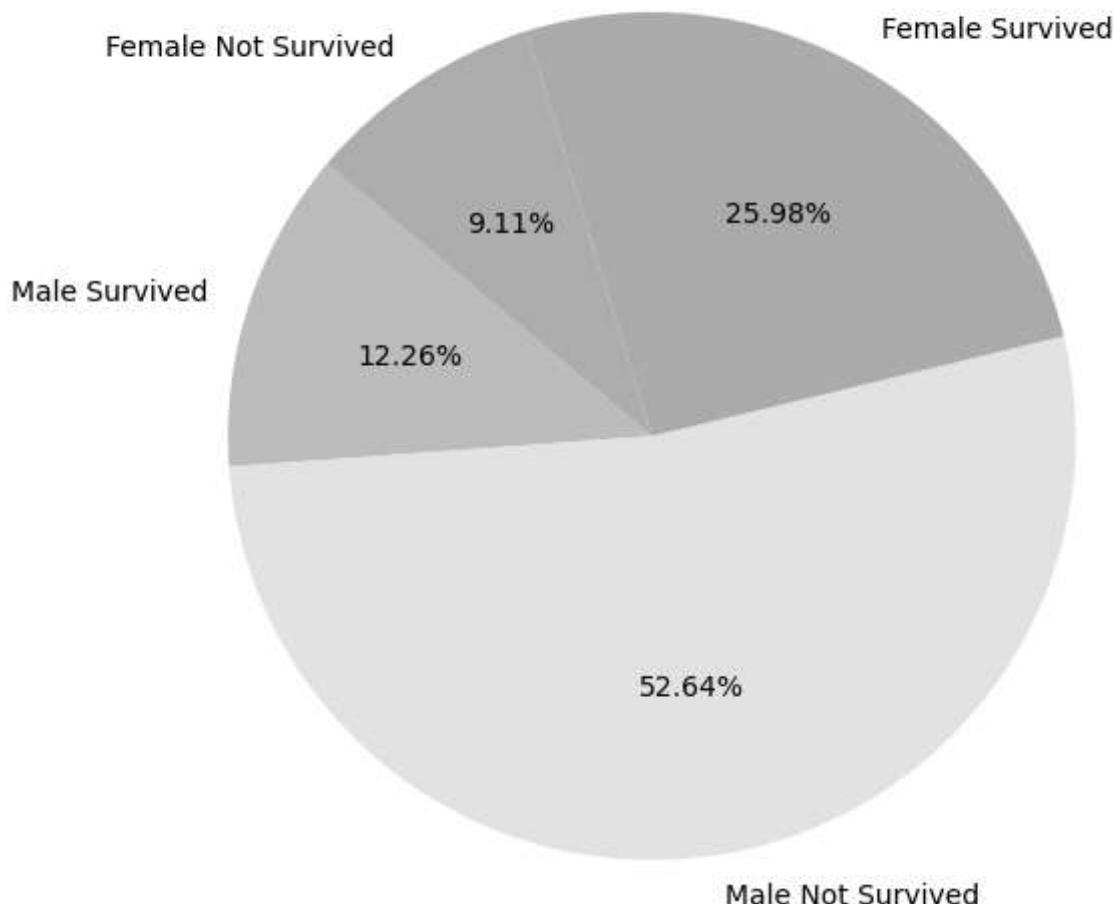
81

```
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3105620411.py:2: UserWarning: Boolean Series key will be reindexed to match DataFrame index.  
    MaleS=titanic[titanic.Sex==1][titanic.Survived==1].shape[0]  
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3105620411.py:4: UserWarning: Boolean Series key will be reindexed to match DataFrame index.  
    MaleN=titanic[titanic.Sex==1][titanic.Survived==0].shape[0]  
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3105620411.py:6: UserWarning: Boolean Series key will be reindexed to match DataFrame index.  
    FemaleS=titanic[titanic.Sex==2][titanic.Survived==1].shape[0]  
C:\Users\ANITHA\AppData\Local\Temp\ipykernel_20296\3105620411.py:8: UserWarning: Boolean Series key will be reindexed to match DataFrame index.  
    FemaleN=titanic[titanic.Sex==2][titanic.Survived==0].shape[0]
```

In [36]:

```
import matplotlib.pyplot as plt  
  
# Data  
sizes = [MaleS, MaleN, FemaleS, FemaleN]  
labels = ['Male Survived', 'Male Not Survived',  
          'Female Survived', 'Female Not Survived']  
  
# Colors similar to your chart  
colors = ['lightskyblue', 'yellow', 'yellowgreen', 'orange']  
  
plt.figure(figsize=(6,6))  
plt.pie(  
    sizes,  
    labels=labels,  
    colors=colors,  
    autopct='%.2f%%',  
    startangle=140)  
  
plt.title('Titanic Survival Distribution')  
plt.axis('equal') # perfect circle  
plt.show()
```

Titanic Survival Distribution



In []: