计算机 22A4 2022015232 孙春辉
2025 年 3 月 24 日 星期一 第五次作业

Report 5

Part 1: Course exercises

1.1:拆分

1.2:by

1.3:可迭代

1.4:groups

1.5:DataFrameBy

2.1:F

2.2:F

2.3:F

2.4:T

2.5:F

3.1:A

3.2:A

3.3:B

3.4D

3.5:C

4.1： 分组的流程主要是根据，其对象的列行索引名称、key 值、自定义的 series 对象索引、字典、函数等进行拆分开的，然后分组展示出来；聚合的流程是将每个分组应用统计运算，并把运算后的结果合并到一起的。

4.2: 其对象的列行索引名称、key 值、自定义的 series 对象索引、字典、函数进行分组的。

5:（1）：

```
import numpy as np
import pandas as pd
from pandas import DataFrame

df_obj = DataFrame({'年纪' : ['大一','大二','大三', '大四','大二','大三', '大一','大三','大四'],
                    '姓名' : ['李宏卓','李思真','张振海','赵鸿飞','白蓉','马腾飞', '张晓凡','金紫
                    '年龄' : [18,19,20,21,19,20,18,20,21],
                    '身高' : [175,165,178,175,160,180,167,170,185],
                    '体重' : [65,60,70,76,55,70,52,53,73],})
df_obj
```

	年纪	姓名	年龄	身高	体重
0	大一	李宏卓	18	175	65
1	大二	李思真	19	165	60
2	大三	张振海	20	178	70
3	大四	赵鸿飞	21	175	76
4	大二	白蓉	19	160	55
5	大三	马腾飞	20	180	70
6	大一	张晓凡	18	167	52
7	大三	金紫萱	20	170	53
8	大四	金烨	21	185	73

（2） ：

```
group_data = df_obj.groupby('年级')
people = group_data.get_group('大一')
people
```

	年级	姓名	年龄	身高	体重
0	大一	李宏卓	18	175	65
6	大一	张晓凡	18	167	52

（3）：

```
data = group_data.apply(max)
del data['年级']
data
```

```
C:\Users\sun\AppData\Local\Temp\ipyke
callable <built-in function max> is c
pandas, the provided callable will be
np.maximum.reduce instead.
  data = group_data.apply(max)
C:\Users\sun\AppData\Local\Temp\ipyke
eGroupBy.apply operated on the groupi
version of pandas the grouping column
ude_groups=False` to exclude the grou
oupby to silence this warning.
  data = group_data.apply(max)
```

年级	姓名	年龄	身高	体重
大一	李宏卓	18	175	65
大三	马腾飞	20	180	70
大二	白蓉	19	165	60
大四	金烨	21	185	76

（4）：

```
junior = dict([x for x in group_data])['大三']
print(people['体重'].apply('mean'))
print(junior['体重'].apply('mean'))
```

```
58.5
64.33333333333333
```

Part 2:

Download the ‘Starbucks.csv’ file from the Moodle and complete the following Pandas practices and screenshot your solutions and results.

1. print 所有宁波星巴克门店信息.

```
import numpy as np
import pandas as pd
from pandas import DataFrame
```

```
df = pd.read_csv('./Report 5-Starbucks.csv')
print(df)
```

```
           Brand  Store Number  ...  Longitude  Latitude
0      Starbucks  47370-257954  ...       1.53     42.51
1      Starbucks  22331-212325  ...      55.47     25.42
2      Starbucks  47089-256771  ...      55.47     25.39
3      Starbucks  22126-218024  ...      54.38     24.48
4      Starbucks  17127-178586  ...      54.54     24.51
...          ...           ...  ...        ...       ...
25595  Starbucks  21401-212072  ...     106.70     10.78
25596  Starbucks  24010-226985  ...     106.71     10.72
25597  Starbucks  47608-253804  ...      28.04    -26.15
25598  Starbucks  47640-253809  ...      28.28    -25.79
```

2. 分析法国的星巴克数量和中国的哪个多.

```
df = pd.read_csv('./Report 5-Starbucks.csv')
df = df.groupby(by='Country')

for dfi in df :
   print(dfi)
```

```
  [96 rows x 13 columns])
  ('CN',           Brand  Store Number
  2091  Starbucks  22901-225145  ...
  2092  Starbucks  32320-116537  ...
  2093  Starbucks  32447-132306  ...
  2094  Starbucks  17477-161286  ...


[8 rows x 13 columns])
('FR',           Brand  Store Number
5209  Starbucks  27750-248649  ...
5210  Starbucks  27752-248647  ...
5211  Starbucks  25626-241740  ...
5212  Starbucks  26289-241644  ...
```

```
cou = df["Brand"].count()
cn = cou['CN']
fr = cou['FR']
```

```
2734
132
```

3. 分析中国每个省份星巴克的数量的情况.

```
df_prv = df_CN.groupby(by='State/Province')
cou = df_prv["State/Province"].count()
print(cou)
```

```
State/Province
11     236
12      58
13      24
14       8
15       8
21      57
22      13
23      16
31     551
32     354
33     315
34      26
```

4. 分析宁波的星巴克和杭州的星巴克数量情况.

```
df_prv = df_CN.groupby(by='State/Province')
df_pro = df_prv.get_group('33')
df_city = df_pro.groupby(by='City')
cou = df_city['City'].count()

hangzhou = cou['杭州市']
ningbo = cou['宁波市']
```

```
print(hangzhou)
print(ningbo)
```

```
117
58
```

```
宁波市    58
慈溪市     5
杭州市   117
```

5. 哪个国家星巴克门店数量最多？哪个国家最少？.

```
df = pd.read_csv('./Report 5-Starbucks.csv')
df = df.groupby(by='Country')
cou = df['Brand'].count()
cou_sorted = cou.sort_values(ascending=False)
print(cou_sorted)
```

```
Country
US    13608
CN     2734
CA     1468
JP     1237
KR      993
      ...
SK        3
ZA        3
LU        2
MC        2
AD        1
```

6. 比较中国星巴克门店最多的省份和美国星巴克门店最多州的数量.

```
china_data = df[df['Country'] == 'CN']  # 中国的数据
us_data = df[df['Country'] == 'US']    # 美国的数据



# 3. 统计中国各省的门店数量
china_province_counts = china_data['State/Province'].value_counts()
most_province_cn = china_province_counts.idxmax()  # 门店最多的省份
```

```
max_count_cn = china_province_counts.max()        # 该省份的门店数量

# 4. 统计美国各州的门店数量
us_state_counts = us_data['State/Province'].value_counts()  # 注意列名可能是 'State/Province' 或 'State'
most_state_us = us_state_counts.idxmax()  # 门店最多的州
max_count_us = us_state_counts.max()      # 该州的门店数量

# 5. 比较结果
print("中国星巴克门店最多的省份:")
print(f"{most_province_cn}: {max_count_cn} 家门店")

print("\n美国星巴克门店最多的州:")
print(f"{most_state_us}: {max_count_us} 家门店")

print("\n比较结果:")
if max_count_cn > max_count_us:
    print(f"中国 {most_province_cn} 的门店数量 ({max_count_cn}) 多于美国 {most_state_us} ({max_count_us})")
elif max_count_cn < max_count_us:
    print(f"美国 {most_state_us} 的门店数量 ({max_count_us}) 多于中国 {most_province_cn} ({max_count_cn})")
else:
    print(f"中国 {most_province_cn} 和美国 {most_state_us} 的门店数量相同 ({max_count_cn})")
```

```
中国星巴克门店最多的省份:
31: 551 家门店

美国星巴克门店最多的州:
CA: 2821 家门店

比较结果:
美国 CA 的门店数量 (2821) 多于中国 31 (551)
```

7. 比较南北半球和东西半球星巴克门店数量.

```
df['Hemisphere_NS'] = df['Latitude'].apply(lambda x: 'Northern' if
x >= 0 else 'Southern')
# 计算东西半球
df['Hemisphere_EW'] = df['Longitude'].apply(lambda x: 'Eastern' if
x >= 0 else 'Western')
# 统计南北半球的门店数量
ns_counts = df['Hemisphere_NS'].value_counts()
# 统计东西半球的门店数量
ew_counts = df['Hemisphere_EW'].value_counts()
print("=== 南北半球星巴克门店数量 ===")
print(ns_counts)
print("\n=== 东西半球星巴克门店数量 ===")
print(ew_counts)
```

```
=== 南北半球星巴克门店数量 ===
Hemisphere_NS
Northern    24898
Southern      702
Name: count, dtype: int64

=== 东西半球星巴克门店数量 ===
Hemisphere_EW
Western    17139
Eastern     8461
Name: count, dtype: int64

Process finished with exit code 0
```

8. 哪个时区的星巴克门店最多？哪个时区最少？.

```
df_timezone = df.groupby(by='Timezone')
dfcou = df_timezone["Brand"].count()
# print(dfcou)
most_time = dfcou.idxmax()
# most_time = dfcou
min_time = dfcou.idxmin()
print(f"最多数量的时区是 ： {most_time}, 最少的失时区是{min_time}")
```

```
最多数量的时区是 ： GMT-05:00 America/New_York, 最少的失时区是GMT+0:00 Atlantic/Canary
```

*Ref: Pandas 数据分组聚合案例 (*https://www.isolves.com/hlw/dsj/2020-06-23/21345.html*)*