In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [27]:
```
df= pd.read_csv("E:\Titanic dataset.csv")
```

In [28]:
```
df.head(10)
```

Out[28]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabi
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN

In [29]:
```
df.drop(["PassengerId","Ticket"],axis =1 , inplace = True)
```

From common sense , columns such as PassengerId , Nmae and Ticket number shouldn't be related to the survival probability . so these columns can be droped. it is also seen that there are missing values in Age and Cabin columns which needs to be handled properly.

for additional field knowledge of titanic survivors : https://titanicfacts.net/titanic-survivors/

In [30]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Survived  891 non-null    int64  
 1   Pclass    891 non-null    int64  
 2   Name      891 non-null    object 
 3   Sex       891 non-null    object 
 4   Age       714 non-null    float64
 5   SibSp     891 non-null    int64  
 6   Parch     891 non-null    int64  
 7   Fare      891 non-null    float64
 8   Cabin     204 non-null    object 
 9   Embarked  889 non-null    object 
dtypes: float64(2), int64(4), object(4)
memory usage: 69.7+ KB
```

It indicates that there are total of 891 passenger details among which 177 people's Age is missing and 687 people's Cabin details are missing. And 2 people's Embarktion details are missing.

In [31]:
```
df.describe()
```

Out[31]:

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In training set :

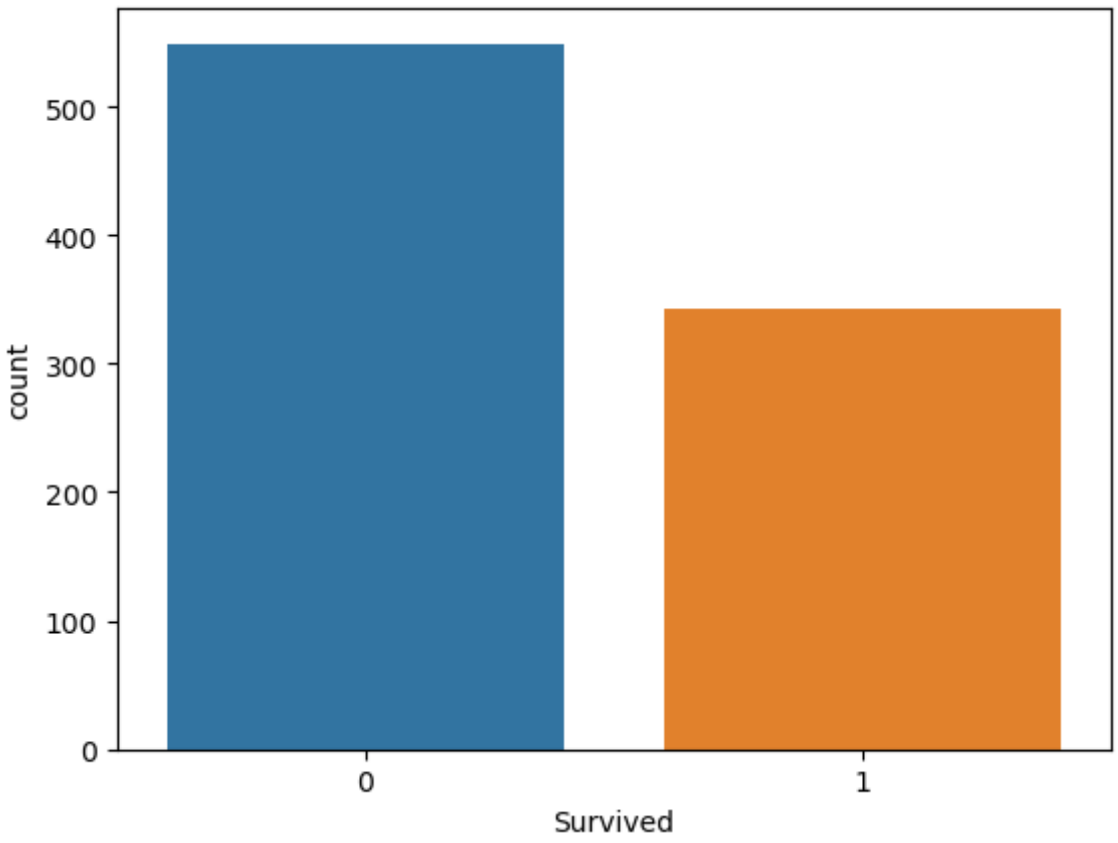
1. 38.3% people survived
2. More number of people were actually in 3rd class.
3. 50% of passengers were in between the age of 20 to 38

Since the survival rate is 0.38, even if decide to give a submission of all passengers being perished , i would still be having a accuracy of 62%. so accuracy cannot br considered as the

only measure in saying how good the model is .

In [32]:
```
sns.countplot(x='Survived' , data=df)
```

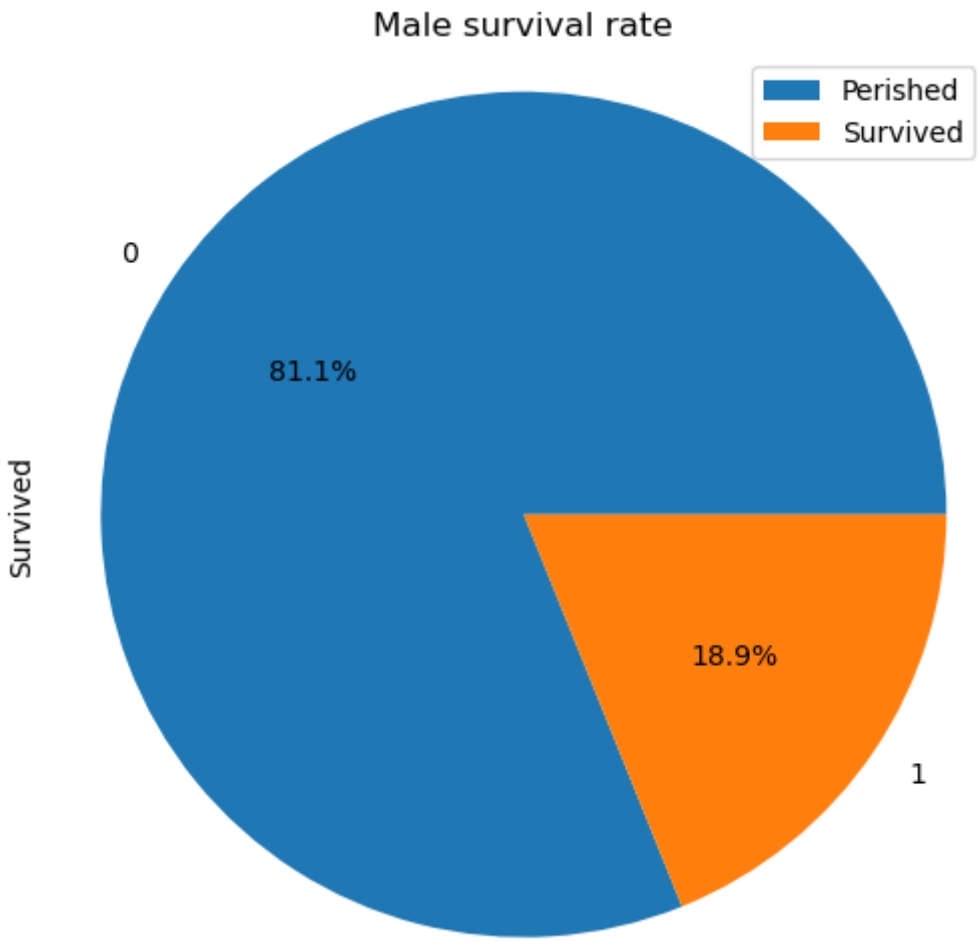
Out[32]:
```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



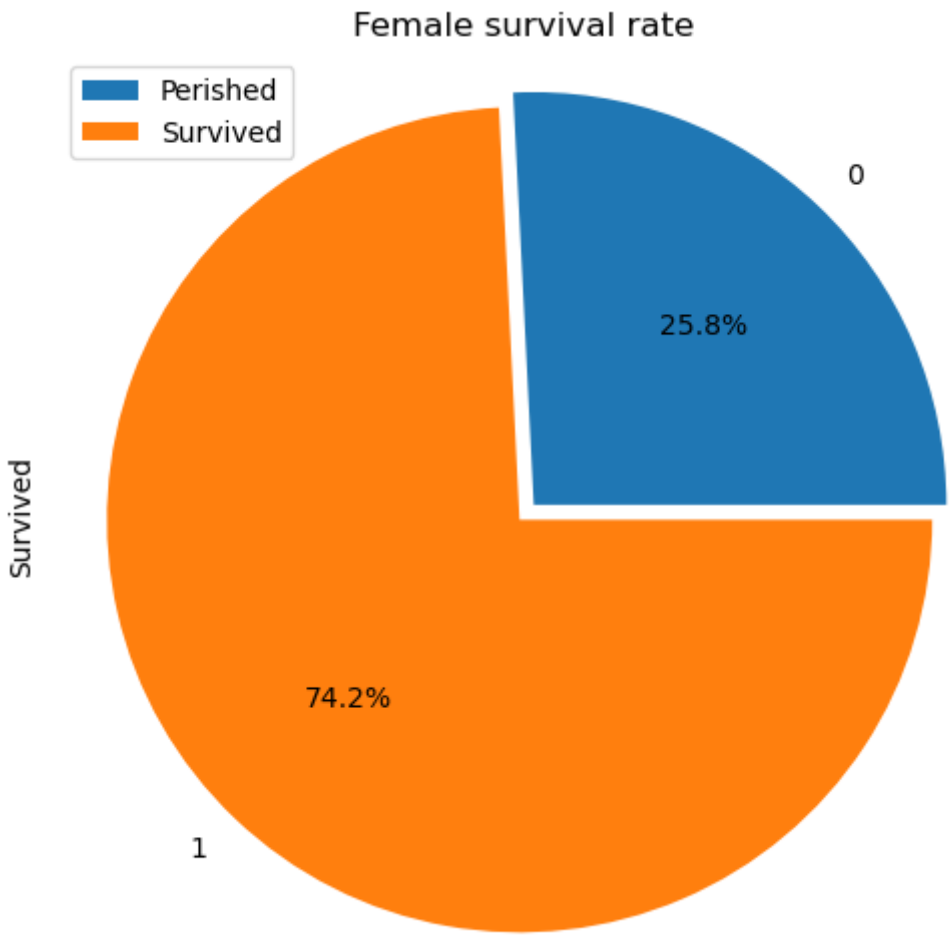
In [33]:
```
df.groupby(['Survived','Sex'])['Survived'].count()
```

Out[33]:
```
Survived  Sex
0         female     81
          male      468
1         female    233
          male      109
Name: Survived, dtype: int64
```

In [35]:
```
df[df['Sex'] =='male'].Survived.groupby(df.Survived).count().plot(kind='pie',figsi
plt.axis('equal')
plt.legend(["Perished","Survived"])
plt.title("Male survival rate")
plt.show()
```



In [39]:
```
df[df['Sex'] == 'female'].Survived.groupby(df.Survived).count().plot(kind='pie',au
plt.axis('equal')
plt.title("Female survival rate")
plt.legend(["Perished","Survived"])
plt.show()
```



The above 2 plots says the females were given more prority than male in the survival process . That too there is a significant difference between the two.

