```
#Importing All Required Libaries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

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

#Loading Datasets
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 Cabin  Embarked
0            1         0       3                              Braund,
Mr. Owen Harris    male  ...      0         A/5 21171   7.2500   NaN
S
1            2         1       1  Cumings, Mrs. John Bradley (Florence
Briggs Th...  female  ...      0          PC 17599  71.2833   C85
C
2            3         1       3
Heikkinen, Miss. Laina  female  ...      0  STON/O2. 3101282   7.9250
NaN         S
3            4         1       1       Futrelle, Mrs. Jacques Heath
(Lily May Peel)  female  ...      0            113803  53.1000  C123
S
4            5         0       3                             Allen, Mr.
William Henry    male  ...      0            373450   8.0500   NaN
S

[5 rows x 12 columns]
```

```
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=True)
```

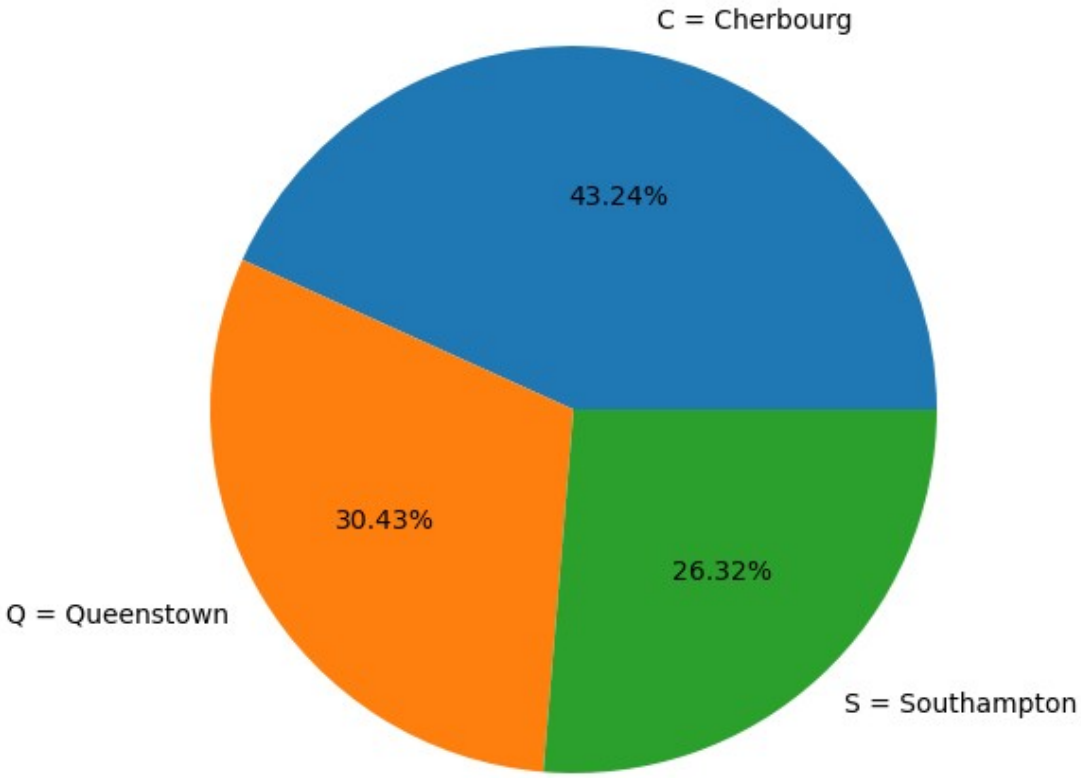
```
      Age  Survived
0    0.42       1.0
1    0.67       1.0
2    0.75       1.0
3    0.83       1.0
4    0.92       1.0
..    ...       ...
83  70.00       0.0
84  70.50       0.0
85  71.00       0.0
86  74.00       0.0
87  80.00       1.0

[88 rows x 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 = Cherbourg', '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        Fare           Cabin Embarked
count    418.000000  418.000000                 418   418
332.000000  ...  418.000000       418  417.000000                91
418
unique          NaN         NaN                 418     2
NaN  ...         NaN       363         NaN                76         3
top             NaN         NaN  Kelly, Mr. James  male
NaN  ...         NaN  PC 17608         NaN  B57 B59 B63 B66         S
freq            NaN         NaN                   1   266
NaN  ...         NaN         5         NaN                3      270
mean    1100.500000    2.265550                 NaN   NaN
30.272590  ...    0.392344       NaN   35.627188                NaN
NaN
std      120.810458    0.841838                 NaN   NaN
14.181209  ...    0.981429       NaN   55.907576                NaN
NaN
min      892.000000    1.000000                 NaN   NaN
0.170000  ...    0.000000       NaN    0.000000                NaN
```

```
NaN
25%      996.250000    1.000000                NaN   NaN
21.000000  ...    0.000000       NaN    7.895800              NaN
NaN
50%     1100.500000    3.000000                NaN   NaN
27.000000  ...    0.000000       NaN   14.454200              NaN
NaN
75%     1204.750000    3.000000                NaN   NaN
39.000000  ...    0.000000       NaN   31.500000              NaN
NaN
max     1309.000000    3.000000                NaN   NaN
76.000000  ...    9.000000       NaN  512.329200              NaN
NaN

[11 rows x 11 columns]
```

```
#Droping Useless Columns
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)
```