一：seaborn介绍与基础案例

Seaborn其实是在matplotlib的基础上进行了更高级的API封装，从而使得作图更加容易，在大多数情况下使用seaborn就能做出很具有吸引力的图

Seaborn作为一个带着定制主题和高级界面控制的Matplotlib扩展包， seaborn不是matplotlib的替代品，只是matplotlib的补充

Seaborn的API：https://www.cntofu.com/book/172/docs/24.md

Iris也称鸢尾花卉数据集，数据集包含150个数据集，分为3类，每类50个数据，每个数据包含4个属性。

属性：花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性

种类：setosa(山鸢尾)，versicolor(杂色鸢尾)，virginica(弗吉尼亚鸢尾)

1、鸢尾花数据读取

代码目标：把鸢尾花数据进行读取

```
import pandas as pd
import seaborn as sns
# 读取鸢尾花的数据
data=pd.read_csv('iris.csv')
data.head()
```

2、主题与配色方案

1. 设置主题
   set_style( )是用来设置主题的，Seaborn有五个预设好的主题： darkgrid , whitegrid , dark , white ,和 ticks 默认： darkgrid
2. seaborn palette参数各配色方案及显示效果
   https://blog.csdn.net/panlb1990/article/details/103851983

3、数据可视化

1. 单变量直方图

distplot( )为hist加强版， displot()集合了hist()与kdeplot的功能，可以绘制密度分布图，密度估计图可以比较直观的看出数据样本本身的分布特征

```
# 设置主题
sns.set(style='darkgrid')
import warnings
warnings.filterwarnings('ignore')
# 绘制单变量观测值分布图(直方图)  数值型 花瓣长度
sns.distplot(data['petal_length'],rug=True,vertical=False)
# 设置字号大小的缩放程度
sns.set(font_scale=1)
list1=[1,2,3,4]
sns.rugplot(list1)
```

(1)、正态分布（Normal distribution）

也称“常态分布”，又名高斯分布，正态曲线呈钟型，两头低，中间高，左右对称因其曲线呈钟形，因此人们又经常称之为钟形曲线。

若随机变量X服从一个数学期望为μ、方差为σ^2的正态分布，记为N(μ，σ^2)。其概率密度函数为正态分布的期望值μ决定了其位置，其标准差σ决定了分布的幅度。当μ = 0,σ = 1时的正态分布是标准正态分布

标准正态分布：标准正态分布（英语：standard normal distribution），是一个在数学、物理及工程等领域都非常重要的概率分布，在统计学的许多方面有着重大的影响力。期望值μ=0，即曲线图象对称轴为Y轴，标准差σ=1条件下的正态分布，记为N(0，1)。

(2)、标准差和方差

$$\text{方差}= \frac{\Sigma(x_i - \bar{x})^2}{n}$$

$$\text{标准差}= \sqrt{\frac{\Sigma(x_i - \bar{x})^2}{n}}$$

我们举个具体的例子，在NBA中，平均数据用来衡量一个球员的战斗力，比如场均得分，盖帽，抢断，助攻等。

那么我们现在想一个问题。如果你是教练，你想知道哪位球员发挥最稳当。因为你需要一支值得信赖的球员队伍，他最不想要的就是表现时好时坏，水平反复无常，波动很大的队员。他需要得是分高，且发挥稳定的球员。

而标准差就是为了描述数据集的波动大小而发明的。

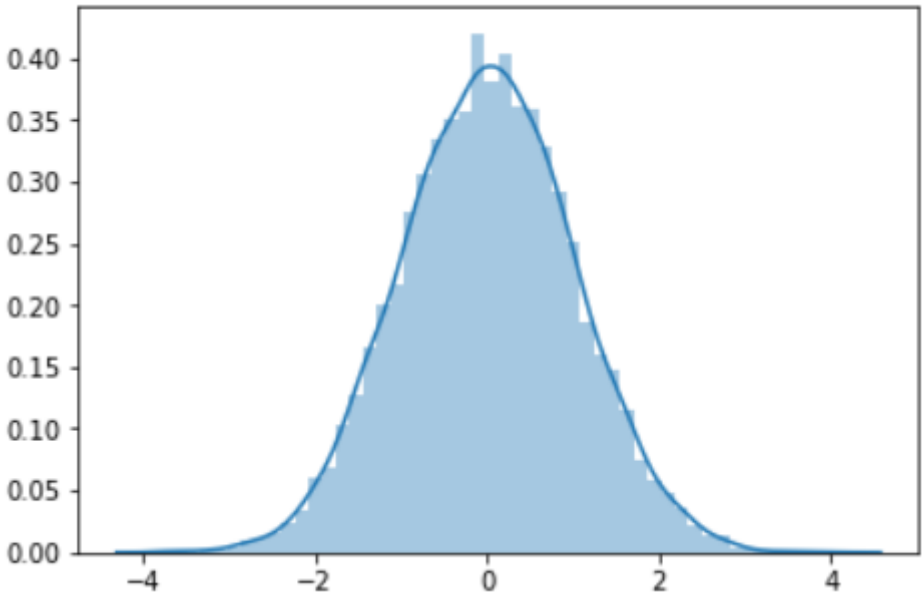
标准差和平均数的量纲（单位）是一致的，在描述一个波动范围时标准差比方差更方便。

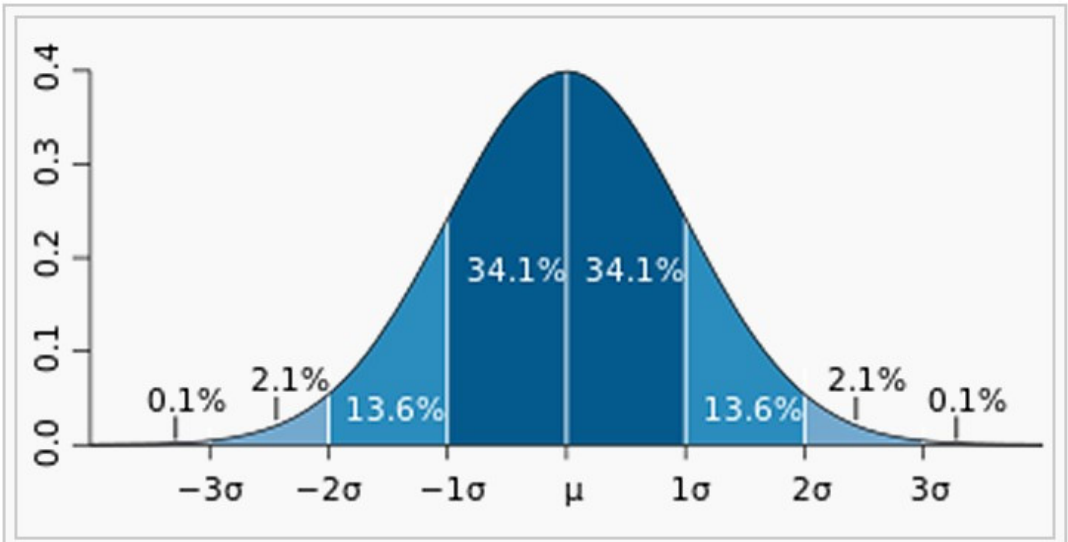
方差被标准化了。开根号，统一单位，取名标准化

简单来说，标准差是一组数值自平均值分散开来的程度的一种测量观念。一个较大的标准差，代表大部分的数值和其平均值之间差异较大；一个较小的标准差，代表这些数值较接近平均值。

标准差通常是相对于样本数据的平均值而定的，表示距离平均值的平均距离

```
import seaborn as sns
import numpy as np
np.random.seed(123)
# a、服从μ=0,σ=1 的正态分布：
data=np.random.randn(10000)
# data
sns.distplot(data)
# b、服从μ=loc,σ=scale 的正态分布：
data2=np.random.normal(loc=2, scale=1, size=10000)
data2
sns.distplot(data2)
```





深蓝色区域是距平均值小于一个标准差之内的数值范围。在**正态分布**中，此范围所占比率为全部数值之**68%**，根据正态分布，两个标准差之内的比率合起来为**95%**；三个标准差之内的比率合起来为**99%**。

(2)、3σ原则识别异常值

又称为拉依达法则。该法则就是先假设一组检测数据只含有随机误差，对原始数据进行计算处理得到标准差，然后按一定的概率确定一个区间，认为误差超过这个区间的就属于异常值，数据的数值分布几乎全部集中在区间(μ-3σ,μ+3σ)内，超出这个范围的数据仅占不到0.3%。故根据小概率原理，可以认为超出3σ的部分数据为异常数据

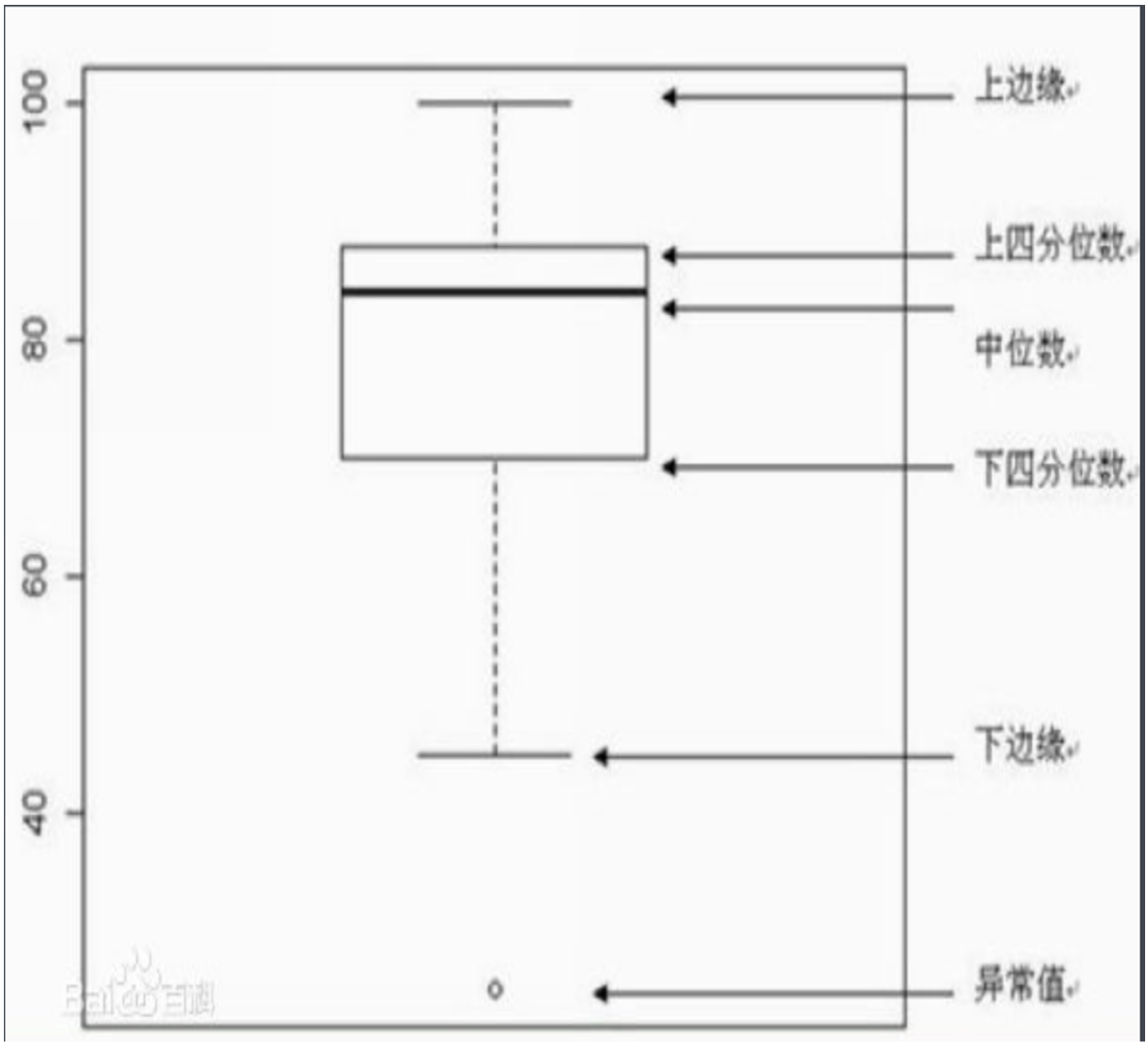
2. 箱线图

箱线图（boxplot）也称箱须图，其绘制需使用常用的统计量，能提供有关数据位置和分散情况的关键信息，尤其在比较不同特征时，更可表现其分散程度差异。

箱线图利用数据中的五个统计量（最小值、下四分位数、中位数、上四分位数和最大值）来描述数据，它也可以粗略地看出数据是否具有对称性、分布的分散程度等信息，特别可以用于对几个样本的比较。

(1)、检测异常值

箱型图提供了识别异常值的一个标准，即异常值通常被定义为小于QL-1.5IQR或大于QU+1.5IQR的值。
QL称为下四分位数，表示全部观察值中有四分之一的数据取值比它小。
QU称为上四分位数，表示全部观察值中有四分之一的数据取值比它大。
IQR称为四分位数间距，是上四分位数QU与下四分位数QL之差，其间包含了全部观察值的一半。



箱线图依据实际数据绘制，真实、直观地表现出了数据分布的本来面貌，且没有对数据做任何限制性要求，其判断异常值的标准以四分位数和四分位数间距为基础。

四分位数给出了数据分布的中心、散布和形状的某种指示，箱形图判断异常值的标准以四分位数和四分位距为基础，箱线图识别异常值的结果比较客观，因此在识别异常值方面具有一定的优越性。

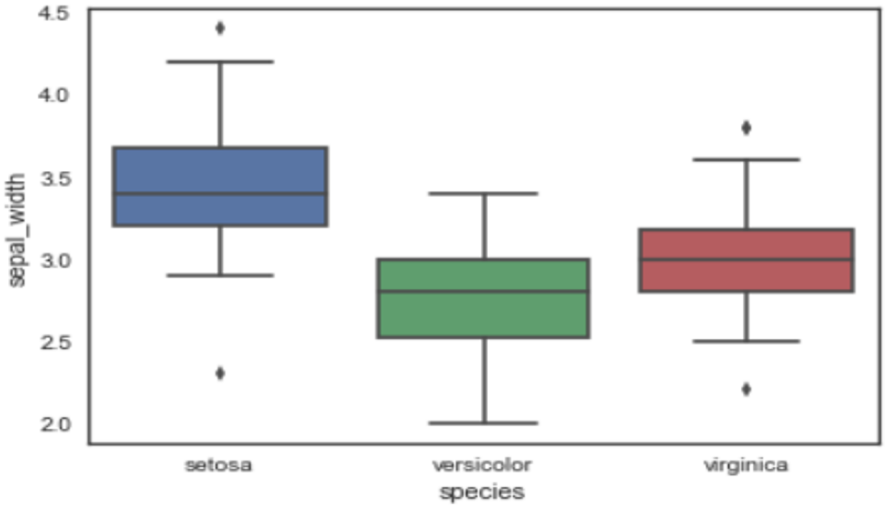
(2)、处理异常值

异常值处理方法	方法描述
删除含有异常值的记录	直接将含有异常值的记录删除
视为缺失值	将异常值视为缺失值，利用缺失值处理的方法进行处理
平均值修正	可用前后两个观测值的平均值修正该异常值
不处理	直接在具有异常值的数据集上进行挖掘建模

boxplot函数( )

```
sns.boxplot(x=df_iris['species'],y=df_iris['sepal_width'])
# plt.show()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e1a1f752b0>
```



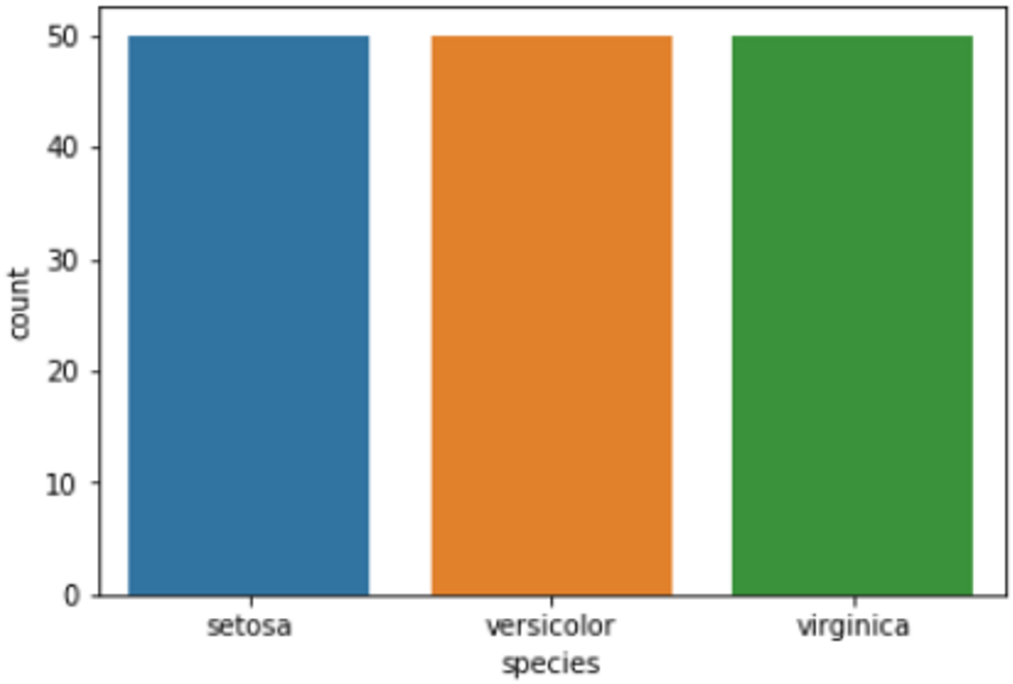

```
sns.boxplot(y=data['petal_length'])
sns.boxplot(x=data['species'],y=data['petal_length'],palette="Paired_r")
```

3. 计数图

countplot( )函数

```
sns.countplot(x=data['species'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1aca0b67198>
```



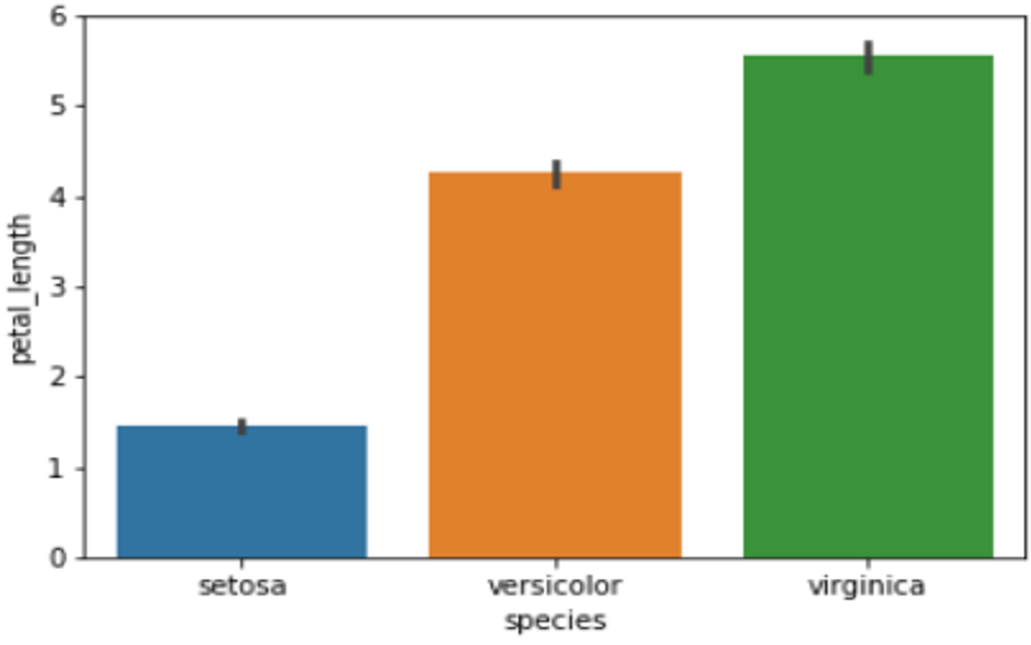

```
sns.countplot(x=data['species'])
```

4. 分组聚合图

barplot( )函数

```
sns.barplot(x=data['species'],y=data['petal_length'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2ca31f0ea20>
```




```
sns.barplot(x='species',y='petal_length',data=data,ci=None)
```

误差线源于统计学，表示数据误差(或不确定性)范围，以更准确的方式呈现数据。当label上有一组采样数据时，一般将这组数据的平均值作为该label上标注的值，而用误差线表示该均值可能的误差范围。误差线可以用标准差(standard deviation,SD)、标准误差(standard error,SE)和置信区间表示，使用时可选用任意一种表示方法并作相应说明即可。当label上值有一个数据时，则不需要标注误差线。

- 标准误差： 当多次进行重复采样时，会得到多组数据，每组数据都有一个平均值，这些平均值间是有差异的，尽管在每组数据量较大时，这个差异会比较小，标准误表示的就是平均值的误差范围
- 由于bar上标明的值是样本均值，这里实际上是对样本均值进行区间估计得到的置信区间

总结

distplot函数：传入单变量数值型数据，得到直方图和密度曲线图

boxplot函数：

1、传入单变量数值型数据，得到x或者y方向的数据分布/分散情况
2、传入两个变量，类别型+数值型，代表不同类别的数据的分散情况，还可以在类比型数据的基础上在输入一个类别型特征
countplot函数：

1、接收类别型数据，不同类别进行计数

2、还可以在类比型数据的基础上在输入一个类别型特征，二次计数

barplot函数：

1、类别型+数值型数据，表示按照不同类别进行分类汇总(默认平均值)

2、还可以在类别型基础上输入一个类别型特征。二次汇总

二：优衣库销售数据分析

优衣库（英文名称：UNIQLO，日文假名发音：ユニクロ），为日本迅销公司的核心品牌,建立于1984年，当年是一家销售西服的小服装店，现已成为国际知名服装品牌。通过独特的商品策划，开发和销售体系来实现店铺运作的低成本化，由此引发优衣库的热卖潮 优衣库的内在涵义是指通过摒弃了不必要装潢装饰的仓储型店铺，采用超市的自助式的自助购物方式，以合理可信的价格提供顾客希望的商品价廉物美的休闲装。

	store_id	city	channel	gender_group	age_group	wkd_ind	product	customer	revenue	order	quant	unit_cost	unit_price
0	658	深圳	线下	Female	25-29	Weekday	当季新品	4	796.0	4	4	59	199
1	146	杭州	线下	Female	25-29	Weekday	运动	1	149.0	1	1	49	149
2	70	深圳	线下	Male	>=60	Weekday	T恤	2	178.0	2	2	49	89
3	658	深圳	线下	Female	25-29	Weekday	T恤	1	59.0	1	1	49	59
4	229	深圳	线下	Male	20-24	Weekend	袜子	2	65.0	2	3	9	22
5	28	武汉	线上	Female	35-39	Weekend	T恤	1	97.0	1	1	49	97
6	649	杭州	线下	Female	25-29	Weekend	短裤	1	33.0	1	1	19	33
7	520	杭州	线下	Male	>=60	Weekend	T恤	2	158.0	2	2	49	79
8	649	杭州	线下	Female	30-34	Weekend	牛仔裤	3	157.0	3	3	69	52
9	21	北京	线下	Female	45-49	Weekend	毛衣	1	199.0	1	1	99	199

store_id 门店随机编号id,无实际意义

city 门店所在城市

channel 门店所产生的销售渠道，

gender_group 客户性别 男女

age_group 客户年龄段

wkd_ind 购买发生的时间（周末，周中）

product 产品类别

customer 客户数量

revenue 销售金额

order 订单数量

quant 购买的产品数量

Unit_cost：单件成本

Unit_price：单件销售

1、优衣库数据读取

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
# 显示中文
plt.rcParams['font.sans-serif']='SimHei'
plt.rcParams['axes.unicode_minus']=False
import warnings
warnings.filterwarnings('ignore')
# 读取数据
data=pd.read_csv('data/unique.csv')
data.head()
```

2、优衣库数据描述性分析

```
# 数据完整性分析，数值型数据分布情况
data.info()
data.describe()
data.select_dtypes('object').describe().T
data['product'].value_counts().plot(kind='barh')
sns.countplot(y=data['product'])
data['wkd_ind'].value_counts()
data['channel'].value_counts()
data['city'].value_counts()
data['gender_group'].value_counts()
data['age_group'].value_counts()
```

3、数据一致性处理

```
# 得到数据中营业额大于零的数据
data_sales=data[data['revenue']>0]
data_sales.head()
# 得到新的一列，产品利润
```

```
data_sales['margin']=(data_sales['revenue']/data_sales['quant'])-
data_sales['unit_cost']
data_sales['margin']=np.round(data_sales['margin'],2)
data_sales.head()
# 没有统计详细信息的性别删除
data_sales=data_sales[data_sales['gender_group'].
                      isin(['Female','Male'])]
data_sales['gender_group'].value_counts()
# 没有统计详细信息的年龄删除
data_sales['age_group'].value_counts()
data_sales=data_sales[~data_sales['age_group']
                      .isin(['Unkown'])]
data_sales['age_group'].value_counts()
data_sales.info()
```

4、业务一：整体销售情况随着时间的变化是怎样的？

题目拆解： 数据中与时间有关的字段仅为类别变量wkd_ind代表的Weekday和Weekend，即购买发生的时间是周中还是周末。

分析内容：对比周末和周中与销售有关的数据，包括产品销售数量quant、销售金额revenue、顾客人数customer的情况，可生成柱状图进行可视化

```
# 分析营业额总和
sns.barplot(x='wkd_ind',y='revenue',data=data_sales,
            estimator=sum,ci=None)
# 分析销量的平均值
sns.barplot(x='wkd_ind',y='quant',data=data_sales)
# 分析顾客的平均值
sns.barplot(x='wkd_ind',y='customer',data=data_sales)
#从销售额和销售数量，顾客数量几个维度来看，优衣库在工作日的整体销售情况比非工作日要好
# 利润比
data_sales.groupby('wkd_ind').margin.sum()
(383100.32/2)/(463584.98/5)
# 销售额比
data_sales.groupby('wkd_ind').revenue.sum()
(1457653.87/2)/(2086397.78/5)
# 销量比
data_sales.groupby('wkd_ind').quant.sum()
(16798/2)/(24433/5)
#虽然总销售额来看工作日高于周末，但是周末其他指标都比工作日要好，从日均销售和客流来看，周末是工作日的2倍，所以还是需要加强周末的销售工作，多搞活动。
```

5、业务二：不同产品的销售情况是怎样的？

题目拆解：不同产品即指product字段中不同类别的产品，销售情况即为销售额和利润，可生成柱状图进行可视化

```
# 分析不同产品营业额
data_sales.groupby('product')['revenue'].describe()
sort_order=data_sales.groupby('product').revenue.sum().sort_values(ascending=Fal
se).index
sort_order
sns.barplot(x='product',y='revenue',estimator=sum,
            data=data_sales,order=sort_order)
```

```
# 从整个销售金额来看，T恤，当季新品，销售额较高，销售重心要放在这里
sort_margin=data_sales.groupby('product').margin.sum().sort_values(ascending=Fal
se).index
sort_margin
# 分析不同产品的总利润
sns.barplot(x='product',y='margin',estimator=sum,
            data=data_sales,order=sort_margin)
# 分析不同产品的平均利润
sns.barplot(x='product',y='margin',data=data_sales)
#毛衣之所以平均利润高，是因为销量不高，T恤平均利润低是因为他是薄利多销的产品
```

6、业务三：顾客偏爱哪一种购买方式？

题目拆解：购买方式只有channel是线上还是线下这一个指标，而顾客可以从不同性别gender_group、年龄段age_group、城市city，产品product四个维度进行分解，
因此这个问题即为探究不同性别、年龄段和城市的顾客以及不同产品对线上、线下两种购买方式的偏好，可生成柱状图进行可视化的呈现

