泰坦尼克数据集项目实战

问题描述：探究哪些因素使乘客生还率更高

一、导入及描述性汇总统计：

In [1]:
```
#导入需要的库
import pandas as pd
import numpy as np
import seaborn as sns


#配置jupyter notebook
%matplotlib inline
%config InlineBackend.figure_format = "retina"
pd.set_option('display.max_columns', 20)
pd.set_option('display.max_rows', 25)

#导入数据
titanic_df  = pd.read_csv("titanic-data.csv")
```

In [2]:
```
#信息
titanic_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Sex            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

数据基本信息

观察到数据主要有12列字段，891行，Age和Cabin字段有较多缺失值，Embanked字段有2个缺失值
数据类型为：float64(2), int64(5), object(5)
PassengerId: 乘客编号
Survived: Survived (1) or died (0)是否存活
Pclass: 船舱
Name: 姓名
Sex: 性别
Age: 年龄
SibSp: 兄弟/姐妹/配偶的数量
Parch: 父母/子女的数量
Ticket: 票号
Fare: 票价
Cabin: 座号
Embarked: 登船港口

In [3]:
```
#查看前10行
titanic_df.head(10)
```

Out[3]:

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

In [4]:
```
#描述
titanic_df.describe()
```

Out[4]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [5]:
```
#皮尔逊相关系数，查看线性相关关系
titanic_df.corr()
```

Out[5]:

	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

乘客生还率与 座舱 和 票价 线性相关程度较高

除此之外，船舱 与 票价 相关程度最高、船舱 与 年龄 相关程度也值得关注

二 、填充缺失值（年龄、座号、港口）

In [6]:
```
#年龄缺失计数
print 'The proportion of Age missing value is {}%'.format((titanic_df['Age'].isnull().sum())/891.0*100)
```

```
The proportion of Age missing value is 19.8653198653%
```

In [7]:
```
#缺失值较多，使用范围随机数填充年龄缺失值
age_avg = titanic_df['Age'].mean()
age_std = titanic_df['Age'].std()
titanic_df.loc[pd.isnull(titanic_df["Age"]),"Age"] = np.random.randint(age_avg-age_std,age_avg+age_std,titanic_df["Age"].isnull().
sum())
titanic_df['Age'].describe()
```

Out[7]:
```
count    891.000000
mean      29.428923
std       13.492765
min        0.420000
25%       21.000000
50%       28.000000
75%       37.000000
max       80.000000
Name: Age, dtype: float64
```

In [8]:
```
#座号缺失值计数
print 'The proportion of Cabin missing value is {}%'.format((titanic_df['Cabin'].isnull().sum())/891.0*100)
```

```
The proportion of Cabin missing value is 77.1043771044%
```

In [9]:
```
#缺失值较多，使用空值填充
titanic_df['Cabin'].fillna('unknow')
```

Out[9]:
```
0      unknow
1         C85
2      unknow
3        C123
4      unknow
5      unknow
6         E46
7      unknow
8      unknow
9      unknow
10         G6
11       C103
        ...
879       C50
880    unknow
881    unknow
882    unknow
883    unknow
884    unknow
885    unknow
886    unknow
887       B42
888    unknow
889      C148
890    unknow
Name: Cabin, Length: 891, dtype: object
```

In [10]:
```
#港口缺失值计数
print 'The proportion of Embarked missing value is {}%'.format((titanic_df['Embarked'].isnull().sum())/891.0*100)
titanic_df["Embarked"].value_counts()
```

```
The proportion of Embarked missing value is 0.224466891134%
```

Out[10]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [11]:
```
#缺失值较少，分类型数据，使用众数填充
titanic_df["Embarked"].fillna("S")
```

Out[11]:
```
0      S
1      C
2      S
3      S
4      S
5      Q
6      S
7      S
8      S
9      C
10     S
11     S
      ..
879    C
880    S
881    S
882    S
883    S
884    S
885    Q
886    S
887    S
888    S
889    C
890    Q
Name: Embarked, Length: 891, dtype: object
```

In [12]:
```
#检查缺失值
titanic_df.isnull().count()
```

Out[12]:
```
PassengerId    891
Survived       891
Pclass         891
Name           891
Sex            891
Age            891
SibSp          891
Parch          891
Ticket         891
Fare           891
Cabin          891
Embarked       891
dtype: int64
```

三、探索性数据分析

之前通过查看皮尔逊相关系数矩阵，船舱 和 票价 与生还率线性相关性高

皮尔逊相关系数矩阵并不能反映非数值型数据的相关关系，因此还要查看 性别，港口

In [13]:
```
#生还者总数
titanic_df['Survived'].sum()
```

Out[13]: 342L

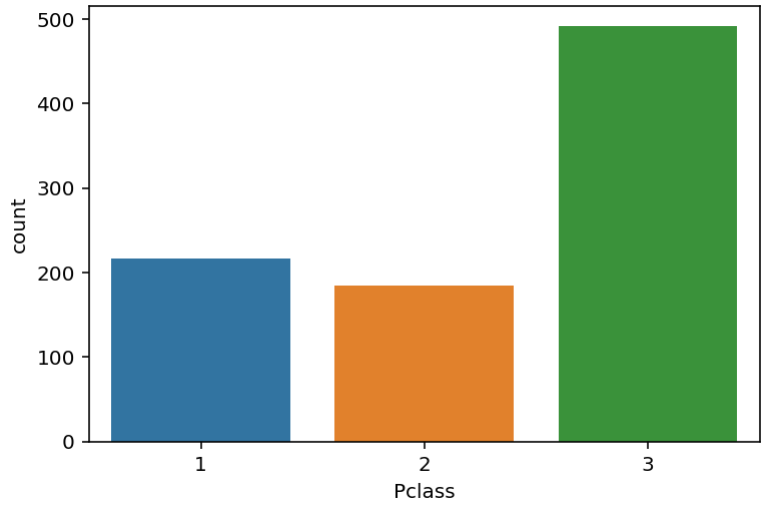
In [14]:
```
#平均生还率
titanic_df['Survived'].mean()
```

Out[14]: 0.3838383838383838

结论1：生还者总数为342人，平均生还率约为38%

In [15]:
```
#各等级船舱乘客总数量统计图:
sns.countplot(titanic_df['Pclass'])
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0xec1dbe0>



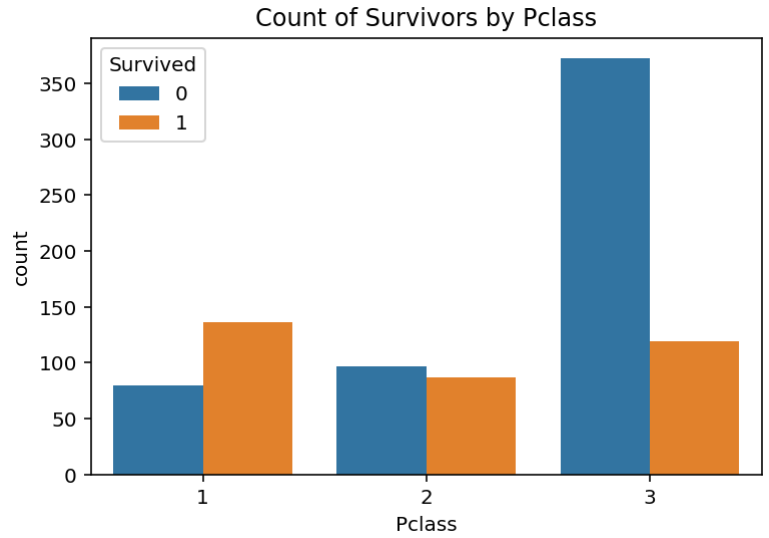
In [16]:
```
#各等级船舱乘客总数量统计:
titanic_df[['Pclass', 'PassengerId']].groupby(['Pclass']).count()
```

Out[16]:

	PassengerId
Pclass	
1	216
2	184
3	491

In [17]:
```
#幸存者数量柱状图函数
import matplotlib.pyplot as plt
def plotCount(feature):
    sns.countplot(data=titanic_df, x=feature, hue='Survived')
    plt.title('Count of Survivors by {}'.format(feature))
```

In [18]:
```
#各等级船舱死亡者与生还者数量统计图:
plotCount('Pclass')
```



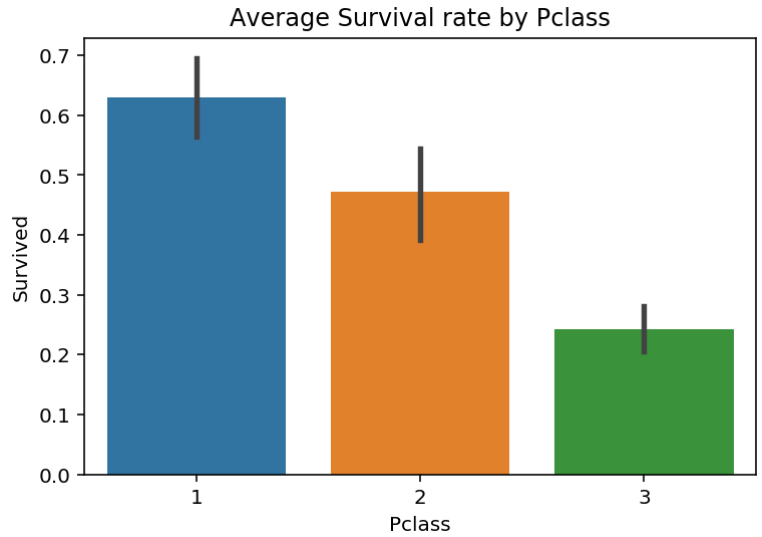
In [19]:
```
#各等级船舱生还数量统计:
titanic_df[['Pclass', 'Survived']].groupby(['Pclass']).sum()
```

Out[19]:

	Survived
Pclass	
1	136
2	87
3	119

In [20]:
```
#平均生还率柱状图函数
def plotBar(feature):
    sns.barplot(data=titanic_df, x=feature, y='Survived')
    plt.title('Average Survival rate by {}'.format(feature))
```

In [21]:
```
#船舱等级与生还率统计图
plotBar('Pclass')
```



In [22]:
```
#各船舱等级对应的平均生还率
titanic_df[['Pclass','Survived']].groupby(['Pclass']).mean()
```

Out[22]:

	Survived
Pclass	
1	0.629630
2	0.472826
3	0.242363

结论2:

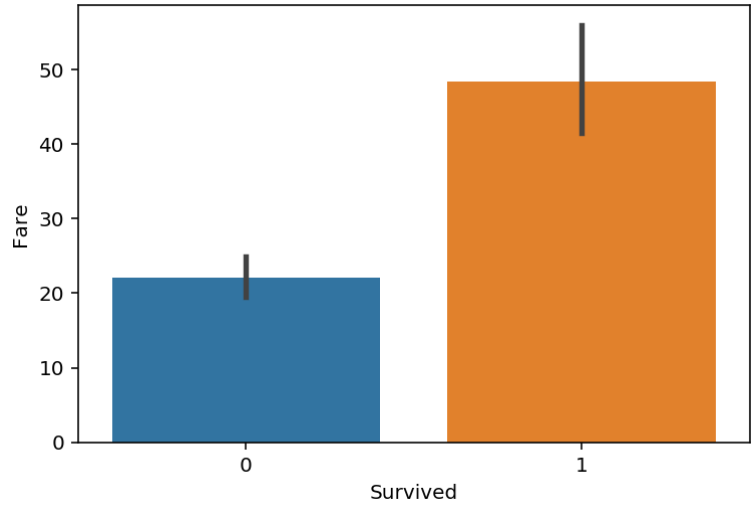
头等舱总人数为216人，二等舱为184人，三等舱为491人。

头等舱生还人数为136人，二等舱为87人，三等舱为119人。

头等舱生还率为62.96%，二等舱为47.28%，三等舱为24.24%。

In [23]:
```
#是否生还与平均票价统计图
sns.barplot(data=titanic_df, x='Survived', y='Fare')
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0xeece828>



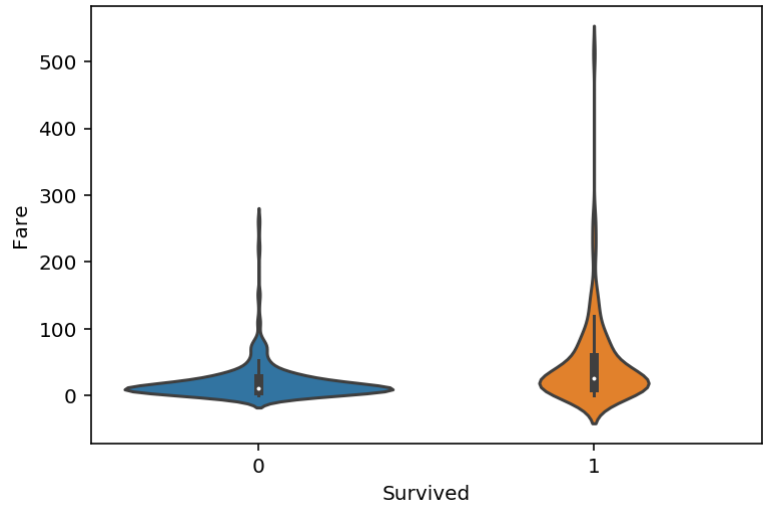
In [24]:
```
#是否生还与票价的统计
titanic_df[['Fare','Survived']].groupby(['Survived']).describe()
```

Out[24]:

	Fare							
	count	mean	std	min	25%	50%	75%	max
Survived								
0	549.0	22.117887	31.388207	0.0	7.8542	10.5	26.0	263.0000
1	342.0	48.395408	66.596998	0.0	12.4750	26.0	57.0	512.3292

In [25]:
```
#了解数据分布
sns.violinplot(data=titanic_df, x='Survived', y='Fare')
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0xf23e940>



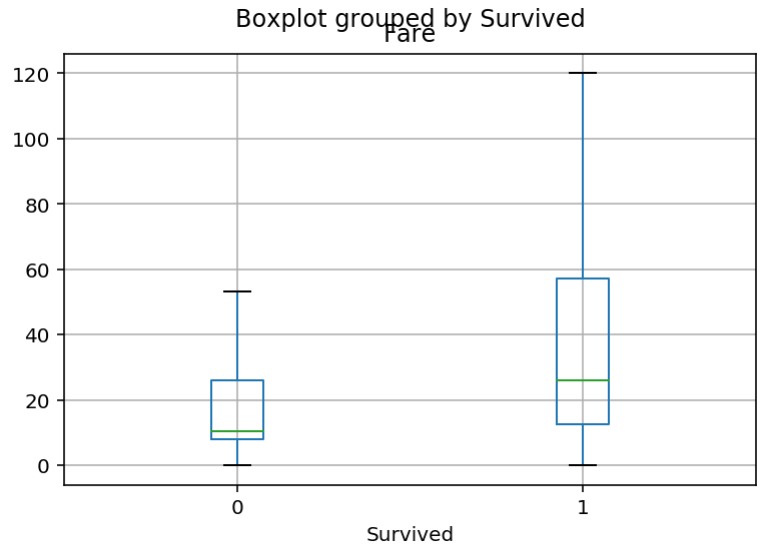
In [27]:
```
#查看前10名异常高的票价
titanic_df.Fare.sort_values(ascending=False).head(n=10)
```

Out[27]:
```
679    512.3292
258    512.3292
737    512.3292
341    263.0000
438    263.0000
88     263.0000
27     263.0000
311    262.3750
742    262.3750
299    247.5208
Name: Fare, dtype: float64
```

In [28]:
```
#主体票价与生还率箱型图
titanic_df.boxplot('Fare', by='Survived',showfliers=False)
```

```
C:\Users\Richa\Anaconda2\lib\site-packages\numpy\core\fromnumeric.py:57: FutureWarning: reshape is deprecated and will raise in a subsequent release. Please use .values.reshape(...) instead
  return getattr(obj, method)(*args, **kwds)
```

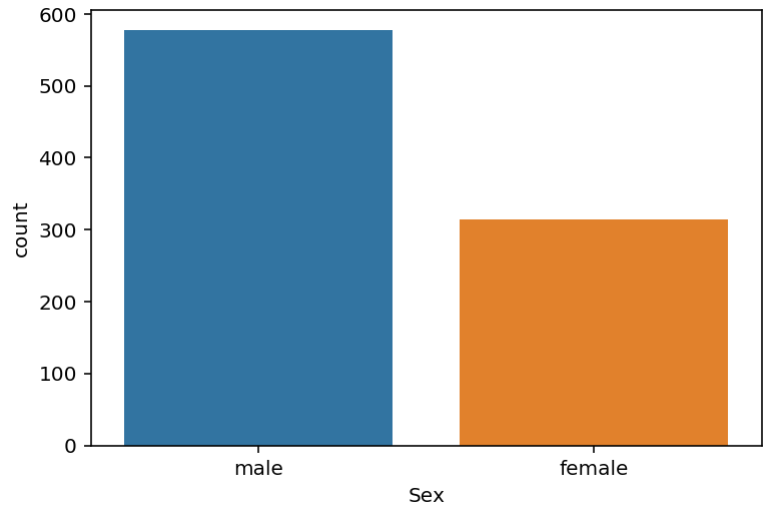
Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xf16e780>
```



结论3：生还者的平均票价更高

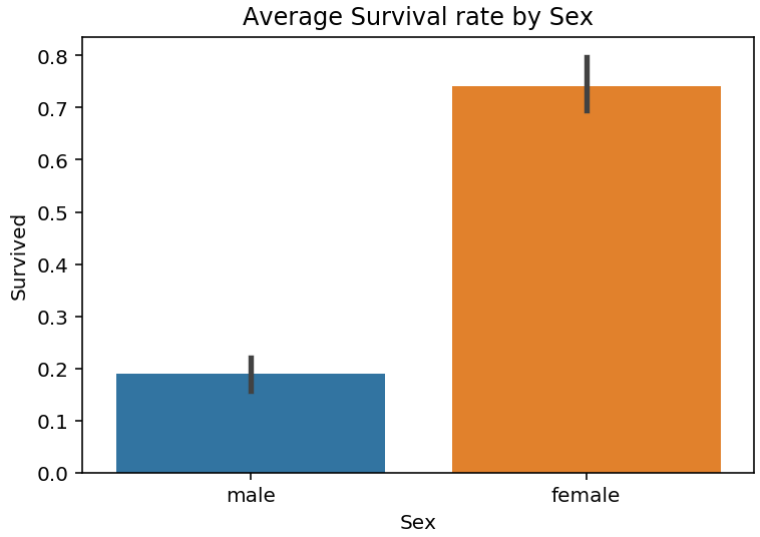
In [29]:
```
#性别分类柱状图
print sns.countplot(titanic_df['Sex'])
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



男性乘客多于女性

In [30]:
```
#生还率和性别的关系
plotBar('Sex')
```



In [31]:
```
titanic_df.groupby(['Sex']).count()
```

Out[31]:

	PassengerId	Survived	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
Sex											
female	314	314	314	314	314	314	314	314	314	97	312
male	577	577	577	577	577	577	577	577	577	107	577

In [32]:
```
#不同性别平均生还率
titanic_df[['Sex','Survived']].groupby(['Sex']).mean()
```

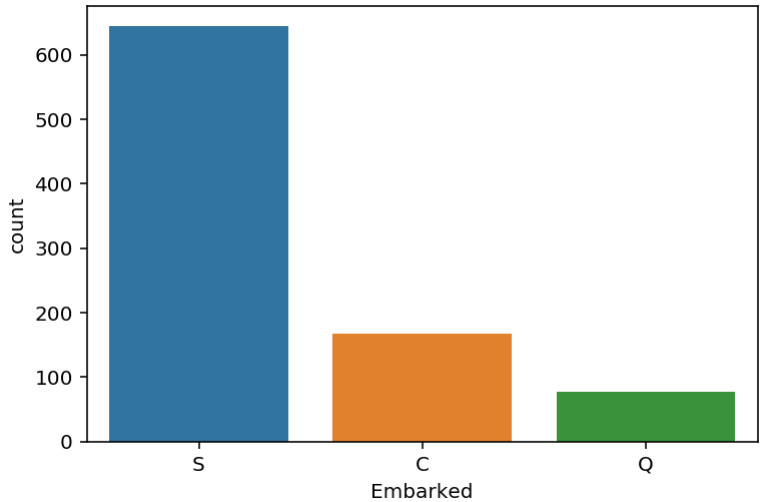
Out[32]:

	Survived
Sex	
female	0.742038
male	0.188908

结论4：乘客中男性共577人，女性314人，女性生还率为74.2%，远高于男性的18.89%。

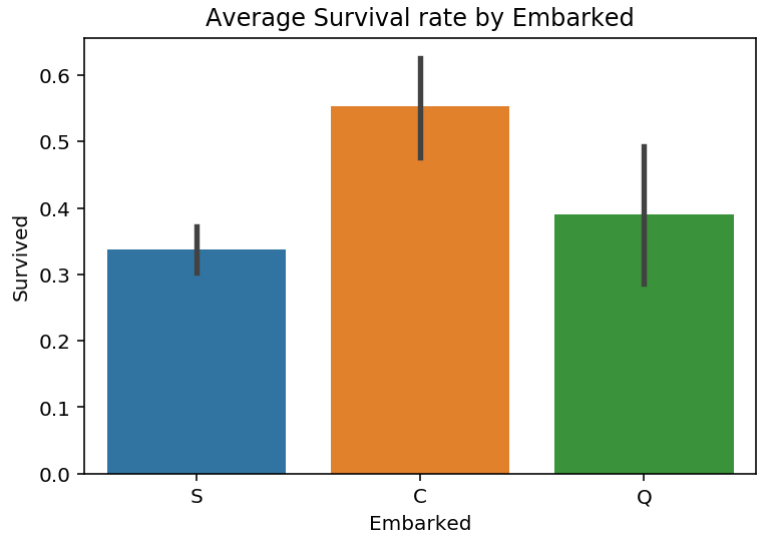
In [33]:
```
#港口分类柱状图
sns.countplot(titanic_df['Embarked'])
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x102b39e8>



S港口乘客远多于CQ港口

In [34]:
```
#生还率和港口的关系
plotBar('Embarked')
```



In [35]:
```
#各港口平均生还率
titanic_df[['Embarked','Survived']].groupby(['Embarked']).mean()
```

Out[35]:

	Survived
Embarked	
C	0.553571
Q	0.389610
S	0.336957

结论5：C港口乘客生还率为55.35%，Q为38.96%，S为33.70%

总结:

本次分析主要探寻泰坦尼克号上的生还率和各因素（客舱等级、性别、票价、上船港口等）的相关关系。

样本数量为 891，海难发生后，生还者还剩 342 人，生还率为 38%。

泰坦尼克号上有一\二\三等舱三种船舱类型，头等舱总人数为216人，二等舱为184人，三等舱为491人。

泰坦尼克号上有一\二\三等舱三种船舱类型，头等舱生还人数为136人，二等舱为87人，三等舱为119人。

泰坦尼克号上有一\二\三等舱三种船舱类型，头等舱生还率为62.96%，二等舱为47.28%，三等舱为24.24%。

泰坦尼克号乘客可以购买不同价格的船票，生还者的平均票价更高。

样本的891人中，男性共577人，女性314人，女性生还率为74.2%，远高于男性的18.89%。

样本的891人中，C港口乘客生还率为55.35%，Q为38.96%，S为33.70%。

最后需要说明的是，此次数据分析的数据集是从总体中抽样而来的，如果抽样无偏，样本是从总体随机选取，根据中心极限定理，分析结果具有代表性，如果不是随机选出，那么分析结果就不可靠了。

In [ ]:

In [ ]: