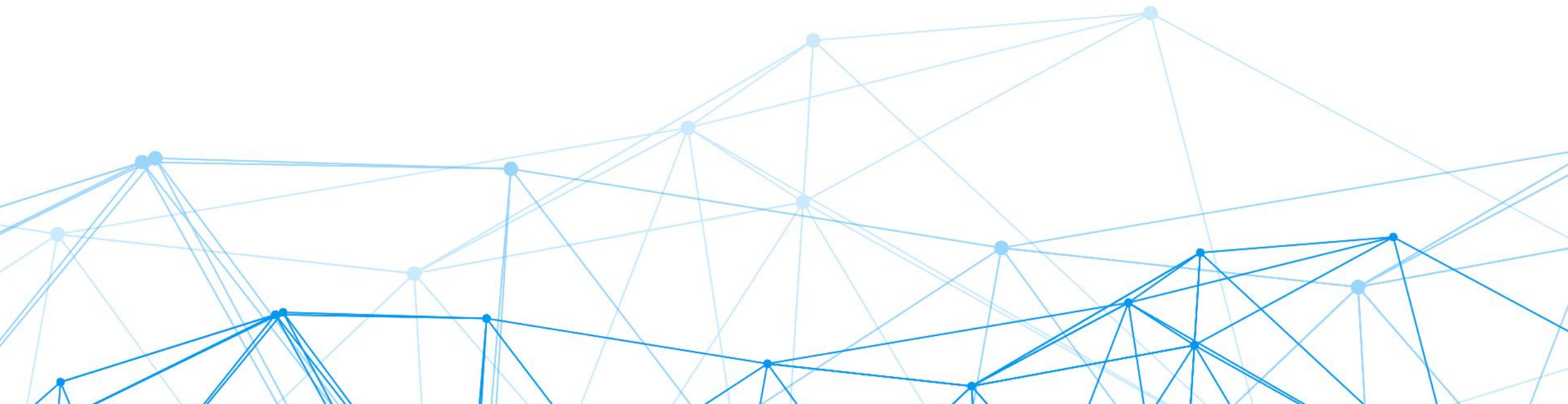
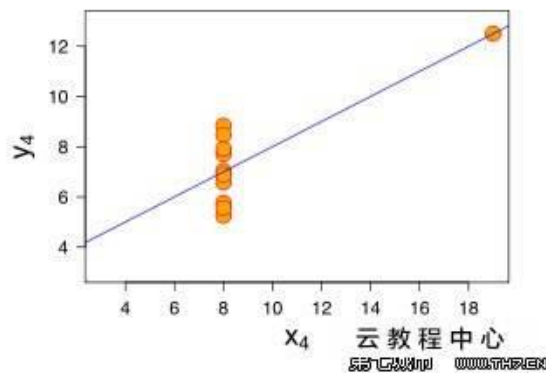
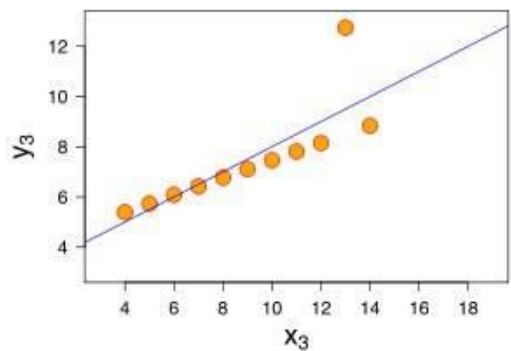
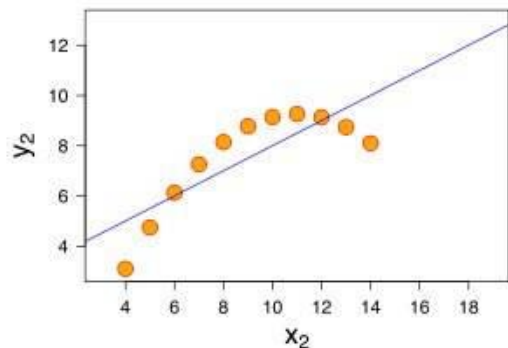
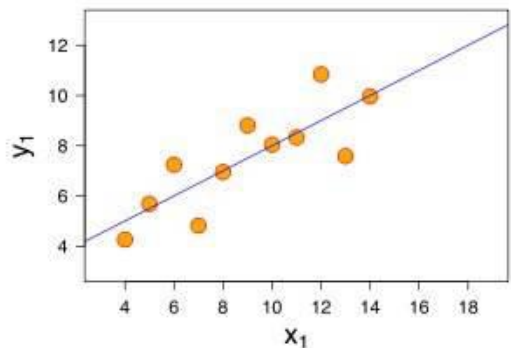


# Python探索性数据分析



## 我们能仅仅依赖统计量吗？



左边四幅图，平均数、标准差、皮尔逊相关系数、线形回归方程都相同。

X1: 正常

X2: 非线性关系

X3: 异常值

X4: 非相关关系+异常值

不能，需要先绘制图表，观察数据分布



## | EDA简介


1. 什么是探索性数据分析（EDA, Exploratory Data Analysis）？

EDA是探索数据的过程，通常会包括探索**数据结构、组成成分、数据分布、变量之间的关系**。在EDA过程中，最重要的工具是可视化图表。

2. 为什么要做EDA？

了解数据，判断数据是否能回答分析问题。

3. EDA的三个目标：

1. 验证数据是否有问题
  2. 判断研究的问题是否能用这些数据来回答
  3. 对研究的问题给一个简易初步的回答
- 



CHAPTER 1

# 第一节 探查数据质量和数据类型





01

## 查看数据类型

规模+用户体验

02

验证有无缺失值、缺失值处理



## 1.1 基础数据结构——数据框(DataFrame)

DataFrame是Pandas中的一个表结构，包括三部分信息，表头（列的名称），表的内容（二维矩阵），索引（每行一个唯一的标记）

	PassengerId	Sex	Pclass	Age	Survived
0	1	male	3	22.0	0
1	2	female	1	38.0	1
2	3	female	3	26.0	1
3	4	female	1	35.0	1
4	5	male	3	35.0	0
5	6	male	3	NaN	0
6	7	male	1	54.0	0

```
#导入pandas
import pandas as pd

#读入数据
df = pd.read_table('titanic.csv',sep=',')
```

```
#取出一列数据
df['PassengerId']

#取出多列数据
df[['PassengerId','Sex','Age']]

#条件索引,取出所有男性数据
df.loc[df.Sex=='male']
```



## 1.1 基础数据类型、基础操作

### 数据类型

#### 1. 分类数据

- 性别
- 专兼职司机
- 乘客购买力

#### 2. 数值型数据(可运算)

- 年龄
- 司机在线时长
- 应答率

### 简单代码实现

#### 1. 查看一个DF各列的数据类型

1. `df.head()`
2. `df.info()` --- 展示df的整体状况, 多少列、列的属性、是否有空值等

#### 2. 数据类型转化 `df.astype()` `df['Pclass2']=df['Pclass'].astype('category')`

	PassengerId	Sex	Pclass	Age
0	1	male	3	22.0
1	2	female	1	38.0
2	3	female	3	26.0
3	4	female	1	35.0
4	5	male	3	35.0
5	6	male	3	NaN
6	7	male	1	54.0
7	8	male	3	2.0
8	9	female	3	27.0
9	10	female	2	14.0

## 1.2 缺失值判断

1. `df.info()`

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 5 columns):
PassengerId    891 non-null int64
Sex            891 non-null object
Age            714 non-null float64
Pclass         891 non-null int64
Pclass2        891 non-null category
dtypes: category(1), float64(1), int64(2), object(1)
```

2. `df.isna().sum()`

```
df2.isnull()
```

	PassengerId	Sex	Age	Pclass	Pclass2
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	True	False	False

```
df2.isnull().sum()
```

```
PassengerId    0
Sex            0
Age            177
Pclass         0
Pclass2        0
```



## 1.3 缺失值处理

### 1. 填充: df.fillna()

#### 1. 用值填充

1. 单一值填充 (0、均值、中位数)
2. 利用字典, 对不同列用不同值填充

#### 2. Method填充

1. 前向填充——空值前面的非空值
2. 后向填充——空值后面的非空值

### 2. 删除: df.dropna(), 可指定按行/列删除含NaN数据

```
df[['Name', 'Sex', 'Age']].head(7)
```

	Name	Sex	Age
0	Braund, Mr. Owen Harris	male	22.0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
2	Heikkinen, Miss. Laina	female	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
4	Allen, Mr. William Henry	male	35.0
5	Moran, Mr. James	male	NaN
6	McCarthy, Mr. Timothy J	male	54.0

```
df[['Name', 'Sex', 'Age']].fillna(method='bfill')
```

	Name	Sex	Age
0	Braund, Mr. Owen Harris	male	22.0
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0
2	Heikkinen, Miss. Laina	female	26.0
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0
4	Allen, Mr. William Henry	male	35.0
5	Moran, Mr. James	male	54.0
6	McCarthy, Mr. Timothy J	male	54.0



CHAPTER 2

# 第二章 探查数据的基础分布情况



A decorative network diagram in the top-left corner, featuring several blue dots connected by thin blue lines, forming a web-like structure.

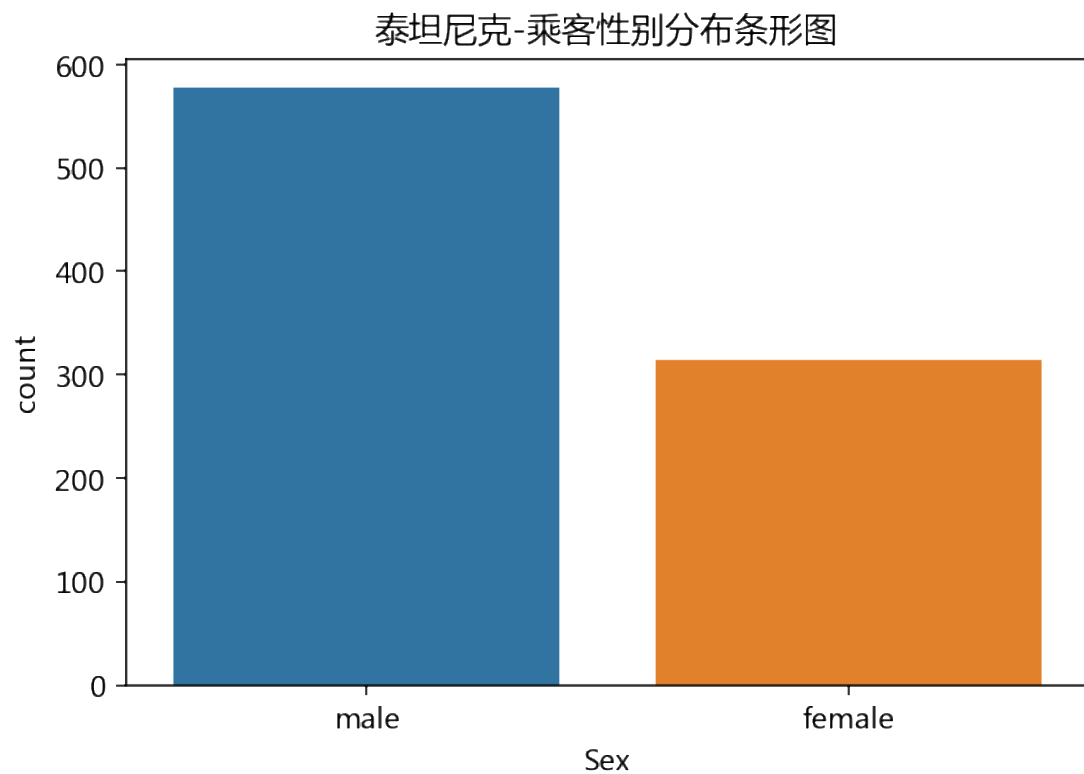
01 分类变量

02 连续变量

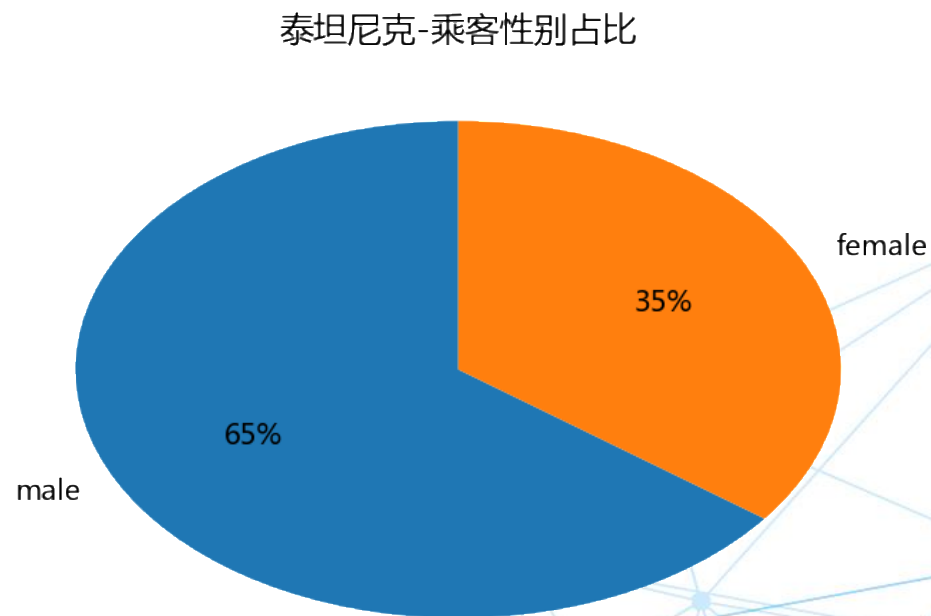
A decorative network diagram in the bottom-right corner, featuring several blue dots connected by thin blue lines, forming a web-like structure.

## 2.1 分类变量

- 分类变量分布
  - 频数——条形图(barplot)



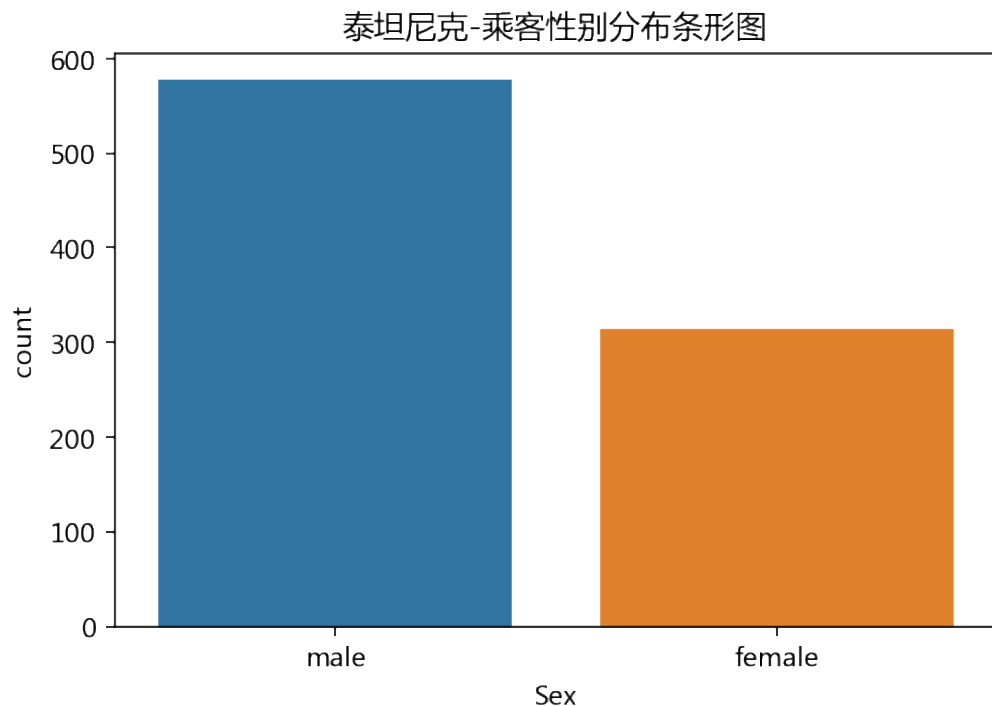
- 分类变量分布
  - 占比——扇形图(pieplot)



## 2.1 分类变量频数——条形图(countplot)

```
sns.countplot(data=df #指定数据
               ,x='Sex' #分类变量, 放在x轴
               )

plt.title('泰坦尼克-乘客性别分布条形图')
```



使用包: seaborn, 简写 sns  
函数: countplot()

用法:

sns.countplot(

data=df, #数据集

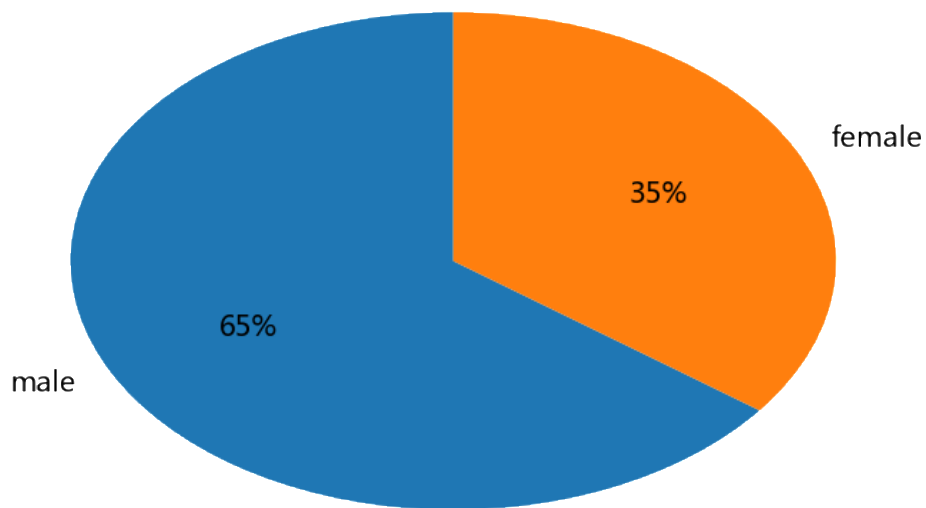
x='Sex' # 要统计的分类变量

)

## 分类变量占比——扇形图(pie)

```
sex_freq = df.Sex.value_counts()
plt.pie(x=sex_freq, #要统计的数据
        labels = sex_freq.index, #分类标签名称
        startangle=90, #起始角度, 90度
        autopct='%1.1f%%' #添加占比, 设置文本格式
    )
plt.title('泰坦尼克-乘客性别占比')
```

泰坦尼克-乘客性别占比



使用包: matplotlib.pylab() 简写为plt  
函数: df.Series.value\_counts、  
plt.pie()

用法:

1. df.Series.value\_counts() 返回分类变量的频数
2. plt.pie 绘制扇形图



## 2.2 连续变量——统计量

常用函数：

`pd.describe()`

返回计数值、均值、标准差、等统计量

```
df.describe()
```

	Age	SibSp	Parch	Fare
count	<u>891.000000</u>	<u>891.000000</u>	<u>891.000000</u>	<u>891.000000</u>
mean	<u>29.870561</u>	0.523008	0.381594	32.204208
std	14.597668	1.102743	0.806057	49.693429
min	0.420000	0.000000	0.000000	0.000000
25%	<u>21.000000</u>	0.000000	0.000000	7.910400
50%	<u>29.000000</u>	0.000000	0.000000	14.454200
75%	39.000000	1.000000	0.000000	<u>31.000000</u>
max	80.000000	8.000000	6.000000	<u>512.329200</u>

常用函数：

`pd.Series.mean()`

`pd.Series.std()`

```
df.Age.mean()
```

29.87056116722783

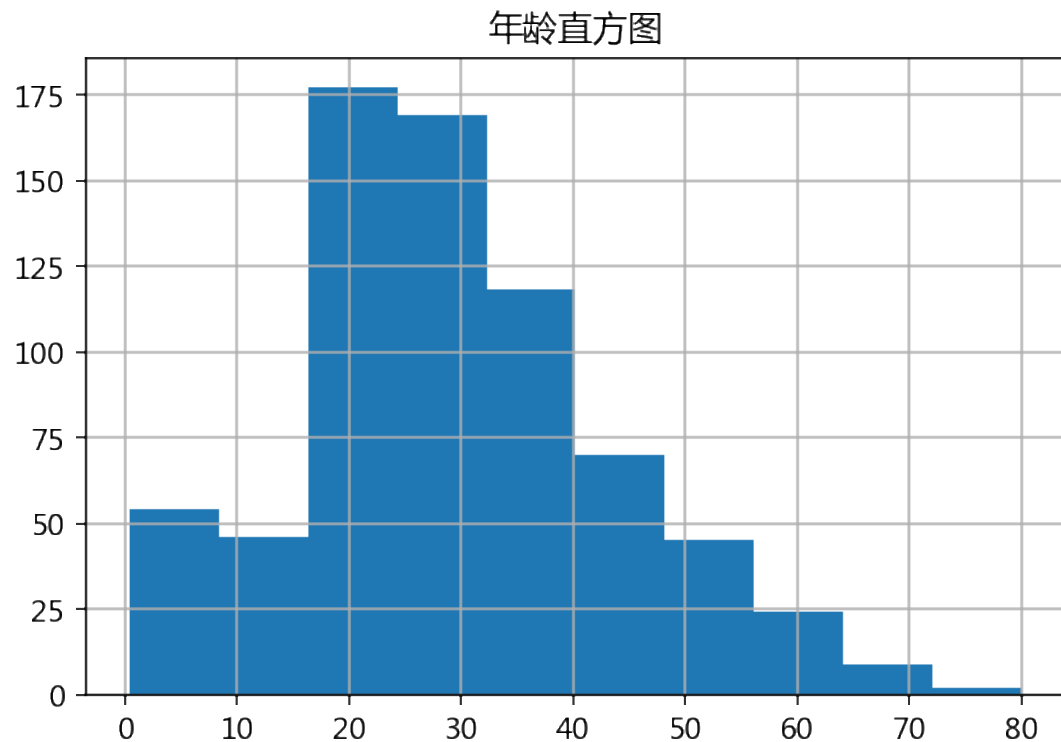
```
df.Age.std()
```

14.597667657302386

## 2.2 连续变量分布——直方图(histogram)

#直方图

```
df.Age.hist(bins=10)  
plt.title('年龄直方图')
```



函数: `pd.Series.hist()`

`df.Age.hist(bins=10 #把Age分为等距10组)`

## 2.2 连续变量分布——条形图(countplot)

### 1. 连续变量离散化

```
df['Age2'] = pd.cut(x=df.Age, #要分组的数据  
                    bins=[0,10,20,40,50,80], # 指定分组位置  
                    labels=['0~10', '10~20', '20~40', '40~50',  
                           '50~80'] #分组名称  
)
```

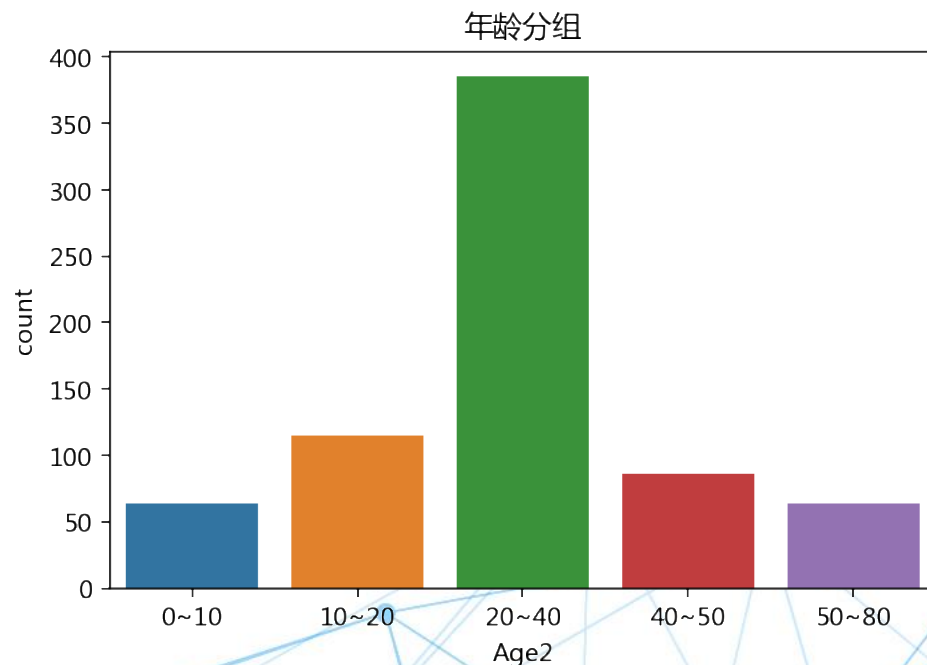
```
df[['Age', 'Age2']].head()
```

	Age	Age2
0	22.0	20~40
1	38.0	20~40
2	26.0	20~40
3	35.0	20~40
4	35.0	20~40

注：根据最大值、最小值确定合适的分组位置

### 2. 分组统计，绘制条形图

```
sns.countplot(data=df, x='Age2')  
plt.title('年龄分组')
```





CHAPTER 3

# 第三章 探索变量之间的关系

01

## 分类 X 分类

条形图、列联表、卡方检验

02

## 分类 X 连续

KDEplot、T检验等均值检验、多因素方差分析

03

## 连续 X 连续

散点图、相关系数

### 3. 探索变量间关系

X变量类型/Y变量类型		二分类	连续
单个变量	二分类	列联表卡方检验	T检验等均值检验
	多重分类	卡方检验	多因素方差分析ANOVA
	连续	T检验等均值检验	相关系数、互信息值等
多个变量	分类	逻辑回归	多因素方差分析ANOVA 线性回归
	连续	逻辑回归	线性回归



## 3.1 分类变量 X 分类变量——列联表

### 1. 分组groupby

```
#支持按列分组
#group1 = df.groupby('key1')

#group2 = df.groupby(['key1', 'key2'])

#分组，分组后g1是groupby 对象
g1 = df.groupby(['Sex', 'Pclass'])
g1
```

```
<pandas.core.groupby.groupby.DataFrameGroupBy object at 0x000002695033CF98>
```

```
#聚合，支持多种聚合函数，mean、std.....
g1.count()
```

		PassengerId
Sex	Pclass	
female	1	94
	2	76
	3	144
male	1	122
	2	108
	3	347

## 3.1 分类变量 X 分类变量——列联表

### 2. 列联表——基于pivot\_table (类似于数据透视表)

```
#pd.pivot_table(df, index='key1', columns='key2', aggfunc=)
#其中参数index指定“行”键, columns指定“列”键, aggfunc 支持很多函数
df.pivot_table(df[['Survived']], index='Sex', columns='Pclass',
               aggfunc=[len, np.mean])
```

	len			mean		
	Survived			Survived		
Pclass	1	2	3	1	2	3
Sex						
female	94	76	144	0.968085	0.921053	0.500000
male	122	108	347	0.368852	0.157407	0.135447

### 3. 列联表——基于crasstab

```
#pd.crosstab(df.key1, df.key2, margins=True)
pd.crosstab(df.Sex, df.Survived, margins=True)
```

Survived	0	1	All
Sex			
female	81	233	314
male	468	109	577
All	549	342	891

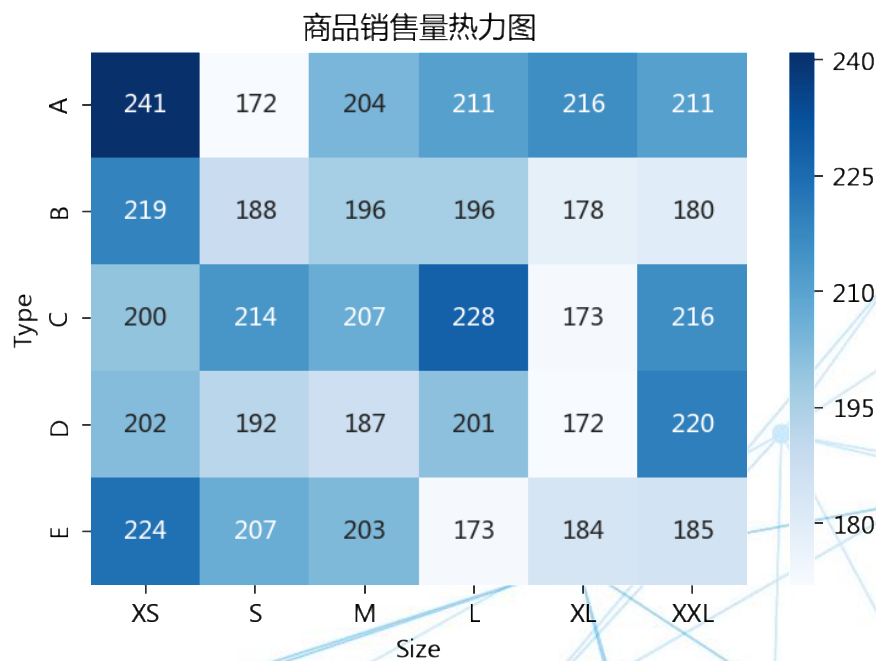
### 3.1 分类变量 X 分类变量——热力图

分类变量类别较多时，绘制heatmap观察其关系

```
x = pd.crosstab(df.Type, df.Size)
x
```

Size	XS	S	M	L	XL	XXL
Type						
A	197	185	216	219	187	209
B	207	192	215	192	186	222
C	190	178	201	203	182	194
D	195	209	225	169	186	196
E	198	214	170	210	222	231

```
sns.heatmap(x,
              annot=True, #设置显示数据标签
              fmt='.0f' #数字格式控制
              #,mask=x.values<200 , #把小于200的区域覆盖掉
              ,cmap='Blues' #设定颜色
              )
plt.title('商品销售量热力图')
```



## 3.1 分类变量 X 分类变量——关联性分析

卡方检验：两个分类变量的关联性分析

核心判断：观察频数与期望是否一致

		PassengerId
Sex	Survived	
female	0	81
	1	233
male	0	468
	1	109

如果性别与存活情况无关，应观察到如下结果

		PassengerId
Sex	Survived	
female	0	193
	1	121
male	0	356
	1	221

```
#存活率与性别列联表
d = df[['PassengerId', 'Survived', 'Sex']].groupby(['Survived', 'Sex']).count()
print(d)

from scipy import stats

#卡方检验
stats.chisquare(d)

