```
import pandas as pd

import numpy as np


titanic = pd.read_csv(r"C:\Users\Hanshu\Desktop\titanic dataset.csv" ,header = 0, dtype={'Age' : np.float64})


titanic.describe()


#Name column can never decide survival of a person, hence we can safely delete it

del titanic['Name']

titanic.head()


del titanic['Ticket']

titanic.head()


del titanic['Fare']

titanic.head()


del titanic['Cabin']

titanic.head()


# Changing Value for "Male, Female" string values to numeric values , male=1 and female=2

def getNumber(str):

    if str == 'male':

        return 1

    else:

        return 2

titanic["Gender"] = titanic['Sex'].apply(getNumber)

titanic.head()


#Deleting Sex column, since no use of it now
```

```
del titanic['Sex']
titanic.head()
```

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

```
### 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###
meanS = titanic[titanic.Survived==1].Age.mean()
meanS
```

```
### 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###
titanic['age']=np.where(pd.isnull(titanic.Age) & titanic['Survived']==1, meanS, titanic['Age'])
```

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

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

```
titanic.age.fillna(meanNS,inplace=True)
titanic.head()
```

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

```
del titanic['Age']
titanic.head()
```

```
### We want to check if "Embarked" column is is important for analysis or not, that is whether survival of the person depends on the Embarked column value or not###
```

```
# 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)
```

```
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)
```

```
#As there are significant changes in the survival rate based on which port the passengers aboard the ship.
#We cannot delete the whole embarked column(It is useful).
#Now the Embarked column has some null values in it and hence we can safely say that deleting some rows from total rows will not affect the result. So rather than trying to fill those null values with some vales. We can simply remove them.
```

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

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

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


titanic.rename(columns={'Gender':'Sex'}, inplace=True)
titanic.head()


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



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


#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 similary for females
females = (titanic['Sex'] == 2).sum()
print(males)
print(females)
p = [males , females]
```

```
plt.pie(p ,      # giving arry
        labels = ['Male', 'Female'], # correspondingly giving labels
        colors = ['green', 'Yellow'],   # Corresponding colors
        explode = (0.15 , 0),    #How much the gap should me there between the pies
        startangle = 0)  #what start angle should be given
plt.axis('equal')
plt.show()



# 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)



chart=[MaleS,MaleN,FemaleS,FemaleN]
colors=['lightskyblue','yellowgreen','Yellow','Orange']
labels=["Survived Male","Not Survived Male","Survived Female","Not Survived Female"]
explode=[0,0.05,0,0.1]
plt.pie(chart,labels=labels,colors=colors,explode=explode,startangle=100,counterclock=False,autopct="%.2f%%")
plt.axis("equal")
plt.show()
```