```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


from warnings import filterwarnings
filterwarnings(action="ignore")


#loading dataset

pd.set_option('display.max_columns',10,"display.width",1000)
train=pd.read_csv("train.csv")
test=pd.read_csv("test.csv")
train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	...	Parch	Ticket	Fare	Cab
0	1	0	3	Braund, Mr. Owen Harris	male	...	0	A/5 21171	7.2500	N
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	...	0	PC 17599	71.2833	C

```
train.shape
```

```
(891, 12)
```

```
test.shape
```

```
(418, 11)
```

```
train.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
test.isnull().sum()
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

```
#descrription of dataset
train.describe(include='all')
```

	PassengerId	Survived	Pclass	Name	Sex	...	Parch	Ticket	
count	891.000000	891.000000	891.000000	891	891	...	891.000000	891	89
unique	NaN	NaN	NaN	891	2	...	NaN	681	
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	...	NaN	347082	
freq	NaN	NaN	NaN	1	577	...	NaN	7	
mean	446.000000	0.383838	2.308642	NaN	NaN	...	0.381594	NaN	3
std	257.353842	0.486592	0.836071	NaN	NaN	...	0.806057	NaN	4
min	1.000000	0.000000	1.000000	NaN	NaN	...	0.000000	NaN	
25%	223.500000	0.000000	2.000000	NaN	NaN	...	0.000000	NaN	
50%	446.000000	0.000000	3.000000	NaN	NaN	...	0.000000	NaN	1
75%	668.500000	1.000000	3.000000	NaN	NaN	...	0.000000	NaN	3

```
train.groupby('Survived').mean()
```

	PassengerId	Pclass	Age	SibSp	Parch	Fare
Survived						
0	447.016393	2.531876	30.626179	0.553734	0.329690	22.117887
1	444.368421	1.950292	28.343690	0.473684	0.464912	48.395408

```
train.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

```
male_ind=len(train[train['Sex']=='male'])
print('No males in titanic:',male_ind)
```

```
     No males in titanic: 577
```

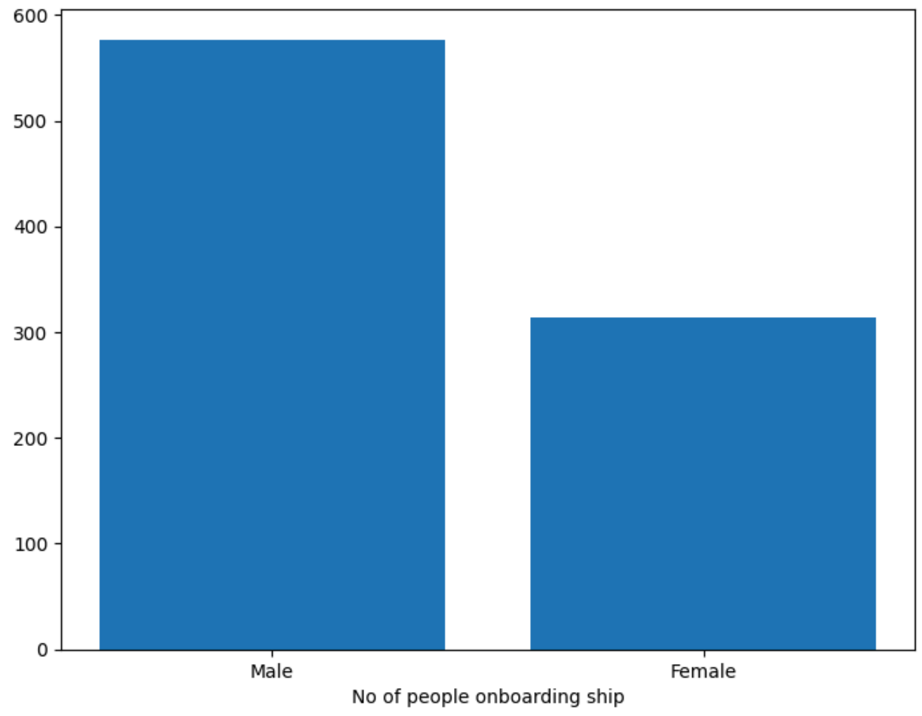
```
female_ind=len(train[train['Sex']=='female'])
print('No females in titanic:',female_ind)
```

```
     No females in titanic: 314
```

```
#plotting
```

```
fig=plt.figure()

ax=fig.add_axes([0,0,1,1])
gender=['Male','Female']
index={577,314}
ax.bar(gender,index)
plt.xlabel("Gender")
plt.xlabel("No of people onboarding ship")
plt.show
```



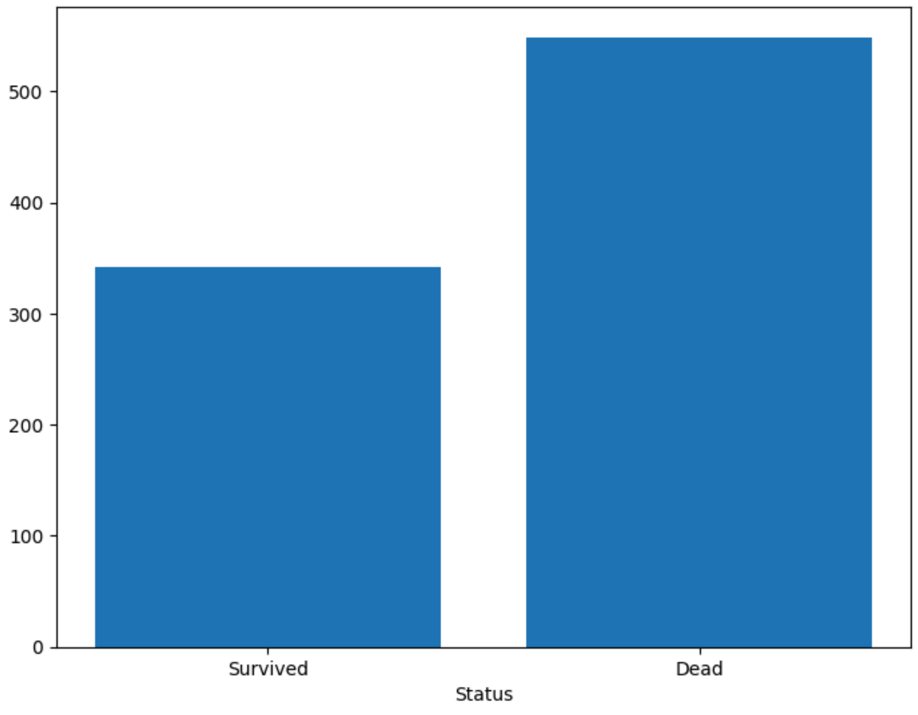
```
alive=len(train[train['Survived']==1])
```

```
dead=len(train[train['Survived']==0])
```

```
train.groupby('Sex')[['Survived']].mean()
```

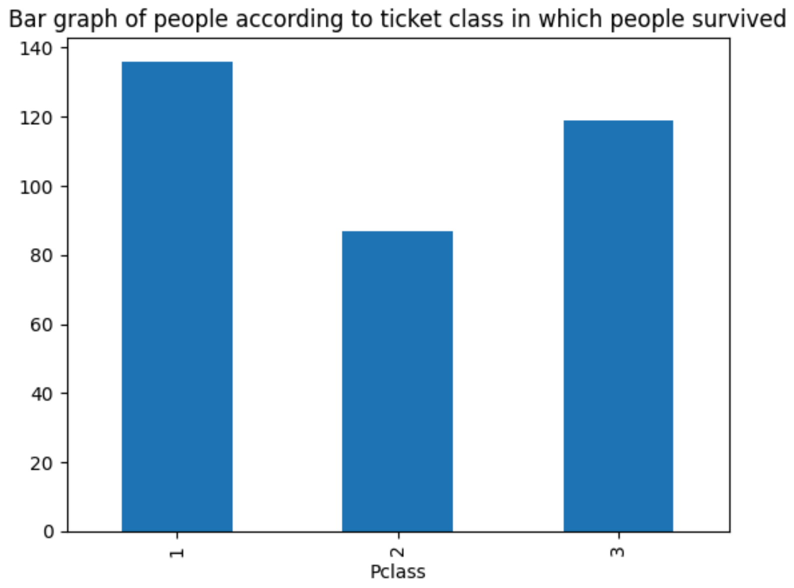
	Survived
Sex	
female	0.742038
male	0.188908

```
fig=plt.figure()
ax=fig.add_axes([0,0,1,1])
Status=['Survived','Dead']
ind=[alive,dead]
ax.bar(Status,ind)
plt.xlabel('Status')
plt.show()
```



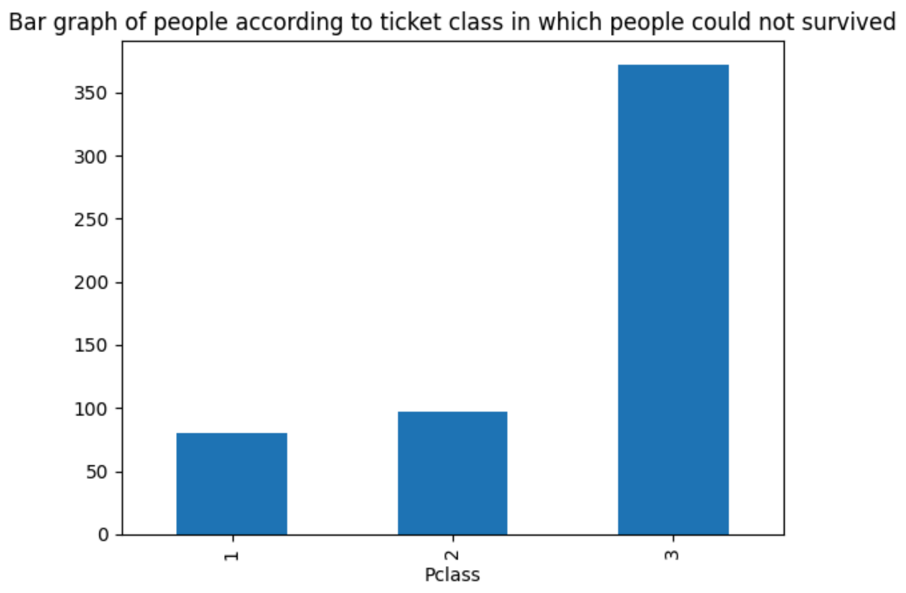
```
plt.figure(1)
train.loc[train['Survived'] == 1,'Pclass'].value_counts().sort_index().plot.bar()
plt.title("Bar graph of people according to ticket class in which people survived")
```

```
Text(0.5, 1.0, 'Bar graph of people according to ticket class in which people survived')
```



```
plt.figure(2)
train.loc[train['Survived'] == 0,'Pclass'].value_counts().sort_index().plot.bar()
plt.title("Bar graph of people according to ticket class in which people could not survived")
```

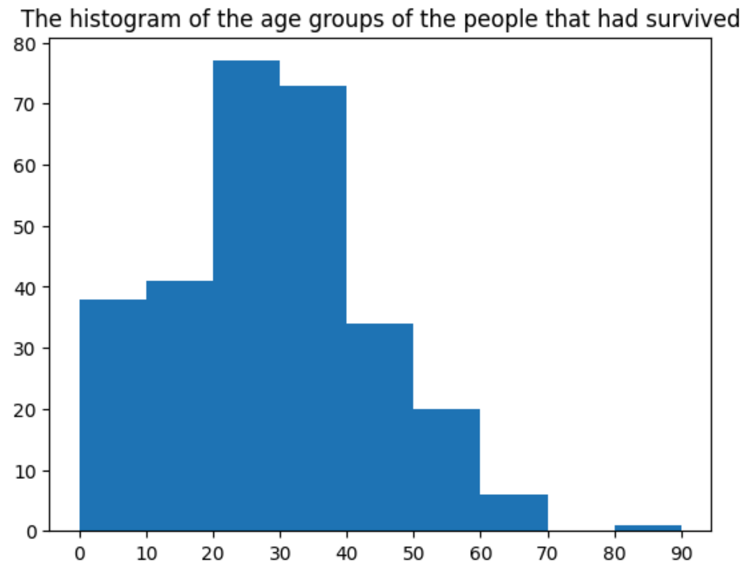
```
Text(0.5, 1.0, 'Bar graph of people according to ticket class in which people could not survived')
```



```
plt.figure(1)
age = train.loc[train.Survived == 1, 'Age']

plt.title("The histogram of the age groups of the people that had survived")
plt.hist(age,np.arange(0,100,10))
plt.xticks(np.arange(0,100,10))
```

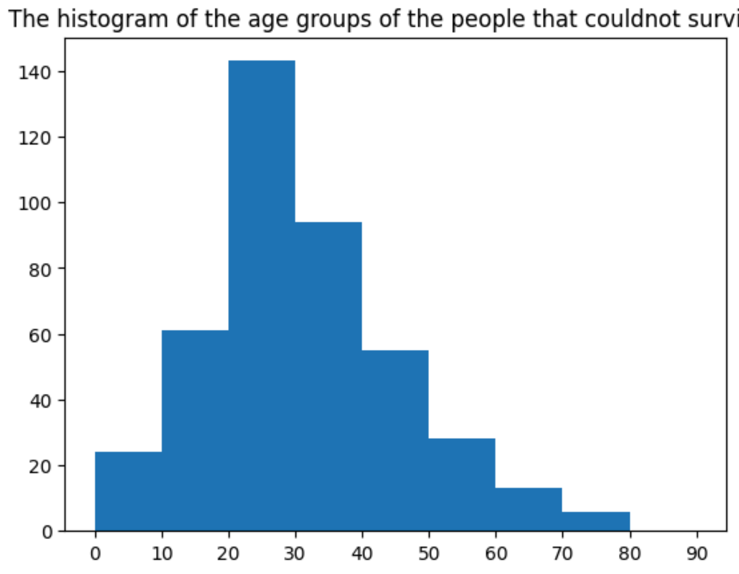
```
([<matplotlib.axis.XTick at 0x7e195df21ff0>,
  <matplotlib.axis.XTick at 0x7e195df22080>,
  <matplotlib.axis.XTick at 0x7e195df22800>,
  <matplotlib.axis.XTick at 0x7e195d9aaf20>,
  <matplotlib.axis.XTick at 0x7e195d9a8310>,
  <matplotlib.axis.XTick at 0x7e195d9ab370>,
  <matplotlib.axis.XTick at 0x7e195da30610>,
  <matplotlib.axis.XTick at 0x7e195d9a89d0>,
  <matplotlib.axis.XTick at 0x7e195da31030>,
  <matplotlib.axis.XTick at 0x7e195da31ae0>],
 [Text(0, 0, '0'),
  Text(10, 0, '10'),
  Text(20, 0, '20'),
  Text(30, 0, '30'),
  Text(40, 0, '40'),
  Text(50, 0, '50'),
  Text(60, 0, '60'),
  Text(70, 0, '70'),
  Text(80, 0, '80'),
  Text(90, 0, '90')])
```



```
plt.figure(1)
age = train.loc[train.Survived == 0, 'Age']

plt.title("The histogram of the age groups of the people that couldnot survived")
plt.hist(age,np.arange(0,100,10))
plt.xticks(np.arange(0,100,10))
```

```
([<matplotlib.axis.XTick at 0x7e195da1dea0>,
  <matplotlib.axis.XTick at 0x7e195da1de70>,
  <matplotlib.axis.XTick at 0x7e195da1da80>,
  <matplotlib.axis.XTick at 0x7e195da5b280>,
  <matplotlib.axis.XTick at 0x7e195e077b20>,
  <matplotlib.axis.XTick at 0x7e195db40640>,
  <matplotlib.axis.XTick at 0x7e195db410f0>,
  <matplotlib.axis.XTick at 0x7e195da598d0>,
  <matplotlib.axis.XTick at 0x7e195db41b10>,
  <matplotlib.axis.XTick at 0x7e195db425c0>],
 [Text(0, 0, '0'),
  Text(10, 0, '10'),
  Text(20, 0, '20'),
  Text(30, 0, '30'),
  Text(40, 0, '40'),
  Text(50, 0, '50'),
  Text(60, 0, '60'),
  Text(70, 0, '70'),
  Text(80, 0, '80'),
  Text(90, 0, '90')])
```



```
train[['SibSp','Survived']].groupby(['SibSp'],as_index=False).mean().sort_values(by='Survived',ascending=False)
```

	SibSp	Survived
1	1	0.535885
2	2	0.464286
0	0	0.345395
3	3	0.250000
4	4	0.166667
5	5	0.000000
6	8	0.000000

```
train[['Pclass','Survived']].groupby(['Pclass'],as_index=False).mean().sort_values(by='Survived',ascending=False)
```

	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.242363

```
train[['Age','Survived']].groupby(['Age'],as_index=False).mean().sort_values(by='Age',ascending=False)
```

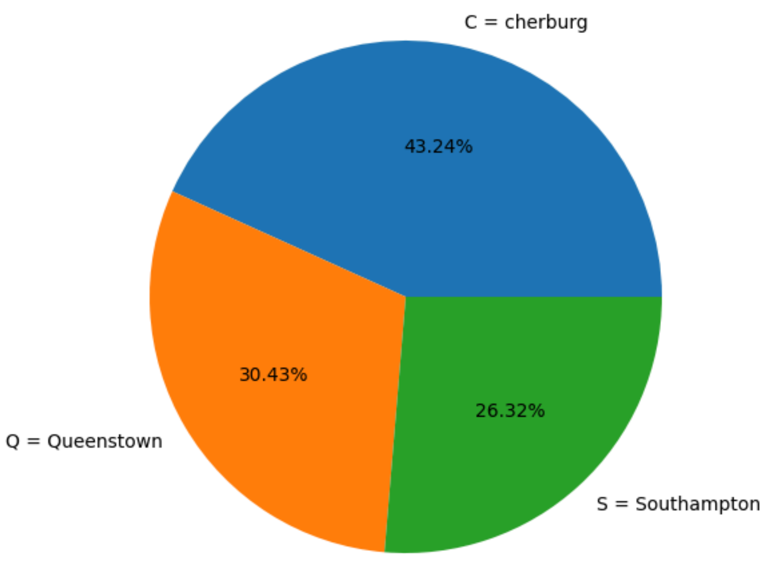
	Age	Survived
87	80.00	1.0
86	74.00	0.0
85	71.00	0.0
84	70.50	0.0
83	70.00	0.0
...	...	...
4	0.92	1.0
3	0.83	1.0
2	0.75	1.0
1	0.67	1.0
0	0.42	1.0

88 rows × 2 columns

```
train[['Embarked','Survived']].groupby(['Embarked'],as_index=False).mean().sort_values(by='Survived',ascending=False)
```

	Embarked	Survived
0	C	0.553571
1	Q	0.389610
2	S	0.336957

```
fig= plt.figure()
ax =fig.add_axes([0,0,1,1])
ax.axis('equal')
l = ['C = cherburg','Q = Queenstown','S = Southampton']
s =[0.553571,0.389610,0.336957]
ax.pie(s,labels= l,autopct='%1.2f%%')
plt.show()
```



```
test.describe(include='all')
```

	PassengerId	Pclass	Name	Sex	Age	...	Parch	Ticket	
count	418.000000	418.000000	418	418	332.000000	...	418.000000	418	417.
unique	NaN	NaN	418	2	NaN	...	NaN	363	
top	NaN	NaN	Kelly, Mr. James	male	NaN	...	NaN	PC 17608	
freq	NaN	NaN	1	266	NaN	...	NaN	5	
mean	1100.500000	2.265550	NaN	NaN	30.272590	...	0.392344	NaN	35.
std	120.810458	0.841838	NaN	NaN	14.181209	...	0.981429	NaN	55.
min	892.000000	1.000000	NaN	NaN	0.170000	...	0.000000	NaN	0.
25%	996.250000	1.000000	NaN	NaN	21.000000	...	0.000000	NaN	7.
50%	1100.500000	3.000000	NaN	NaN	27.000000	...	0.000000	NaN	14.
75%	1204.750000	3.000000	NaN	NaN	39.000000	...	0.000000	NaN	31.

```
train= train.drop(['Ticket'],axis=1)
test= test.drop(['Ticket'],axis=1)
```

```
train= train.drop(['Cabin'],axis=1)
test= test.drop(['Cabin'],axis=1)
```

```
train= train.drop(['Name'],axis=1)
test= test.drop(['Name'],axis=1)
```

```
#feutures selection
column_train=['Age','Pclass','SibSp','Parch','Fare','Sex','Embarked']

#training values

X=train[column_train]

#target value

Y=train['Survived']
```

```
X['Age'].isnull().sum()
X['Pclass'].isnull().sum()
X['SibSp'].isnull().sum()
X['Parch'].isnull().sum()
X['Fare'].isnull().sum()
X['Sex'].isnull().sum()
X['Embarked'].isnull().sum()
```

```
2
```

```
X['Age']=X['Age'].fillna(X['Age'].median())
X['Age'].isnull().sum()
```

```
0
```

```
X['Embarked']=train['Embarked'].fillna(method='pad')
X['Embarked'].isnull().sum()
```

```
0
```

```
d={'male':0,'female':1}
X['Sex']=X['Sex'].apply(lambda x:d[x])
X['Sex'].head()
```

```
0    0
1    1
```

```
     2    1
     3    1
     4    0
     Name: Sex, dtype: int64
```

```
e={'C':0,'Q':1,'S':2}
X['Embarked']=X['Embarked'].apply(lambda x:e[x])
X['Embarked'].head()
```

```
     0    2
     1    0
     2    2
     3    2
     4    2
     Name: Embarked, dtype: int64
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=7)
```

```
from sklearn.linear_model import LogisticRegression
model= LogisticRegression()
model.fit(X_train,Y_train)
Y_pred=model.predict(X_test)

from sklearn.metrics import accuracy_score
print("Accuracy Score;",accuracy_score(Y_test,Y_pred))
```

```
     Accuracy Score; 0.753731343283582
```

```
from sklearn.metrics import accuracy_score,confusion_matrix
confusion_mat = confusion_matrix(Y_test,Y_pred)
print(confusion_mat)
```

```
     [[133  23]
      [ 43  69]]
```

```
from sklearn.svm import SVC

model1=SVC()
model1.fit(X_train,Y_train)

pred_y = model1.predict(X_test)
from sklearn.metrics import accuracy_score
```