#p值在显著性水平0.05下小于0.05
```

```
PassengerId
Survived Sex
0         female      81
          male     468
1         female     233
          male     109
```

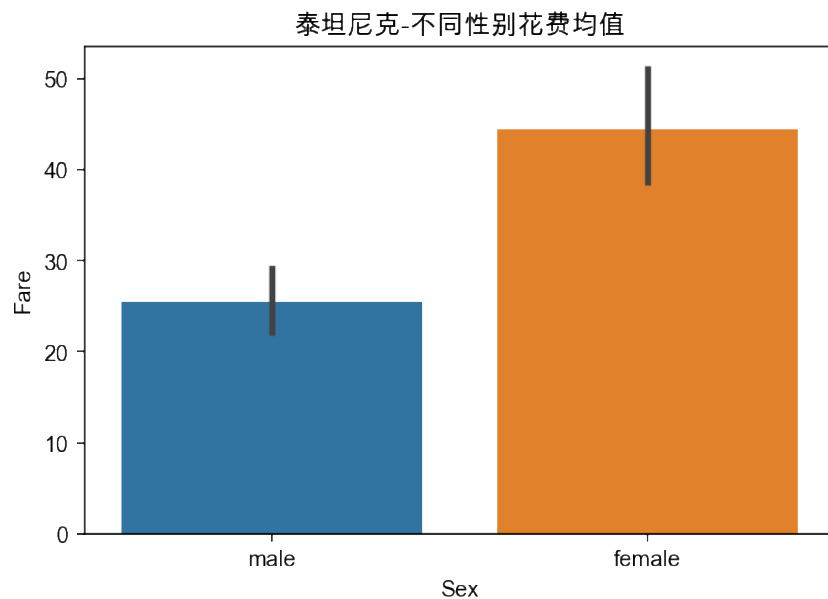
```
Power_divergenceResult(statistic=array([418.78675645]), pvalue=array([1.88617143e-90]))
```

卡方检验要求样本足够大，样本过小可能出现错误结果

## 3.2 分类变量 X 连续变量——分布情况

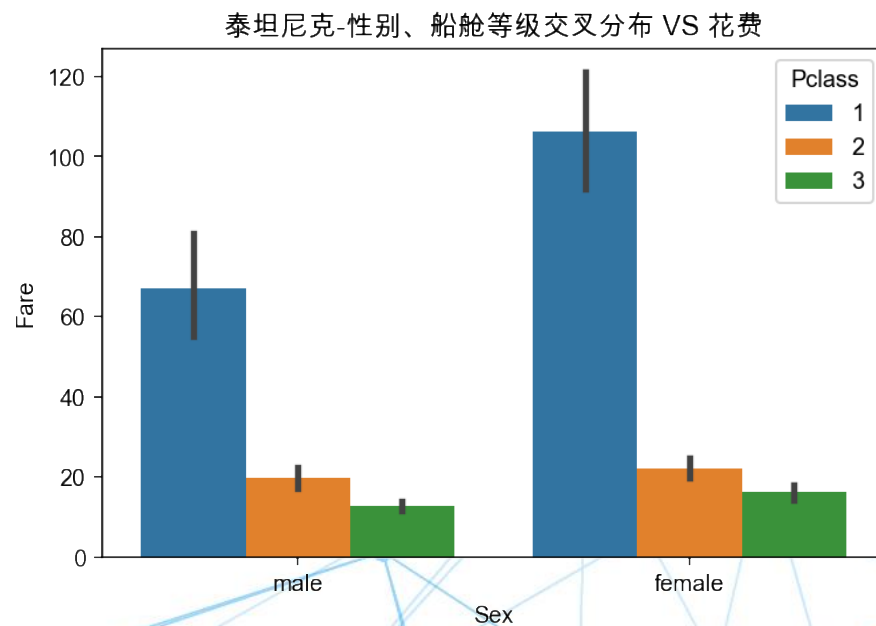
### 1. 基于barplot观察不同分类类别下连续变量的分布差异

```
#按照x分组后, 按estimator统计, 以结果作为条形图的高度
sns.barplot(data=df,
            x='Sex', #按x分组
            y='Fare', #对y做统计
            #estimator=np.mean #支持多种统计量, 默认为均值
            )
plt.title('泰坦尼克-不同性别船费均值')
```



```
#可以支持多变量分组
```

```
sns.barplot(data=df, x='Sex',
            hue='Pclass', #次级分类变量
            y='Fare'
            )
plt.title('泰坦尼克-性别、船舱等级交叉分布 VS 船费')
```



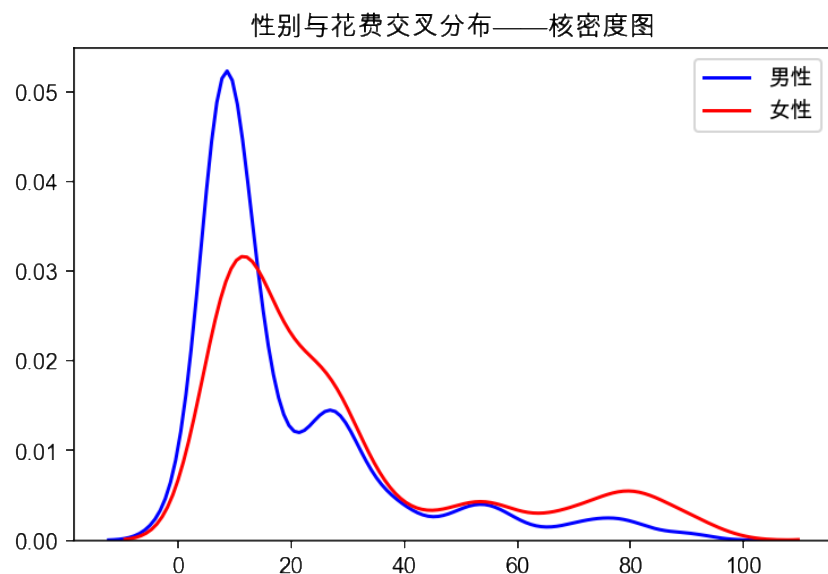


## 3.2 分类变量 X 连续变量——分布情况

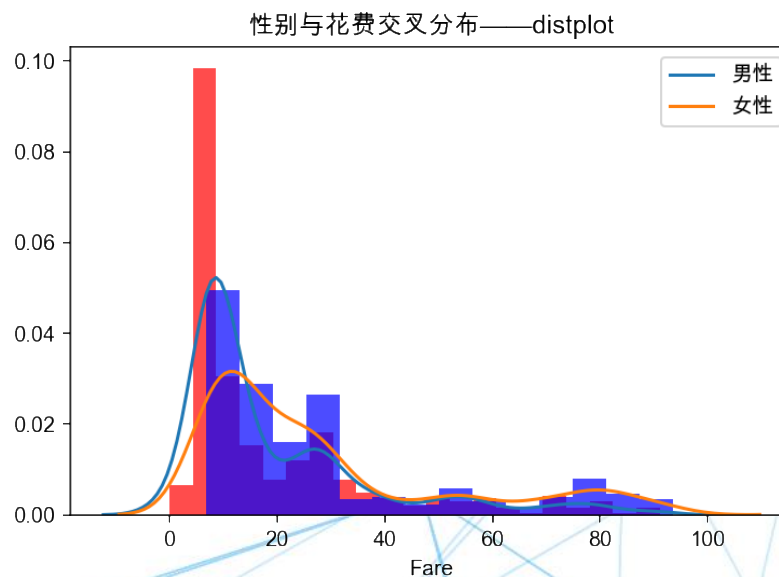
### 2. 堆叠kdeplot、distplot不同分类类别下连续变量的分布差异

```
df2 = df.loc[df.Fare <= 100]

ae = sns.kdeplot(df2.loc[df2.Sex == 'male'].Fare,
                 color='b', label='男性')
ax = sns.kdeplot(df2.loc[df2.Sex == 'female'].Fare,
                 color='r', label='女性')
plt.title('性别与花费交叉分布——核密度图')
```



```
ae = sns.distplot(df2.loc[df2.Sex == 'male'].Fare,
                  kde_kws={"label": "male"}, #设置图例
                  hist_kws={"linewidth": 2,
                             "alpha": 0.7, "color": "r"
                            } #设置格式
                  )
ax = sns.distplot(df2.loc[df2.Sex == 'female'].Fare,
                  kde_kws={"label": "female"},
                  hist_kws={"linewidth": 2,
                             "alpha": 0.7, "color": "b"
                            })
plt.title('性别与花费交叉分布——distplot')
```





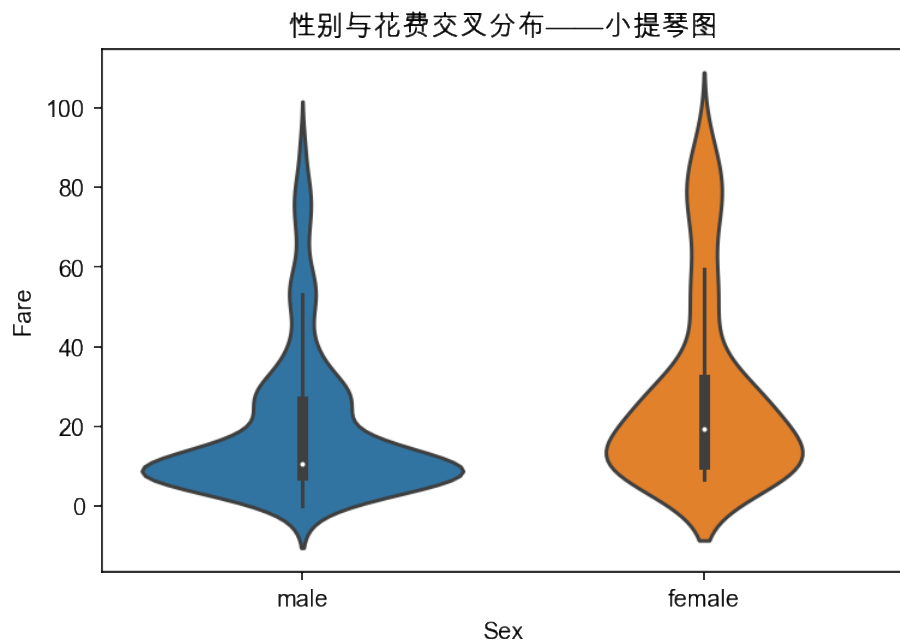
## 3.2 分类变量 X 连续变量——分布情况

### 3. 基于小提琴图（violinplot）观察分类后连续变量的分布

箱线图中所有绘图组件都对应于实际数据点，小提琴绘图以基础分布的核密度估计为特征

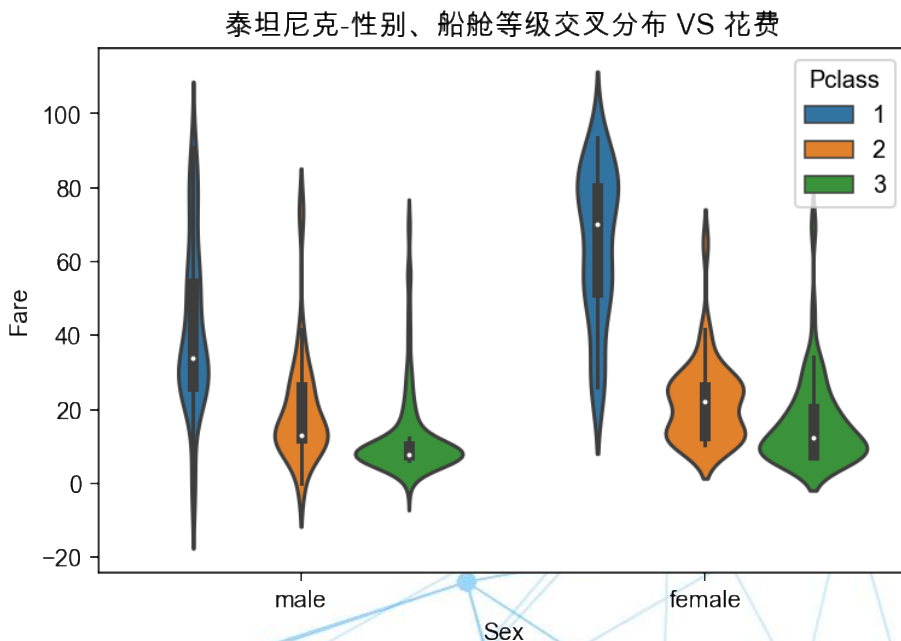
#与boxplot类似

```
sns.violinplot(data=df2,x='Sex',y='Fare')  
plt.title('性别与花费交叉分布——小提琴图')
```



#多个分类变量

```
sns.violinplot(data=df2,x='Sex',y='Fare',hue='Pclass')  
plt.title('泰坦尼克-性别、船舱等级交叉分布 vs 花费')
```



## 3.2 分类变量 X 连续变量——均值检验判断变量关系

### 两个类别分类变量——Z检验、T检验（均值）

选择T检验、Z检验：

1. 样本量大——Z检验、T检验

2. 样本量小

1. 总体方差已知——Z检验

2. 总体方差未知——T检验

*#假设男性、女性花费相同*

```
male_fare = df2.loc[df2.Sex=='male'].Fare  
female_fare = df2.loc[df2.Sex=='female'].Fare
```

*#用ttest\_ind做T检验，要求输入原始样本数据*

```
t_stats, p_value = stats.ttest_ind(male_fare, female_fare)
```

```
print("P value is %.10f" %(p_value)) # 双边检验
```

```
P value is 0.0000000132
```

注：在大样本情况下，t检验和z检验是近似的，因此也可使用t检验

## 3.2 分类变量 X 连续变量——方差分析

多个类别分类变量——方差分析（ANOVA）的核心思想：总误差=组内误差+组间误差

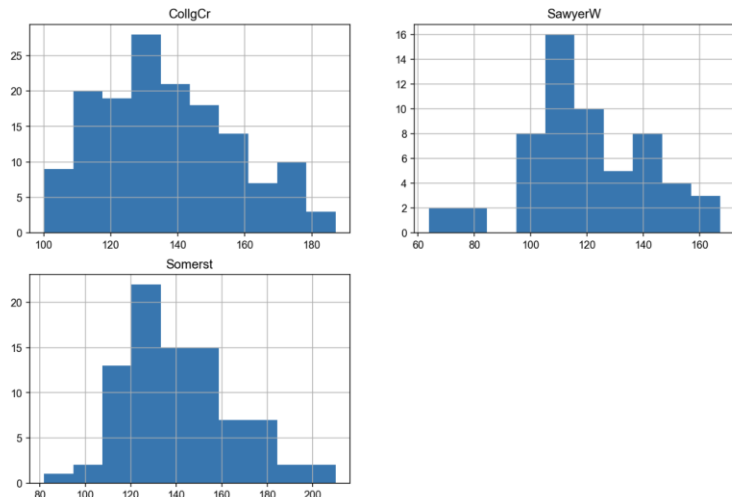
方差分析条件：1、每个总体都服从正态分布；2、总体方差相同；3、观测值独立  
需要做：正态性检验、方差齐性检验

#绘制直方图观察分布

```
plt.figure(figsize=(12,8))
plt.subplot(221)
df3.loc[df3.Neighborhood=='CollgCr'].Avg_price.hist()
plt.title('CollgCr')
plt.subplot(222)
df3.loc[df3.Neighborhood=='SawyerW'].Avg_price.hist()
plt.title('SawyerW')

plt.subplot(223)
df3.loc[df3.Neighborhood=='Somerst'].Avg_price.hist()
plt.title('Somerst')
```

Text(0.5, 1.0, 'Somerst')



#kstest 是一个很强大的检验模块，除了正态性检验，  
#还能检验 scipy.stats 中的其他数据分布类型

```
from scipy import stats
c1=df3.loc[df3.Neighborhood=='CollgCr'].SalePrice
c2=df3.loc[df3.Neighborhood=='SawyerW'].SalePrice
c3=df3.loc[df3.Neighborhood=='Somerst'].SalePrice
```

#标准化

```
normed_c1 = (c1-c1.mean())/c1.std()
normed_c2 = (c2-c2.mean())/c2.std()
normed_c3 = (c3-c3.mean())/c3.std()
```

```
print(stats.kstest(normed_c1, 'norm'))
print(stats.kstest(normed_c2, 'norm'))
print(stats.kstest(normed_c3, 'norm'))
# 结果返回两个值: statistic → D值, pvalue → P值
# p值>0.05,可以认为服从整天分布
```

```
KstestResult(statistic=0.07876189252392499, pvalue=0.2988435199271138)
KstestResult(statistic=0.0788643581597126, pvalue=0.86346997450627)
KstestResult(statistic=0.08572863015016824, pvalue=0.5342204597289284)
```

## 3.2 分类变量 X 连续变量——方差分析

- 方差齐性检验（莱文检验，levene）
- 单因素方差分析

#方差齐性检验

```
import scipy
```

```
c1=df3.loc[df3.Neighborhood=='CollgCr'].SalePrice  
c2=df3.loc[df3.Neighborhood=='SawyerW'].SalePrice  
c3=df3.loc[df3.Neighborhood=='Somerst'].SalePrice
```

```
scipy.stats.levene(c1,c2,c3)
```

#检验结果为 $p>0.05$ 所以，可以认为方差是相等的

```
LeveneResult(statistic=1.2156177816078617, pvalue=0.2980331746348337)
```

#单因素方差分析

```
from statsmodels.formula.api import ols  
from statsmodels.stats.anova import anova_lm
```

#不同社区 (Neighborhood)

