In [2]:
```
#import the required libraries
import numpy as np
import pandas as pd
```

In [4]:
```
titanic = pd.read_csv(r"D:\DATAS SCIENCE NIT\SEPTEMBER\17th - ML\TITANIC PROJECT
```

In [5]:
```
titanic
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450
...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376

891 rows × 12 columns

In [6]:
```
titanic.tail()
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7

Performing Data Cleaning and Analysis

In [7]:
```
titanic.describe() #Descriptive statistics
```

Out[7]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329

In [8]:
```
#Name column can never decide survival of a person, hence we can safely delete i
del titanic["Name"] #delete the name column
titanic.head()
```

Out[8]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
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 [9]:
```
del titanic["Ticket"] #delete the ticket column
titanic.head()
```

Out[9]:

	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 [10]:
```
del titanic["Fare"] # delete the fare column
titanic.head()
```

Out[10]:

	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 [11]:
```
del titanic['Cabin'] #delete the cabin column
titanic.head()
```

Out[11]:

	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 [12]:
```
# Changing Value for "Male, Female" string values to numeric values , male=1 and
def getNumber(str): #def function
    if str=="male":  # conditional statement
        return 1
    else:
        return 2
titanic["Gender"]=titanic["Sex"].apply(getNumber) #created a new column
#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[12]:

	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 [13]:
```
#Deleting Sex column, since no use of it now
del titanic["Sex"]
titanic.head()
```

Out[13]:

	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 [14]:
```
titanic.isnull().sum() #see the missing value
```

Out[14]:
```
PassengerId      0
Survived         0
Pclass           0
Age            177
SibSp            0
Parch            0
Embarked         2
Gender           0
dtype: int64
```

In [15]:
```
meanS= titanic[titanic.Survived==1].Age.mean()
meanS
```

Out[15]:
```
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 [16]:
```
titanic["age"]=np.where(pd.isnull(titanic.Age) & titanic["Survived"]==1  ,meanS,
titanic.head()
```

Out[16]:

	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 [18]:
```
titanic.isnull().sum() #see the missing values
```

Out[18]:
```
PassengerId      0
Survived         0
Pclass           0
Age            177
SibSp            0
Parch            0
Embarked         2
Gender           0
age            125
dtype: int64
```

In [19]:
```
# Finding the mean age of "Not Survived" people
meanNS=titanic[titanic.Survived==0].Age.mean()
meanNS
```

Out[19]: 30.62617924528302

In [20]:
```
titanic.age.fillna(meanNS,inplace=True)
titanic.head()
```

```
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\1157731433.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

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


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

Out[20]:

	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 [21]:
```
titanic.isnull().sum() #see the missing value
```

Out[21]:
```
PassengerId      0
Survived         0
Pclass           0
Age            177
SibSp            0
Parch            0
Embarked         2
Gender           0
age              0
dtype: int64
```

In [22]:
```
del titanic['Age'] #delete the Age column
titanic.head()
```

Out[22]:

	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

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

In [23]:
```
# 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\arati\AppData\Local\Temp\ipykernel_20520\3300902897.py:4: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  survivedQ = titanic[titanic.Embarked == 'Q'][titanic.Survived == 1].shape[0]
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\3300902897.py:5: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  survivedC = titanic[titanic.Embarked == 'C'][titanic.Survived == 1].shape[0]
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\3300902897.py:6: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  survivedS = titanic[titanic.Embarked == 'S'][titanic.Survived == 1].shape[0]
```

In [24]:
```
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)
```

```
47
75
427
```

```
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\3240960939.py:1: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  survivedQ = titanic[titanic.Embarked == 'Q'][titanic.Survived == 0].shape[0]
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\3240960939.py:2: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  survivedC = titanic[titanic.Embarked == 'C'][titanic.Survived == 0].shape[0]
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\3240960939.py:3: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  survivedS = titanic[titanic.Embarked == 'S'][titanic.Survived == 0].shape[0]
```

In [25]:
```
titanic.dropna(inplace=True) #drop the missing value
titanic.head()
```

Out[25]:

	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 [26]:
```
titanic.isnull().sum()
```

Out[26]:
```
PassengerId    0
Survived       0
Pclass         0
SibSp          0
Parch          0
Embarked       0
Gender         0
age            0
dtype: int64
```

In [27]:
```
#Renaming "age" and "gender" columns
titanic.rename(columns={'age':'Age'}, inplace=True)
titanic.head()
```

Out[27]:

	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 [28]:
```
titanic.rename(columns={'Gender':'Sex'}, inplace=True) # rename the gender colum
titanic.head()
```

Out[28]:

	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 [29]:
```
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[29]:

	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 [30]:
```
del titanic['Embarked']
titanic.rename(columns={'Embark':'Embarked'}, inplace=True)
titanic.head()
```

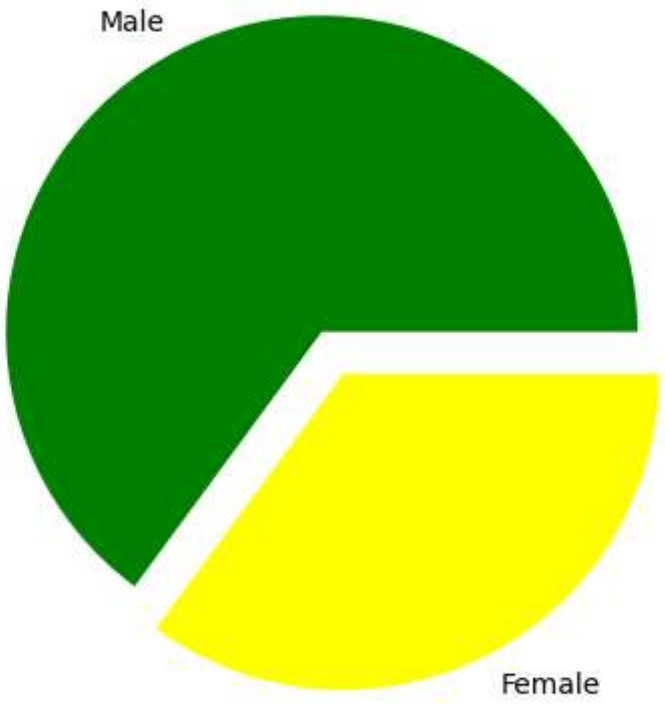
Out[30]:

	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 [31]:
```
#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 array
        labels = ['Male', 'Female'], #Correspndingly giving labels
        colors = ['green', 'yellow'],   # Corresponding colors
        explode = (0.15, 0),    #How much the gap should me there between the pie
        startangle = 0)  #what start angle should be given
plt.axis('equal')
plt.show()
```

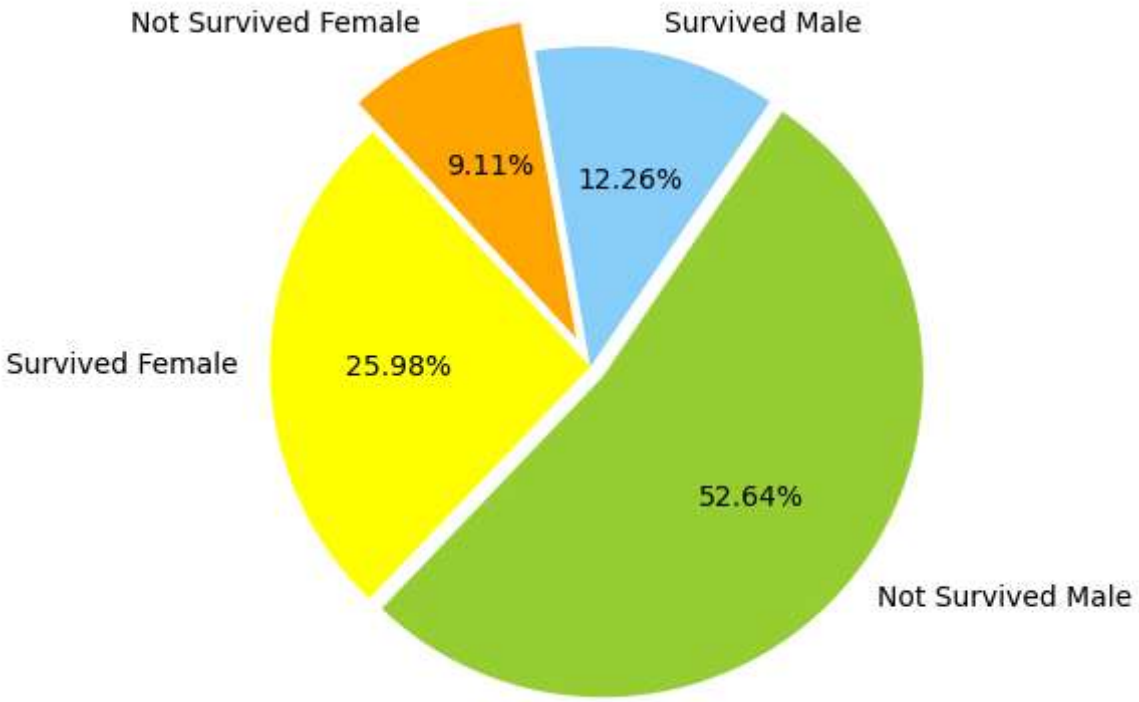
```
577
312
```



In [32]:
```
# 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\arati\AppData\Local\Temp\ipykernel_20520\3105620411.py:2: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  MaleS=titanic[titanic.Sex==1][titanic.Survived==1].shape[0]
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\3105620411.py:4: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  MaleN=titanic[titanic.Sex==1][titanic.Survived==0].shape[0]
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\3105620411.py:6: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  FemaleS=titanic[titanic.Sex==2][titanic.Survived==1].shape[0]
C:\Users\arati\AppData\Local\Temp\ipykernel_20520\3105620411.py:8: UserWarning: B
oolean Series key will be reindexed to match DataFrame index.
  FemaleN=titanic[titanic.Sex==2][titanic.Survived==0].shape[0]
```

In [33]:
```
chart=[MaleS,MaleN,FemaleS,FemaleN]
colors=['lightskyblue','yellowgreen','Yellow','Orange']
labels=["Survived Male","Not Survived Male","Survived Female","Not Survived Fema
explode=[0,0.05,0,0.1]
plt.pie(chart,labels=labels,colors=colors,explode=explode,startangle=100,counter
plt.axis("equal")
plt.show()
```


Not Survived Female
Survived Male
9.11%
12.26%
Survived Female
25.98%
52.64%
Not Survived Male

In [ ]: