第四章　数据整理与计算（pandas 和 matplotlib）

【课时目标】

1. 掌握 Pandas 模块的两种数据结构 Series 和 DataFrame。
2. 学习使用 Pandas 模块对数据进行编辑、计算、统计、分析。
3. 会使用 Python 进行简单数据处理，并能从其中提取有用信息形成结论。

【知识梳理】

1. 常用的 Python 扩展模块有 Numpy、Scipy、Pandas 和 Matplotlib 等，Pandas 模块主要用于数据的处理和分析。
2. Python 中引入 Pandas 模块的方法：import pandas as pd，pd 是用户为导入模块取的别名。
3. pandas 提供了 Series 和 DataFrame 两种数据结构。

①Series 是一种一维的数据结构，包含一个数组的数据和一个与数据关联的索引(index)，索引值默认是从 0 起递增的整数，数据可以是不同类型的元素。列表、字典等可以用来创建 Series 数据结构。

（1）创建 Series 对象（默认索引）

```
import pandas as pd
s1=pd.Series([45,30,35,28])
print (s1)
```

对应结构

index	values
0	45
1	30
2	35
3	28

运行结果

```
0    45
1    30
2    35
3    28
dtype: int64
```

Series 对象 s1 创建好之后：s1[0] = 45,s1[1] = 30, s1[2] = 35, s1[3] = 28

（2）创建 Series 对象（指定索引）

```
import pandas as pd
s2=pd.Series([8,12,18,20], index=["s01","s02","s03","s04"])
print(s2)
```

对应结构

index	values
s01	8
s02	12
s03	18
s04	20

运行结果

```
s01     8
s02    12
s03    18
s04    20
dtype: int64
```

Series 对象 s2 创建好之后：s2["s01"] = 8, s2["s02"] = 12，……

需要注意的是默认从 0 开始的索引，还是可以使用：s2[0] = 8,s2[1]= 12，……

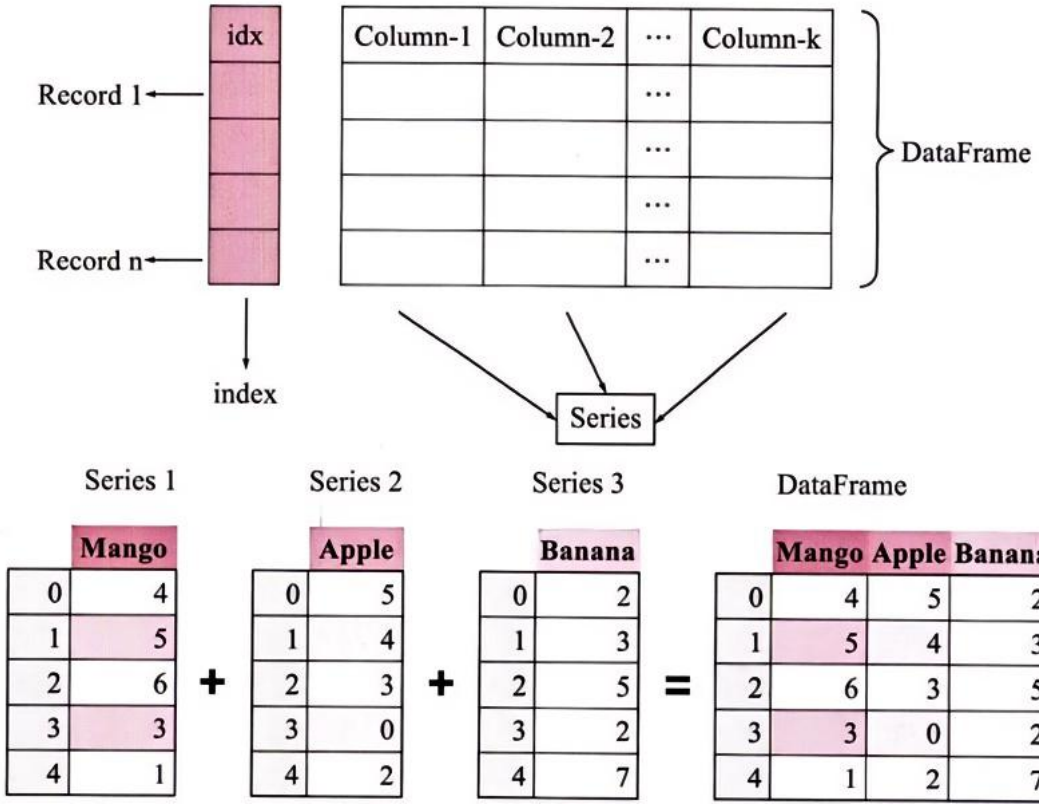
(3)Series 对象的遍历。

遍历 Series 对象的索引	运行结果
import pandas as pd s2 = pd.Series([8, 12, 18, 20], index = ["s01", "s02", "s03", "s04"]) for x in **s2.index :** 　　print(x)	s01 s02 s03 s04

遍历 Series 对象的值	运行结果
import pandas as pd s2 = pd.Series([8, 12, 18, 20], index = ["s01", "s02", "s03", "s04"]) for x in **s2.values :**　　**#等同于** for x in **s2 :** 　　print(x)	8 12 18 20

注意：遍历 Series 对象的值时，s2.values 可以省略成 s2,因为 Series 对象默认属性就是 values。

②DataFrame 是一种二维的数据结构，由 1 个索引列(index)和若干个数据列组成，每个数据列可以是不同的类型。DataFrame 可以看作是共享同一个 index 的 Series 的集合。



DataFrame 对象常用属性如下:

属性	说明
index	DataFrame 的行索引
columns	存放各列的列标题
values	存放值的二维数据
T	行列转置

【注意】index 与 columns 的使用:代码中 index = [0,1,2], columns =["学号","性别","年龄"]可以省略,因为默认索引就是[0,1, 2],列标题 columns 默认就是字典的键。

(1)利用字典创建 DataFrame 对象。

```
import pandas as pd
dic={"学号":["s01","s02","s03"], "性别":["男","女","男"],"年龄":[16,17,16]}
df=pd.DataFrame(dic, index=[0,1,2], columns=["学号","性别","年龄"])
print(df)
```

对应结构

index	学号	性别	年龄
0	s01	男	16
1	s02	女	17
2	s03	男	16

运行结果

```
   学号 性别  年龄
0  s01  男   16
1  s02  女   17
2  s03  男   16
```

(1) DataFrame 对象遍历及转置(以上面数据为例)。

遍历 DataFrame 对象的索引	运行结果
for i in df.index : print(i)	0 1 2

遍历 DataFrame 对象的列标题	运行结果

for i in df.columns: #df.columns 可写成 df
  print(i)
	学号
	性别
	年龄

遍历 DataFrame 对象的值	运行结果
for i in df.values : print(i)	['s01' '男' 16] ['s02' '女' 17] ['s03' '男' 16]

DataFrame 行列转置	运行结果			
df2 = df.T print(df2)		0	1	2
	学号	s0l	s02	s03
	性别	男	女	男

(2) 其他创建 DataFrame 对象方式

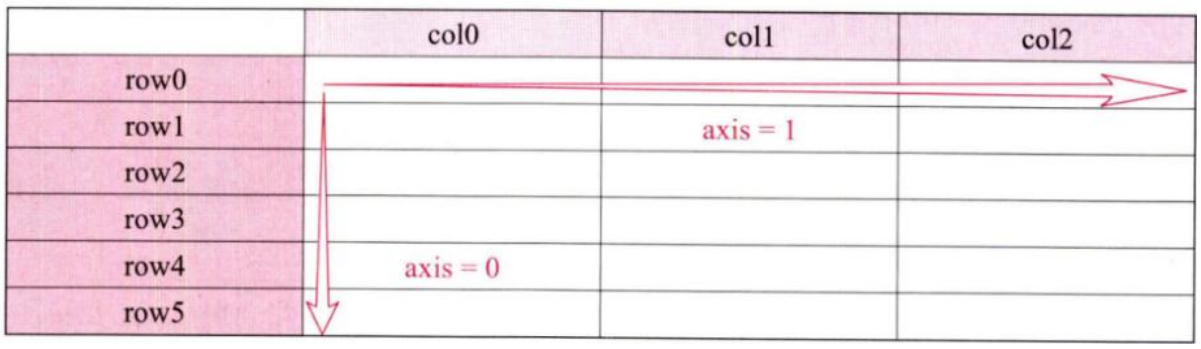
由二维列表创建	**由 Excel 表格创建(高考中常见形式)**
import pandas as pd data = [["s01","男",16], ["s02","女",17], ["s003","男",16]] df= pd.DataFrame(data, columns =["学号", "性别", "年龄"])	**import pandas as pd** **df = pd.read_excel("成绩.xlsx")**

(3) DataFrame 常见函数

函数	说明
count()	返回非空(NaN)数据项的数量, 通过 axis = 0/1 确定行列
sum()、mean()	求和、求平均值,通过 axis = 0/1 确定行列
max()、min()	返回最大、最小值,通过 axis = 0/1 确定行列
head()、tail()	返回 DataFrame 的前 n 个、后 n 个数据记录，若 n 省略，则默认返回前/后 5 行的数据
groupby()	对各列或各行中的数据进行分组,参数 as_index=True/False 决定是否将分组项作为 index
sort_values()	排序，通过 axis = 0/1 确定行列,ascending=True/False 表示升/降序
drop()	删除数据，通过 axis = 0/1 确定行列。df.drop("性别″，axis = 1)删除“性别”列数据
append()	在指定元素的结尾插入内容
insert()	在指定位置插入列
rename()	修改列名或者索引
concat()	合并 DataFrame 对象
set_value()	根据行标签和列标签设置单个值
plot()	绘图

axis = 0：数据在纵向发生变化，沿着每一列向下执行(默认 axis = 0)。

axis = 1：数据在横向发生变化，沿着每一行横向执行。



例如：df.mean()或 df.mean(axis=0) 按**列**计算平均值(计算**列**的平均值)

df.mean(axis=1) 按**行**计算的均值(计算**行**的平均值)。在删除列操作时需要指定 axis=1

4.DataFrame 的基本操作（一）

某 DataFrame 对象 df 的数据结构如图所示：

index	品牌	销量
0	小米	900
1	华为	2700
2	OPPO	800
3	苹果	2500

(1) 获取某列数据：获取列数据可以通过**属性记法**或**字典记法**。例如,获取品牌列的数据：**df.品牌**或**df["品牌"]（引号用单引号或双引号都可以）**。如果列名是数字开头，则只能用第二种方式(字典记法)。

(2) 获取行数据。例如，获取第 1 行数据:**df[0:1]**或 df.loc[0]。(**注意：获取行数据不能写成 df[行号]格式**)

df.head(n)获取前 n 行数据，**df.tail(n)**获取后 n 行数据。

(3) 获取单个值。例如，获取第二行销量的值 2700: **df.at[1,"销量"]或 df["销量"][1]**。

(4) 按条件筛选数据行。例如，以筛选销量大于 1000 的数据行为条件：**df.销量＞1000 或 df["销量"]＞1000,所以筛选的语句为 df[df.销量＞1000]或 df[df["销量"]>1000]。**

(5) 数据统计。

① 计算品牌列数据个数: df.品牌.count() 或 df["品牌"].count()。

② 计算销量列平均值: df.销量.mean() 或 df["销量"].mean()。

③ 计算销量列数据之和: df.销量.sum() 或 df["销量"].sum()。

DataFrame 的基本操作(二)

（1）读取 Excel 文件中的数据到 DataFrame 对象。

```
import pandas as pd
df = pd.read_excel("成绩.xlsx")
```

	A	B	C	D
1	班级	姓名	语文	数学
2	高二(1)班	沈佳	110	128
3	高二(2)班	王伟佳	99	105
4	高二(3)班	钱小平	93	73
5	高二(1)班	钟明	105	110
6	高二(2)班	朱乐乐	94	116
7	高二(3)班	陈王佳	102	109
8	高二(1)班	朱丽萍	106	97
9	高二(2)班	沈丽丽	102	126
10	高二(3)班	孙一可	98	101
11	高二(1)班	罗杰	99	98

成绩.xlsx

index	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
2	高二(3)班	钱小平	93	73
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98

df 结构

	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
2	高二(3)班	钱小平	93	73
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98

运行结果

(2) **删除 DataF rame 对象的数据列**

```
import pandas as pd
df = pd.read_excel("成绩.xlsx")
df2 = df.drop("姓名", axis = 1)
print(df2)
```

	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
2	高二(3)班	钱小平	93	73
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98

df

删除列

	班级	语文	数学
0	高二(1)班	110	128
1	高二(2)班	99	105
2	高二(3)班	93	73
3	高二(1)班	105	110
4	高二(2)班	94	116
5	高二(3)班	102	109
6	高二(1)班	106	97
7	高二(2)班	102	126
8	高二(3)班	98	101
9	高二(1)班	99	98

df2

【注意】 drop()函数的使用。

① df2 = df.drop("姓名",axis=1),
删除列时,数据在横向发生变化,列减少了,相当于变瘦了,所以要指定 axis=1。

② df2 = df.drop("姓名",axis=1),删除“姓名”列数据,得到新的 DataFrame 对象 df2。
注意：df 不变,因为没有给 df 赋值。

(3) **删除 DataFrame 对象的数据行。**

```
import pandas as pd
df = pd.read_excel("成绩.xlsx")
df2 = df.drop(2) #axis = 0 可以省略
print(df2)
```

	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
2	高二(3)班	钱小平	93	73
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98

df

删除行

	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98

df2

【注意】 df2 = df.drop(2)删除行,因为删除了行,相当于变矮了,纵向发生了变化,所以指定axis=0(axis=0 可以省略)。

(4)添加 DataFrame 对象的数据行

```
import pandas as pd
df=pd.read_excel("成绩.xlsx")
df2=df.append({"班级":"高二(8)班","姓名:"杜宗飞","语文":119,"数学":118},ignore_index=True)
print(df2)
```

	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
2	高二(3)班	钱小平	93	73
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98

df

添加行

	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
2	高二(3)班	钱小平	93	73
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98
10	高二(8)班	杜宗飞	119	118

df2

(5) 数据排序。

```
import pandas as pd
df = pd.read_excel("成绩.xlsx")
df_sort = df.sort_values("语文")   #按语文成绩排序,默认升序 ascending = True
print(df_sort)
```

	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
2	高二(3)班	钱小平	93	73
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98

df

排序

	班级	姓名	语文	数学
2	高二(3)班	钱小平	93	73
4	高二(2)班	朱乐乐	94	116
8	高二(3)班	孙一可	98	101
1	高二(2)班	王伟佳	99	105
9	高二(1)班	罗杰	99	98
5	高二(3)班	陈王佳	102	109
7	高二(2)班	沈丽丽	102	126
3	高二(1)班	钟明	105	110
6	高二(1)班	朱丽萍	106	97
0	高二(1)班	沈佳	110	128

df_sort

降序排序代码:df_sort = df.sort_values("语文",ascending = False)

(6) 数据分组统计。

```
import pandas as pd
df = pd.read_excel("成绩.xlsx")
g = df.groupby("班级", as_index = False).mean()
print(g)
```

班级	姓名	语文	数学
高二(1)班	沈佳	110	128
高二(2)班	王伟佳	99	105
高二(3)班	钱小平	93	73
高二(1)班	钟明	105	110
高二(2)班	朱乐乐	94	116
高二(3)班	陈王佳	102	109
高二(1)班	朱丽萍	106	97
高二(2)班	沈丽丽	102	126
高二(3)班	孙一可	98	101
高二(1)班	罗杰	99	98

成绩.xlsx

班级	姓名	语文	数学
高二(1)班	沈佳	110	128
高二(1)班	钟明	105	110
高二(1)班	朱丽萍	106	97
高二(1)班	罗杰	99	98

班级	姓名	语文	数学
高二(2)班	王伟佳	99	105
高二(2)班	朱乐乐	94	116
高二(2)班	沈丽丽	102	126

班级	姓名	语文	数学
高二(3)班	钱小平	93	73
高二(3)班	陈王佳	102	109
高二(3)班	孙一可	98	101

按班级分组

	班级	语文	数学
0	高二(1)班	105.000000	108.250000
1	高二(2)班	98.333333	115.666667
2	高二(3)班	97.666667	94.333333

分组统计得到新的 DataFrame 对象 g

注意：如果 as_index = False 不写，代码 g = df.groupby("班级").mean()分组 统计得到新的 DataFrame 对象 g 如下所示。此时，“班级”列数据将被作为索引，要获取“班级”列数据的方式为“g.index”，而不是“g.班级”。

此列为 index，但不会有 index 字样，“班级”对应行内容为空。

班级	语文	数学
高二(1)班	105	108.25
高二(2)班	98.333333	115.666667
高二(3)班	97.666667	94.333333

4.利用 matplotlib 模块绘图。

matplotlib 是一个绘图库，使用其中的 pyplot 子库所提供的函数可以快速绘图和设置图表的坐标轴、坐标轴刻度、图例等。

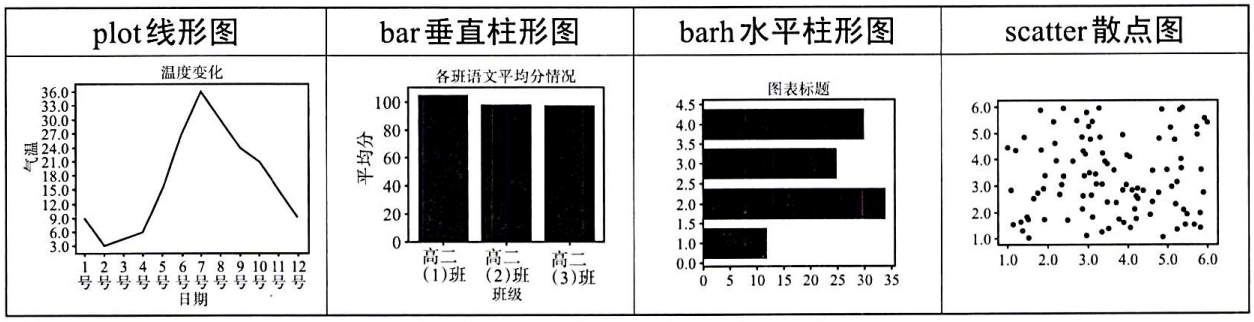
(1)常用绘图函数。

函数	说明
figure()	创建一个新的图表对象，并设置为当前绘图对象。直接调用 plot 等绘图函数进行绘图，matplotlib 会自创建一个 figure 对象
plot()	绘制线形图
bar()	绘制垂直柱形图
barh()	绘制水平柱形图
scatter()	绘制散点图
title()	设置图表的标题
xlim()、ylim()	设置 X、Y 轴的取值范围
xlabel()、ylabel()	设置 X、Y 轴的标签
legend()	显示图例
show()	显示创建的所有绘图对象

(4)常用绘图函数的部分属性。

函数	属性说明
plot() bar() barh()	linestyle：线条样式 color：线条颜色 linewidth：线条宽度 label :指定线条标签
scatter()	S：指定散点的大小 C：指定散点的颜色 linewidths：指定散点边框线的宽度 marker ：指定散点的图形样式

(5)常用图表的形状。



(6)利用 matplotlib 模块绘图实例。

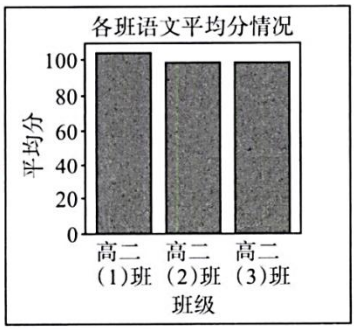
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel("成绩.xlsx")      # 读取 Excel 数据创建 DataFrame 对象 df
g = df.groupby("班级", as_index = False).mean()    #按班级分组并求平均分
plt.bar(g.班级,g.语文)      #根据班级和语文两列数据生成柱形图
plt.title("各班语文平均分情况")#图表标题
plt.xlabel("班级")      #x 轴标签
plt.ylabel("平均分") #y 轴标签
plt.show()      #显示绘图结果
```

index	班级	姓名	语文	数学
0	高二(1)班	沈佳	110	128
1	高二(2)班	王伟佳	99	105
2	高二(3)班	钱小平	93	73
3	高二(1)班	钟明	105	110
4	高二(2)班	朱乐乐	94	116
5	高二(3)班	陈王佳	102	109
6	高二(1)班	朱丽萍	106	97
7	高二(2)班	沈丽丽	102	126
8	高二(3)班	孙一可	98	101
9	高二(1)班	罗杰	99	98

成绩.xlsx

	班级	语文	数学
0	高二(1)班	105.000000	108.250000
1	高二(2)班	98.333333	115.666667
2	高二(3)班	97.666667	94.333333

分组求平均得到g



生成的图表

(5)DataFrame 函数 plot()结合 pyplot 相关函数绘图(了解)。

除了使用 matplotlib 的 pyplot 子库所提供的函数可以快速绘图外，还可以使用 DataFrame 函数 plot()结合 pyplot 相关函数绘图。

【课堂练习 1】

1.有如下 Python 代码：

```
import pandas as pd
s1=pd.Series([166,178,180],index=["s01","s02","s03"])
print(s1.values)      #①
print(s1.index)       #②
print(s1)             #③
```

语句①代码运行结果是(　　)

语句②代码运行结果是(　　)

语句③代码运行结果是(　　)

A.	B.	C.	D.	E.	F.	G.
166 178 180	[166,178,180]	Index(['s01','s02','s03'], dtype='object')	"s01" "s02" "s03"	s01 s02 s03	0 166 1 178 2 180 dtype: int64	s01 166 s02 178 s03 180 dtype: int64

2.有如下 Python 代码：

```
import pandas as pd
s1＝pd.Series([166,178,180], index＝["s01","s02","s03"])
s1[1]＝168
print(s1[s1<179])
```

执行代码后，输出的结果是(　　)

A.	B.	C.	D.
166 168 180	s01 168 s02 178	s01 166 s02 168	166 168

3.有 python 代码如下：

```
import pandas as pd
x＝pd.DataFrame({"x1":[1，2，3，4]，"x2":[5，6，7，8]，"x3":[9，10，11，12]})
```

下列说法正确的是(　　)

A.运行该程序代码后 print(x)，不会输出索引列

B.构造的 x 对象除索引列外，还有 4 列数据

C.x 的 values 中是既有字符类数据，还有数值类数据

D.执行语句 print(x[x["x2"]>6])，将输出除标题外的 2 行数据

4.有 python 代码如下：

```
import pandas as pd
data＝{"姓名":["王静怡","张佳妮","李臣武"]，"性别":["女","女","男"]，
        "借阅次数":[28，56，37]}
df1＝pd.DataFrame(data，columns＝["姓名","借阅次数","性别"])
p＝df1[1:2]
t＝df1.at[2,"姓名"]
print(df1)
```

下列说法正确的是(　　)

A.变量 t 的值为"张佳妮"

B.对象 p 中只有一行记录

C.输出信息中，第 2 列(除索引列)为性别

D.执行语句“df1.借阅次数＝40”，将把第一个人的借阅次数修改为 40

【课堂练习 2】

1.小明将一些 App 的活跃人数（单位：万人）数据存在文件“app.xlsx”中，部分界面如图所示。

	A	B	C	D	E
1	APP名称	应用领域	10月人数	11月人数	12月人数
2	360手机卫士	安全管理	13855.8	14422.3	16017.2
3	腾讯手机管家	安全管理	17992.5	18135.2	19688.5
4	微信	即时通讯	74257	75101.6	75616.5
5	QQ	即时通讯	53931.5	54913.7	55216.7
6	QQ浏览器	浏览器	20759.1	20540.3	22438.6
7	手机百度	搜索	18401.2	18571.3	19960.2
8	QQ音乐	移动音乐	13707.9	13709.1	13691.5
9	酷狗音乐	移动音乐	19407.5	19237.8	20847.05
10	酷我音乐	移动音乐	4935.37	4864.13	5480.75
11	支付宝	支付	29897	31408.9	33146.8
12	京东	综合电商	6441.72	7596.09	6854.08
13	淘宝	综合电商	27153.9	27737.6	29807.4
14	爱奇艺视频	综合视频	20319.4	20543.8	22047.9
15	芒果tv	综合视频	5283.03	5165.8	5108.08
16	腾讯视频	综合视频	19833.5	20412.6	21332.2
17	优酷视频	综合视频	12576.4	12264.5	13840.4

```
import pandas as pd
#读取文件“app.xlsx”数据
book1=__①__
 #计算 10 月人数之和
book1_sum=__②__
#计算 11 月人数平均值
book1_aver=__③__
#按应用领域分组统计，分别对各月人数求和
book1_g=__④__
#按 11 月人数值降序排序
book1_sort=__⑤__
print("10 月人数之和：",book1_sum)
print("11 月人数平均值：",book1_aver)
print("各应用领域各月人数和：",book1_g)
print("11 月人数降序排序情况：",book1_sort)
```

2.小明将某天 24 小时的楼层停靠数据导出，并保存在“data.csv”文件中，部分数据如图所示（时间格式为“时:分:秒”）。现分析各小时时段停靠次数最多的楼层（1 楼不参与统计）的部分 Python 代码如下，请在划线处填入合适的代码。

```
时间,楼层
01:17:00,6
01:33:00,6
01:45:00,5
02:11:00,5
02:23:00,6
02:25:00,5
```

```
#导入模块，代码略
df=pd.read_csv("data.csv")
df.insert(0,"小时","")
for i in df.index:
    t=df.at[i,"时间"]
    __①__=int(t[0:2])
xs=[]
cnt=[]
for i in range(24):
    dft=df[df["小时"] = = i]
    if len(dft)>0 :
        dfg=dft.groupby(__②__,as_index=False).count()
        dfg=__③__#筛出除 1 楼外楼层
        dfg=dfg.sort_values(__④__)
        xs.append(i)
        cnt.append(dfg["楼层"].values[0])        #获取第 i 小时停靠最多的楼层数
```

【课后练习 1】Series 和 DataFrame

1. 有如下 Python 程序段：

```
import pandas as pd
s=pd.Series(["猕猴桃","水蜜桃",1,2])
print(s[1])
```

执行该程序段后，输出的结果是

A.1　　B."猕猴桃"　　C."水蜜桃"　　D.水蜜桃

2. 有如下 Python 程序段：

```
import pandas as pd
s1=pd.Series([203,199,288],index=["蔷薇花","茉莉花","向日葵"])
for i in s1.index:
    print(i)
```

执行该程序段后，输出的结果是

A.	B.	C.	D.
0 1 2	蔷薇花 茉莉花 向日葵	203 199 288	蔷薇花　203 茉莉花　199 向日葵　288

3.有如下 Python 程序段：

```
import pandas as pd
s=pd.Series([179,178,182],index=["s01","s02","s03"])
for i in s.values:
    print(i)
```

执行该程序段后，输出的结果是

A.	B.	C.	D.
0 1 2	s01 s02 s03	179 178 182	s01 179 s02 178 s03 182

4. 有如下 Python 程序段：

```
import pandas as pd
data=[[6,7,8],[2,1,1]]
s=pd.Series(data,index=["小红","小明"])
s[1]=[9,8,5]
print(s["小明"])
```

执行该程序段后，输出的结果是

A.1　[2,1,1]　　B.小明　[2,1,1]　　C.小明　[9,8,5]　　D. [9,8,5]

5. 有如下 Python 程序段：

```
import pandas as pd
grade=pd.Series([112,122,97],index=["数学","语文","信息"])
grade[1]=132
print(grade)
```

执行该程序段后，输出"语文"对应的值为

A.112　　B.122　　C.132　　D.97

6. 有如下 Python 程序段：

```
import pandas as pd
k=pd.Series([123,34,135,51],index=["qwef ","2af ","wefe","weqr"])
k["2af "]=134
print(k[1]+k[3])
```

执行该程序段后，输出的结果是

A.85　　B.185　　C.175　　D.258

7. 有如下 Python 程序段：

```
import pandas as pd
m=pd.Series([63,74,65,71],index=["a","b","c","d"])
m["c"]=77
n=m.max()//2%3
print(n)
```

执行该程序段后，输出的结果是

A.0　　B.1　　C.2　　D.3

8. 有如下 Python 程序段：

```
import pandas as pd
s1=pd.Series([10,6,5,15,1,4])
s2=pd.Series([20,2,10,5,3,0])
res=0
for i in range(len(s1)):
    if s1[i]%5==0 and s2[i]%5==0:
        res+=1
print(res)
```

执行该程序段后，输出的结果是

A.1　　B.2　　C.3　　D.5

9. 有如下 Python 程序段：

```
import pandas as pd
data=[[1,2,3],[4,5,6],[7,8,9]]
df1=pd.Series(data)
df1[1][2]=0
print(df1)
```

下列说法正确的是

A. 数据“6”被修改为“0”　　B. 执行该程序段后，列表 data 的值并没有被改变

C. 对象 df1 是一个二维数据结构　D.执行语句“print(df1.values[2])”,输出的结果是 3

10.某 DataFrame 对象 df 中包含“id-code”、“name”、“age”等数据列和 5 个数据行，下列语句中，能输出 df 对象中“name”数据列所有数据的是

A.print(df["name"])　　B.print(df.tail(2))　　C.print(df.columns)　　D.print(df[1:2])

11.有如下 Python 程序段：

```
import pandas as pd
data=[[1,2,3],[3,4,5],[6,7,8]]
df1=pd.DataFrame(data,index=["a","b","c"],columns=["e","f","g"])
print(df1)
```

下列说法正确的是

A.df1 对象的列标题是 a、b、c　　B.print(df1["e"]) 用于输出“e”行的数据

C.df1.drop("a",axis=1) 用于删除“a”行数据　D.df1["d"]=[4,2,6] 用于增添 d 列数据

12.有如下 Python 程序段：

```
import pandas as pd
data={"姓名":["小杜","小张","小李","小孙"],"性别":["女","女","男","男"],"比赛分数":
      [8,6,9,6]}
df1=pd.DataFrame(data,columns=["姓名 ","性别 ","比赛分数"])
a=df1["比赛分数"].max()
b=df1.比赛分数 .min()
print(a+b)
```

执行该程序段后，输出的结果是

A.9　　B.14　　C.15　　D.17

13.有如下 Python 程序段：

```
import pandas as pd
data={"学号":["2110301","2110302","2110303"],"成绩":[401,450,420]}
df1=pd.DataFrame(data,columns=["学号","成绩"])
print(df1)
```

若要让数据按“成绩”列降序排列，则下列操作正确的是

A.df1.sort_values(["成绩"])　　B.df1.sort_values("成绩",ascending=False)

C.df1.sort_values(data)　　D.df1.sort_values("成绩",ascending=True)

14.小王收集了某时刻全国各省部分城市的 PM2.5 和 AQI 数据保存在“kqzl.xlsx”中，部分数据如图所示。编写 Python 程序，统计当时 AQI 值（AQI 值不含重复数据）最小的 5 个城市。

1	时间	省份	城市	PM2.5	AQI
83	2023.10.11 08:00	甘肃省	酒泉市	22μg/m³	66
84	2023.10.11 08:00	甘肃省	平凉市	71μg/m³	95
85	2023.10.11 08:00	广东省	潮州市	3μg/m³	8
86	2023.10.11 08:00	海南省	海口市	11μg/m³	16
87	2023.10.11 08:00	黑龙江省	齐齐哈尔市	21μg/m³	57
88	2023.10.11 08:00	黑龙江省	哈尔滨市	26μg/m³	63
89	2023.10.11 08:00	湖北省	宜昌市	16μg/m³	28
90	2023.10.11 08:00	湖北省	十堰市	26μg/m³	41

```
import pandas as pd
df=pd.read_excel("kqzl.xlsx")
df1=df.sort_values("AQI",ascending=True)
print(_____)
```

要实现上述功能，程序横线处实现输出 AQI 值最小的 5 个城市，下列语句可行的有

①df1.head(5) ②df1.tail(5) ③df1[0:5] ④df1[0:6] ⑤df1.head()

A.①④　　B.②③　　C.①③⑤　　D.①④⑤

15.文件“disease. csv”中包含“Country”“Cumulative”“Existing”“Cure”等字段以及若干个数据行，执行如下程序段后，对象 disease_data 中的数据将

```
import pandas as pd
disease_data=pd.read_csv("disease.csv")
disease_data.drop("Cumulative",axis=1)
disease_data.sort_values("Existing",inplace=True)  #inplace=True 在原数据上修改
```

A.按“Existing”升序排列　　B.不再包含“Cumulative”数据列

C.减少“Existing”数据列　　D.增加了一个数据行

16.某 DataFrame 对象 score 包含“准考证号”“学校名称”“姓名”“总分”“排名”等数据列，下列语句中，实现以学校为单位，输出各校学生“总分”平均值的是

A.print(score.groupby("学校名称",as_index=False).mean())

B.print(score.groupby("总分",as_index=False).mean())

C.print(score.groupby("学校名称",as_index=False).排名 .mean())

D.print(score.sort_index("学校名称", as_index=False).sum())

【课后练习 2】pandas 综合练习

1. 小李收集了某奶茶门店 2024 年 1 月的销售数据，如图 a 所示。

	A	B	C	D	E	F	G	H
1	日期	订单号	点单状态	饮品名称	品类	单价	数量	金额
2	2024/1/1	1	成功	芋泥厚厚牛乳	芋泥	14		0
3	2024/1/1	1	成功	龙井香青团	轻牛乳	15	1	15
4	2024/1/1	1	成功	杨枝甘露	果茶	15	1	15
5	2024/1/1	2	退单	茉莉奶芙	轻牛乳	15	2	30
6	2024/1/1	2	退单	大叔奶茶	奶茶	11	1	11
7	2024/1/1	3	成功	生椰榴莲	果茶	20	1	20
8	2024/1/1	4	成功	芝士莓莓	果茶	22	1	22
312	2024/1/30	16	成功	酒酿小丸子	奶茶	12	1	12
313	2024/1/30	17	成功	云岭茉莉椰	轻牛乳	12	2	24
314	2024/1/30	18	成功	布蕾脆碗奶芙	奶茶	16	1	16

为统计分析该门店不同品类饮品的销售情况。编写 Python 程序，请回答下列问题：

（1）读取文件，筛选出点单成功的数据，代码如下。

```
import pandas as pd
df=pd.read_excel("salelist.xlsx")
df1=B df[df.点单状态=="成功"]  或 df[df["点单状态"]=="成功"]
```

划线处应填入的代码为_____（单选，填字母）。

A.df[df["金额"]>0]　　B.df[df.点单状态=="成功"]

C.df1[df1["金额"]>0]　　D.df1[df1.点单状态=="成功"]

（2）统计该月不同品类饮品销售数量，并绘制柱形图（图 b），部分 Python 入合适的代码。

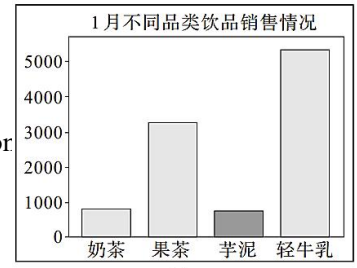


图 b

```
import matplotlib.pyplot as plt
df2=df1.groupby("品类①",as_index=False)
df2=df2.数量 .sum()
plt.figure()
plt.bar(df2.品类,② df2["金额"])
'''设置绘图参数，显示结果如图 b 所示，代码略'''
plt.show()
```

index	品类	日期	订单号	点单状态	饮品名称	单价	数量	金额
0	奶茶	…	…	…	…	…	…	800
1	果茶	…	…	…	…	…	…	3200
2	芋泥	…	…	…	…	…	…	810
3	轻牛乳	…	…	…	…	…	…	5200

2.小明收集了浙江省某些热门景点 3 月 15 日的温度数据，存储在"weather.xlsx"文件中，部分数据如图 a 所示。现利用 pandas 模块对该表进行数据分析，按“地级市”分类统计各市各时间点热门景点的平均温度，最后绘制线形图如图 b 所示。请在划线处填入合适的代码。

	A	B	C	D	E	F	G	H	I	J
1	地级市	景点	08时	11时	14时	17时	20时	23时	02时	05时
2	宁波	松兰山	12	15	16	14	13	12	12	12
3	杭州	宋城	13	18	22	21	17	17	16	15
4	绍兴	沈园	13	20	22	20	17	17	16	15
5	杭州	龙门古镇	12	20	25	24	18	17	15	15
6	绍兴	鲁迅故里	13	20	22	20	17	17	16	15
7	温州	楠溪江	14	21	24	20	15	14	14	14
8	杭州	西湖	14	22	24	24	18	16	16	14
9	宁波	五龙潭	11	20	22	20	13	12	12	12
10	绍兴	西施故里	13	20	24	24	20	18	16	16
11	温州	江心屿	18	21	23	21	17	16	17	16
12	宁波	天一阁	13	18	20	19	16	15	14	14
13	温州	雁荡山	12	18	21	17	13	12	12	12

图 a

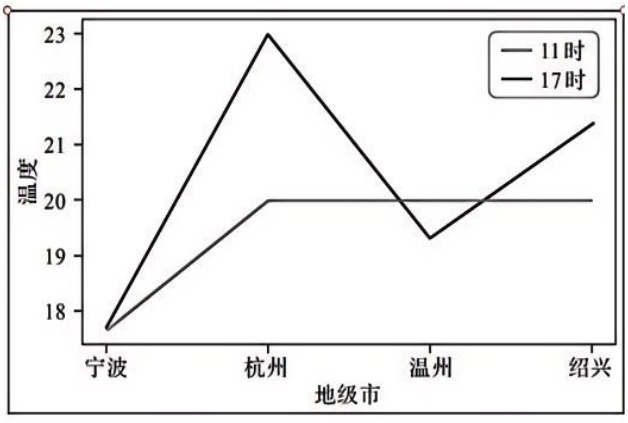
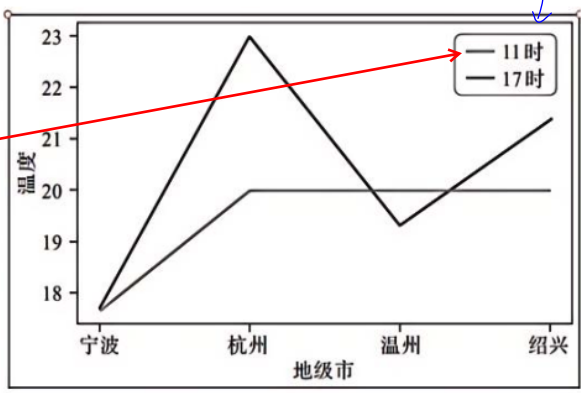


图 b

```
import pandas as pd
import matplotlib.pyplot as plt
plt.rc("font",family="SimSun")
df=pd.read_excel("weather.xlsx")
df_group=df.groupby(___"地级市"_____,as_index=False)
df_ave=df_group.mean()
x=df_ave["地级市"]
y1=df_ave["11 时"]
y2=df_ave["17 时"]
plt.plot(x,y1,label="11 时")
plt.plot(x,y2,_label="17时"_)
plt.xlabel("地级市")
plt.ylabel("温度")
_plt.legend()___#显示图例
plt.show()
```

图例

把groupby分两行代码操作



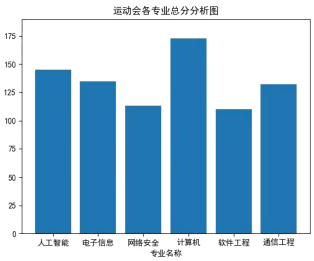
3. 某学院举行运动会，比赛设跳高、100 米等项目，每个项目分男子组和女子组。现要进行报名数据处理和比赛成绩分析。请回答下列问题：

	专业	学号	姓名	性别	项目
0	软件工程	S10111	钱*然	男	跳高
1	软件工程	S20212	石*如	女	100米
2	软件工程	S30212	宋*尘	男	100米
…	…	…	…	…	…
26	软件工程	S10622	王*娟	女	400米
27	软件工程	S10919	王*栩	女	200米
28	软件工程	S30110	叶*涛	男	100米

图 a

专业	学号	姓名	性别	项目	名次	得分
软件工程	S30110	叶*涛	男	跳远	5	
计算机	D30101	朱*奕	女	100米	2	
电子信息	C10522	赵*宇	男	铁饼	4	
…	…	…	…	…	…	
电子信息	C30211	郑*珥	女	跳远	15	
人工智能	A20109	裘*晨	女	跳高	16	
人工智能	A20109	裘*晨	女	跳远	7	

图 b



图c

（1）运动会报名规则为：对于每个项目的男子组和女子组，每个专业最多各报 5 人(如“软件工程”专业在男子跳高项目中最多报 5 人)。软件工程专业的报名数据保存在 DataFrame 对象 df 中,如图 a 所示。若要编写 Python 程序检查该专业男子跳高项目报名是否符合规则,下列方法中，正确的是_____（单选，填字母）。 3次筛选： 1软件工程(已做) 2.男子 3.跳高 （筛选顺序可变）

A. 从 df 中筛选出性别为“男”的数据 dfs，再从 dfs 中筛选出项目为“跳高”的数据，判断筛选出的数据行是否超过 5 行✓

B. 对 df 中数据按性别排序并保存到 dfs 中，再从 dfs 中筛选出项目为“跳高”的数据，判断筛选出的数据行是否超过 5 行 ×

C.从 df 中筛选出项目为“跳高”的数据 dfs，判断 dfs 中是否有连续 5 行以上的男生数据 ×

（2）运动员比赛成绩的部分数据如图 b 所示。根据已有名次计算得分，1 名至第 8 名分别计 9,7,6,5,4,3,2,1 分,第 8 名之后计 0 分。实现上述功能的部分 Python 程序如下，请在程序中划线处填入合适的代码。

```
import pandas as pd
import matplotlib.pyplot as plt
'''读取如图 b 所示数据，保存到 DataFrame 对象 df1 中，代码略'''
f=[9,7,6,5,4,3,2,1]  没有8分的情况，所以不是线性关系，只能用列表进行对应
for i in range(0,len(df1)):
    rank=df1.at[i,"名次"]        #通过行、列标签取单个值
    score=0
    if rank<=8:
        score=f[rank-1]
        df1.at[i,"得分"]=score
```

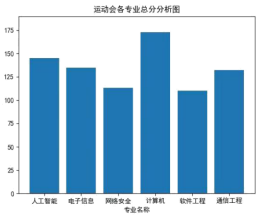
（3）根据上述 df1 中的得分数据，统计各专业总分，绘制如图 c 所示的柱形图，实现该功能的部分

Python 程序如下：

```
#分组求和
df2=df1.groupby("___专业___",as_index=False).sum()
```

问题：

专业	学号	姓名	性别	项目	名次	得分
软件工程	S30110	叶*涛	男	跳远	5	
计算机	D30101	朱*奕	女	100 米	2	
电子信息	C10522	赵*宇	男	铁饼	4	
…	…	…	…	…	…	…
电子信息	C30211	郑*珥	女	跳远	15	
人工智能	A20109	裘*晨	女	跳高	16	
人工智能	A20109	裘*晨	女	跳远	7	




```
#设置绘图参数，代码略
plt.bar(x,y)  #绘制柱形图
```

①请在程序中划线处填入合适的代码。

②程序的方框中应填入的正确代码为___B___

A.x=df1["专业"] y=df1["总分"]	B.x=df2["专业"] y=df2["得分"]	C.df1["专业"]="专业" df1["总分"]="总分"	D.df2["专业"]="专业" df2["得分"]="得分"

4. 学校气象站的小江同学收集了本地 2022 年全年的天气数据，数据按日期顺序存储在“tqsj.xlsx”文件中，部分数据如图 a 所示。为分析全年各月份天气情况，编写 Python 程序，请回答下列问题：

（1）计算 2022 年每天的温差（最高气温℃-最低气温℃），找出最大温差，如有相同温差，输出所有符合要求的日期，输出结果如图 b 所示。程序代码如下，请在划线①②处填入相应的代码。

	A	B	C	D	E
1	日期	最高气温℃	最低气温℃	天气	风向
2	2022-01-01	14	7	晴	西南风 1级
3	2022-01-02	18	3	晴	北风 1级
4	2022-01-03	17	5	晴	东北风 1级
5	2022-01-04	19	8	阴	西风 1级
6	2022-01-05	1	0	阴	西北风 1级
7	2022-01-06	14	6	多云	东北风 2级
8	2022-01-07	15	6	多云	东北风 1级
9	2022-01-08	14	7	小雨	南风 1级
10	2022-01-09	9	6	阴	北风 1级
…	…	…	…	…	…
362	2022-12-27	11	5	雾	东北风 1级
363	2022-12-28	12	5	多云	北风 1级
364	2022-12-29	9	6	雾	西北风 1级
365	2022-12-30	12	6	晴	北风 2级
366	2022-12-31	13	3	晴	西风 1级

图 a

```
import pandas as pd
df=pd.read_excel("tqsj.xlsx")
df["温差"]=①___df.最高气温℃-df.最低气温℃___
df_wch=df.sort_values("温差",ascending=False)  df["最高气温℃"]-df["最低气温℃"]
max=df_wch.温差.values[0]
rq=[]
for i in df_wch.index:
    if df_wch.at[i,"温差"]==max:
        rq.append(②___df["日期"] 或 df.日期___)
print("最大温差：",max,"℃")
print("日期：",rq)
```

最大温差： 18 ℃
日期： ['2022-09-19', '2022-12-22']

图 b

（2）统计各月平均温差并绘制线形图，部分 Python 程序如下，请

index	日期	温差
0	2026-01-01	…
1	2026-01-02	
…	…	…
30	2026-01-31	…
31	2026-02-01	…
…	…	…
58	2026-02-28	…
59	2026-03-01	

begin

```
avewch = [0]*12
mdays = [31,28,31,30,31,30,31,31,30,31,30,31]  #2022 年每月天数
begin=0
for m in range(12):
    total=0
    for d in range(begin,③___begin+mdays[m]___):
        total+=df.at[d,"温差"]
    avewch[m]= ④___total/m___
    ⑤___begin=begin+mdays[m]___
x=[i+1 for i in range(12)]
y=avewch
plt.plot(x,y,label="平均温差")
#设置绘图参数，显示如图 c 所示，代码略
```

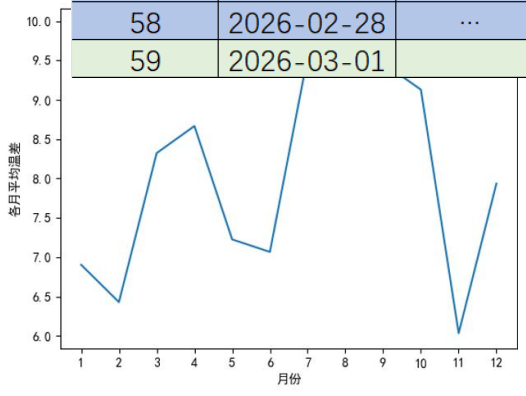


图 c

5. 某校高三首考后，汇总了学生 7 选 3 成绩以及次考科目弃考情况（注意：1 表示次考弃考，0 表示参加次考），ckqk.xlsx 文件部分数据图 a 所示，现要统计各班人均弃科目考门数和各科次考弃考比例，编写如下 Python 程序。

	A	B	C	D	E	F	G	H	I	J	K
1	班级	姓名	科目1	科目2	科目3	科目1赋分	科目2赋分	科目3赋分	弃考科目1	弃考科目2	弃考科目3
2	1	陈[illegible]侯	物理	化学	生物	85	92	93	0	0	0
3	1	陈[illegible]炜	物理	化学	生物	98	92	99	1	0	1
45	2	鲍[illegible]漪	政治	生物	地理	82	74	60	0	0	0
46	2	邓[illegible]	政治	历史	地理	94	93	85	0	0	0
86	3	包[illegible]妍	物理	地理	技术	93	91	98	1	0	1
87	3	陈[illegible]溯	物理	地理	技术	93	84	98	1	0	1

图 a

	班级	弃考门数
0	1	1.16
1	2	1.10
2	3	0.75
3	4	1.03
4	5	1.19
5	6	1.54
6	7	1.42

图 b

请回答下列问题：

（1）下列代码读取 ckqk.xlsx 文件，输出各班每人平均弃考门数，输出格式如图 b 所示，划线处填写的代码。

```
import pandas as pd
df = pd.read_excel('ckqk.xlsx')
df['弃考门数'] = df[['弃考科目_1','弃考科目_2','弃考科目_3']].sum(axis = 1)
dfg = df.groupby("班级",as_index=False).弃考门数.mean()
print(dfg)
```

新增加1列， 弃考门数

（2）统计 7 选 3 每门科目选考人数和弃考人数，并计算各科目弃考比例，Python 程序如下，请在划线处填写合适的代码。

```
courses = {}
for i in df.index :
    for k in range(1,4):
        subject = df.at[i,'科目%d' %k]
        if subject not in courses :
            courses[subject] = [1,0] #[1,0]表示当前为本科目第 1 个人，弃考人数初值置 0
        else:
            courses[subject][0] += 1
        abandon = df.at[i,'弃考科目%d' %k]
        if abandon == 1 :
            courses[subject][1]+=1
dfs = pd.DataFrame(courses)
dfs = dfs.T
dfs = dfs.rename(columns={0:'总人数',1:'弃考人数'})
dfs['弃考比例'] = round(dfs['弃考人数'] / dfs['总人数'] * 100,1)
dfs = dfs.sort_values('弃考比例',ascending = False)
```

字典数据类型 courses:{"物理":[5,1],"化学":[6,2],"生物":[7,3]....}

df.at[i,"科目1"] df.at[i,"科目2"] .....

index	物理	化学	生物	政治	历史	地理	技术
0	5	8	7	2	9	6	3
1	1	2	3	4	5	6	3

dfs.T dfs.rename()

index	0	1
物理	5	1
化学	8	2
生物	7	3
政治	2	4
历史	9	5
地理	6	6
技术	3	3

→

index	总人数	弃考人数
物理	5	1
化学	8	2
生物	7	3
政治	2	4
历史	9	5
地理	6	6
技术	3	3

字典方式形成dfs数据

行列转置

（3）编写代码绘制如图 c 所示图表，则 7 选 3 科目中次考弃考比例超 30%的有 ① 5 门。为实现该功能，请在下面划线处填入合适代码。

```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei']
plt.figure('chart',figsize=(6,4))
plt.title('各科次考弃考比例分析')
plt.bar(dfs.index, dfs.弃考比例 ,label='弃考比例')
plt.ylabel('各科弃考百分比')
plt.xlabel('7 选 3 科目')
plt.legend()
plt.show()
```

② dfs["弃考比例"]

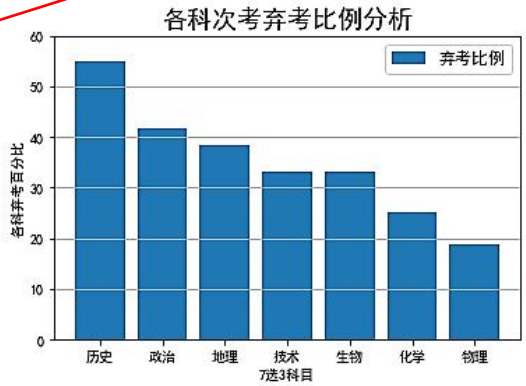


图 c

思考：不是很合理的数据结构

班级 姓名 物理 化学 生物 政治 历史 地理 技术...