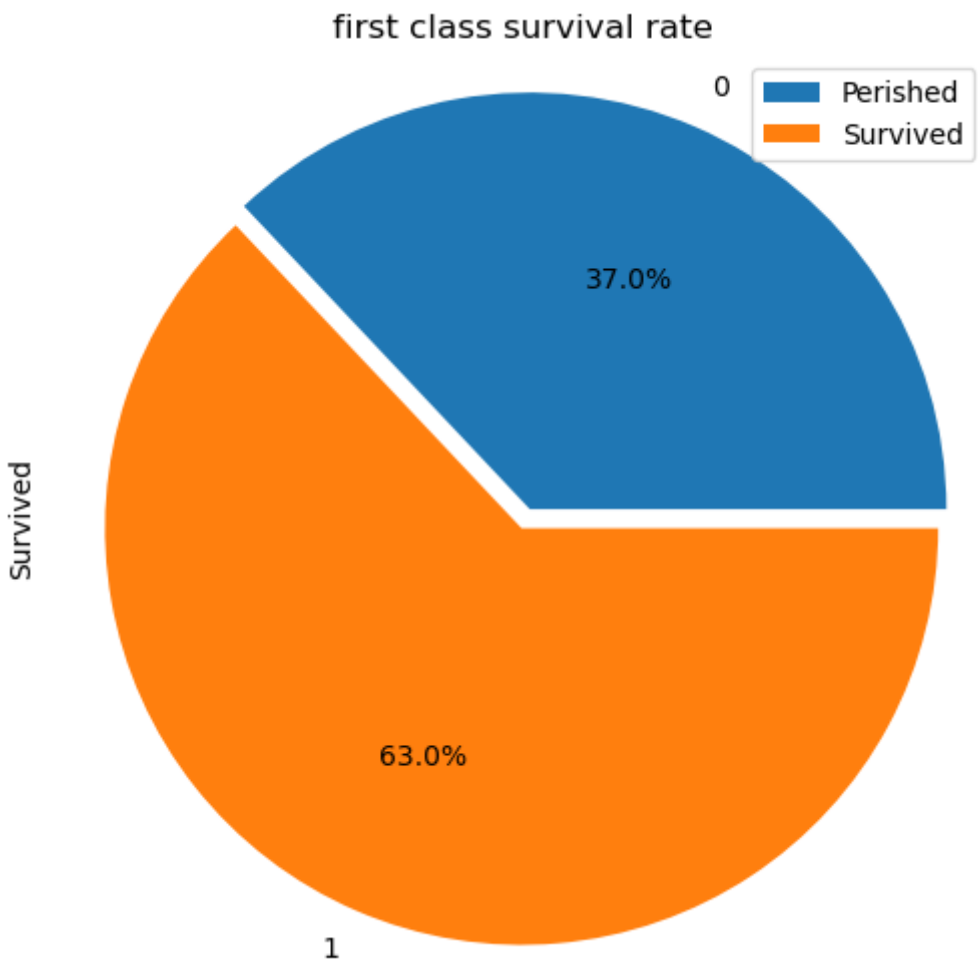
so now if we choose just Sex as the only feature and say all females survived and all men Perished , then we would end up with an accuracy of 78.67%

In [41]:
```
pd.crosstab(df.Pclass, df.Survived, margins=True)
```

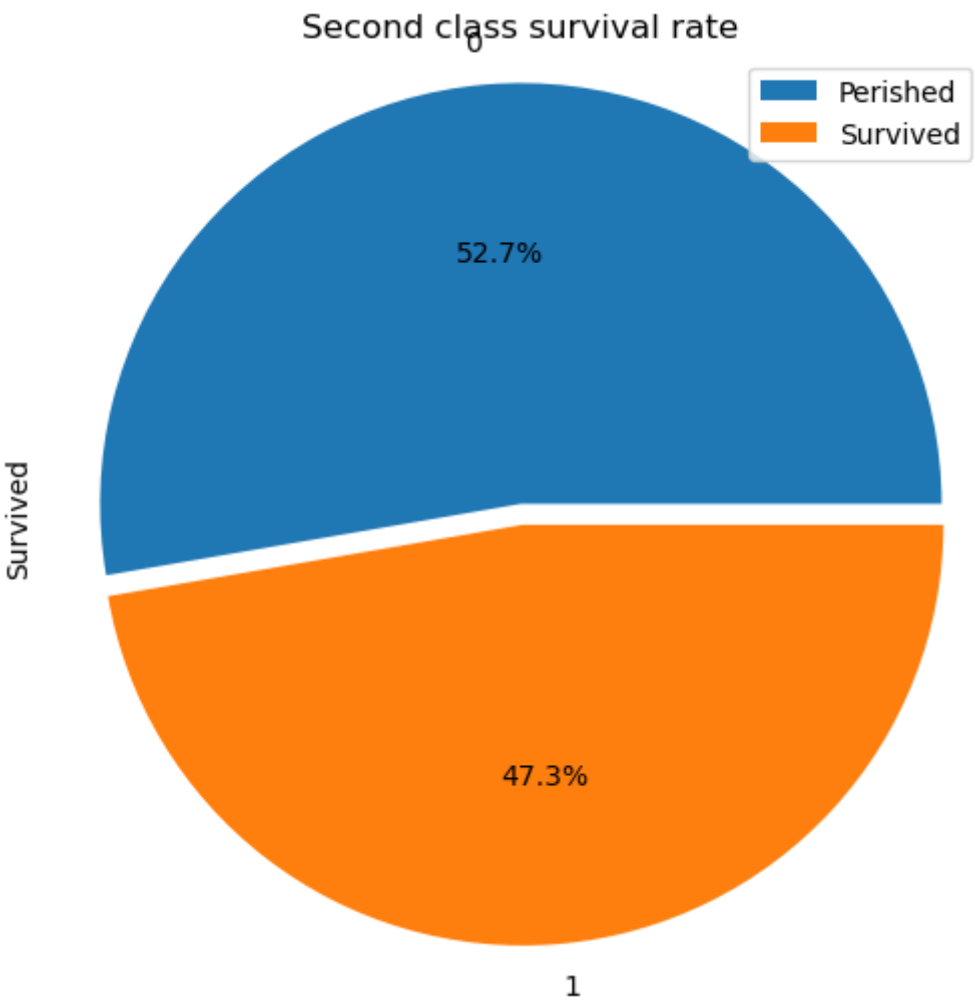
Out[41]:

Survived	0	1	All
Pclass			
1	80	136	216
2	97	87	184
3	372	119	491
All	549	342	891

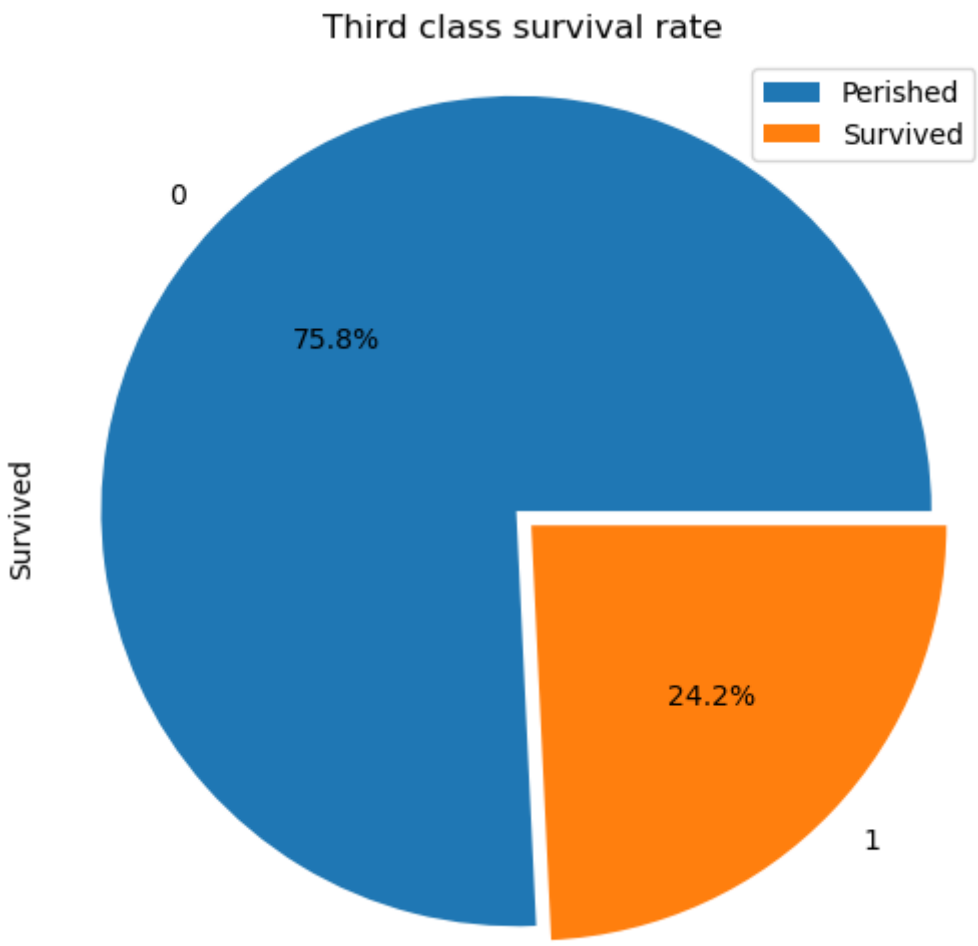
In [43]:
```
df[df['Pclass'] == 1].Survived.groupby(df.Survived).count().plot(kind='pie',figsiz
plt.axis('equal')
plt.legend(["Perished", "Survived"])
plt.title("first class survival rate")
plt.show()
```



In [44]:
```
df[df['Pclass'] ==2].Survived.groupby(df.Survived).count().plot(kind='pie',figsize
plt.axis('equal')
plt.legend(["Perished","Survived"])
plt.title("Second class survival rate")
plt.show()
```



In [45]:
```
df[df['Pclass'] ==3].Survived.groupby(df.Survived).count().plot(kind='pie',figsize
plt.axis('equal')
plt.legend(["Perished","Survived"])
plt.title("Third class survival rate")
plt.show()
```



In [46]:
```
pd.crosstab([df.Sex,df.Survived], df.Pclass, margins= True)
```

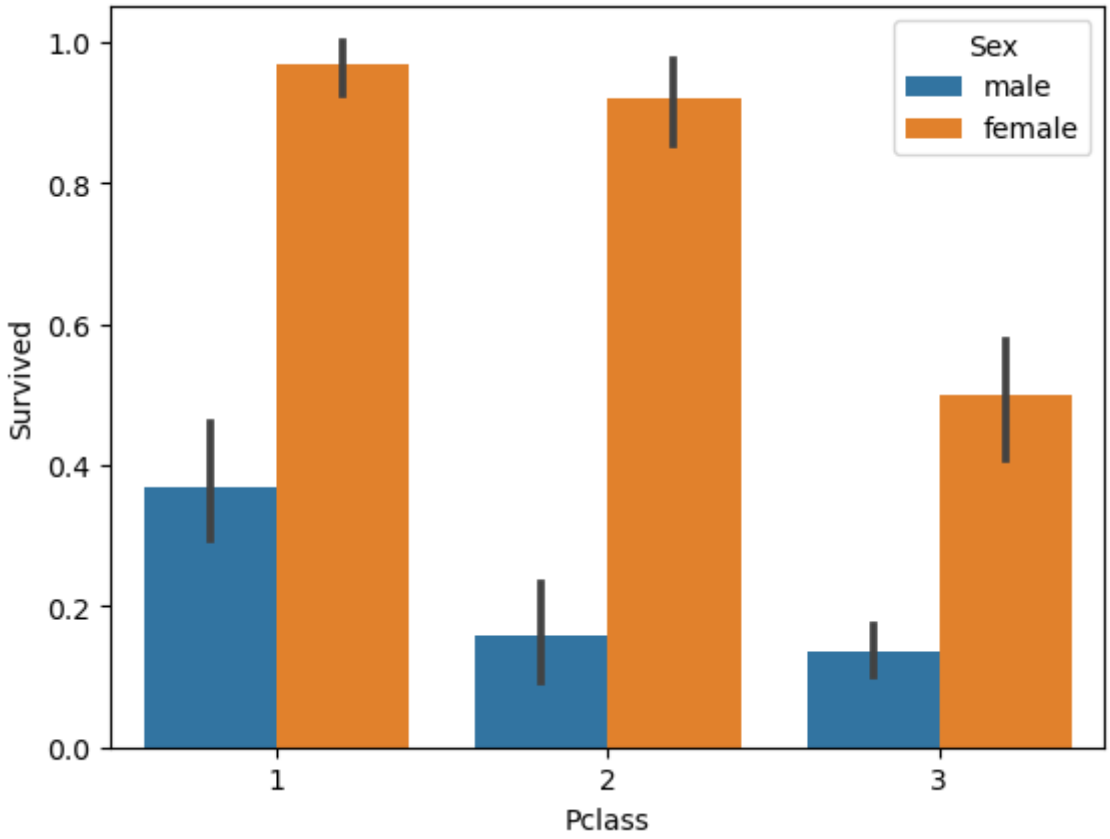
Out[46]:

Sex	Survived	Pclass 1	2	3	All
female	0	3	6	72	81
	1	91	70	72	233
male	0	77	91	300	468
	1	45	17	47	109
All		216	184	491	891

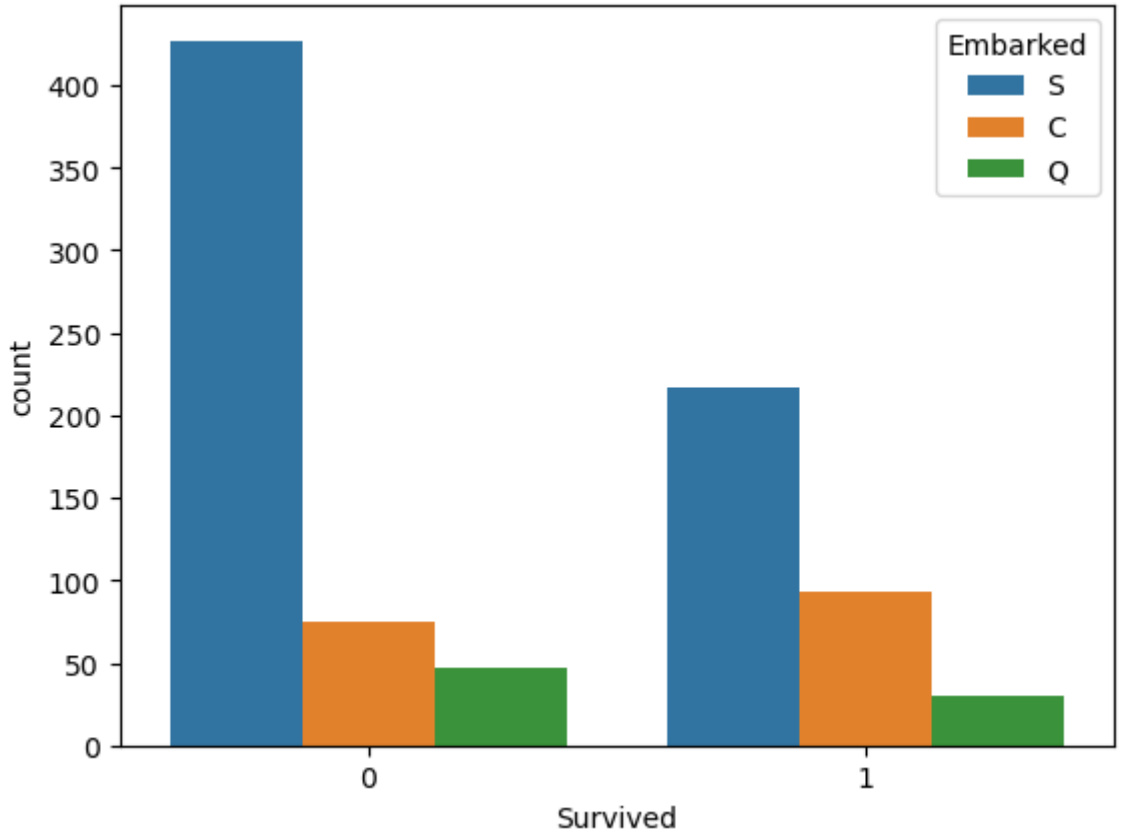
In [47]:
```
sns.barplot('Pclass','Survived',hue='Sex', data = df)
```

```
C:\Users\meanu\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[47]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Survived'>
```



In [48]:
```
sns.countplot(x='Survived', data = df, hue = 'Embarked');
```



In [53]:
```
pd.crosstab([df.Sex, df.Survived], [df.SibSp, df.Pclass], margins=True)
```

Out[53]:

	SibSp	0			1			2			3	4	5	8	All
	Pclass	1	2	3	1	2	3	1	2	3	1	2	3	3	3	3	
Sex	Survived																
female	0	1	3	33	2	3	21	0	0	3	0	0	7	4	1	3	81
	1	48	41	48	38	25	17	3	3	4	2	1	1	2	0	0	233
male	0	59	67	235	16	20	35	1	4	7	1	0	4	11	4	4	468
	1	29	9	35	15	7	10	1	1	1	0	0	0	1	0	0	109
All		137	120	351	71	55	83	5	8	15	3	1	12	18	5	7	891

The above crosstab indicates 2 things:

1. Most of the passengers didn't had sibling onboard and the majority had atmost 1 sibling onboard.
2. Not much of priority was given to the passengers who had sibelings onboard in the resue operation.

In [54]:
```
pd.crosstab([df.Sex, df.Survived], [df.Parch , df.Pclass], margins = True)
```

Out[54]:

	Parch	0			1			2			3		4	5	6	All
	Pclass	1	2	3	1	2	3	1	2	3	2	3	1	3	3	3	
Sex	Survived																
female	0	1	5	35	0	1	13	2	0	17	0	1	0	2	3	1	81
	1	63	40	50	17	17	12	11	11	8	2	1	0	0	1	0	233
male	0	63	81	260	10	7	22	3	3	15	0	1	1	1	1	0	468
	1	36	8	36	4	7	8	5	2	3	0	0	0	0	0	0	109
All		163	134	381	31	32	55	21	16	43	2	3	1	3	5	1	891

The above crosstab indicate 2 things:

1. The age was not a priority in the rescue operation similat to the sibellings and parents column as correlation with the target varible is very low.

2 There should have been a higher correlation between the fare and Pclass.

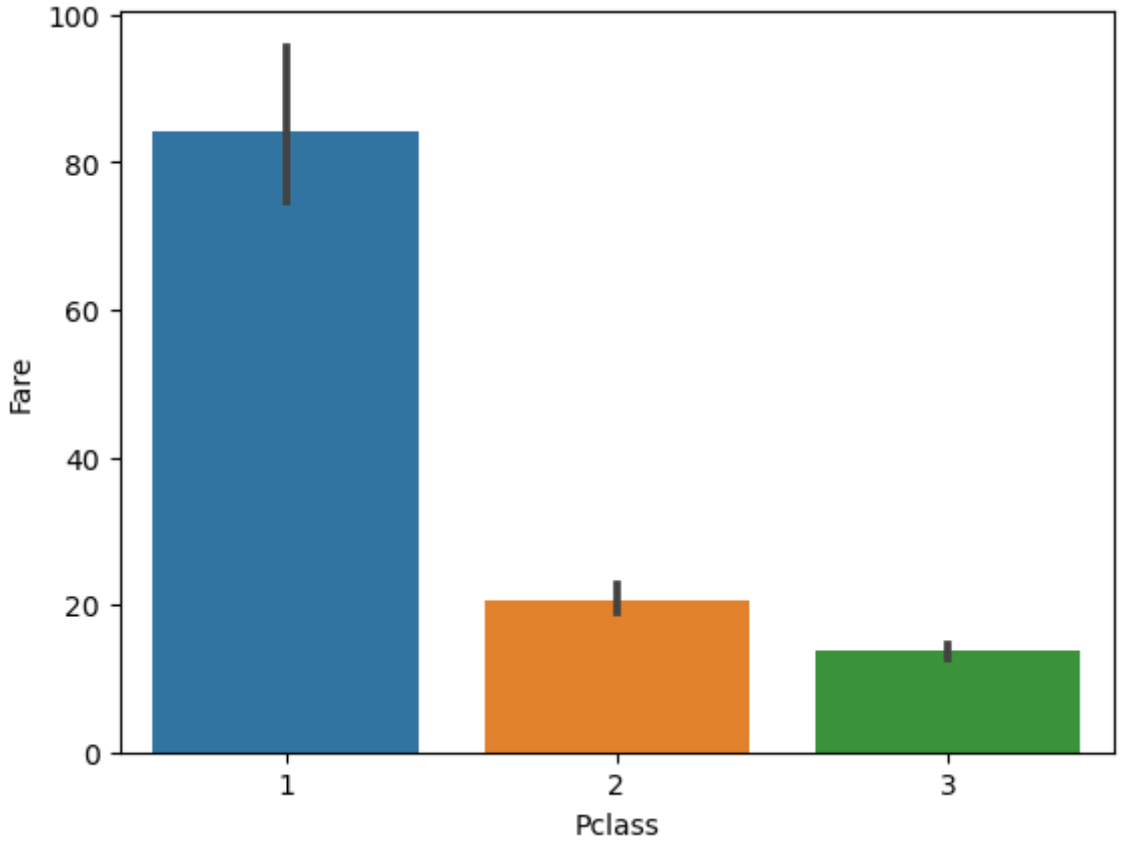
In [55]:
```
df.head(10)
```

Out[55]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S
5	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	NaN	Q
6	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	E46	S
7	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	NaN	S
8	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	NaN	S
9	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	NaN	C

In [56]:
```
sns.barplot(y= "Fare", x= "Pclass",data= df)
```

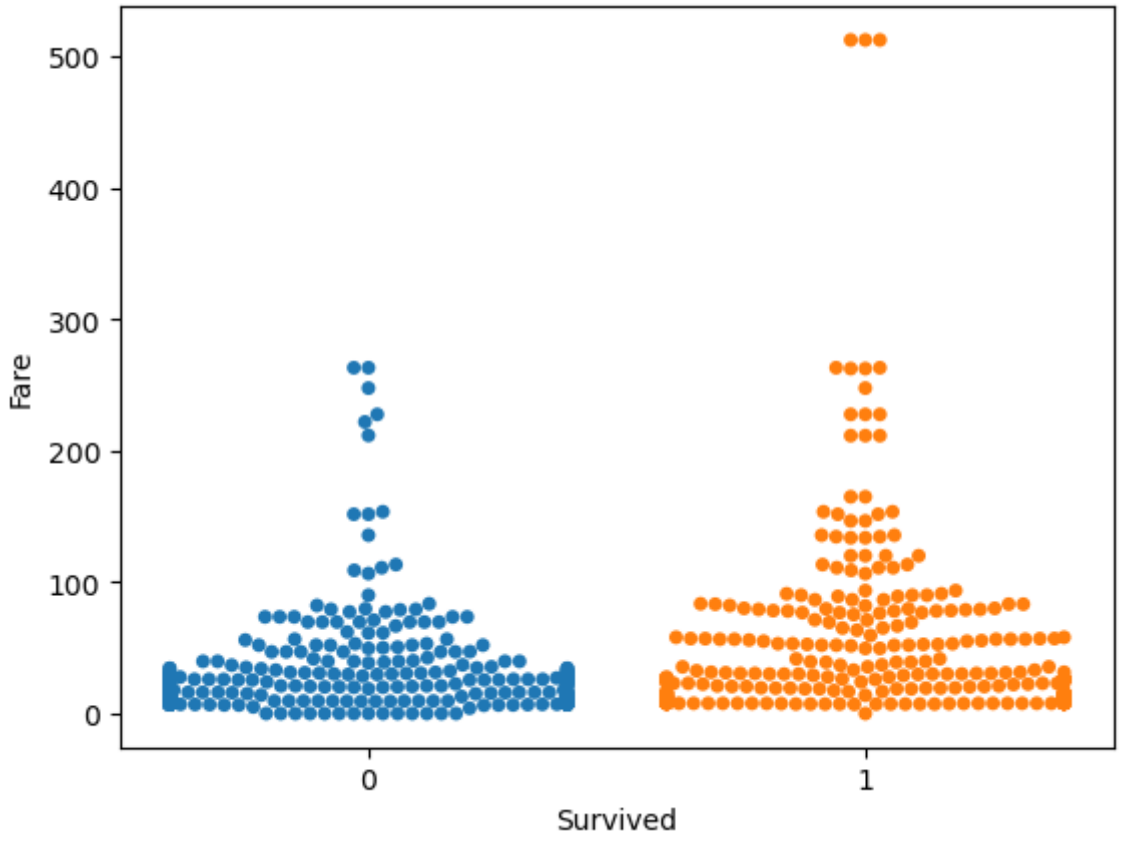
Out[56]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Fare'>
```



In [57]:
```
sns.swarmplot(x='Survived', y = 'Fare', data = df)
```

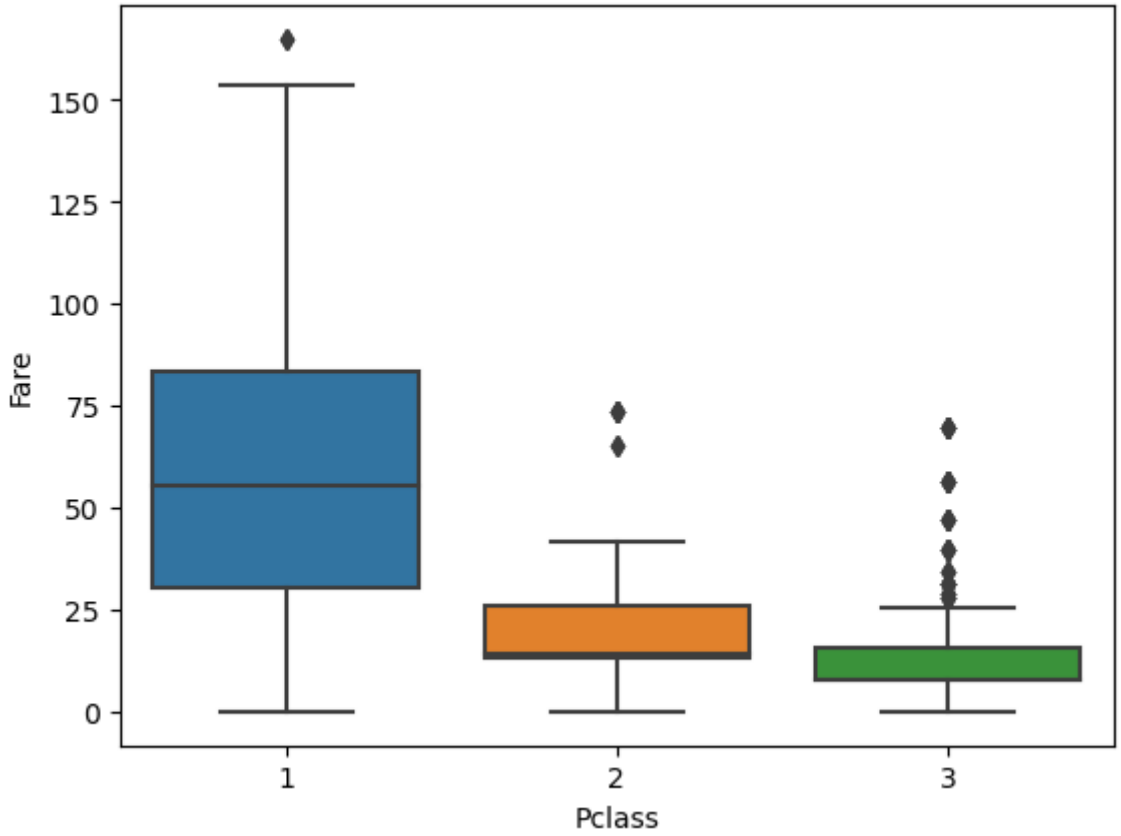
```
C:\Users\meanu\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 68.5% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\meanu\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarnin
g: 41.2% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[57]:
```
<AxesSubplot:xlabel='Survived', ylabel='Fare'>
```



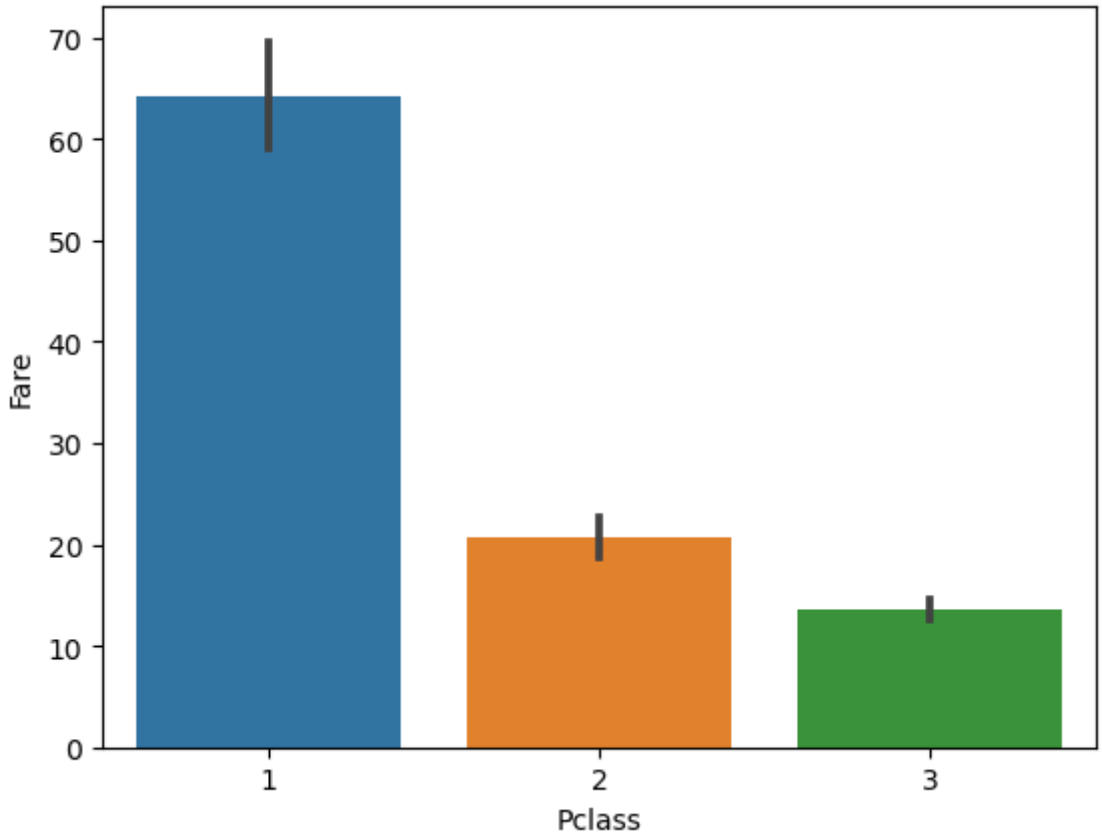
In [58]:
```
sns.boxplot(y = "Fare", x= "Pclass", data=df[df["Fare"]<200])
```

Out[58]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Fare'>
```



In [59]:
```
sns.barplot(y="Fare",x= "Pclass", data = df[df["Fare"]<200])
```

Out[59]:
```
<AxesSubplot:xlabel='Pclass', ylabel='Fare'>
```



In [60]:
```
sns.pairplot(df.drop("Name",axis = 1).dropna(),hue = "Survived")
```

Out[60]:
```
<seaborn.axisgrid.PairGrid at 0x264ecf95100>
```

Not much information Could be extrcted from the correlation table.

Now lets see how we can handle the missing values of Age.

1. By filling with mean value

i.e df.fillna(value = df.mean())

2. By filling mean value of corresponding Survived category.

In [61]:
```
df.groupby('Survived').describe()['Age']
```

Out[61]:

Survived	count	mean	std	min	25%	50%	75%	max
0	424.0	30.626179	14.172110	1.00	21.0	28.0	39.0	74.0
1	290.0	28.343690	14.950952	0.42	19.0	28.0	36.0	80.0

Both values actually look very similar.

Now let's try something special . if we see the name column , there is data which correspond to the age of the person. yes: Mr.Mrs .Master.Miss. So lets use that in filling the NA values for age.

In [62]:
```
df.head(5)
```

Out[62]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S

In [81]:
```
def extract(x):
    temp = x.split(" ")
    if "Mr." in temp:
        return "Mr"
    elif "Mrs." in temp:
        return "Mrs"
    elif "Miss." in temp:
        return "Miss"
    elif "Master." in temp:
        return "Master"
    elif "Dr." in temp:
        return "Dr"
    else:
        return None
```

In [88]:
```
df["Category"] = train["Name"].apply(extract)
```

In [89]:
```
df.head()
```

Out[89]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked	Category
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	NaN	S	Mr
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C85	C	Mrs
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	NaN	S	Miss
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	C123	S	Mrs
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	NaN	S	Mr

In [90]:
```
df["Category"].unique()
```

Out[90]:
```
array(['Mr', 'Mrs', 'Miss', 'Master', None, 'Dr'], dtype=object)
```

In [91]:
```
print("Mr." , np.mean(df[df["Category"] == "Mr"]["Age"]))
print("Mrs." , np.mean(df[df["Category"] == "Mrs"]["Age"]))
print("Miss." , np.mean(df[df["Category"] == "Miss"]["Age"]))
print("Master." , np.mean(df[df["Category"] == "Master"]["Age"]))
print("Dr." , np.mean(df[df["Category"] == "Dr"]["Age"]))
```

```
Mr. 32.368090452261306
Mrs. 35.898148148148145
Miss. 21.773972602739725
Master. 4.574166666666667
Dr. 42.0
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: