一、数据挖掘概述

1、什么是数据挖掘

数据挖掘一般是指从大量的数据中通过算法搜索隐藏于其中信息的过程。数据挖掘通常与计算机科学有关，并通过统计、在线分析处理、情报检索、机器学习、专家系统（依靠过去的经验法则）和模式识别等诸多方法来实现上述目标。

2、数据挖掘流程

（一）数据读取：

- 读取数据，并进行展示
- 统计数据各项指标
- 明确数据规模与要完成任务

（二）特征理解分析

- 单特征分析，逐个变量分析其对结果的影响
- 多变量统计分析，综合考虑多种情况影响
- 统计绘图得出结论

（三）数据清洗与预处理

- 对缺失值进行填充
- 特征标准化/归一化
- 筛选有价值的特征
- 分析特征之间的相关性

（四）建立模型

- 特征数据与标签准备
- 数据集切分
- 多种建模算法对比
- 集成策略等方案改进

3、工具的选择

Python是数据挖掘的默认语言，在Python提供了丰富的数据挖掘库：

- Numpy-科学计算库，主要用来做矩阵运算。在数据分析计算中，数据就是行（样本）和列（特征）组成的，那么数据本身就是一个矩阵，Numpy则是·矩阵运算的佼佼者。
- Pandas-数据分析处理库。 大家都说用python处理数据很容易，那么容易在哪呢？有了pandas很

复杂的操作我们也可以一行代码去解决。

- Matplotlib-可视化库。无论是分析还是建模，通过可视化库把结果和过程可视化的展示出来。
- Seaborn-可视化库。更简单的可视化库封装上Matplot基础之上。
- Scikit-Learn 机器学习库 。非常实用的机器学习算法库，这里面包含了基本你觉得你能用上所有机器学习算法。但还远不止如此，还提供了很多预处理和评估的模块等。

二、泰坦尼克获救预测项目

1、项目概述

1912年4月14日，星期天晚上，一个风平浪静的夜晚。泰坦尼克号撞上了冰山，“永不沉没的”泰坦尼克号面临沉船的命运。船长发话了『lady and kid first！』，我们通过数据分析的技术，看看这个悲伤的故事是不是女士获得了优先生存权呢。

2、泰坦尼克数据挖掘实现

1)数据读入

- **导入相关的库**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('fivethirtyeight')
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

- **加载数据**

```
data=pd.read_csv('train.csv') #读入数据
data.head() #显示开头的几行
```

显示数据如下：

	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

以上各列数据分别是：

PassengerId:乘客ID

Survived:是否获救

Pclass:乘客等级

Name:乘客姓名

Sex:性别

Age:年龄

SibSp:堂兄弟妹个数

Parch:父母与小孩个数

Ticket:船票信息

Fare:票价

Cabin:客舱

Embarked:登船港口

- **观察数据**

观察null值数量

```
data.isnull().sum() #统计各列null值数据的数量
```

```
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
```

其中age、Cabin数据缺失比较多。后面在分析时候，需要根据需要考虑是否需要做数据的填充。

观察数据整体情况（平均值、标准差、总数等）

```
data.describe()
```

	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

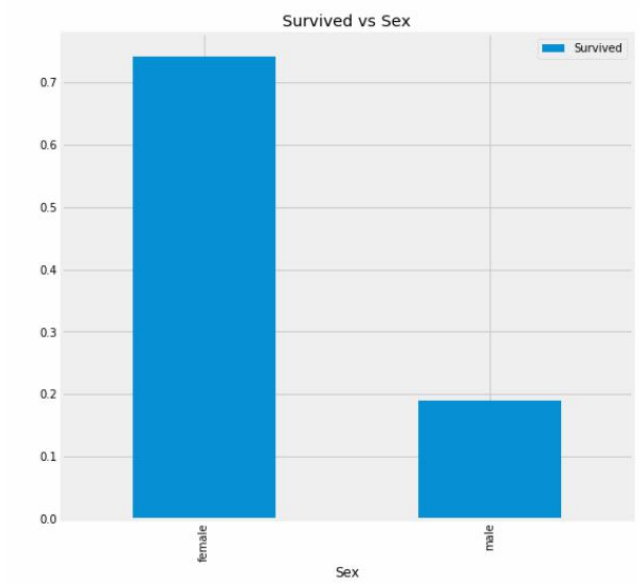
2)统计数据各项指标

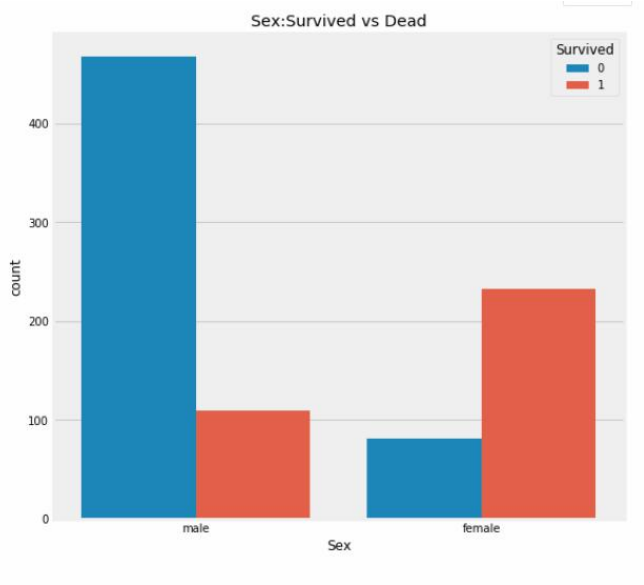
统计获救指标

画图分析获救指标

```
f,ax=plt.subplots(1,2,figsize=(18,8)) #设置1行2列，图的宽和高
data[['Sex','Survived']].groupby(['Sex']).mean().plot.bar(ax=ax[0]) #第1个图
ax[0].set_title('Survived vs Sex')
sns.countplot('Sex',hue='Survived',data=data,ax=ax[1])#第2个图
ax[1].set_title('Sex:Survived vs Dead')
plt.show()
```

画图效果如下：





这看起来很有趣。船上的男人比女人多得多。不过，挽救的女性人数几乎是男性的两倍。生存率为一个女人在船上是 75%左右，而男性在 18-19%左右。（让妇女和儿童先走，虽然电影忘得差不多了，这句话还记着，确实是这样的）

这看起来是一个非常重要的特性。一会我们建模会用上他的！

3)船舱等级跟获救情况的关系分析

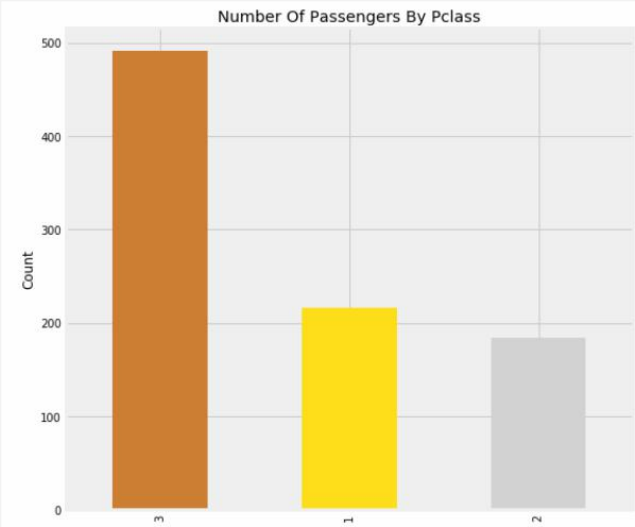
交叉表分析船舱等级和获救之间的关系：

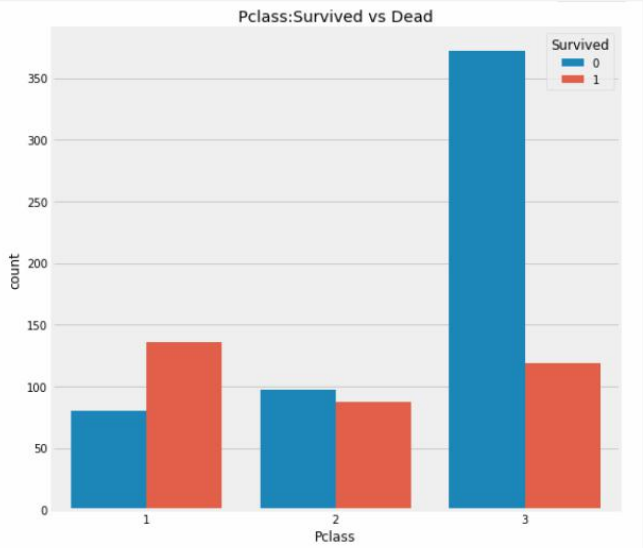
```
pd.crosstab(data.Pclass,data.Survived,margins=True).style.background_gradient(cmap='summer_r')
```

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

```
f,ax=plt.subplots(1,2,figsize=(18,8)) #创建1行两列的图，设置图的宽和高
#第一个图：使用柱状图统计各个等级的人数
data['Pclass'].value_counts().plot.bar(color=['#CD7F32','#FFDF00','#D3D3D3'],ax=ax[0])
ax[0].set_title('Number Of Passengers By Pclass')
ax[0].set_ylabel('Count')
sns.countplot('Pclass',hue='Survived',data=data,ax=ax[1]) #第二个图：统计各个船舱等级存活和死亡情况
ax[1].set_title('Pclass:Survived vs Dead')
plt.show()
```

统计结果：





人们说金钱不能买到一切。但我们可以清楚地看到，船舱等级为1的被给予很高的优先级而救援。尽管数量在pClass 3乘客高了很多，仍然存活数从他们是非常低的，大约25%。
对于pClass1来说存活是63%左右，而pclass2大约是48%。所以金钱和地位很重要。

4) 船舱等级和性别跟获救情况的关系分析

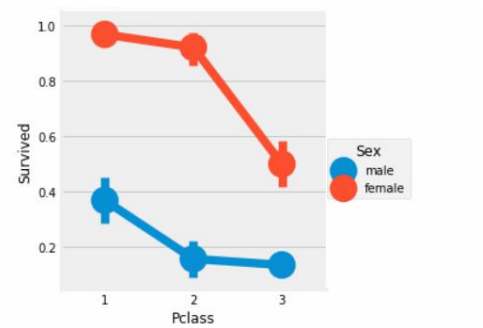
交叉表分析船舱等级和性别对结果的影响

```
pd.crosstab([data.Sex,data.Survived],data.Pclass,margins=True).style.background_gradient(cmap='summer_r')
```

	Pclass	1	2	3	All
Sex	Survived				
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

使用因子图分析

```
sns.factorplot('Pclass','Survived',hue='Sex',data=data)
plt.show()
```



我们可以很容易地推断，从pclass1女性生存是95-96%，如94人中只有3的女性从pclass1没获救。
显而易见的是，不论pClass，女性优先考虑。
看来Pclass也是一个重要的特征。

5) 年龄分析

获取最大、最小年龄和平均年龄

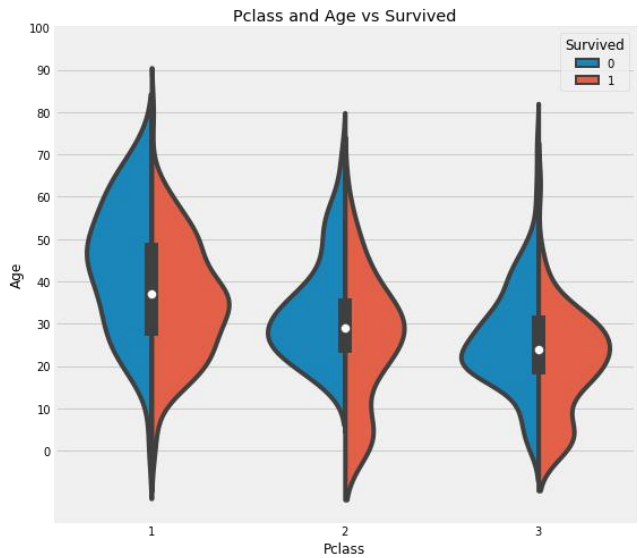
```
print('Oldest Passenger was of:',data['Age'].max(),'Years') #最大
print('Youngest Passenger was of:',data['Age'].min(),'Years') #最小
print('Average Age on the ship:',data['Age'].mean(),'Years') #平均
```

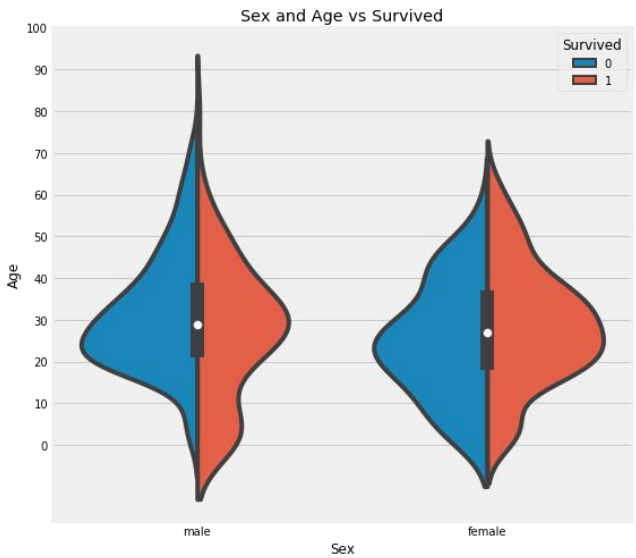
结果：

```
Oldest Passenger was of: 80.0 Years
Youngest Passenger was of: 0.42 Years
Average Age on the ship: 29.69911764705882 Years
```

```
f,ax=plt.subplots(1,2,figsize=(18,8))
#图一：船舱等级和年龄 跟获救的关系分析
sns.violinplot("Pclass","Age", hue="Survived", data=data,split=True,ax=ax[0])
ax[0].set_title('Pclass and Age vs Survived')
ax[0].set_yticks(range(0,110,10))
#图二：性别和年龄跟获救的关系分析
sns.violinplot("Sex","Age", hue="Survived", data=data,split=True,ax=ax[1])
ax[1].set_title('Sex and Age vs Survived')
ax[1].set_yticks(range(0,110,10))
plt.show()
```

通过小提琴图分析结果 ：





结论：
1）10 岁以下儿童的存活率随船舱人数的增加而增加。

2）年龄为 20-50 岁获救几率更高一些。

3）对男性来说，随着年龄的增长，存活率降低。

6) 年龄缺失值填充

数据缺失如何填充，主要有这样的一些方法：

- 平均值 ： 使用平均值进行填充
- 经验值：使用经验值填充，
- 回归模型预测：通过回归模型预测填充
- 剔除掉：直接删除空值的行

年龄特征有177个空值。为了替换这些缺失值，我们可以直接给它们分配数据集的平均年龄。但是有没有办法更加的接近实际值的年龄呢？
我们可以检查名字特征。根据这个特征，我们可以看到名字有像先生或夫人这样的称呼，这样我们就可以把先生和夫人的平均值分配给各自的组。这样的填充比直接填充全部的人的平均值会更加的接近实际值。

数据填充的处理步骤如下：

1、分析称呼和性别之间的关系：抽取所有名字的称呼，使用正则表达式。

```
data['Initial']=0 #增加列 Initial
for i in data:
    data['Initial']=data.Name.str.extract('([A-Za-z]+)\.')  #获取每行的称呼数据

pd.crosstab(data.Initial,data.Sex).T.style.background_gradient(cmap='summer_r') #分析称呼和性别的关系
```

结果：

Initial	Capt	Col	Countess	Don	Dr	Jonkheer	Lady	Major	Master	Miss	Mlle	Mme	Mr	Mrs	Ms	Rev	Sir
Sex																	
female	0	0	1	0	1	0	1	0	0	182	2	1	0	125	1	0	0
male	1	2	0	1	6	1	0	2	40	0	0	0	517	0	0	6	1

替换称呼为Mr.、Mrs、Ms.、Miss、Other

```
data['Initial'].replace(['Mlle','Mme','Ms','Dr','Major','Lady','Countess','Jonkheer','Col','Rev','Capt','Sir','Don'],['Miss','Miss','Miss','Mr','Mr','Mrs','Mrs','Other','Other','Other','Mr','Mr','Mr'],inplace=True)

#根据替换后的性别分析平均年龄
data.groupby('Initial')['Age'].mean()
```

结果：

```
Initial
Master     4.574167
Miss      21.860000
Mr        32.739609
Mrs       35.981818
Other     45.888889
Name: Age, dtype: float64
```

使用均值填充缺失了性别的数据：

```
## 使用每组的均值来进行填充
data.loc[(data.Age.isnull())&(data.Initial=='Mr'),'Age']=33
data.loc[(data.Age.isnull())&(data.Initial=='Mrs'),'Age']=36
data.loc[(data.Age.isnull())&(data.Initial=='Master'),'Age']=5
data.loc[(data.Age.isnull())&(data.Initial=='Miss'),'Age']=22
data.loc[(data.Age.isnull())&(data.Initial=='Other'),'Age']=46

#观察填充结果
data.Age.isnull().any() #看看填充完了咋样
```

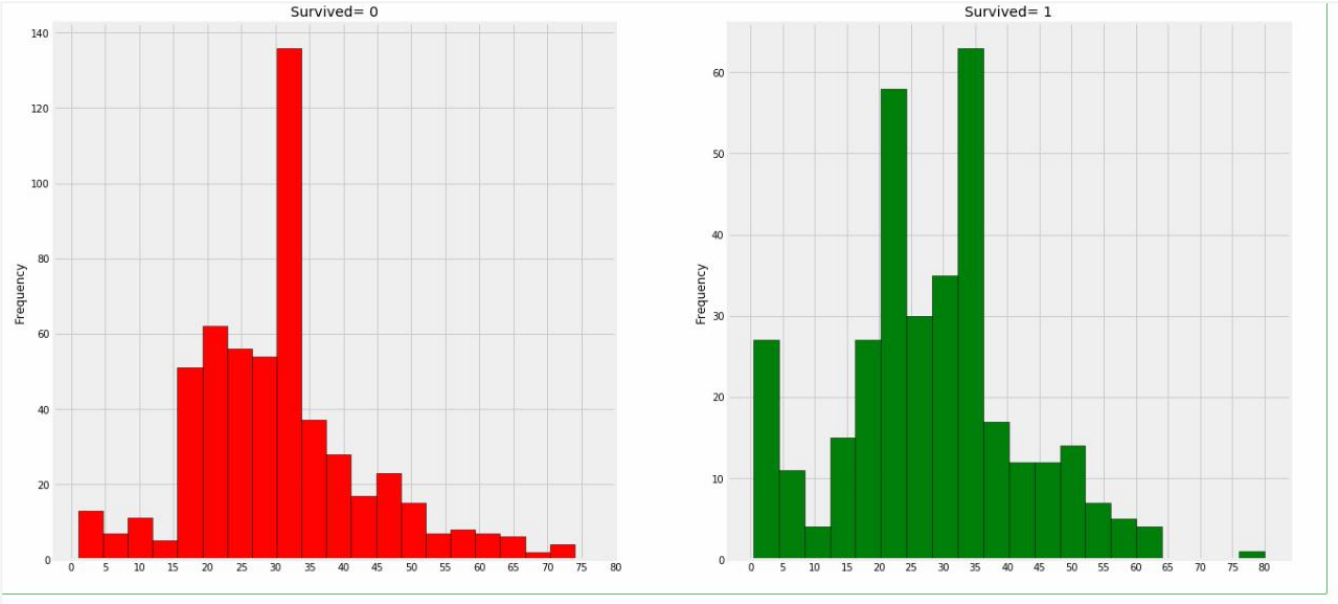
```
False
```

7) 填充后年龄和获救分析

```
f,ax=plt.subplots(1,2,figsize=(20,10))
#图1：死亡和年龄关系分析
data[data['Survived']==0].Age.plot.hist(ax=ax[0],bins=20,edgecolor='black',color='red')
ax[0].set_title('Survived= 0')
x1=list(range(0,85,5))
ax[0].set_xticks(x1)

#图2：存活和年龄关系分析
data[data['Survived']==1].Age.plot.hist(ax=ax[1],color='green',bins=20,edgecolor='black')
ax[1].set_title('Survived= 1')
x2=list(range(0,85,5))
ax[1].set_xticks(x2)
plt.show()
```

分析结果：



结论:

1）幼儿（年龄在 5 岁以下）获救的是比多的（妇女和儿童优先政策）。

2）最老的乘客得救了（80 年）。

3）死亡人数最高的是 30-40 岁年龄组。

8) 登船地点分析（Embarked）

交叉分析

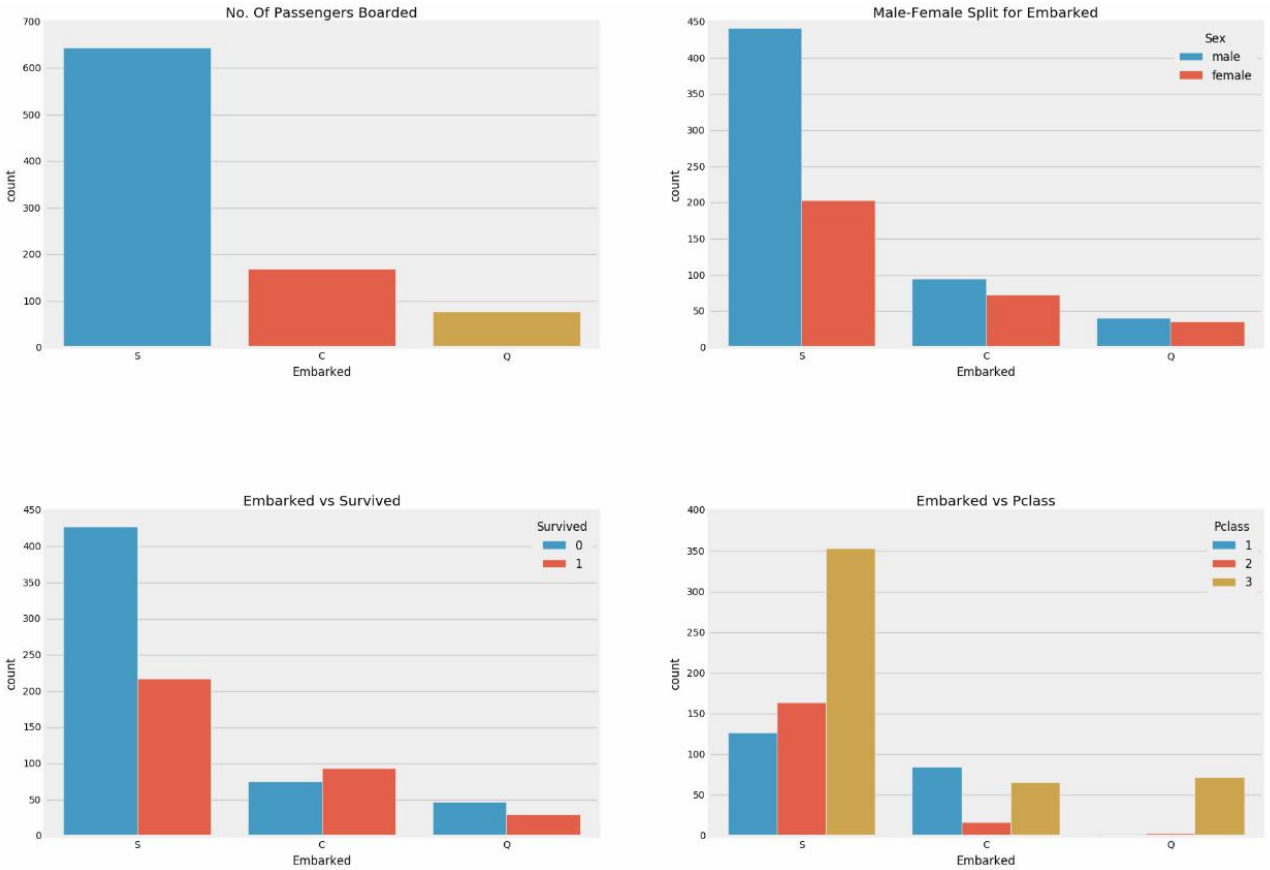
```
pd.crosstab([data.Embarked,data.Pclass],[data.Sex,data.Survived],margins=True).style.backgro
und_gradient(cmap='summer_r')
```

	Sex	female		male		All
	Survived	0	1	0	1	
Embarked	Pclass					
C	1	1	42	25	17	85
	2	0	7	8	2	17
	3	8	15	33	10	66
Q	1	0	1	1	0	2
	2	0	2	1	0	3
	3	9	24	36	3	72
S	1	2	46	51	28	127
	2	6	61	82	15	164
	3	55	33	231	34	353
All		81	231	468	109	889

```
sns.factorplot('Embarked','Survived',data=data)
fig=plt.gcf()
fig.set_size_inches(5,3)
plt.show()
```

C港生存的可能性最高在0.55左右，而S的生存率最低。

```
f,ax=plt.subplots(2,2,figsize=(20,15))
sns.countplot('Embarked',data=data,ax=ax[0,0])
ax[0,0].set_title('No. Of Passengers Boarded')
sns.countplot('Embarked',hue='Sex',data=data,ax=ax[0,1])
ax[0,1].set_title('Male-Female Split for Embarked')
sns.countplot('Embarked',hue='Survived',data=data,ax=ax[1,0])
ax[1,0].set_title('Embarked vs Survived')
sns.countplot('Embarked',hue='Pclass',data=data,ax=ax[1,1])
ax[1,1].set_title('Embarked vs Pclass')
plt.subplots_adjust(wspace=0.2,hspace=0.5)
plt.show()
```



分析：

1）大部分人的船舱等级是3。

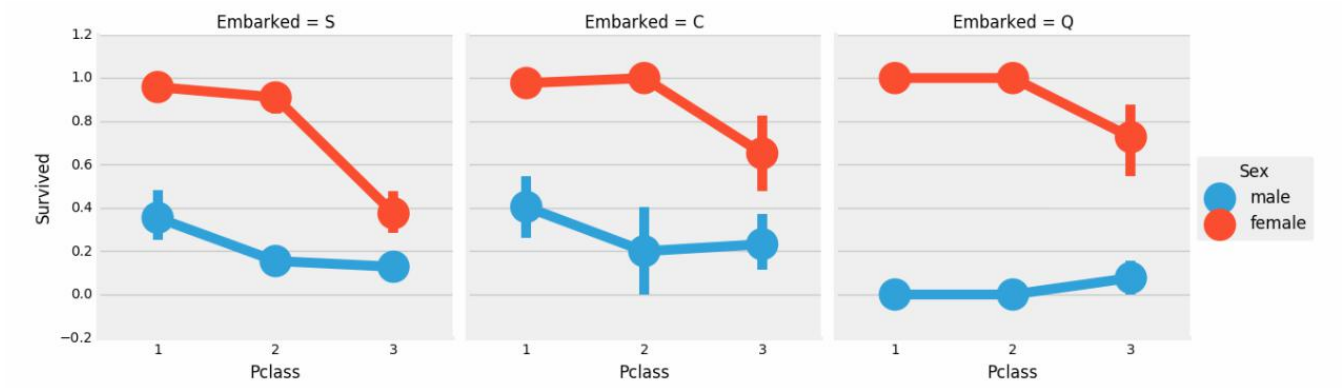
2）C的乘客看起来很幸运，他们中的一部分幸存下来。

3）S港口的富人蛮多的。仍然生存的机会很低。

4）港口Q几乎有95%的乘客都是穷人。

各个港口的不同等级船舱等级获救率分析

```
sns.factorplot('Pclass','Survived',hue='Sex',col='Embarked',data=data)
plt.show()
```



结果分析：

1）存活的几率几乎为 1 在 pclass1 和 pclass2 中的女人。

2）pclass3 的乘客中男性和女性的生存率都是很偏低的。

3）端口 Q 很不幸，因为那里都是 3 等舱的乘客。

9) 登船地点数据填充

港口中也存在缺失值，在这里我用众数来进行填充了，因为 S 登船人最多。

```
data['Embarked'].fillna('S',inplace=True)
data.Embarked.isnull().any()
```

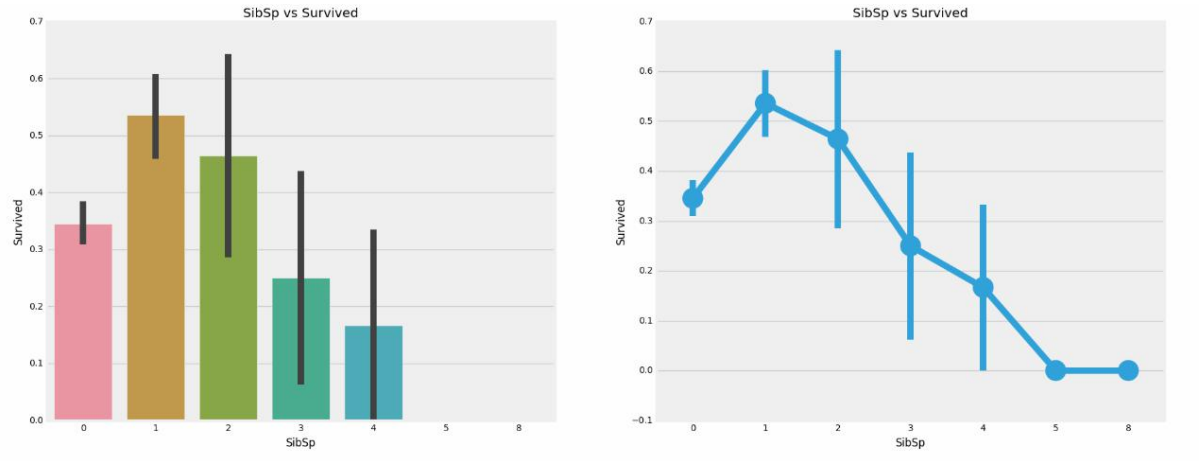
10) 兄弟姐妹的数量（sibsip）分析

交叉分析

```
pd.crosstab([data.SibSp],data.Survived).style.background_gradient(cmap='summer_r')
```

Survived	0	1
SibSp		
0	398	210
1	97	112
2	15	13
3	12	4
4	15	3
5	5	0
8	7	0

```
f,ax=plt.subplots(1,2,figsize=(20,8))
sns.barplot('SibSp','Survived',data=data,ax=ax[0])
ax[0].set_title('SibSp vs Survived')
sns.factorplot('SibSp','Survived',data=data,ax=ax[1])
ax[1].set_title('SibSp vs Survived')
plt.close(2)
plt.show()
```




```
pd.crosstab(data.SibSp,data.Pclass).style.background_gradient(cmap='summer_r')
```

Pclass	1	2	3
SibSp			
0	137	120	351
1	71	55	83
2	5	8	15
3	3	1	12
4	0	0	18
5	0	0	5
8	0	0	7

barplot（条形图）和factorplot（因子图）表明，如果乘客是孤独的船上没有兄弟姐妹，他有34.5%的存活率。如果兄弟姐妹的数量增加，该图大致减少。这是有道理的。也就是说，如果我有一个家庭在船上，我会尽力拯救他们，而不是先救自己。但是令人惊讶的是，5-8名成员家庭的存活率为0%。原因可能是他们在pclass=3的船舱。

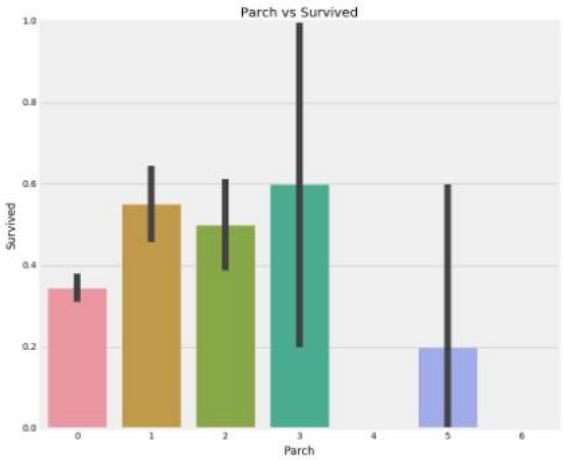
11) 父母和孩子的数量（Parch ）

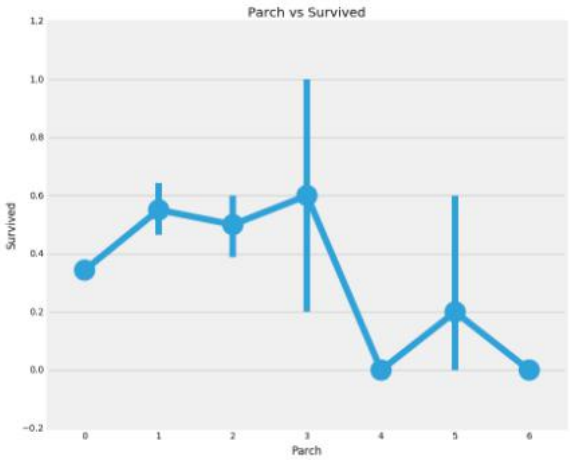
```
pd.crosstab(data.Parch,data.Pclass).style.background_gradient(cmap='summer_r')
```

Pclass	1	2	3
Parch			
0	163	134	381
1	31	32	55
2	21	16	43
3	0	2	3
4	1	0	3
5	0	0	5
6	0	0	1

表明，大家庭都在pclass3。

```
F,ax=plt.subplots(1,2,figsize=(20,8))
sns.barplot('Parch','Survived',data=data,ax=ax[0])
ax[0].set_title('Parch vs Survived')
sns.factorplot('Parch','Survived',data=data,ax=ax[1])
ax[1].set_title('Parch vs Survived')
plt.close(2)
plt.show()
```





这里的结果也很相似。带着父母的乘客有更大的生存机会。然而，它随着数字的增加而减少。
在船上的家庭父母人数中有1-3个的人的生存机会是好的。独自一人也证明是致命的，当船上有4个父母时，生存的机会就会减少。

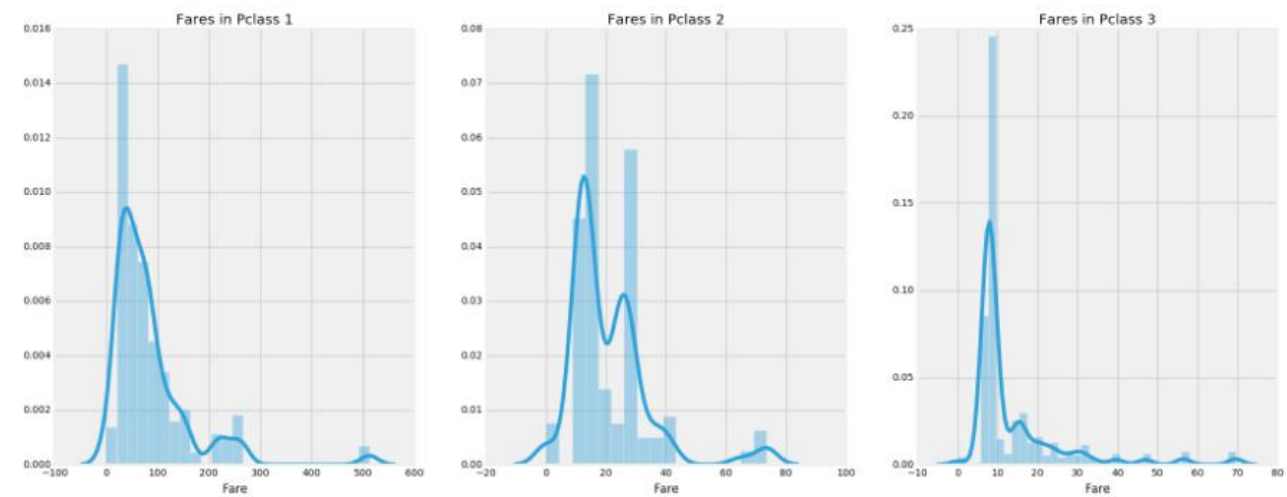
12) 船票的价格（Fare）

```
print('Highest Fare was:',data['Fare'].max())
print('Lowest Fare was:',data['Fare'].min())
print('Average Fare was:',data['Fare'].mean())
```

```
Highest Fare was: 512.3292
Lowest Fare was: 0.0
Average Fare was: 32.2042079685746
```

最低票价是0英镑，最高是512.33。

```
f,ax=plt.subplots(1,3,figsize=(20,8))
sns.distplot(data[data['Pclass']==1].Fare,ax=ax[0])
ax[0].set_title('Fares in Pclass 1')
sns.distplot(data[data['Pclass']==2].Fare,ax=ax[1])
ax[1].set_title('Fares in Pclass 2')
sns.distplot(data[data['Pclass']==3].Fare,ax=ax[2])
ax[2].set_title('Fares in Pclass 3')
plt.show()
```



不同等级仓位价格分布。

13) 分析总结

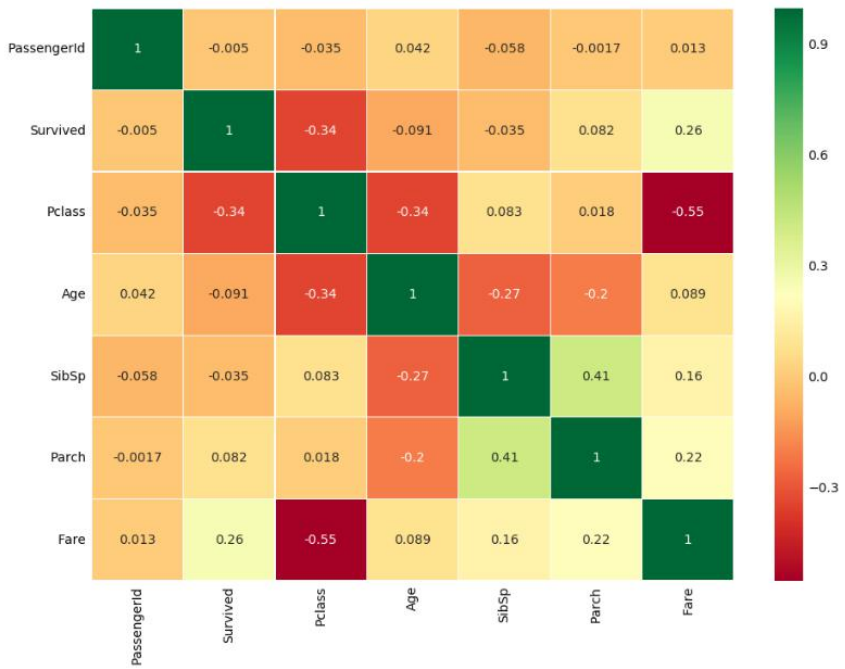
概括地观察所有的特征：

- 性别，与男性相比，女性的生存机会很高。
- Pclass：有，第一类乘客给你更好的生存机会的一个明显趋势。对于pclass3成活率很低。对于女性来说，从pclass1生存的机会几乎是。
- 年龄：小于5-10岁的儿童存活率高。年龄在15到35岁之间的乘客死亡很多。
- 港口：上来的仓位也有区别，死亡率也很大！
- 家庭：有1-2的兄弟姐妹、配偶或父母上1-3显示而不是独自一人或有一个大家庭旅行，你有更大的概率存活。

14) 特征之间的相关性—使用热力图（热度图）分析

热度图：

```
sns.heatmap(data.corr(),annot=True,cmap='RdYlGn',linewidths=0.2)
#data.corr()-->correlation matrix
fig=plt.gcf()
fig.set_size_inches(10,8)
plt.show()
```



使用特征相关热度图，分析特征之间的相关程度。

首先要注意的是，只有数值特征进行比较。
正相关：如果特征A的增加导致特征b的增加，那么它们呈正相关。值1表示完全正相关。
负相关：如果特征A的增加导致特征b的减少，则呈负相关。值-1表示完全负相关。

现在让我们说两个特性是高度或完全相关的，所以一个增加导致另一个增加。这意味着两个特征都包含高度相似的信息，并且信息很少或没有变化。这样的特征对我们来说是没有价值的。
在制作或训练模型时，我们应该尽量减少冗余特性，这样可以减少了训练时间和减少内存损耗。

从上面的图，我们可以看到，特征不显著相关。