```
formula = 'Avg_price~Neighborhood'  
anova_results = anova_lm(ols(formula,df3).fit())  
print(anova_results)
```

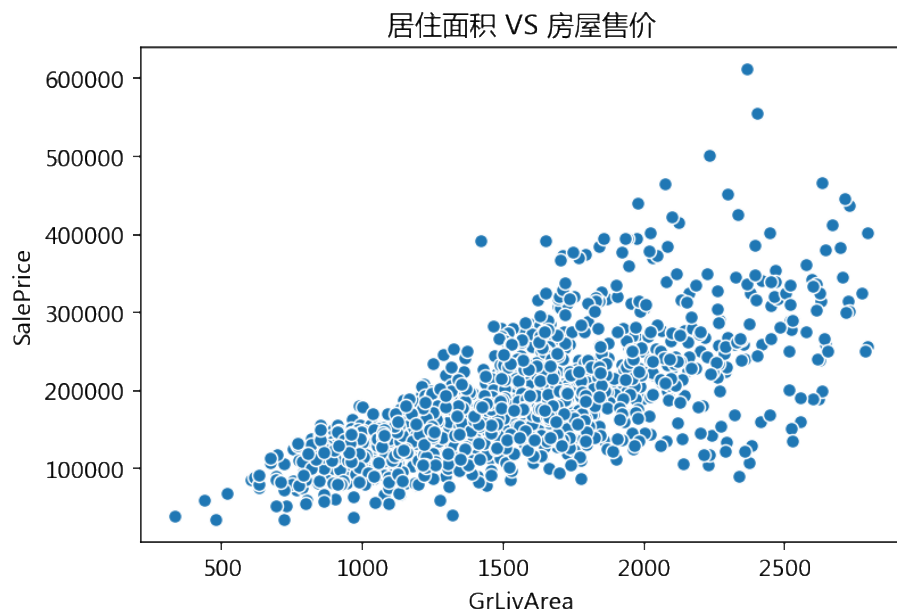
#P值小于0.05，可以拒绝不同社区对房价没有影响的假设，可以认为社区对售价有显著影响

	df	sum_sq	mean_sq	F	PR(>F)
Neighborhood	3.0	59384.757781	19794.919260	40.065289	2.853460e-23
Residual	512.0	252962.073253	494.066549	NaN	NaN

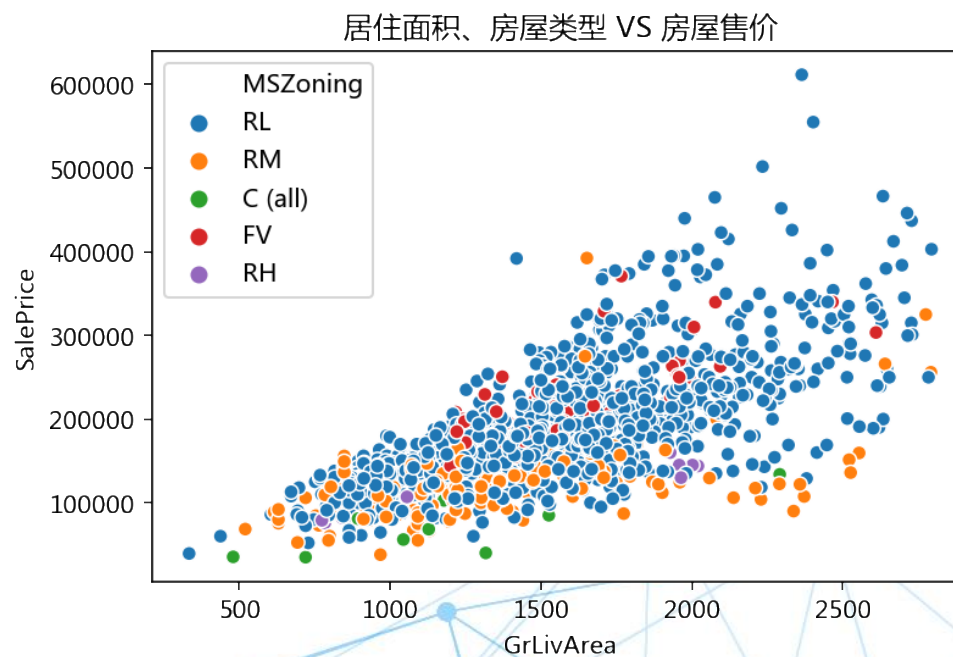
### 3.3 连续变量 X 连续变量——散点图

散点图：观察2个连续变量的趋势，发现潜在规律。

```
sns.scatterplot(data=df2,x='GrLivArea',y='SalePrice')  
plt.title('居住面积 VS 房屋售价')
```



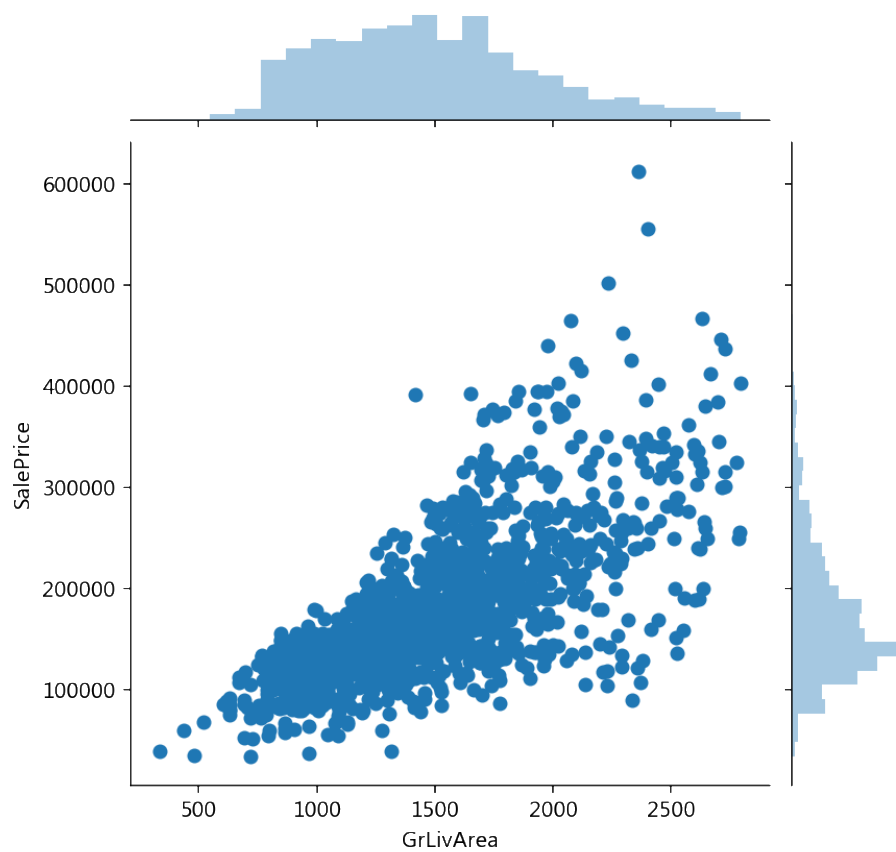
```
#分组散点图  
sns.scatterplot(data=df2,x='GrLivArea',y='SalePrice',  
                hue='MSZoning')  
plt.title('居住面积、房屋类型 VS 房屋售价')
```





### 3.3 连续变量 X 连续变量——jointplot

```
g = sns.jointplot(data=df2, x='GrLivArea', y='SalePrice')
```





### 3.3 连续变量 X 连续变量——相关系数

#### 线性相关——Pearson相关系数

定义：两个变量 X、Y 之间的协方差和标准差的比值

```
from scipy.stats import pearsonr

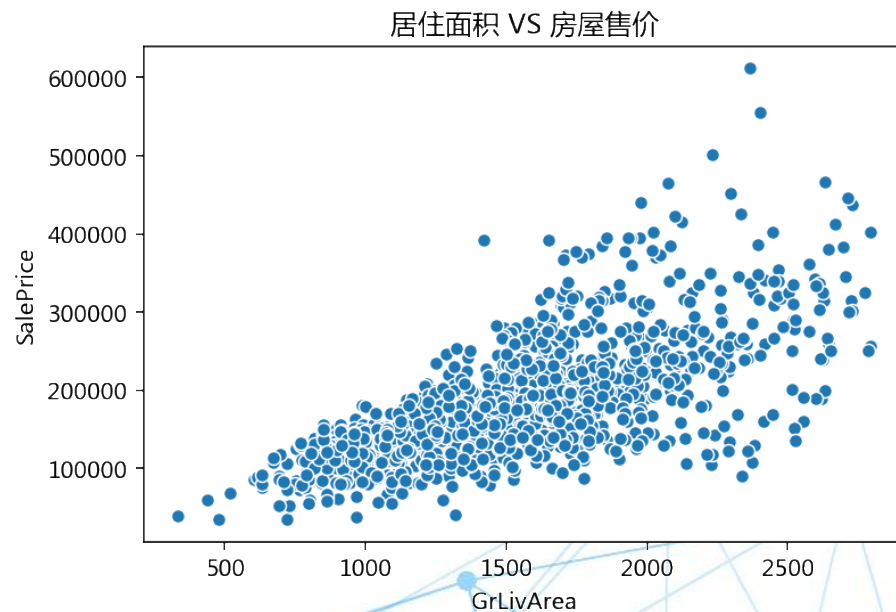
x = df2.SalePrice #房价
y = df2.GrLivArea #居住面积

#计算pearson相关系数
r_row, p_value = pearsonr(x, y)
print(r_row.round(3)) #pearson 相关系数
print(p_value.round(3)) #P值
```

0.699

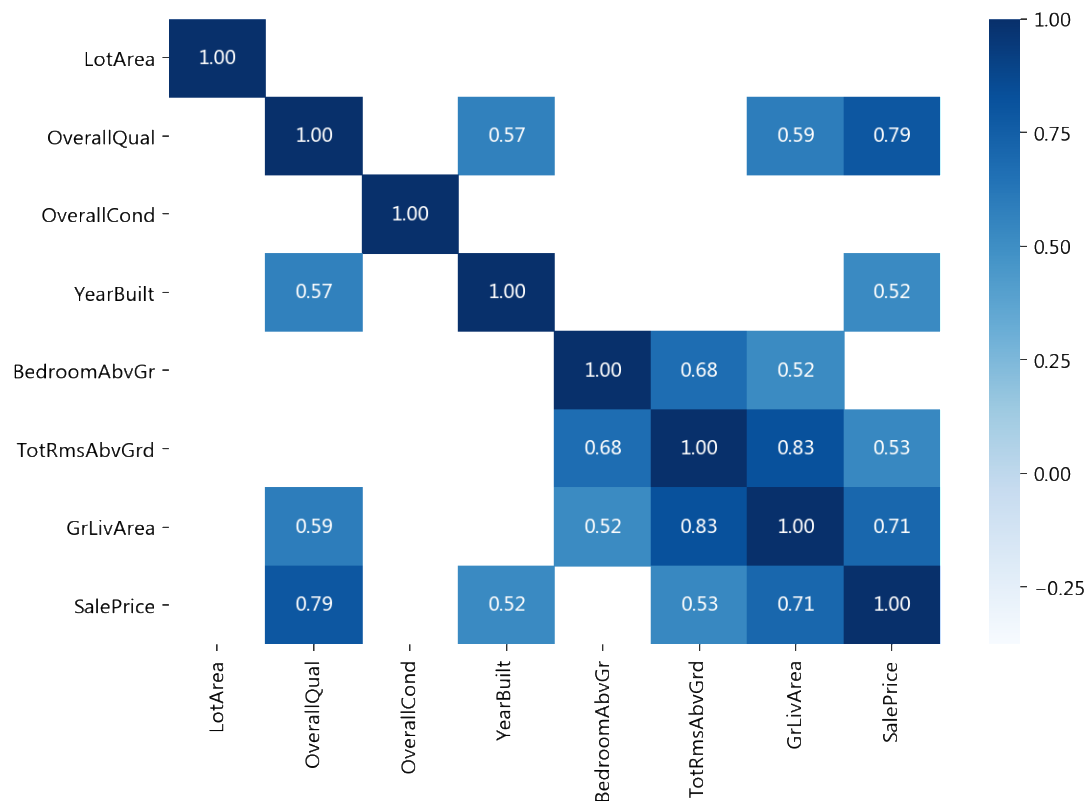
0.0

```
sns.scatterplot(data=df2, x='GrLivArea', y='SalePrice')
plt.title('居住面积 VS 房屋售价')
```



### 3.3 连续变量 X 连续变量——热力图

```
plt.figure(figsize=(9,6)) #设置图片大小
sns.heatmap(df.corr(), #相关系数矩阵
            annot=True,
            fmt='.2f',
            mask=df.corr().values<0.5 #小于0.5的值不展示
            ,cmap='Blues'
            )
```




热力图——对绘制相关系数矩阵绘制热力图，观察变量间相关性强弱



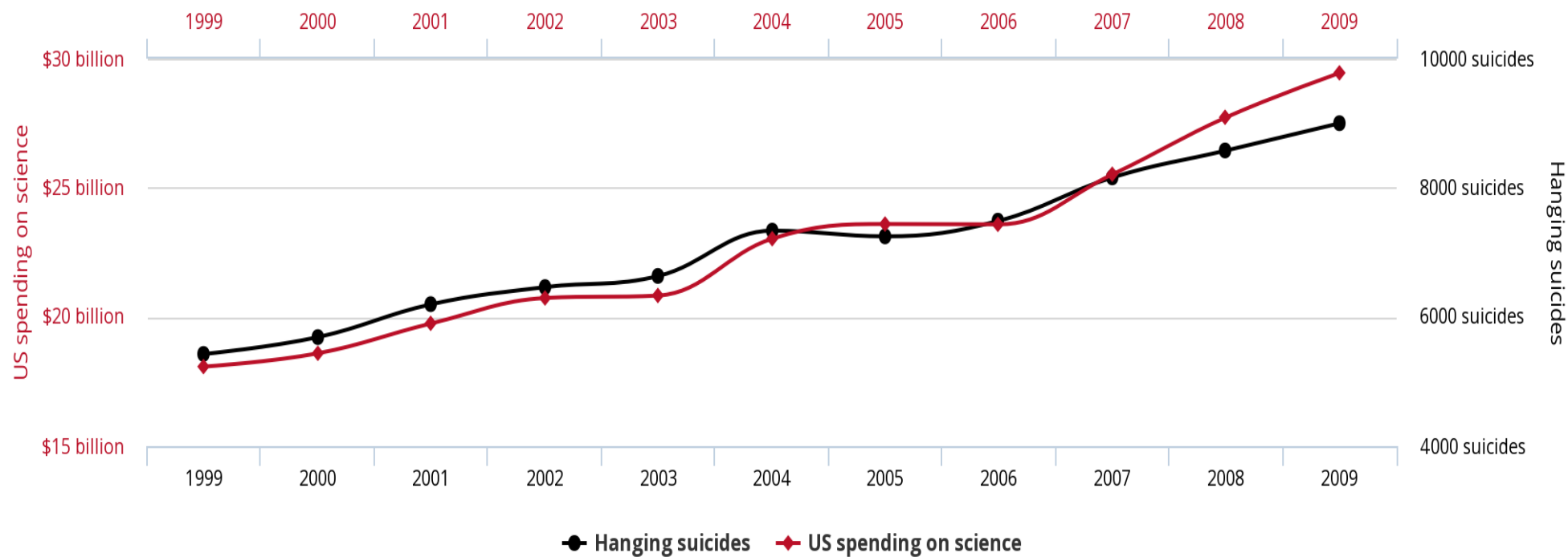
### 3.3 连续变量 X 连续变量——相关性与因果性

如果A和B相关，有至少五种可能性：

1. A导致B
  2. B导致A
  3. C导致A和B
  4. A和B互为因果
  5. 小样本引起的巧合
- 

### 3.3 相关性与因果性——小样本引起的巧合

## US spending on science, space, and technology correlates with Suicides by hanging, strangulation and suffocation





## 参考资料

课程推荐好用的新包及几篇实用EDA案例:

- 1.用于探索缺失值和变量量相关性等: <https://github.com/ResidentMario/missingno>
  - 2.用于快速可视化诊断整个数据集的质量:<https://github.com/pandas-profiling/pandas-profiling>
  - 3.各种连续变量量分布的探查:<http://seaborn.pydata.org/tutorial/distributions.html>
  - 4.Kaggle Extensive EDA方法示例1:<https://www.kaggle.com/kabure/lending-club-extensive-eda>
  - 5.Kaggle Extensive EDA方方法示例例2  
Byplotly:<https://www.kaggle.com/shivamb/homecreditrisk-extensive-eda-baseline-0-772>
- 