



文本分析

第6讲

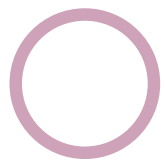
李田雨





给数据分个班





给数据分个班



上一节我们讲解了如何对数据的合并、筛选和数据的排序的技能，大家要记得经常的复习和练习呦！

俗话说：“人与类聚，物以群分”，这一节我们将讲解数据的分组以及分组后统计。Pandas 的分组相对于 Excel 会更加简单和灵活。



本节知识点

数据的分组

对分组进行遍历

分组后统计






数据的分组



本节我们将以福布斯**2018**年度亿万富翁数据为实验数据，探索数据分组的奥秘，运行下面的代码，来了解一下数据的基本情况：



```
In [1]: import pandas as pd
df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
print(df.head())
print(df.shape)
```


	name	lastName	age	country	gender	wealthSource
0	Jeff Bezos	Bezos	54	United States	M	Amazon
1	Bill Gates	Gates	62	United States	M	Microsoft
2	Warren Buffett	Buffett	87	United States	M	Berkshire Hathaway
3	Bernard Arnault	Arnault	69	France	M	LVMH
4	Mark Zuckerberg	Zuckerberg	34	United States	M	Facebook

(2031, 6)

数据详情：name-名字、lastName-姓、age-年龄、country-国家、gender-性别、wealthSource-财富来源。

根据结果我们了解到，共有**2031**条数据，那么在这些富翁中男女比例是多少呢？

要解决这个问题，我们最好的办法就是根据性别分成男女两组，然后分别计算他们的人数，从而计算他们的占比。



Pandas提供了一个灵活高效的groupby功能，它使你能以一种自然的方式对数据集进行切片、切块、摘要等操作。我们一起看下如何使用groupby()方法根据性别将富翁们进行分组，运行下方代码，查看结果。



```
In [1]: import pandas as pd
df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
# 根据gender列进行分组
groups = df.groupby('gender')
print(groups)

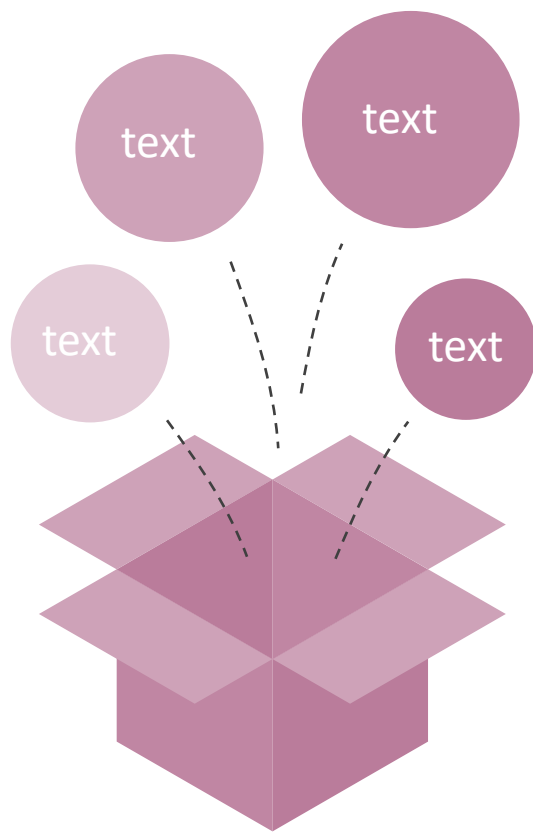
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7f02d007c990>
```

根据结果可以发现，分组后的结果为DataFrameGroupBy object，是一个分组后的对象。



01

数据的分组



用groupby的size方法可以查看分组后每组的数量，并返回一个含有分组大小的Series：

代码片段`print(groups.size())`

```
gender
F      221
M     1810
dtype:int64
```


根据上面的方法，你是否已经有了如何获取男女的占比是多少的思路呢？别犹豫，在下面的代码框中一试便知。

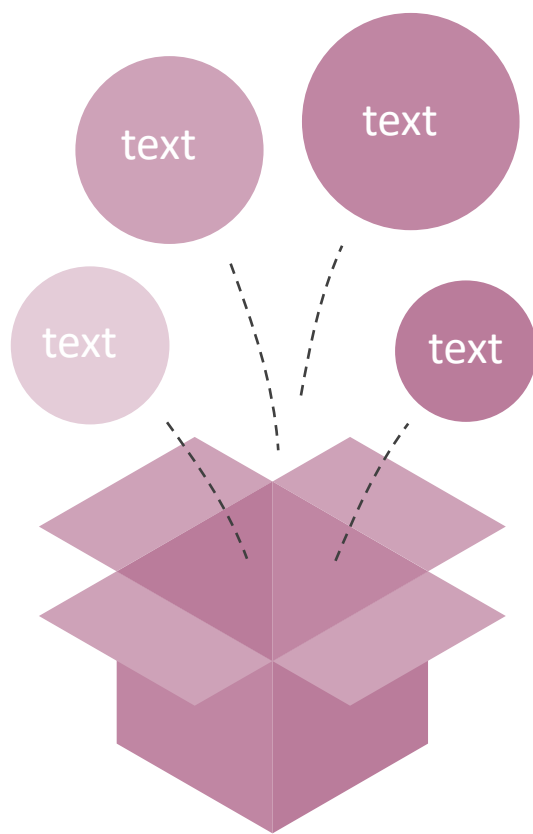
```
In [4]: 1 import pandas as pd
2 df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
3 # 请根据你的思路用代码在下面实现

In [6]: 1 import pandas as pd
2 df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
3 group = df.groupby('gender')
4 for gender, value in group.size().items():
5     # 计算每组的占比
6     accounted = value/df.shape[0]
7     # 将小数转化成百分数
8     bb = "%.2f%" % (accounted * 100)
9     print('福布斯2018年度亿万富翁中{}共{}位，占比是{}'.format(gender, value, bb))
```

```
福布斯2018年度亿万富翁中F共221位，占比是10.88%
福布斯2018年度亿万富翁中M共1810位，占比是89.12%
```

01

数据的分组



`df.groupby('gender')`是根据`gender`列对整个数据进行分组，同样我们也可以只对一列数据进行分组，只保留我们需要的列数据。

例如：我们通过性别`gender`，只对`age`列数据进行分组。

代码片段

```
1 group = df['age'].groupby(df['gender'])
2 # 查看分组
3 print(group.groups)
4 # 根据分组后的名字选择分组
5 print(group.get_group('F'))
```



```
In [1]: 1 import pandas as pd
2 df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
3 group = df.groupby('gender')
4 for gender, value in group.size().items():
5     # 计算每组的占比
6     accounted = value/df.shape[0]
7     # 将小数转化成百分数
8     bb = "%.2f%%" % (accounted * 100)
9     print('福布斯2018年度亿万富翁中{}共{}位, 占比是{}'.format(gender, value, bb))
```

福布斯2018年度亿万富翁中F共221位, 占比是10.88%
福布斯2018年度亿万富翁中M共1810位, 占比是89.12%

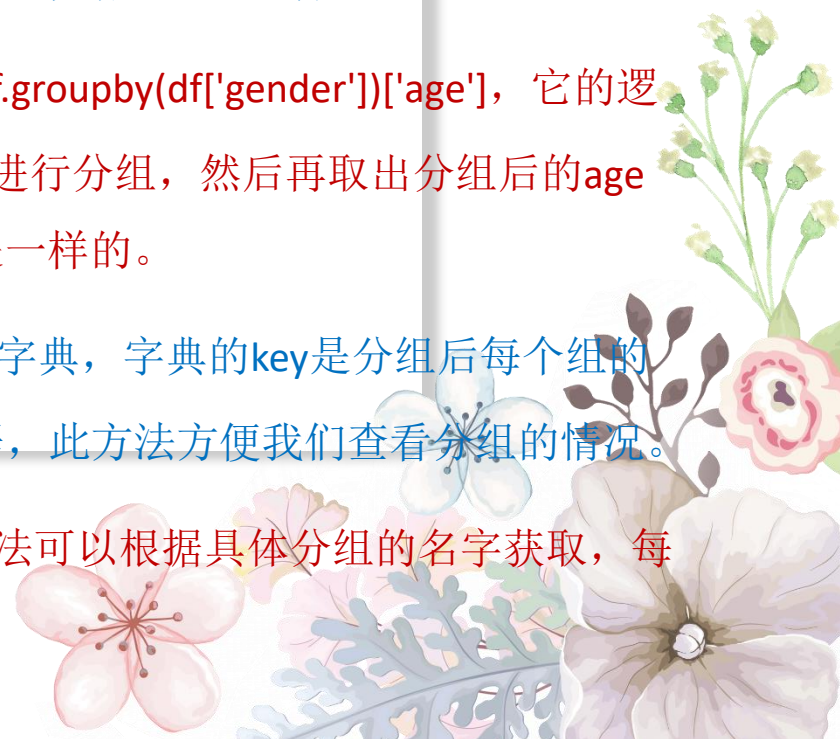
01.将上面代码补全复制到左面的代码框，运行查看结果：

02.代码`df['age'].groupby(df['gender'])`的逻辑是：取出`df`中`age`列数据，并且对该列数据根据`df['gender']`列数据进行分组操作。

03.上一步的代码也可改写成`df.groupby(df['gender'])['age']`，它的逻辑是：将`df`数据通过`df['gender']`进行分组，然后再取出分组后的`age`列数据。两种写法达到的效果是一样的。

04. `group.groups`的结果是一个字典，字典的`key`是分组后每个组的名字，对应的值是分组后的数据，此方法方便我们查看分组的情况。

05. `group.get_group('F')`这个方法可以根据具体分组的名字获取，每个组的数据。





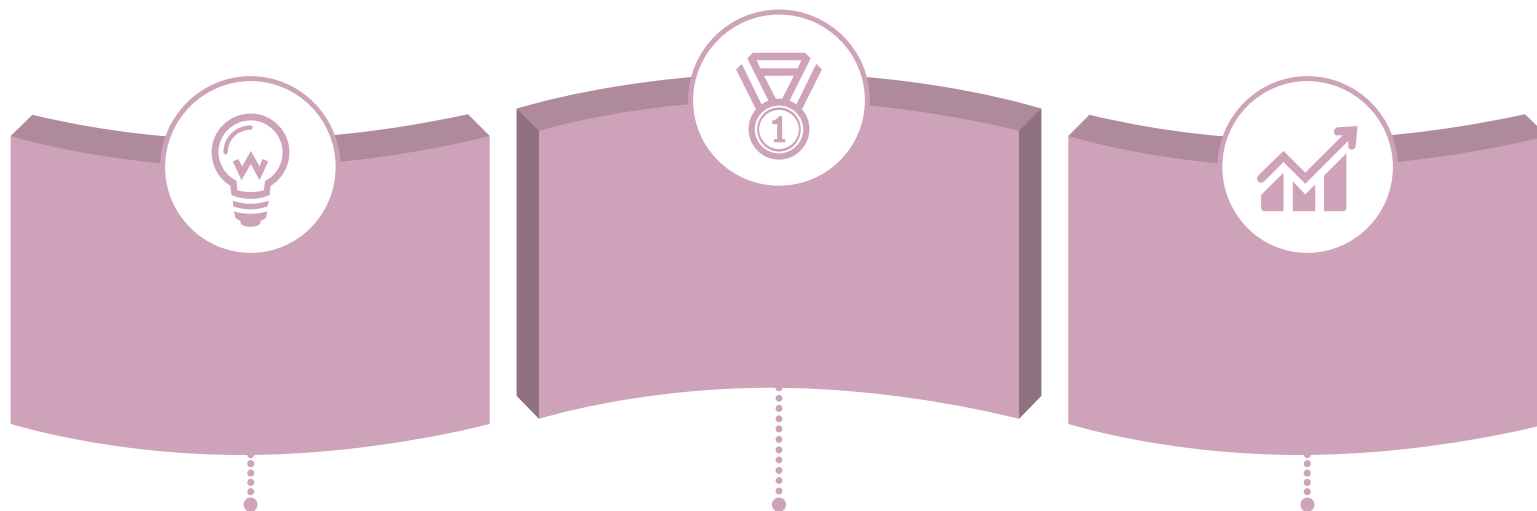
二

对分组进行遍历



02

对分组进行遍历



上面我们通过`groupby()`和`size()`两个方法以及以前所学的一些技能计算出了富豪的男女占比。

如果我们还想要分别查看富豪中男、女的最大年纪，最小年纪以及平均年龄，看看我们是不是还有机会成为他们中的一员。

`groups.get_group('F')`可以获取分组后某一个组的数据，'F'为组的名字，这样我们就可以对某一个组进行处理。

02

对分组进行遍历

```
In [1]: import pandas as pd
df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
groups = df.groupby('gender')
# 获取F组的数据
f_group = groups.get_group('F')
# 获取平均值
f_mean = f_group['age'].mean()
# 获取最大值
f_max = f_group['age'].max()
# 获取最小值
f_min = f_group['age'].min()
print(f_mean, f_max, f_min)
```

```
60.470588235294116 94 21
```

下面的代码实现了获取 'F' 组的最大年纪，最小年纪以及平均年龄，运行代码并观察结果。

代码中我们使用 `get_group()` 获取了 F 组的数据，并使用 `mean()`、`max()`、`min()` 等统计函数快速获取我们的指标值。



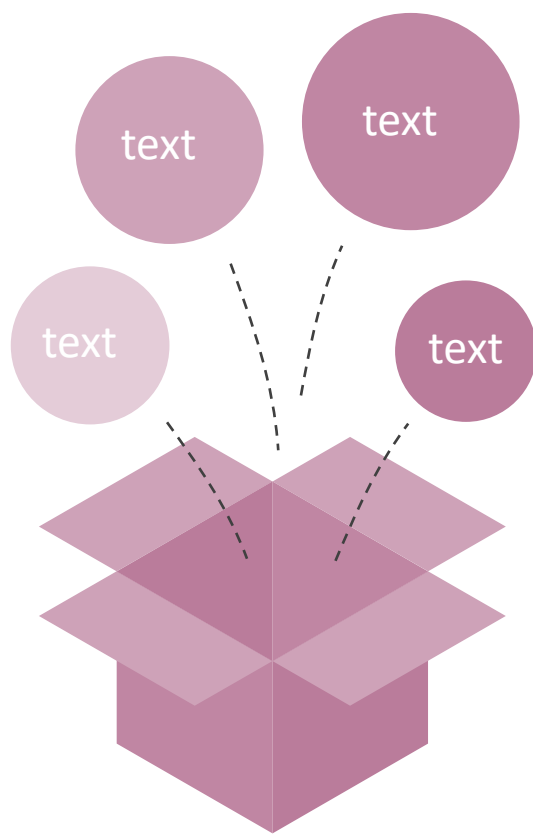
Pandas常用统计函数

函数	意义
count():	统计列表中非空数据的个数
nunique():	统计非重复的数据的个数
sum():	统计列表中所有数值的和
mean():	计算列表中数据的平均值
median():	统计列表中数据的中位数
max():	求列表中数据的最大值
min():	求列表中数据的最小值



02

对分组进行遍历



上面的代码成功的计算出了我们想要的数
据，我们也可以遍历分组后的数据，并获取他们的最大
年纪，最小年纪以及平均年龄。

运行下面的代码，看一下如何遍历分组后的数据。

```
1 import pandas as pd
2 df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
3 groups = df.groupby('gender')
4 for group_name, group_df in groups:
5     print(group_name, group_df.shape)
```

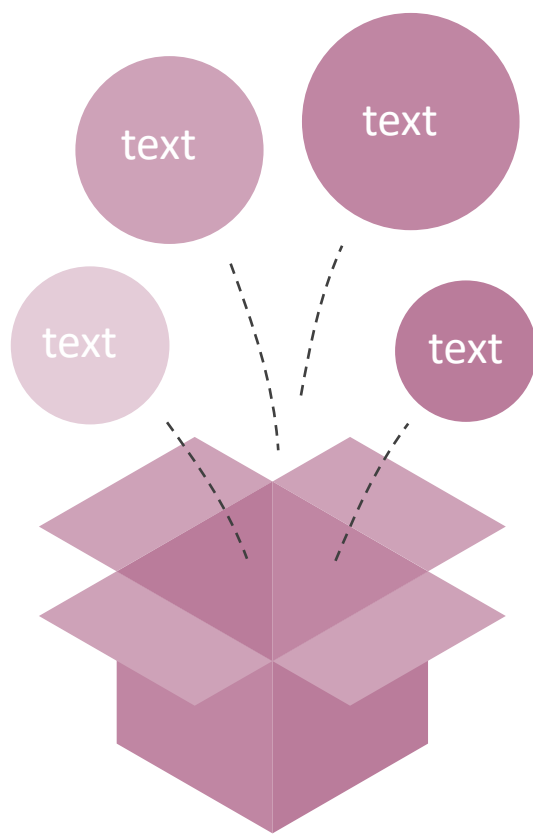
```
F (221, 6)
M (1810, 6)
```

```
1
```

上面代码中的将分组后的对象groups进行遍历，可
以获取到group_name每个组的名字，group_df每个
组的数据。

02

对分组进行遍历



接下来我们自己在下面代码框中练习使用遍历的方法，计算出每一组中的最大年纪，最小年纪以及平均年龄。

```
1 import pandas as pd
2 df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
3 groups = df.groupby('gender')
4 for group_name, group_df in groups:
5     f_mean = group_df['age'].mean()
6     f_max = group_df['age'].max()
7     f_min = group_df['age'].min()
8     print("{}组的最大年龄是{}, 最小年龄是{}, 平均年龄是{}".format(group_name, f_max, f_min, f_mean))
```

F组的最大年龄是94，最小年龄是21，平均年龄是60.470588235294116
M组的最大年龄是99，最小年龄是25，平均年龄是64.32099447513812



三

按多列进行分组



03

按多列进行分组

```
1 import pandas as pd
2 df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
3 group=df.groupby(['country','gender'])
4 df1 = group.size()
5 print(df1)
```

country	gender	
Algeria	M	1
Angola	F	1
Argentina	M	5
Australia	F	9
	M	31
		...
United States	M	498
Venezuela	M	2
Vietnam	F	1
	M	3
Zimbabwe	M	1

Length: 103, dtype: int64

运行上面的代码，
看下**groupby()**是
如何进行多列分
组的：

按照上面的
分析，难道
我们要写两
次**groupby**的
分组操作？
NO，我们强
大的
groupby() 方
法是支持按
照多列进行
分组。



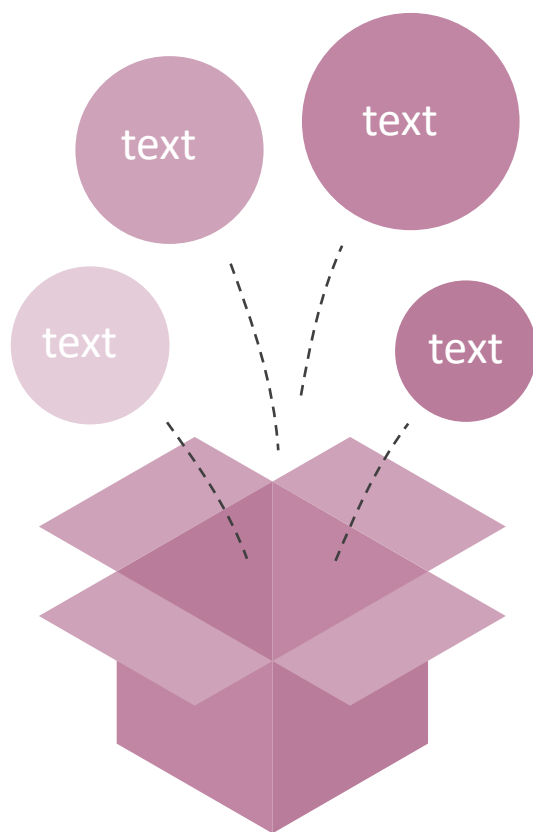
刚刚我们完成了将
富豪以性别进行分
组，并拿到了年龄
的最大值和最小值
以及平均值等信息。



现在我们完成
一个相对复杂
的需求，需要
查看每个国家
男女的富豪的
数量。那就需
要我们将富豪
们先按国家分
组，然后在按
性别分组。。

03

按多列进行分组



当需要按多列进行分组的时候，`groupby`方法里面我们传入的一个列表，列表中分别存储分组依据的列名。

注意：列表中列名的顺序，确定了先按`country`列进行分组，然后再按`gender`列分组。不同的顺序，产生的分组名字是不同的。

`group.size()`返回的结果中发现索引值是多层的，那么对于多层索引的值我们如何去获取呢？

```
In [1]: import pandas as pd
df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
group=df.groupby(['country','gender'])
df1 = group.size()
size = df1['Austria']['F']
print(size)
```

1

通过代码，我们发现对于多层索引值的获取，只需要从外往里一层一层的取就可以了，就像我们睡觉之前，需要先脱外衣再脱掉内衣是一样的。



四

对分组后数据进行统计



04

对分组后数据进行统计

接下来我们来体验一下，`agg()`方法的使用：

```
1 import pandas as pd
2 df = pd.read_excel('data/course_data/data_analysis/forbes_2018.xlsx')
3 groups = df.groupby('gender')
4 for group_name, group_df in groups:
5     f_se = group_df['age'].agg(['max', 'min', 'mean'])
6     print('{}组的最大年龄是{}, 最小年龄是{}, 平均年龄是{}'.format(group_name, f_se[0], f_se[1], f_se[2]))
```

f组的最大年龄是94.0, 最小年龄是21.0, 平均年龄是60.470588235294116
M组的最大年龄是99.0, 最小年龄是25.0, 平均年龄是64.32099447513812

观察上面的代码，可以发现在使用`agg()`函数时，我们可以将多个统计函数一起放到一个`agg()`函数中。

并且需要注意的是，如果是统计函数是pandas提供的，我们只需将函数的名字以字符串的形势存储到列表中即可，例如：将`max()`改成'`max`'。



代码实现

为大家使用更为灵活，pandas提供了一个`agg()`方法用来对分组后的数据进行统计。



代码实现

上面我们已经了解了一些Pandas提供好的统计函数，例如：`mean()`、`max()`等函数。



代码实现

数据统计（也称为数据聚合）是数据处理的最后一步，通常是要使每一个数组生成一个单一的数值。

04

对分组后数据进行统计

这样不仅简化了我们的代码，在添加和删减统计函数的时候我们只需更改agg()函数中list就可以了。是不是很方便。

它的好处还不止这些，比如现在又有新的需求，要计算年龄的最大值和最小值的差值。但是，pandas并没有提供这样统计函数，所以就需要我们进行自己定义一个统计函数：

01

Creative



text

02

text



Creative

03

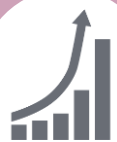
Creative



text

04

text



Creative

代码片段

```
1 def peak_range(df):
2     """
3     返回数值范围
4     """
5     return df.max() - df.min()
```

04

对分组后数据进行统计

现在我们看一下自己定义的统计函数，如何使用

```
import pandas as pd
df = pd.read_excel('/data/course_data/data_analysis/forbes_2018.xlsx')
groups = df.groupby('gender')
def peak_range(df):
    """
    返回数值范围
    """
    return df.max() - df.min()
for group_name, group_df in groups:
    f_se = group_df['age'].agg(['max', 'min', 'mean', peak_range])
    print(f_se[0], f_se[1], f_se[3])
```

```
94.0 21.0 73.0
99.0 25.0 74.0
```

`peak_range(df)`函数是我们自定义的函数，并设置了一个df参数,为了接收group_df['age']的值。

注意：自定义的函数名字在传入`agg()`函数中时不需要转换成字符串。



本节总结

第六课 知识点总结



本节我们通过groupby方法对数据进行分组以及根据多列进行分组，并且对分组后的数据进行统计。



知识点回顾



数据的分组

- 1.使用`groupby()`方法进行分组
- 2.`group.size()`查看分组后每组的数量
- 3.`group.groups`查看分组情况
- 4.`group.get_group('F')`根据分组后的名字选择分组数据

知识点回顾



对分组 进行遍历

- 1.使用for...in...可以对分组后的对象进行遍历
- 2.遍历时刻获取到两个对象，分组后的名字和对应组的数据

知识点回顾



按多列 进行分组

1. 使用groupby()方法进行按多列分组
2. 将多个列名放到列表中传给groupby做参数
3. 分组后的数据会有多层索引，获取数据需要从外到里逐层获取



练习

好色的不止是女人

题目要求

本次练习采用的是网易考拉海淘网口红一天的销售数据。每条数据都包含了品牌、折扣价、原价、是否自营、评论数、国家共6列信息。文件的路径为data/course_data/data_analysis/lipsticks.xls

x



03 解析

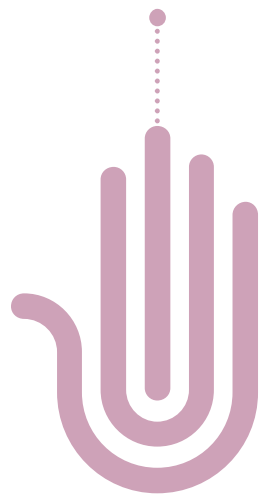
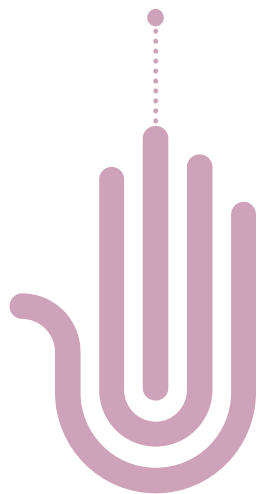
第一步:明确目标 ▲

本次练习采用的是网易考拉海淘网口红一天的销售数据。每条数据都包含了品牌、折扣价、原价、是否自营、评论数、国家共6列信息。文件路径为data/course_data/data_analysis/lipsticks.xlsx

第二步:分析过程 ▲

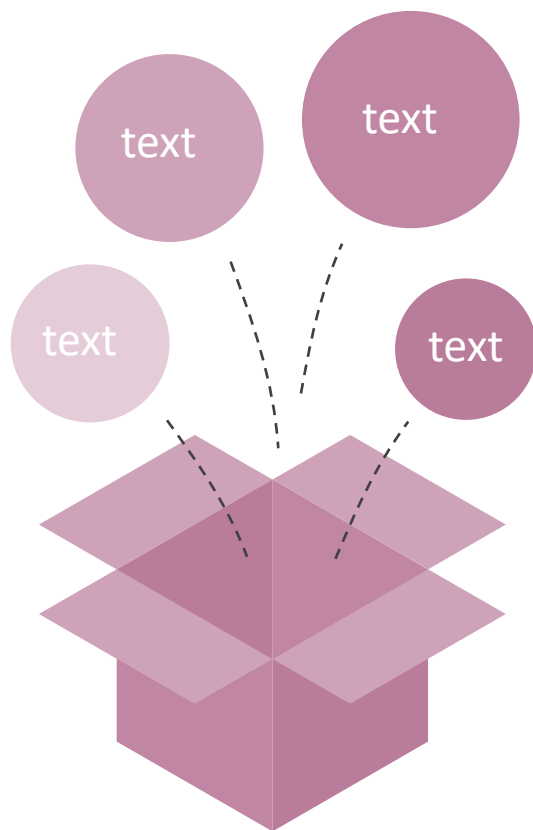
1. 统计每种口红的平均折扣价。 2. 分别统计每种口红自营评论数总和和非自营的评论数总和。

```
in [1]: ▶ 1 import pandas as pd
2 df = pd.read_excel('/data/course_data/data_analysis/lipsticks.xlsx')
3 print(df.head())
4
5 # 1. 统计每种口红的平均折扣价。
6 # 根据品牌进行分类
7 groups = df.groupby('品牌')
8 for group_name, group_df in groups:
9     mean = group_df['折扣价'].mean()
10    str_mean = '{} 的平均折扣价为{}'.format(group_name, mean)
11    print(str_mean)
12
13 # 2. 分别统计每种口红自营评论数总和和非自营的评论数总和
14 # 根据品牌列和是否自营列进行分组
15 groups = df.groupby(['品牌', '是否自营'])
16 for group_name, group_df in groups:
17     group_sum = group_df['评论数'].sum()
18     str_sum = '{} {} 的评论数为{}'.format(group_name[0], group_name[1], group_sum)
19     print(str_sum)
```



05

运行结果



	品牌	折扣价	原价	是否自营	评论数	国家
0	ESTÉE LAUDER 雅诗兰黛	109.0	400	自营	165	美国
1	MAC 魅可	89.0	280	自营	67	美国
2	MAC 魅可	89.0	280	自营	60	美国
3	ESTÉE LAUDER 雅诗兰黛	129.0	400	自营	2037	美国
4	MARIE DALGAR 玛丽黛佳	99.0	300	非自营	888	中国

BareMinerals的平均折扣价为158.12698412698413
CHANEL 香奈儿的平均折扣价为270.42105263157896
Dior 迪奥的平均折扣价为276.54545454545456
ESTÉE LAUDER 雅诗兰黛的平均折扣价为247.6
GIORGIO ARMANI 乔治·阿玛尼的平均折扣价为298.72727272727275
GIVENCHY 纪梵希的平均折扣价为271.94594594594594
GUERLAIN 娇兰的平均折扣价为228.92592592592592
KIKO MILANO的平均折扣价为68.75
L'ORÉAL 欧莱雅的平均折扣价为107.04347826086956
LANCÔME 兰蔻的平均折扣价为324.75
MAC 魅可的平均折扣价为128.41666666666666
MARIE DALGAR 玛丽黛佳的平均折扣价为92.83333333333333
MENTHOLATUM 曼秀雷敦的平均折扣价为47.0
Mamonde 梦妆的平均折扣价为69.0
Manuka Bee 小蜜坊的平均折扣价为55.65833333333333
Maybelline 美宝莲的平均折扣价为90.84615384615384
SAINT LAURENT PARIS 圣罗兰的平均折扣价为267.1617647058824
SHISEIDO 资生堂的平均折扣价为207.0
TOM FORD 汤姆·福特的平均折扣价为354.51851851851853
wet n wild的平均折扣价为65.0
BareMinerals自营的评论数为140
BareMinerals非自营的评论数为0
CHANEL 香奈儿自营的评论数为4999
Dior 迪奥自营的评论数为89329
Dior 迪奥非自营的评论数为12
ESTÉE LAUDER 雅诗兰黛自营的评论数为6761
GIORGIO ARMANI 乔治·阿玛尼自营的评论数为4961
GIORGIO ARMANI 乔治·阿玛尼非自营的评论数为1
GIVENCHY 纪梵希自营的评论数为15302
GUERLAIN 娇兰自营的评论数为3277
GUERLAIN 娇兰非自营的评论数为34
KIKO MILANO自营的评论数为7083
L'ORÉAL 欧莱雅自营的评论数为720
L'ORÉAL 欧莱雅非自营的评论数为7141
LANCÔME 兰蔻自营的评论数为7045
MAC 魅可自营的评论数为30597
MARIE DALGAR 玛丽黛佳非自营的评论数为2855
MENTHOLATUM 曼秀雷敦非自营的评论数为1873
Mamonde 梦妆自营的评论数为326
Mamonde 梦妆非自营的评论数为60
Manuka Bee 小蜜坊自营的评论数为553
Manuka Bee 小蜜坊非自营的评论数为851
Maybelline 美宝莲自营的评论数为582
Maybelline 美宝莲非自营的评论数为2547
SAINT LAURENT PARIS 圣罗兰自营的评论数为23698
SAINT LAURENT PARIS 圣罗兰非自营的评论数为32
SHISEIDO 资生堂自营的评论数为34486
SHISEIDO 资生堂非自营的评论数为11
TOM FORD 汤姆·福特自营的评论数为7200
wet n wild自营的评论数为1416
wet n wild非自营的评论数为655



练习

那些年错过的电影

题目要求

本次练习采用的是爱奇艺视频数据。共有6万多条电影数据，每条数据包含12列信息，文件的路径为/data/course_data/data_analysis/aiqiyi.xlsx，以下获取的前五条数据：



02

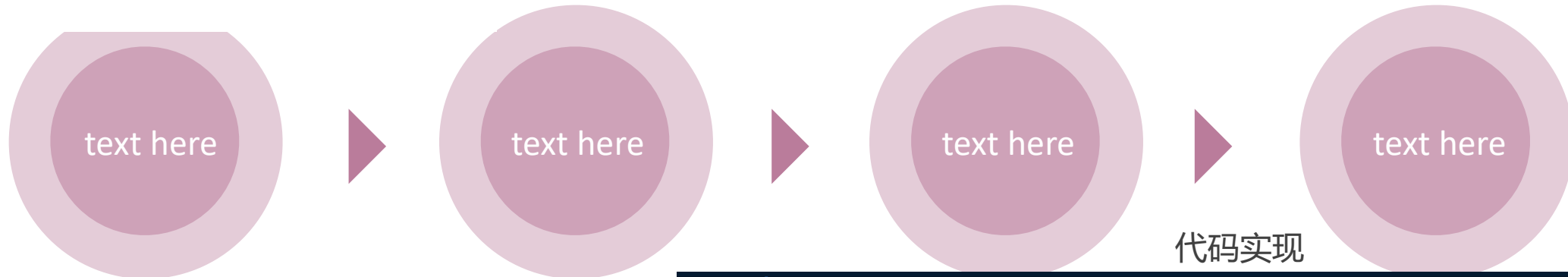
课题研究的思路与方法 Ideas And Methods

第一步:明确目标 ▲

本次练习采用的是爱奇艺视频数据。共有6万多条电影数据，每条数据包含12列信息，文件的路径为/data/course_data/data_analysis/aiqiyi.xlsx，以下获取的前五条数据：

第二步:分析过程 ▲

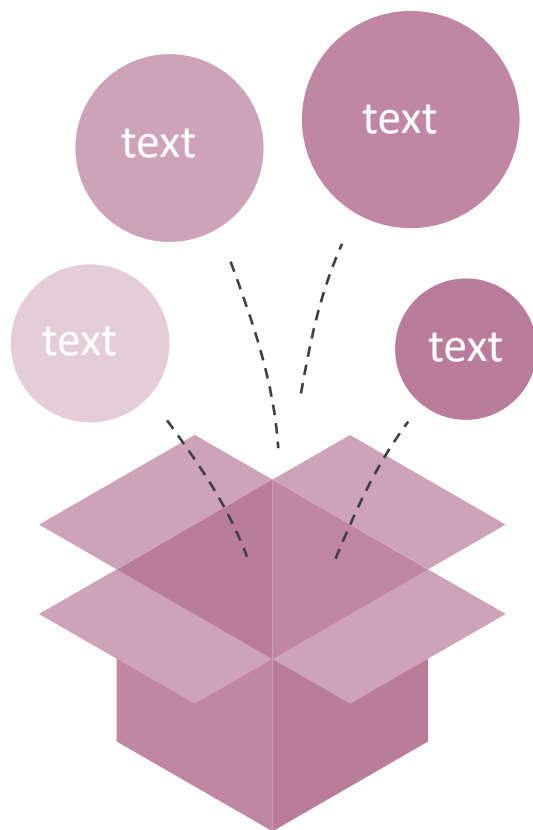
1. 取出每年电影评分前两名电影的名字
2. 哪一年的电影总评分最高



```
In [1]: #第一题答案
import pandas as pd
df = pd.read_excel('/data/course_data/data_analysis/aiqiyi.xlsx')
groups = df.groupby('上映时间')
for group_name, group_df in groups:
    result = group_df.sort_values(by='评分', ascending=False)[0:2]
    print(group_name, result['整理后剧名'])
#第二题答案
import pandas as pd
df = pd.read_excel('./data/aiqiyi.xlsx')
groups = df.groupby('上映时间')
year=groups.sum().sort_values(by='评分', ascending=False).index.to_list()[0]
print(year)
```

05

运行结果



```
1988 1195 4世同堂
1231 1 少奇同志在东北
Name: 整理后剧名, dtype: object
1986 156 钟鼓楼
603 凯旋在子夜
Name: 整理后剧名, dtype: object
1988 1222 少奇同志在武汉
788 小镇总理
Name: 整理后剧名, dtype: object
1989 700 李大利
Name: 整理后剧名, dtype: object
1990 441 渴望
1157 吴晗
Name: 整理后剧名, dtype: object
1991 742 家有仙妻
1144 让我们荡起双桨
Name: 整理后剧名, dtype: object
1992 358 皇城根儿
195 风华绝代
Name: 整理后剧名, dtype: object
1993 1999 少奇同志在天津
Name: 整理后剧名, dtype: object
1994 901 过把瘾
1463 新书剑恩仇录
Name: 整理后剧名, dtype: object
1995 898 无悔追踪
162 东边日出西边雨
Name: 整理后剧名, dtype: object
1996 626 宰相刘罗锅
138 大索腔
Name: 整理后剧名, dtype: object
1997 1069 寇老西儿
630 鸦片战争演义
Name: 整理后剧名, dtype: object
1998 521 聊斋先生
704 快排
Name: 整理后剧名, dtype: object
1999 1275 永不瞑目
699 刑警本色
Name: 整理后剧名, dtype: object
2000 1012 上错花轿嫁对郎
653 都是天使惹的祸
Name: 整理后剧名, dtype: object
2001 27 爱情宝典
322 天下第一丑
Name: 整理后剧名, dtype: object
2002 652 我的淘气天使
379 白领公寓
Name: 整理后剧名, dtype: object
2003 1292 双响炮
481 火帅
Name: 整理后剧名, dtype: object
2004 1020 天龙八部
469 铁齿铜牙纪晓岚3
Name: 整理后剧名, dtype: object
2005 1291 我爹河东狮
464 宋莲生坐堂
Name: 整理后剧名, dtype: object
2006 527 沂蒙新传
884 士兵突击
Name: 整理后剧名, dtype: object
2007 209 我们生活的年代
1442 睡龙神探之情爱保险
Name: 整理后剧名, dtype: object
2008 106 防火墙5788
320 所谓婚姻
Name: 整理后剧名, dtype: object
2009 1372 青春舞台
463 我们的队伍向太阳
Name: 整理后剧名, dtype: object
2010 407 尖刀
384 大女当嫁
Name: 整理后剧名, dtype: object
2011 808 闹海
203 盘龙卧虎高山顶
Name: 整理后剧名, dtype: object
2012 969 闭嘴花美男乐队金明珠cut集锦
1249 闭嘴花美男乐队金明珠cut集锦
Name: 整理后剧名, dtype: object
2013 516 恋歌
610 邻家花美男
Name: 整理后剧名, dtype: object
2014 838 保卫孙子
224 犀利仁师之药不能停路云霏
Name: 整理后剧名, dtype: object
2015 667 我们办人事
449 红色青橄榄
Name: 整理后剧名, dtype: object
2016 331 皇天不负有心人
Name: 整理后剧名, dtype: object
```



THANK YOU

欢迎进入下一章节的学习

