棕榈学院

10 天 Python 训练营讲义

第七讲

2018 年 10 月 22 日-2018 年 10 月 31 日

目录

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

注：

① matplotlib.pyplot 是一个用于画图的包。

② seaborn 也是一个用于画图的包，它能使画出来的图更加美观，且它有更多功能，生成的图比 matplotlib.pyplot 有更复杂。

开始画图

基本格式：

```
plt.figure(figsize=(10,5))
plot ...
(add labels, adjust axes, legend etc.)
plt.show()
```

注：

① figsize 用于定义图片的大小。10 指的是图的宽度，5 指的是图的高度。

② plt.show()表示显示图片。

%matplotlib inline

注：在最前面运行这行代码，之后不打 plot.show()也能把图片显示出来。

一、 线型图

例 1：

x = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

y = [102,134,154,122,143,243,355,342,276,299,241,287,260,231,100]

plt.figure(figsize=(10,5))

plt.plot(x,y)

plt.title('Weight change in 15 months')

plt.xlabel('Month')

plt.ylabel('kg')

plt.show()

注：

① plt.plot()默认画一个线型图。

② plt.title()表示给这个图添加一个名字。

③ plt.xlabel()和 plt.ylabe()表示对 x 轴和 y 轴标注。

例 2：

```
y1 = [102,134,154,122,143,243,355,342,276,299,241,287,260,231,100]
y2 = [244,250,245,256,234,241,230,267,266,255,248,239,233,221,227]

plt.figure(figsize=(10,5))
plt.plot(x,y1,label = 'A')
plt.plot(x,y2,label = 'B')
plt.title('Weight change in 15 months')
plt.xlabel('Month')
plt.ylabel('kg')
plt.legend(fontsize = 10)
plt.show()
```

注：

① 当我们需要在一张图上展示两个数据时，直接在第一个数据 plt.plot(x,y1,label = 'A')后，输入第二个数据 plt.plot(x,y2,label = 'B')，且它会自动选择不同颜色。

② plt.legend()可以用于给图片添加标签，标注不同的线表示的数据。第一种方式可以在括号内直接输入 A,B，这里填入 A,B 的顺序一定要和前面放数据的顺序一致。还可以在括号内用 fontsize 设置标签的大小，还可以用 loc 设置标签所在的位置，1 表示右上角，2 表示左上角，3 表示左下角，4 表示右下角。当我们要画很多线在同一个图上面时，为了避免出错，也可以在 plt.plot 内用 label 来设置标注的名字。

二、 柱状图

例 1：

```
plt.figure(figsize=(10,5))
plt.bar(x,y,color = '#9999ff',width = 0.5)
plt.title('Weight change in 15 months')
plt.xlabel('Month')
plt.ylabel('kg')
plt.show()
```

注：

① 使用 plt.bar()来绘制柱状图。

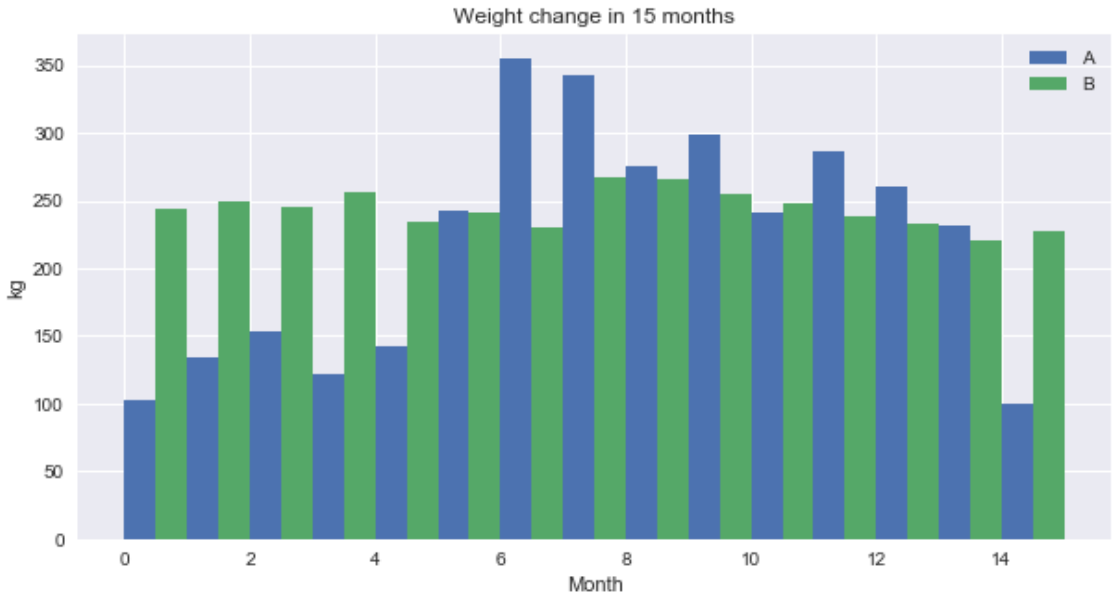
② 当我们不用关注趋势的变化，而关注每一个点量的高低时使用柱状图会更加直观。或者我们要比较两个数据在同一个时间点量上的变化时，使用柱状图也更多。

③ 在 plt.bar 括号内可以通过 color 设置颜色，通过 width 设置每根柱子的宽度。大家如果想要更改其他颜色的话，可以去网上搜索 Python 内颜色的色卡。

例 2：

```
x1 = [0.25,1.25,2.25,3.25,4.25,5.25,6.25,7.25,8.25,9.25,10.25,11.25,12.25,13.25,14.25]
x2 = [0.75,1.75,2.75,3.75,4.75,5.75,6.75,7.75,8.75,9.75,10.75,11.75,12.75,13.75,14.75]

plt.figure(figsize=(10,5))
plt.bar(x1,y1,width = 0.5,label = 'A')
plt.bar(x2,y2,width = 0.5,label = 'B')
plt.title('Weight change in 15 months')
plt.xlabel('Month')
plt.ylabel('kg')
plt.legend()
plt.show()
```

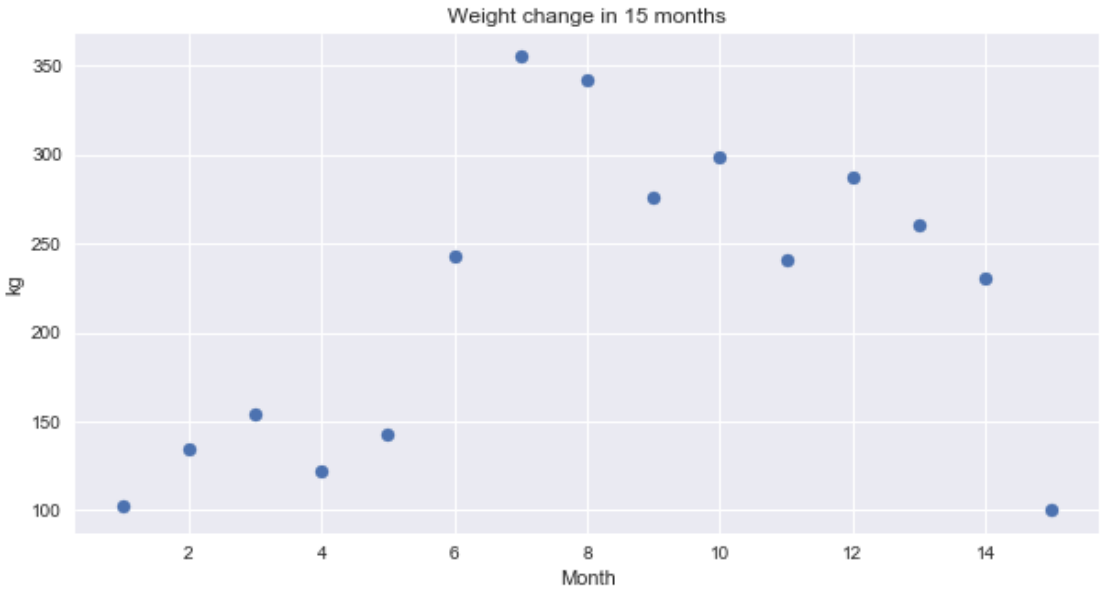


注：当我们要展示多个数据在一个柱状图上时，如果直接运用画线型图的方法，我们会发现 y 直接覆盖在了 x 上，所以这个时候我们可以调整两个数据 x 轴坐标，我们可以把 x1 的坐标调整成从 0-1 直接的前 0.25 开始画，把 x2 调整成从前 0.75 的位置开始画，这里的 0.25 与 0.75 表示的是这个柱子的中心点在哪里。同时我们也要利用 width 调整一下每根柱子的宽度。

三、 点状图

例 1：

```
plt.figure(figsize=(10,5))
plt.scatter(x,y)
plt.title('Weight change in 15 months')
plt.xlabel('Month')
plt.ylabel('kg')
plt.show()
```



注：

① 使用 plt.scatter()来画点状图。

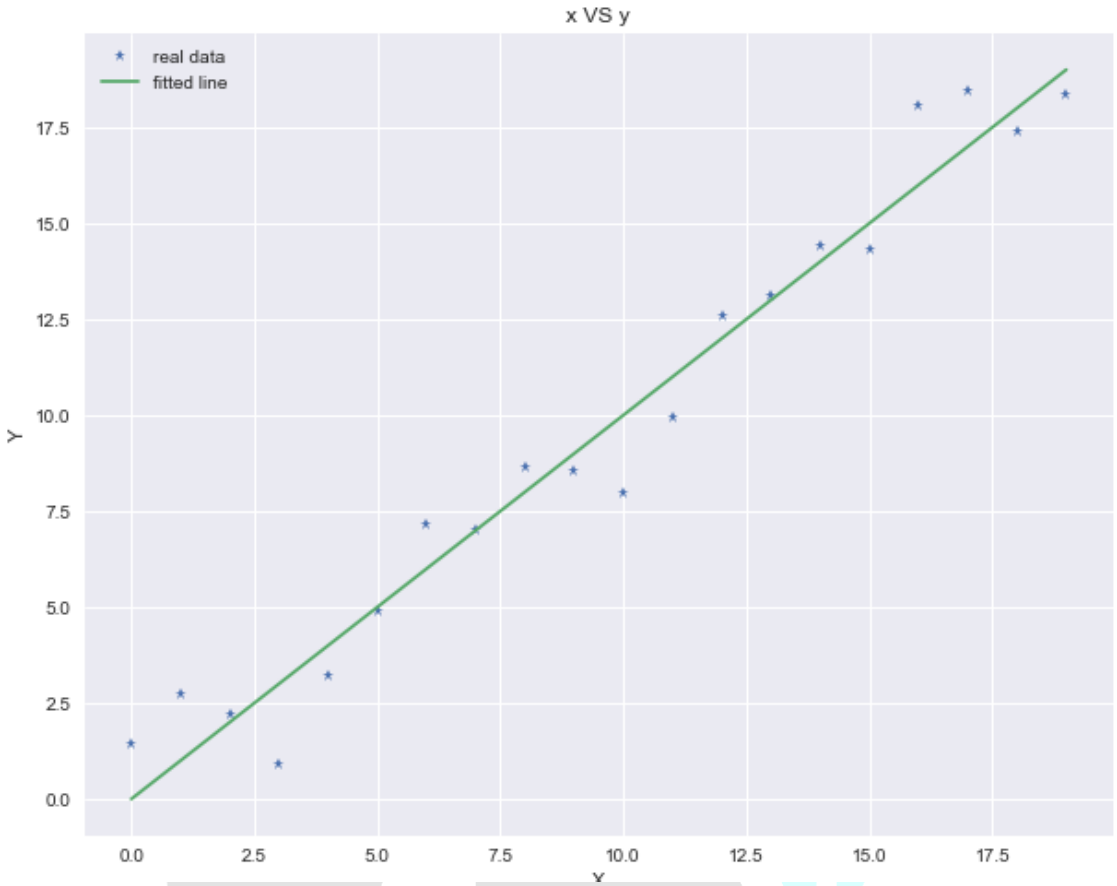
② 当我们使用点状图时，主要看 x 与 y 之间有没有什么关系。

例 2：

```
x = range(20)
y = x + np.random.randn(20)*1.05

plt.figure(figsize=(10,8))
# plt.scatter(x,y)
plt.plot(x,y,'*')
plt.plot(x,x)
```

```
plt.title('x VS y')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend(('real data','fitted line'))
plt.show()
```



注:

① range()会返回一串数，从 0 开始一直到你在括号内填入的数字。

② np.random.randn()是指会返回一个或一串正态分布的随机数。

③ 设置一个 y，与 x 存在正相关。

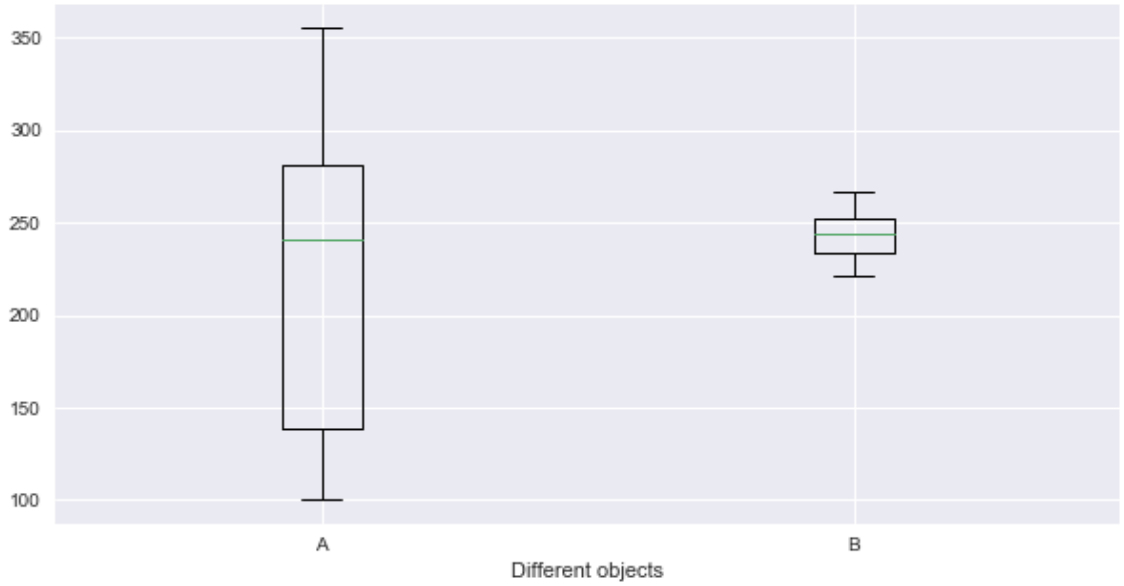
④ 将点状图与线型图相结合，更好地看到 x 与 y 之间的关系。

⑤ 两种画点状图的方式，第一种是# plt.scatter(x,y)，第二种是 plt.plot(x,y,'.')，在这里我们还可以在引号内填入别的字符串，就可以已输入的字符串为样子，表示每一个点。

四、 盒状图

例：

```
plt.figure(figsize=(10,5))
plt.boxplot([y1,y2])
plt.xticks([1,2],['A','B'])
plt.xlabel('Different objects')
plt.show()
```



注：

① 使用 plt.boxplot()来画盒状图。

② 盒状图主要看一组数据的集中值，以及有哪些异常数字。

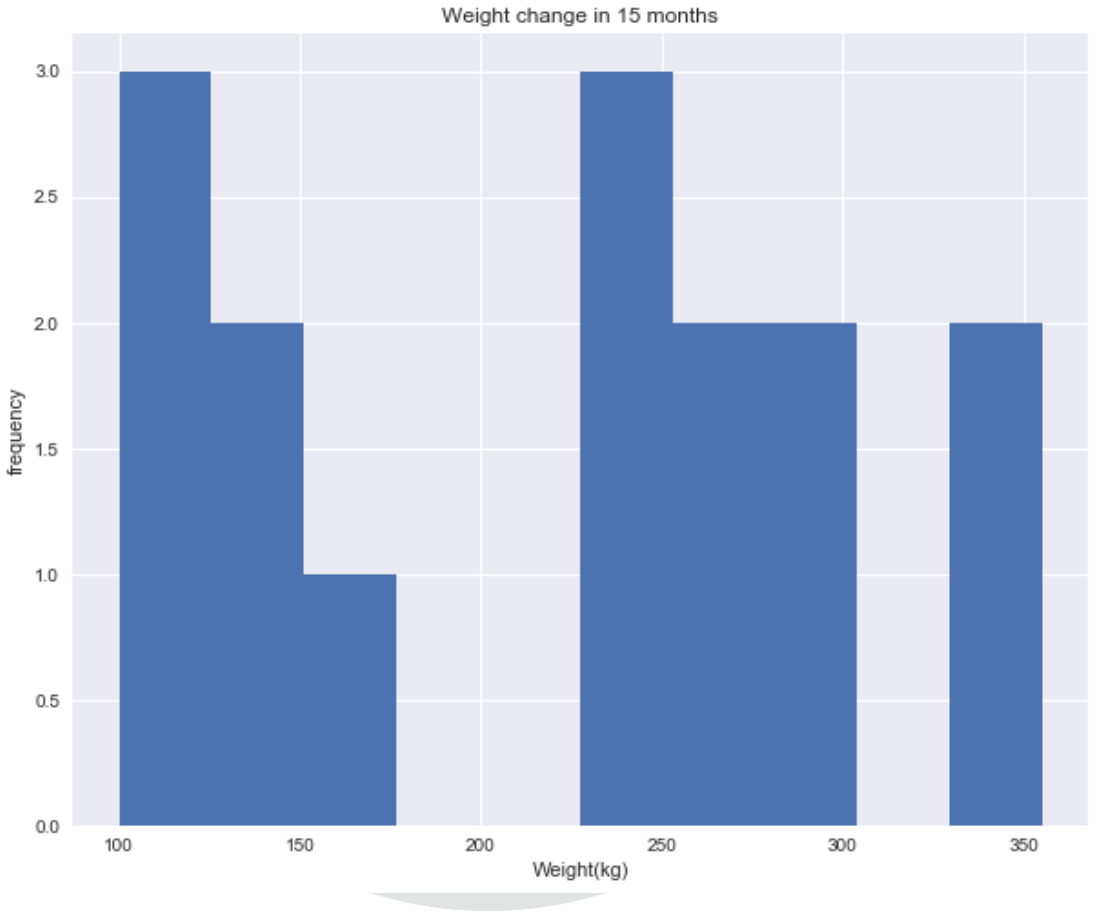
③ 当我们需要比较两组数据之间的分布时，只要同时把两组数据都填入 boxplot 里面

④ 使用 plt.xticks()来更改 x 轴上的输入值，它是一个有替换功能的函数，把原来横坐标上的 1 和 2 替换成想要输入的字。逗号的左边填入原来的值，逗号的右边填入想要输入的值。

五、 直方图

例 1：

```
plt.figure(figsize=(10,8))
plt.hist(y1)
plt.title('Weight change in 15 months')
plt.xlabel('Weight(kg)')
plt.ylabel('frequency')
plt.show()
```



注：

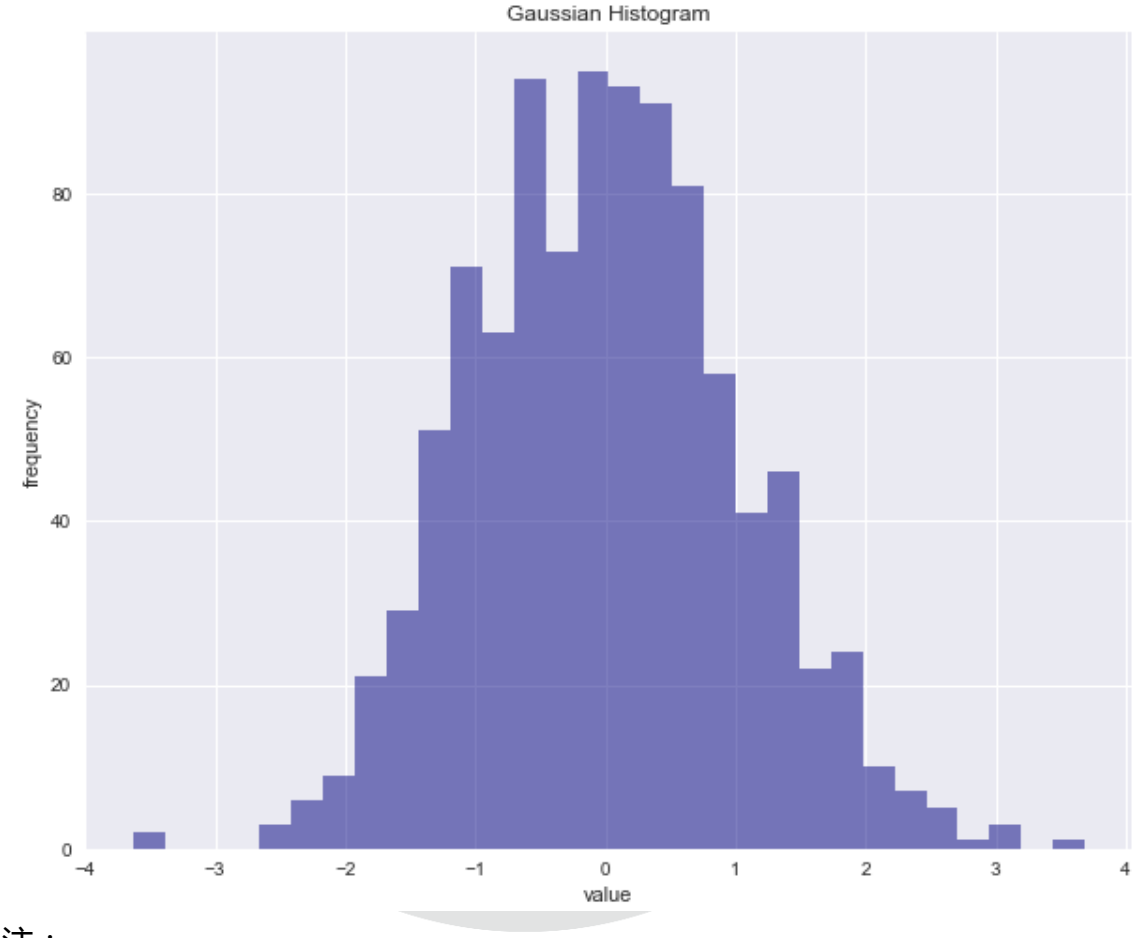
① 使用 plt.hist()来绘制直方图。

② 直方图用于研究单一数值分布变化。

③ 这里 y1 体现在 x 轴的坐标上，y 轴表示的是这个数字出现的频率，这里指的的每个区间内数据出现了多少次。

例 2：

```
plt.figure(figsize=(10,8))
gaussian_numbers = np.random.randn(1000)
plt.hist(gaussian_numbers, 30 ,color = 'navy',alpha = 0.5)
plt.title('Gaussian Histogram')
plt.xlabel('value')
plt.ylabel('frequency')
plt.show()
```



注：

① 这里利用 np.random.randn(1000)随机输入了 1000 个正态分布的随机数。

② plt.hist(gaussian_numbers, 30 ,color = 'navy',alpha = 0.5)这里的 30 表示要画多少个柱子的意思。也可以直接告诉 Python 你要用的颜色是什么（用于常见的颜色），alpha 调节颜色的色彩饱和度。

总结

	matplotlib
Line chart:	plt.plot(x , y)
Bar chart:	plt.bar(x , y)
Scatter plot:	plt.scatter(x , y)
Boxplot:	plt.boxplot([data,..])
Histogram:	plt.hist(data)

① 如果要使用多个数据叠加，我们可以多打几个代码，同时可以将多种图形组合起来。

② 盒状图和直方图都是用来研究单一数据分布的，如果要把多组数据画在盒状图中，要用列表把它们框起来。

今日作业：

用自己的数据运行今天学习的代码并画图。

第六讲作业答案

Q1.

```
d1 = pd.DataFrame([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]],columns = ['A','B','C','D'])
d1['E'] = ['H','J','K','L']
d2 = pd.DataFrame({'1':['H','L','J','J'],'2':['S','S','S','T'],'3':[6,2,5,3]})

d3_1 = pd.merge(d1,d2,left_on = d1['E'], right_on = d2['1'], how = 'outer')
```

d3_1

	key_0	A	B	C	D	E	1	2	3
0	H	1	2	3	4	H	H	S	6.0
1	J	5	6	7	8	J	J	S	5.0
2	J	5	6	7	8	J	J	T	3.0
3	K	9	10	11	12	K	NaN	NaN	NaN
4	L	13	14	15	16	L	L	S	2.0

注：

① 关于这到题的'key_0'是怎么出来的，主要是因为我在 left_on 还有 right_on 的参数上指定了 dataframe. 如果两个 dataframe 需要被合并，并且两个

dataframe 的 column name 有重复的名字但是正好不是要被合并的列的时候可以这么做。不过这道题并没有这个问题，所以同学们做的也都是对哒~~~👆

② 还有就是有些同学用数字做了某些列的名字，然后发现在切片的时候有可能遇到问题。原因是 pandas 的 dataframe 有自己默认的 index,列也是。这个的应用就是 iloc，所以如果我们自定义列的名字的时候，最好不要用数字而是用字符串。

Q2.

```
d2.index = d2['1']
d1.index = d1['E']
d3_2 = d1.join(d2)
d3 = d3_2.dropna(how = 'any')
d3.index = ['01','02','03','04']
```

Q3.

```
L = []
for i in d3.index:
    if d3.loc[i,'2'] == 'S':
        p = d3.loc[i,'B'] + d3.loc[i,'C'] - d3.loc[i,'3']
    if d3.loc[i,'2'] == 'T':
        p = d3.loc[i,'B'] - d3.loc[i,'C'] + d3.loc[i,'3']
    L.append(pd.DataFrame({'Type': d3.loc[i,'2'], 'Value': p}, index = [0]))
result = pd.concat(L, ignore_index = True)

result
```

	Type	Value
0	S	-1.0
1	S	8.0
2	T	2.0
3	S	27.0