```
# 按照不同性别分析购买偏好
sns.countplot(y='gender_group',hue='channel',
              data=data_sales)
# 从不同产品分析顾客偏好
sns.countplot(y='product',hue='channel',
              data=data_sales)
# 从不同的年龄段分析顾客喜好
data_sales['age_group'].value_counts().index
sns.countplot(y='age_group',hue='channel',data=data_sales,
              order=data_sales['age_group'].value_counts().index)
# 看出不管是哪个年龄段的人群，都是以线下购买为主，且顾客年龄集中在20-40岁之间，建议优衣库服装设计应偏向年轻群体，后期应该加大对于线上商城的搭建和运营
# 从不同的城市分析顾客喜好
sns.countplot(y='city',hue='channel',data=data_sales,
              order=data_sales['city'].value_counts().index)
#可以看出各个城市依然是以线下为主，但是个别城市线上购买量超过线下，比如广州。优衣库线上业务在深圳，杭州，成都，北京，南京，这几个城市还有很大的拓展及上升空间，可以效仿广州和武汉的线上业务营销推广及运营方式
```

7、业务四：销售额和产品成本之间的关系怎么样？

题目拆解：

思路一：margin是如何分布的？是否存在亏本销售的产品？

思路二：探究实际销售额和产品成本之间的关系，即为求它们之间的相关，若成正相关，则产品成本越高，销售额越高，或许为高端商品；若成负相关，则成本越高，销售额越低，为薄利多销的模式。

还可以拆分得更细，探究不同城市和门店中成本和销售额的相关性

```
# 整组数据的利润分布情况
# 直方图绘制
sns.distplot(data_sales['margin'])
# 对于优衣库产品利润直方图来看，整个利润跨度较大，有存在亏本销售的产品，也存在利润超过100的产品，整体10-50的较多。说明优衣库大部分产品是薄利多销的
sns.boxplot(y=data_sales['margin'])
data_sales['margin'].describe()
```

```
# 不同产品的利润情况分析
sns.boxplot(x='margin',y='product',data=data_sales)
#牛仔裤最有可能是亏本销售产品，部分的毛衣和T恤也存在亏本销售
#T恤的盈利波动比较大，-50-200元
#裙子和配件是盈利比较高的两类商品
# 将利润这一列的数据进行离散化，变为类别型
# 划分离散点 相邻的两个数据组成一个区间
bins=[-100,-50,0,50,100,150,200,250,300]
# 把利润这一列数据的每个值划分到一个区间
data_sales['margin_level']=pd.cut(x=data_sales.margin,
                                  bins=bins)
# data_sales.head()
data_sales['margin_level'].value_counts()
sns.countplot(y=data_sales['margin_level'])
# 分组聚合
data_margin=data_sales.groupby(['product','margin_level'])
['store_id'].count().reset_index()
data_margin.to_csv('data_margin.csv')
# 不同产品的利润分布条形图或者柱状图
plt.figure(figsize=(10,8))
sns.countplot(x='product',hue='margin_level',data=data_sales)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
plt.xlabel('product',fontsize=18)
plt.ylabel('count',fontsize=18)
plt.ylim((0,2000))
# 分析不同城市的利润分布
plt.figure(figsize=(8,6))
sns.boxplot(x='margin',y='city',data=data_sales)
# 分析单件销售额和单件成本之间的关系 相关系数矩阵
data_sales[['unit_price','unit_cost']].corr()
sns.heatmap(data_sales[['unit_price','unit_cost']].corr(),annot=True,vmin=0,vmax
=1)
# 探究不同城市和门店中成本和销售额的相关性
data_sales.groupby('city')[['unit_price','unit_cost']].corr()
```