In [10]:
```
#Importing All Required Libaries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

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

In [12]:
```
pd.set_option('display.max_columns',10,'display.width',1000)
test = pd.read_csv(r'C:\Users\Lenovo\Desktop\SURAJ TASK 2 DATASET\train.csv')
test = pd.read_csv(r'C:\Users\Lenovo\Desktop\SURAJ TASK 2 DATASET\test.csv')
```

In [3]:
```
train.shape
```

Out[3]:
```
(891, 12)
```

In [4]:
```
test.shape
```

Out[4]:
```
(418, 11)
```

In [5]:
```
train.isnull().sum()
```

Out[5]:
```
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
```

In [6]:
```
test.isnull().sum()
```

Out[6]:
```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [7]:
```
#Description of dataset
train.describe(include="all")
```

Out[7]:

	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
max	891.000000	1.000000	3.000000	NaN	NaN	...	6.000000	NaN	51

11 rows × 12 columns

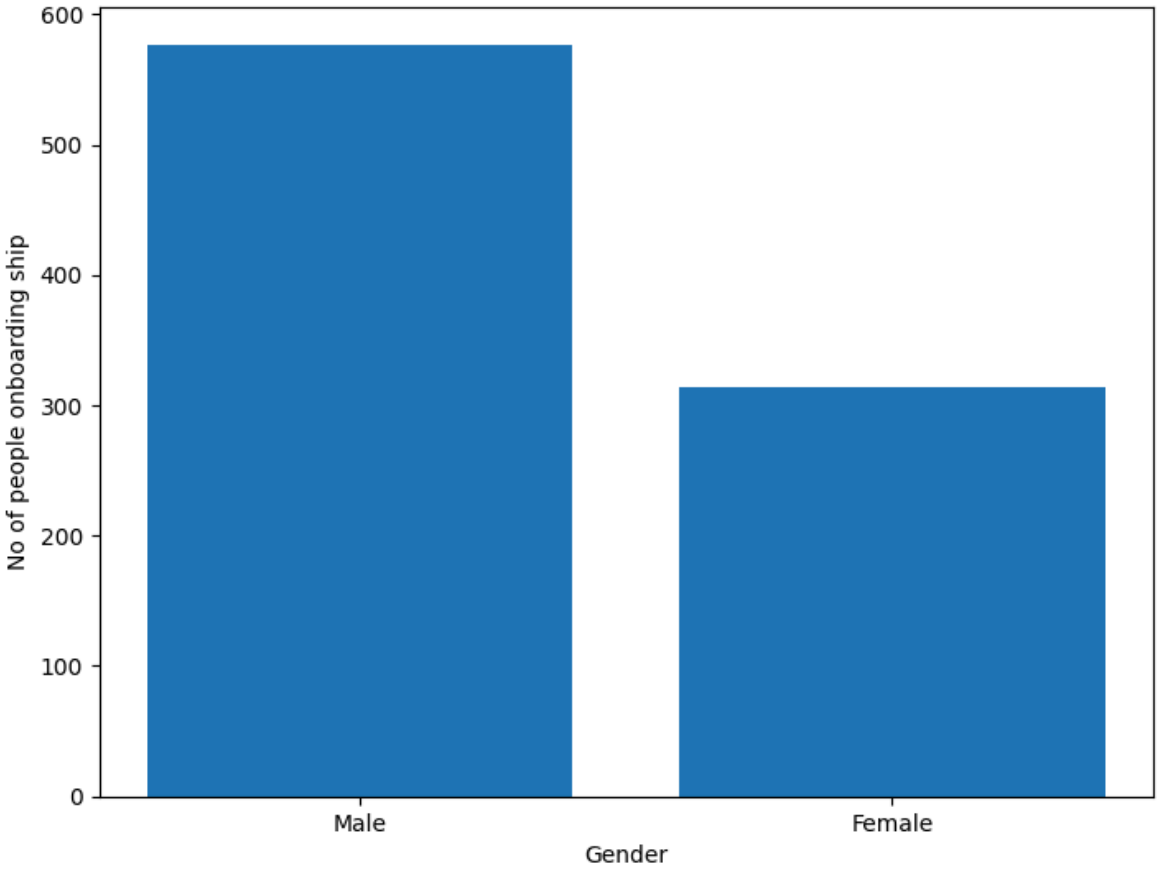
In [21]:
```
male_ind = len(train[train['Sex'] == 'male'])
print("No of Males in Titanic:",male_ind)
```

```
No of Males in Titanic: 577
```

In [22]:
```
female_ind = len(train[train['Sex'] == 'female'])
print("No of Females in Titanic:",female_ind)
```

```
No of Females in Titanic: 314
```

In [23]:
```
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.ylabel("No of people onboarding ship")
plt.show()
```



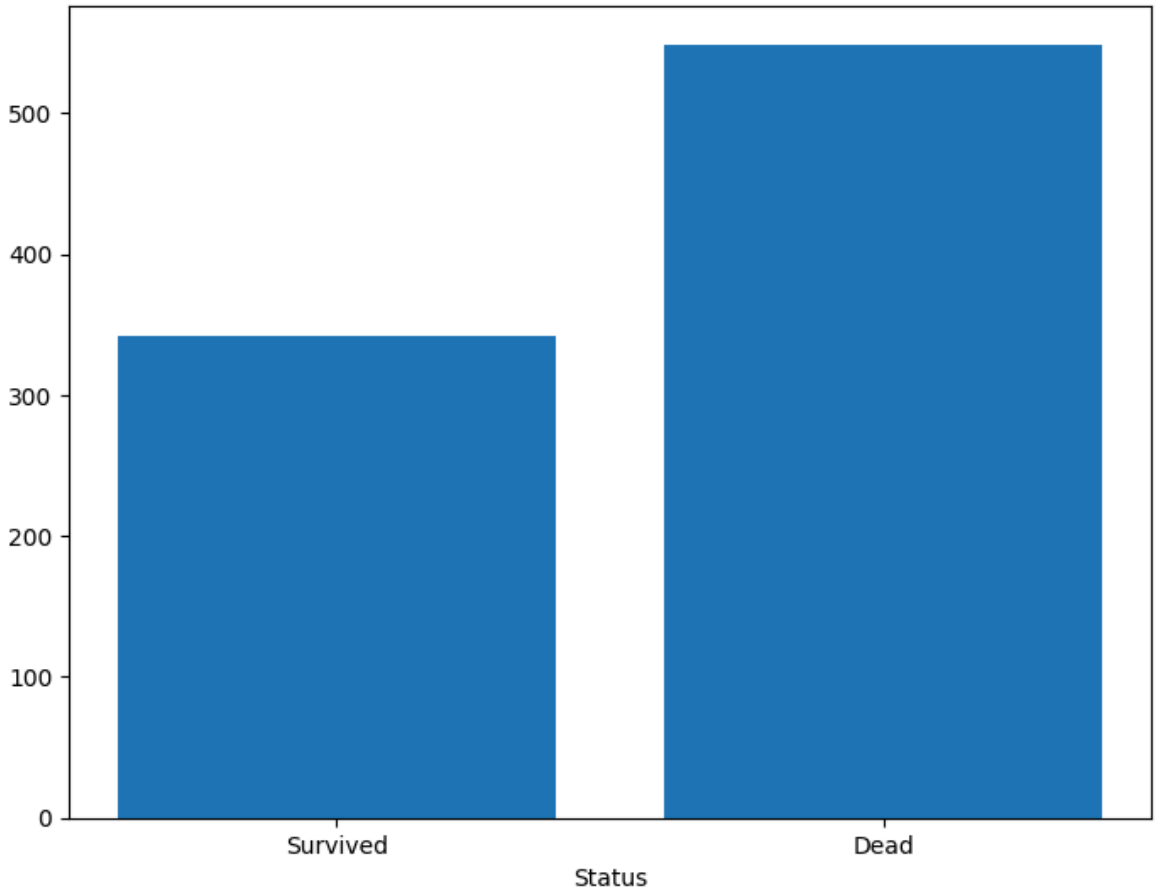
In [24]:
```
alive = len(train[train['Survived'] == 1])
dead = len(train[train['Survived'] == 0])
```

In [25]:
```
train.groupby('Sex')[['Survived']].mean()
```

Out[25]:

Sex	Survived
female	0.742038
male	0.188908

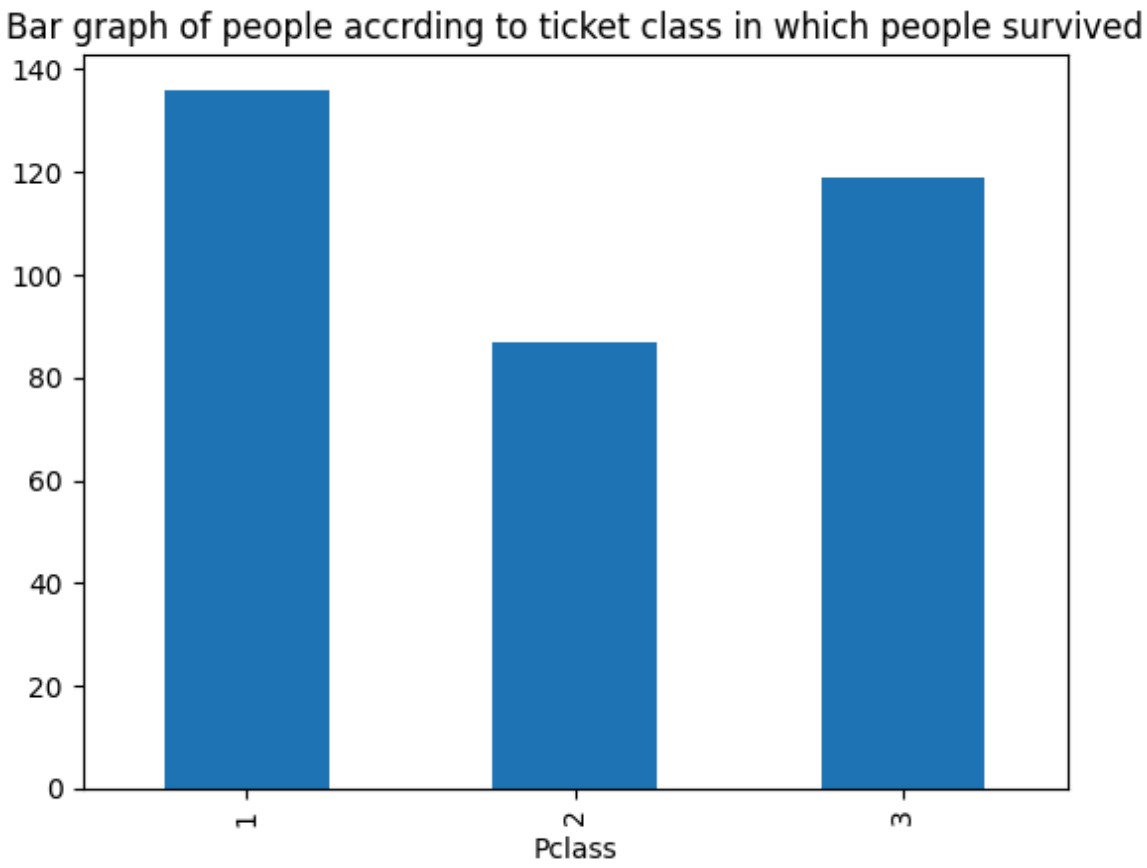
In [26]:
```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
status = ['Survived','Dead']
ind = [alive,dead]
ax.bar(status,ind)
plt.xlabel("Status")
plt.show()
```

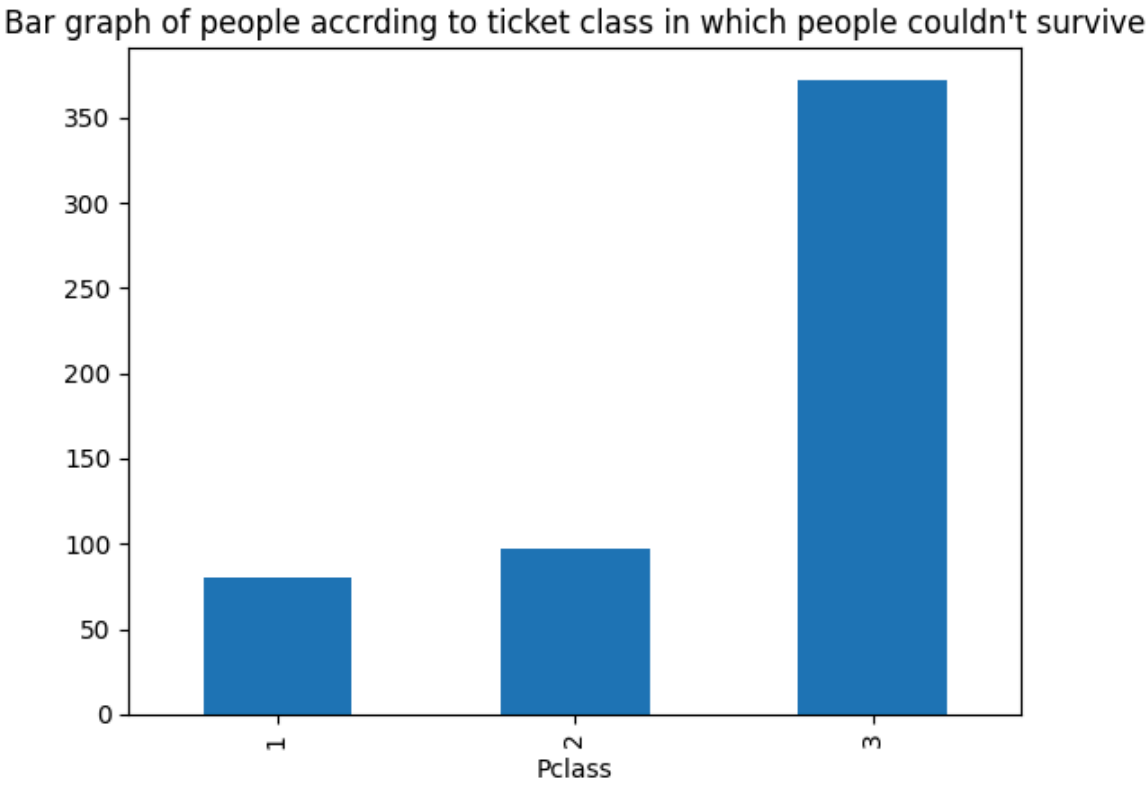


In [27]:
```
plt.figure(1)
train.loc[train['Survived'] == 1, 'Pclass'].value_counts().sort_index().plot.bar
plt.title('Bar graph of people accrding to ticket class in which people survived


plt.figure(2)
train.loc[train['Survived'] == 0, 'Pclass'].value_counts().sort_index().plot.bar
plt.title('Bar graph of people accrding to ticket class in which people couldn\'
```

Out[27]:
```
Text(0.5, 1.0, "Bar graph of people accrding to ticket class in which people co
uldn't survive")
```



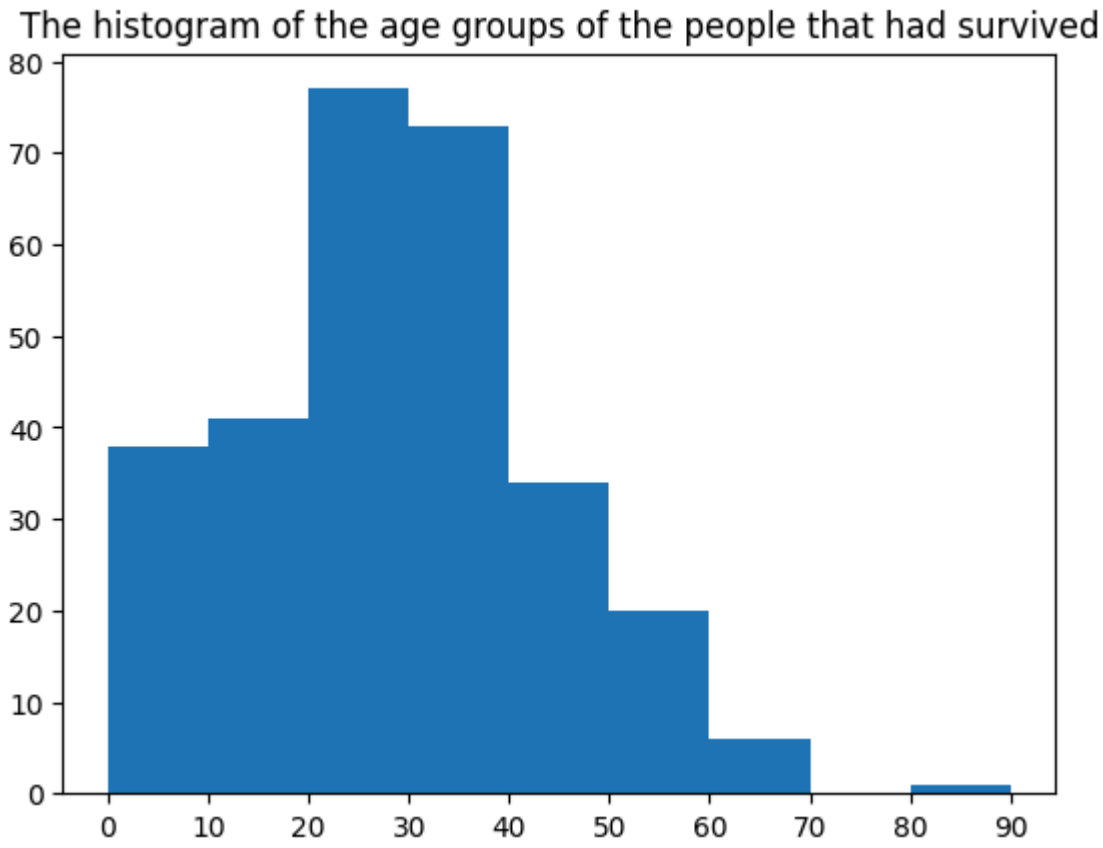


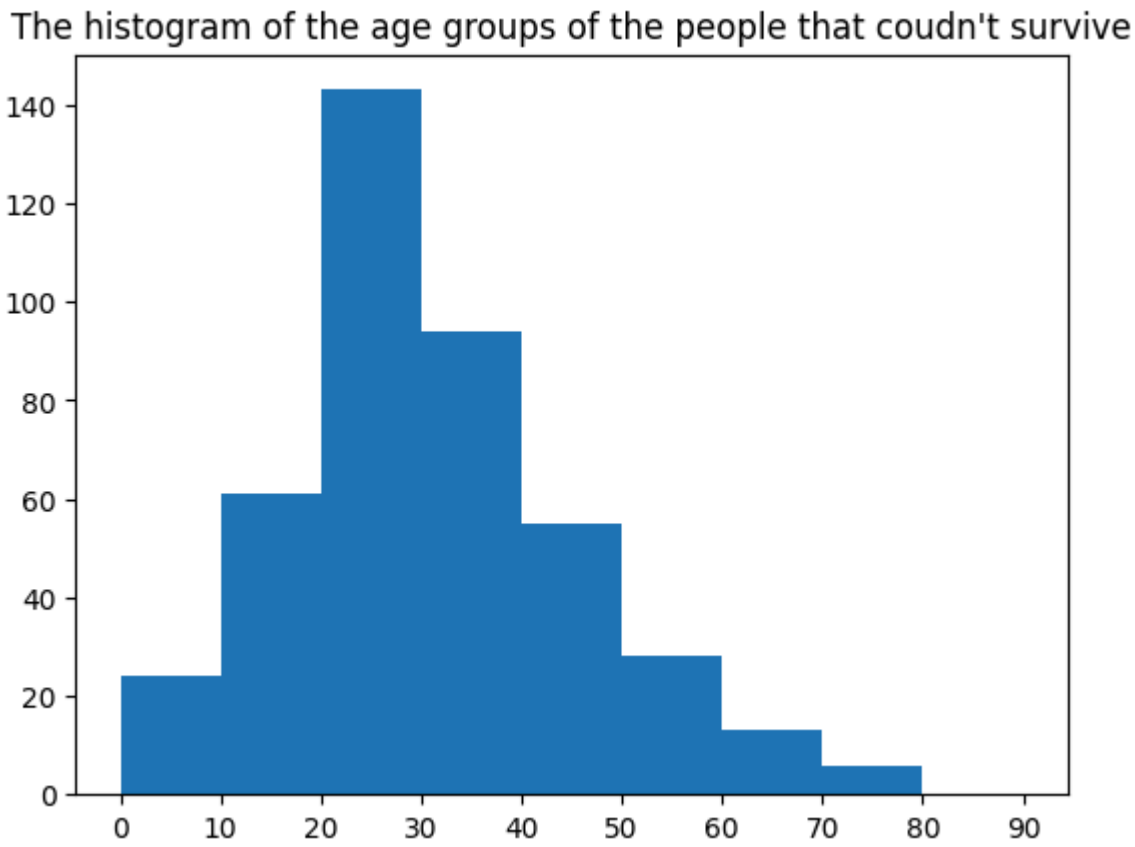
In [28]:
```
plt.figure(1)
age  = train.loc[train.Survived == 1, 'Age']
plt.title('The histogram of the age groups of the people that had survived')
plt.hist(age, np.arange(0,100,10))
plt.xticks(np.arange(0,100,10))


plt.figure(2)
```

```
age  = train.loc[train.Survived == 0, 'Age']
plt.title('The histogram of the age groups of the people that coudn\'t survive')
plt.hist(age, np.arange(0,100,10))
plt.xticks(np.arange(0,100,10))
```

Out[28]:
```
([<matplotlib.axis.XTick at 0x2196d658a40>,
  <matplotlib.axis.XTick at 0x2196d6683b0>,
  <matplotlib.axis.XTick at 0x2196d666390>,
  <matplotlib.axis.XTick at 0x2196d66d490>,
  <matplotlib.axis.XTick at 0x2196d66de50>,
  <matplotlib.axis.XTick at 0x2196d66e810>,
  <matplotlib.axis.XTick at 0x2196d66f1a0>,
  <matplotlib.axis.XTick at 0x2196d66fa70>,
  <matplotlib.axis.XTick at 0x2196d66f4a0>,
  <matplotlib.axis.XTick at 0x2196d670260>],
 [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')])
```





In [29]:
```
train[["SibSp", "Survived"]].groupby(['SibSp'], as_index=False).mean().sort_valu
```

Out[29]:

	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

In [30]:
```
train[["Pclass", "Survived"]].groupby(['Pclass'], as_index=False).mean().sort_va
```

Out[30]:

	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.242363

In [31]:
```
train[["Age", "Survived"]].groupby(['Age'], as_index=False).mean().sort_values(b
```

Out[31]:

	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

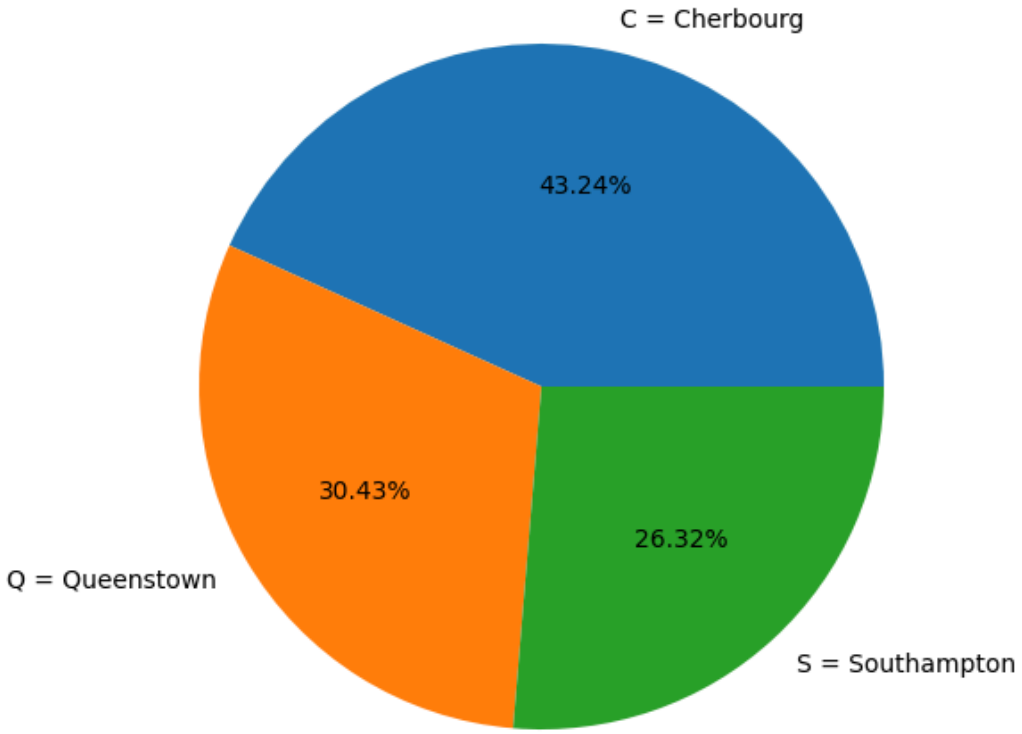
88 rows × 2 columns

In [32]:
```
train[["Embarked", "Survived"]].groupby(['Embarked'], as_index=False).mean().sor
```

Out[32]:

	Embarked	Survived
0	C	0.553571
1	Q	0.389610
2	S	0.336957

In [33]:
```
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()
```



In [34]:
```
test.describe(include="all")
```

Out[34]:

	PassengerId	Pclass	Name	Sex	Age	...	Parch	Ticket	
count	418.000000	418.000000	418	418	332.000000	...	418.000000	418	417.0
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	35.6
std	120.810458	0.841838	NaN	NaN	14.181209	...	0.981429	NaN	55.9
min	892.000000	1.000000	NaN	NaN	0.170000	...	0.000000	NaN	0.0
25%	996.250000	1.000000	NaN	NaN	21.000000	...	0.000000	NaN	7.8
50%	1100.500000	3.000000	NaN	NaN	27.000000	...	0.000000	NaN	14.4
75%	1204.750000	3.000000	NaN	NaN	39.000000	...	0.000000	NaN	31.5
max	1309.000000	3.000000	NaN	NaN	76.000000	...	9.000000	NaN	512.3

11 rows × 11 columns

In [35]:
```
train = train.drop(['Ticket'], axis = 1)
```

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

In [36]:
```
train = train.drop(['Cabin'], axis = 1)
test = test.drop(['Cabin'], axis = 1)
```

In [37]:
```
train = train.drop(['Name'], axis = 1)
test = test.drop(['Name'], axis = 1)
```

In [40]:
```
X['Age']=X['Age'].fillna(X['Age'].median())
X['Age'].isnull().sum()
```

Out[40]:
```
0
```

In [41]:
```
X['Embarked'] = train['Embarked'].fillna(method ='pad')
X['Embarked'].isnull().sum()
```

Out[41]:
```
0
```

In [42]:
```
d={'male':0, 'female':1}
X['Sex']=X['Sex'].apply(lambda x:d[x])
X['Sex'].head()
```

Out[42]:
```
0    0
1    1
2    1
3    1
4    0
Name: Sex, dtype: int64
```

In [22]:
```
results = pd.DataFrame({
    'Model': ['Logistic Regression','Support Vector Machines', 'Naive Bayes','KN
    'Score': [0.75,0.66,0.76,0.66,0.74]})

result_df = results.sort_values(by='Score', ascending=False)
result_df = result_df.set_index('Score')
result_df.head(9)
```

Out[22]:

Score	Model
0.76	Naive Bayes
0.75	Logistic Regression
0.74	Decision Tree
0.66	Support Vector Machines
0.66	KNN

In [ ]:
```

```