文本分析

李田雨

第3讲

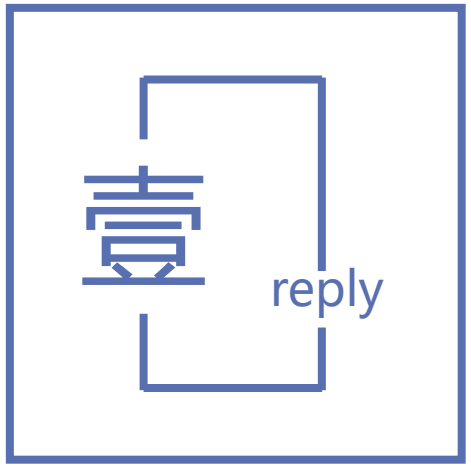
再见Excel，你好Pandas！

在上一节我们了解了Series的常用属性和方法以及DataFrame的数据获取和遍历。

在做数据分析的时候，Excel是我们最常用的工具，但是当数据量比较大的时，Excel光把数据文件打开就要很久很久，那么利用Pandas就会非常高效。

本节我们将开始新的征程，学习如何利用Pandas读取和写入Excel。

I Love You



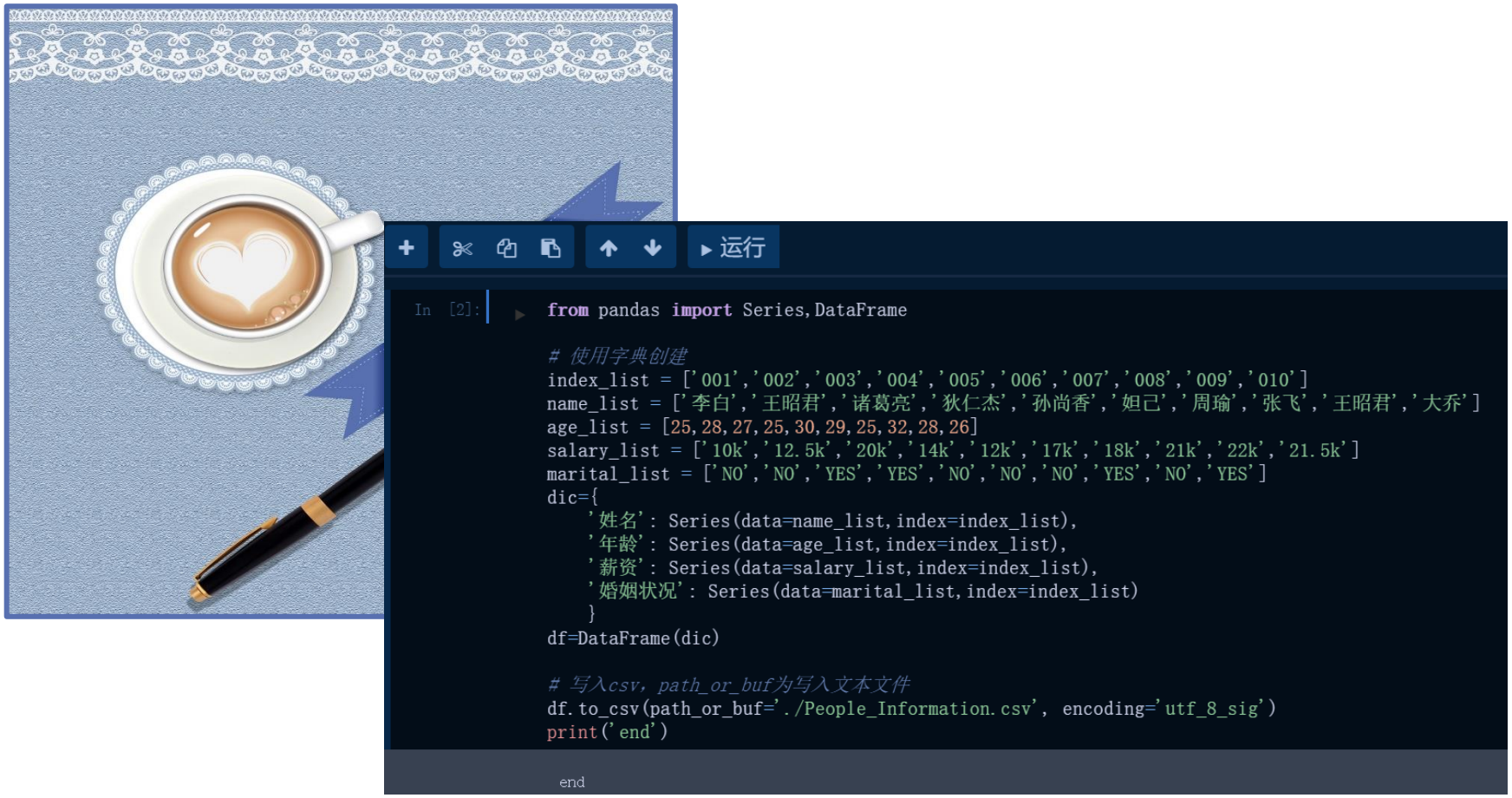
数据的写入

01 数据的写入

我们可以将数据写入到文件中进行永久性的保存，支持的文件格式有HTML、CSV、JSON、Excel。

csv是最为常见的以纯文本文件存储数据文件的格式，它的优点是通用性很强，不受操作系统以及具体的软件的限制。我们以写入csv为例，看一下pandas是如何是将数据写入csv文件中。

01 数据的写入



```
In [2]:
from pandas import Series,DataFrame

# 使用字典创建
index_list = ['001','002','003','004','005','006','007','008','009','010']
name_list = ['李白','王昭君','诸葛亮','狄仁杰','孙尚香','妲己','周瑜','张飞','王昭君','大乔']
age_list = [25,28,27,25,30,29,25,32,28,26]
salary_list = ['10k','12.5k','20k','14k','12k','17k','18k','21k','22k','21.5k']
marital_list = ['NO','NO','YES','YES','NO','NO','NO','YES','NO','YES']
dic={
    '姓名': Series(data=name_list,index=index_list),
    '年龄': Series(data=age_list,index=index_list),
    '薪资': Series(data=salary_list,index=index_list),
    '婚姻状况': Series(data=marital_list,index=index_list)
    }
df=DataFrame(dic)

# 写入csv，path_or_buf为写入文本文件
df.to_csv(path_or_buf='./People_Information.csv', encoding='utf_8_sig')
print('end')

end
```

在上面的代码里，我们创建了一个DataFrame，接着通过to_csv()方法将DataFrame保存为csv文件。从结果中可以发现，to_csv()保存数据时，df的行索引作为一列被输出到csv文件中。

01 数据的写入

如何在保存csv文件的时候，不存储DataFrame的行索引信息呢，我们看下面的解决方法。

代码片段

```
1 df.to_csv(path_or_buf='./People_Information.csv',index=False,enc
```

在to_csv方法中将参数index设置为False就可以不存储DataFrame的行索引信息。

在to_csv方法参数中设置encoding='utf_8_sig'，此举为何呢？

I Love You

濮	骞撮			濠濮荤跺
浆25	10k		NO	
		28	12.5k	NO
璇歌浜		27	20k	YES
浠＿艾25	14k		YES	
瀛灏棣		30	12k	NO
濡插箒		29	17k	NO
—		25	18k	NO
寮 椋		32	21k	YES
		28	22k	NO
澶т		26	21.5k	YES

因为to_csv()方法生成csv文件时，打开文件时都是乱码，encoding参数设置“utf_8_sig”后乱码就会消失。

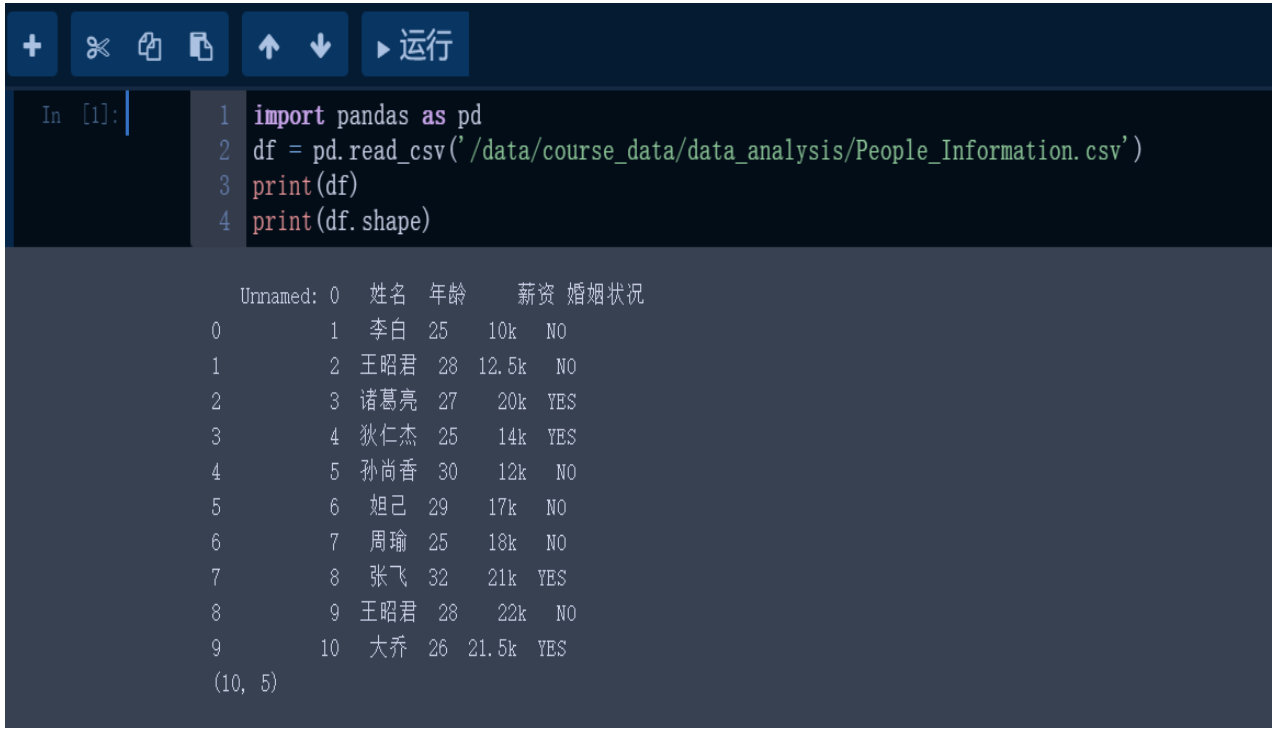
I Love You

贰

reply

数据的读取

02 数据的读取



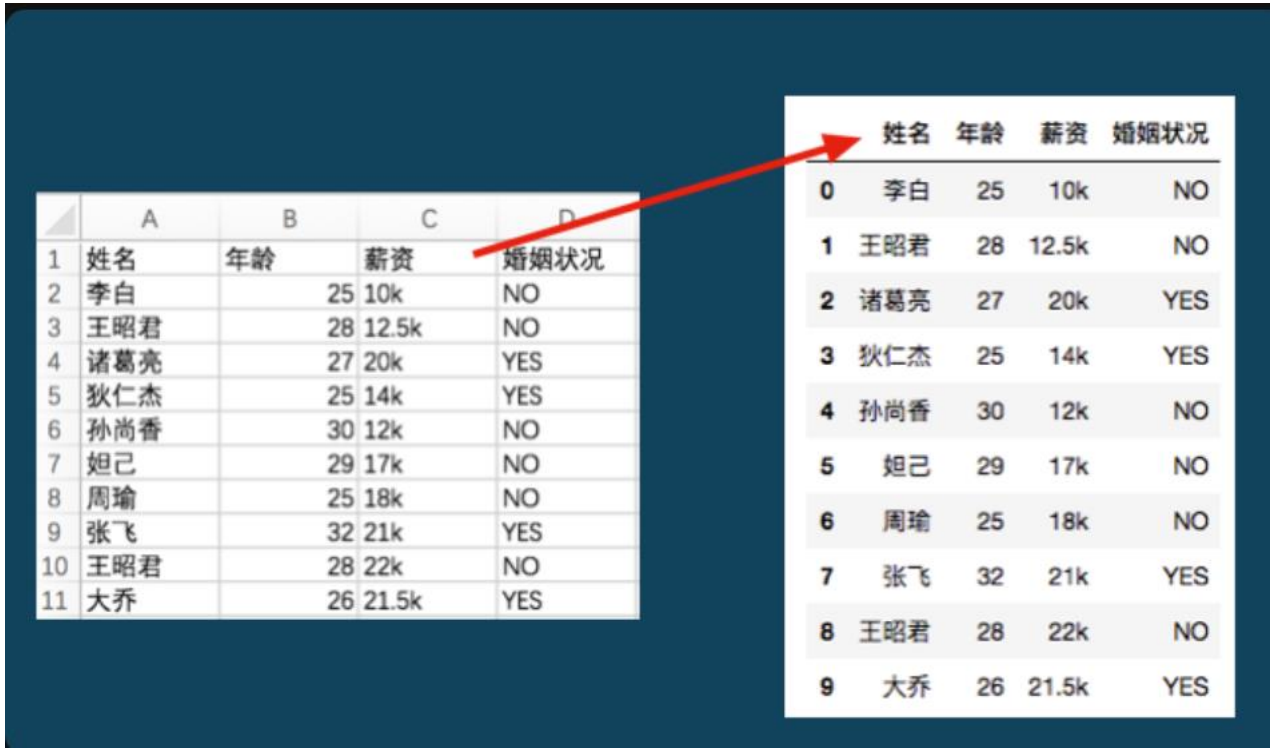
数据的存储我们发现很简单，调用to_csv()后设置文件存储路径后就可以了。

人生就是要反复的折腾，现在我们看看是如何从csv文件将数据读取出来的。

运行下面的代码，看一下与上面保存的数据是否一致。

02 数据的读取

根据结果我们可以看出，调用read_csv()方法并传入文件的路径，就可以将数据读取出来并且是DataFrame类型。



	A	B	C	D
1	姓名	年龄	薪资	婚姻状况
2	李白	25	10k	NO
3	王昭君	28	12.5k	NO
4	诸葛亮	27	20k	YES
5	狄仁杰	25	14k	YES
6	孙尚香	30	12k	NO
7	妲己	29	17k	NO
8	周瑜	25	18k	NO
9	张飞	32	21k	YES
10	王昭君	28	22k	NO
11	大乔	26	21.5k	YES

	姓名	年龄	薪资	婚姻状况
0	李白	25	10k	NO
1	王昭君	28	12.5k	NO
2	诸葛亮	27	20k	YES
3	狄仁杰	25	14k	YES
4	孙尚香	30	12k	NO
5	妲己	29	17k	NO
6	周瑜	25	18k	NO
7	张飞	32	21k	YES
8	王昭君	28	22k	NO
9	大乔	26	21.5k	YES

还可以看出，read_csv()默认会将文件中的第一行作为数据的列索引。

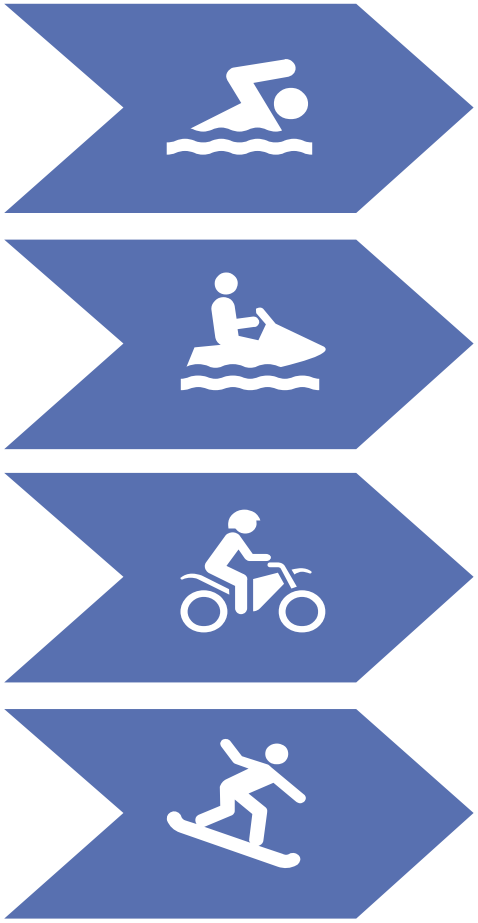
如果csv文件的第一行或者其他行不满足我们的需求时，我们就不能再屈服在它的淫威下了，我们要自己修改。

I Love You

aa	bb	bb	dd	ee	ff
ID	Type	Title	FirstName	MiddleName	LastName
1	Employee	NULL	Ken	J	Sánchez
2	Employee	NULL	Terri	Lee	Duffy
3	Employee	NULL	Roberto	NULL	Tamburello
4	Employee	NULL	Rob	NULL	Walters
5	Employee	Ms.	Gail	A	Erickson
6	Employee	Mr.	Jossef	H	Goldberg
7	Employee	NULL	Dylan	A	Miller
8	Employee	NULL	Diane	L	Margheim
9	Employee	NULL	Gigi	N	Matthew
10	Employee	NULL	Michael	NULL	Raheem
11	Employee	NULL	Ovidiu	V	Cracium
12	Employee	NULL	Thierry	B	D'Hers
13	Employee	Ms.	Janice	M	Galvin
14	Employee	NULL	Michael	I	Sullivan
15	Employee	NULL	Sharon	B	Salavaria

当csv数据的第一行是一条脏数据，不符合我们要求，如上图。

02 数据的读取



可以利用read_csv()中的header参数进行选择哪一行作为我们的列索引。

代码片段

```
1 import pandas as pd
2 people = pd.read_csv('/data/course_data/data_analysis/People1.csv
3 print(people.columns)
```

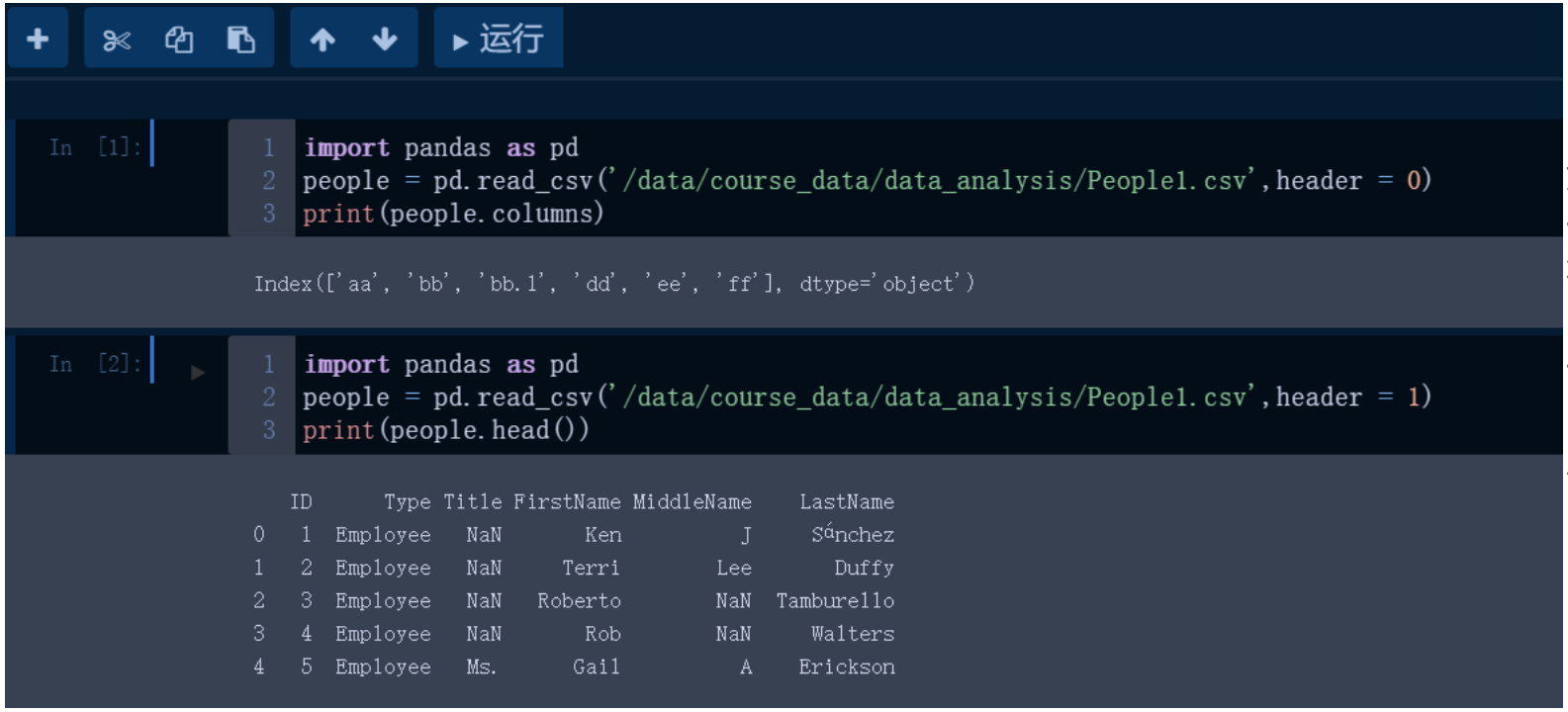
代码片段

```
1 import pandas as pd
2 people = pd.read_csv('/data/course_data/data_analysis/People1.csv
3 print(people.head())
```

将上面的两个代码块分别放到运行框中运行，对比结果。

02 数据的读取

read_csv()的header参数默认是0，取第一行的值，可以根据具体的要求设置header的值来确定列索引。



```
In [1]:
1 import pandas as pd
2 people = pd.read_csv('/data/course_data/data_analysis/People1.csv',header = 0)
3 print(people.columns)
```

```
Index(['aa', 'bb', 'bb.1', 'dd', 'ee', 'ff'], dtype='object')
```

```
In [2]:
1 import pandas as pd
2 people = pd.read_csv('/data/course_data/data_analysis/People1.csv',header = 1)
3 print(people.head())
```

```
   ID      Type Title FirstName MiddleName    LastName
0   1  Employee   NaN       Ken          J     Sánchez
1   2  Employee   NaN     Terri        Lee       Duffy
2   3  Employee   NaN   Roberto        NaN  Tamburello
3   4  Employee   NaN       Rob        NaN     Walters
4   5  Employee   Ms.      Gail          A    Erickson
```

如果都不满足的你的要求，可以将header设置为None，列索引值会使用默认的1、2、3、4，之后在自行设置。

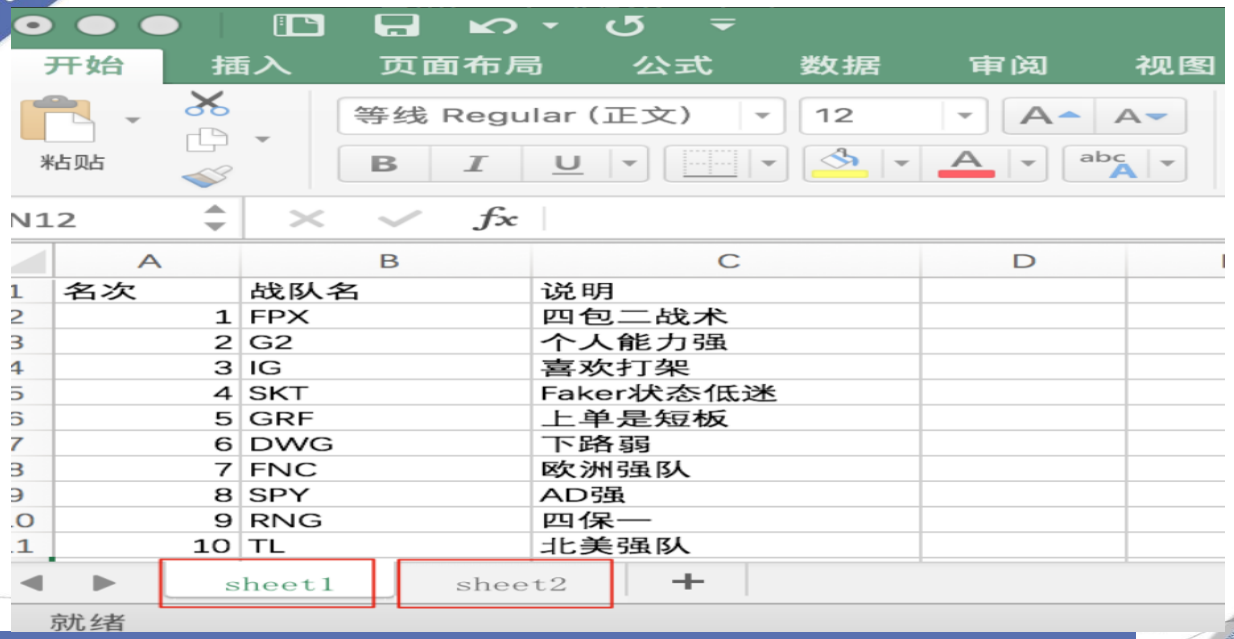
江湖秘籍

I Love You

当指定了header的值，读出来的数据就是从该行开始向下切片，该行以上的数据会被忽略。

一个Excel文件可以创建多个表，然后在不同的表中存储不同数据，这种形式的文件很常见。但是要注意csv文件不存在多个sheet的问题。

I Love You



	A	B	C	D
1	名次	战队名	说明	
2	1	FPX	四包二战术	
3	2	G2	个人能力强	
4	3	IG	喜欢打架	
5	4	SKT	Faker状态低迷	
6	5	GRF	上单是短板	
7	6	DWG	下路弱	
8	7	FNC	欧洲强队	
9	8	SPY	AD强	
10	9	RNG	四保一	
11	10	TL	北美强队	

所以，如果是Excel文件就需要考虑，如何从Excel中读取出其中的一个表。

02 数据的读取

text

text

text

text

Excel文件的读取和csv的读取方式相似，read_csv()读取csv文件，read_excel()读取Excel文件。

```
1 import pandas as pd
2 sheet = pd.read_excel('/data/course_data/data_analysis/sheet.xls
3 print(sheet.head())
```

将上面的代码，复制到下面运行，对比一下Excel和csv的文件读取。

```
In [1]:
1 import pandas as pd
2 sheet = pd.read_excel('/data/course_data/data_analysis/sheet.xlsx')
3 print(sheet.head())
```

```
   ID Name
0   0   zs
1   1   li
```

02 数据的读取

```
In [1]:
1 import pandas as pd
2 sheet1 = pd.read_excel('/data/course_data/data_analysis/sheet.xlsx',sheet_name='sheet1')
3 print(sheet1.head())
4
5 sheet2 = pd.read_excel('/data/course_data/data_analysis/sheet.xlsx',sheet_name='sheet2')
6 print(sheet2.head())
```

```
   ID Name
0   0   zs
1   1   li
   ID  age
0   0   18
1   1   19
```

to_csv()会比to_excel()少一个sheet_name的参数，这个参数就是可以指定表的名字。

在上面的代码里，我们引入了带有两个表的sheet.xlsx的Excel文件，两个表名分别为'sheet1'，'sheet2'，然后我们通过指定sheet_name的值，获取不同表中的数据。

01 本节总结


I Love You


第三课
知识点总结
数据的写入
永久性保存，支持的文件根式有HTML、CSV、JSON、 Excel
to_csv()的path_ or_ buf参数:需要指定的文件的本地路径
to_ csv()的index参数: 没置为False就可以不存储Dataframe的行索引信息
to_csv()的encoding参数:没置“utf_8_sig"后解决文件打开后乱码
数据的读取
使用read_csv()、read_excel()等方法读取
read_csv()第一个参数填文件的路径
默以会讲文件中的第一行作力数据的列索引, 使用header参数没置
(指定行位置索引、None)
一个Excel文件可以存多个表，一个csv只有一个表
读取某个表时，使用sheet_name指定

数据的写入

I Love You

数据写入

1.永久性保存，支持的文件格式有HTML,CSV,JSON,EXCEL

2.to_csv()的path_or_buf参数：需要指定的文件的本地路径

3.to_csv()的index参数：设置为False就可以不储存DataFrame的行索引信息

4.to_csv()的encoding参数：设置“utf_8_sig”后解决文件打开后乱码

数据的读取

数据读取

1.使用read_csv(),read_excel等方法读取

2.read_csv()第一个参数填文件的路径

3.默认会将文件中的第一行作为数据的列索引，使用header参数设置（指定行位置索引，None）

4.一个Excel文件可以存很多个表，一个CSV只有一个表

5.读取某个表时，使用sheet_name指定

练习一

I Love You

组建自己的球队

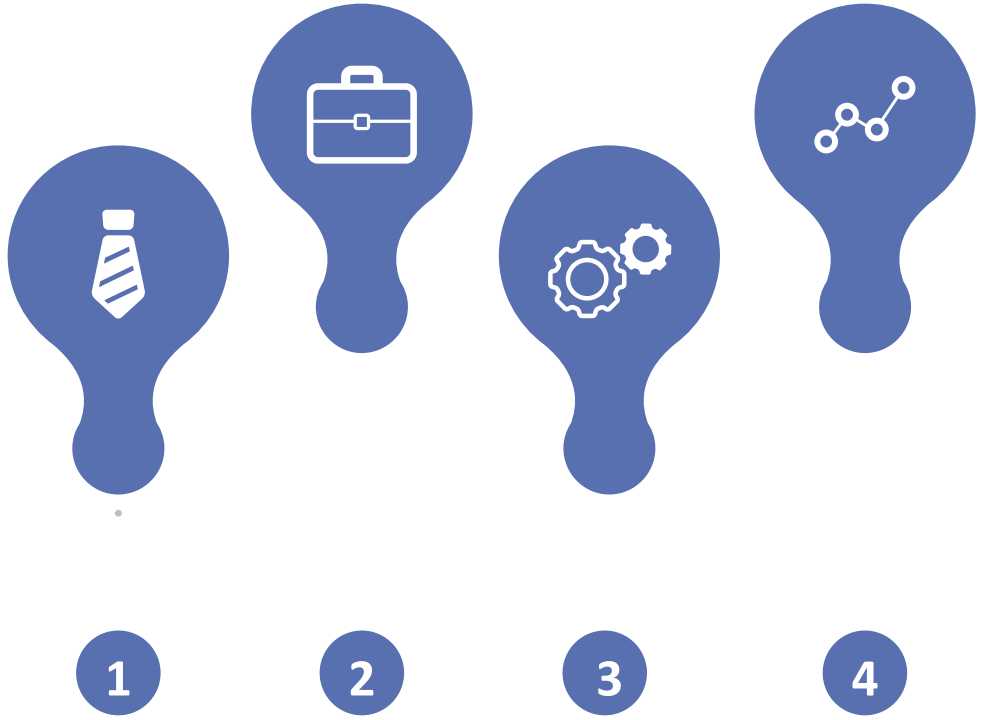
题目要求

从数据集中选取5个球员，组成自己的球队。

练习一

分析过程

1. 球员信息存储在csv中，路径为/data/course_data/data_analysis/players.csv
2. 打印前5条了解数据的基本信息
3. 随机获取5条数据
4. 代码实现



练习一

代码实现


```
In [1]:
1  import pandas as pd
2  import random
3  # 1. 读取数据
4  players = pd.read_csv('/data/course_data/data_analysis/players.csv')
5  # 2. 打印前5条了解数据的基本信息
6  print(players.head())
7  # 3. 随机获取5条信息
8  index_list = players.index.tolist()
9  for i in range(0,5):
10     value = index_list[random.randint(0,len(index_list))]
11     msg = players.iloc[value]
12     print(msg)
```

练习一

```
              player  height  weight                          college  born
0  Curly Armstrong      180      77              Indiana University  1918
1     Cliff Barker      188      83            University of Kentucky  1921
2    Leo Barnhorst      193      86          University of Notre Dame  1924
3       Ed Bartels      196      88  North Carolina State University  1925
4      Ralph Beard      178      79            University of Kentucky  1927

    birth_city birth_state
0          NaN         NaN
1    Yorktown      Indiana
2          NaN         NaN
3          NaN         NaN
4  Hardinsburg    Kentucky
player              Russell Cross
height                        208
weight                         97
college         Purdue University
born                         1961
birth_city                Chicago
birth_state              Illinois
Name: 1599, dtype: object
player                   Henry Dickerson
height                               193
weight                                86
college         University of Charleston
born                                1951
birth_city                        Berkley
birth_state                 West Virginia
Name: 1080, dtype: object
player          Joe Graboski
height                   201
weight                    88
college                  NaN
born                    1930
birth_city               NaN
birth_state              NaN
Name: 65, dtype: object
player                   Jason Sasser
height                            201
weight                            102
college         Texas Tech University
born                             1974
birth_city                     Denton
birth_state                     Texas
Name: 2501, dtype: object
player                                          Allan Bristow
height                                                    201
weight                                                     95
college         Virginia Polytechnic Institute and State Unive...
born                                                      1951
birth_city                                            Richmond
birth_state                                           Virginia
Name: 980, dtype: object
```

运行结果

练习二

I Love You

进军好莱坞

题目要求

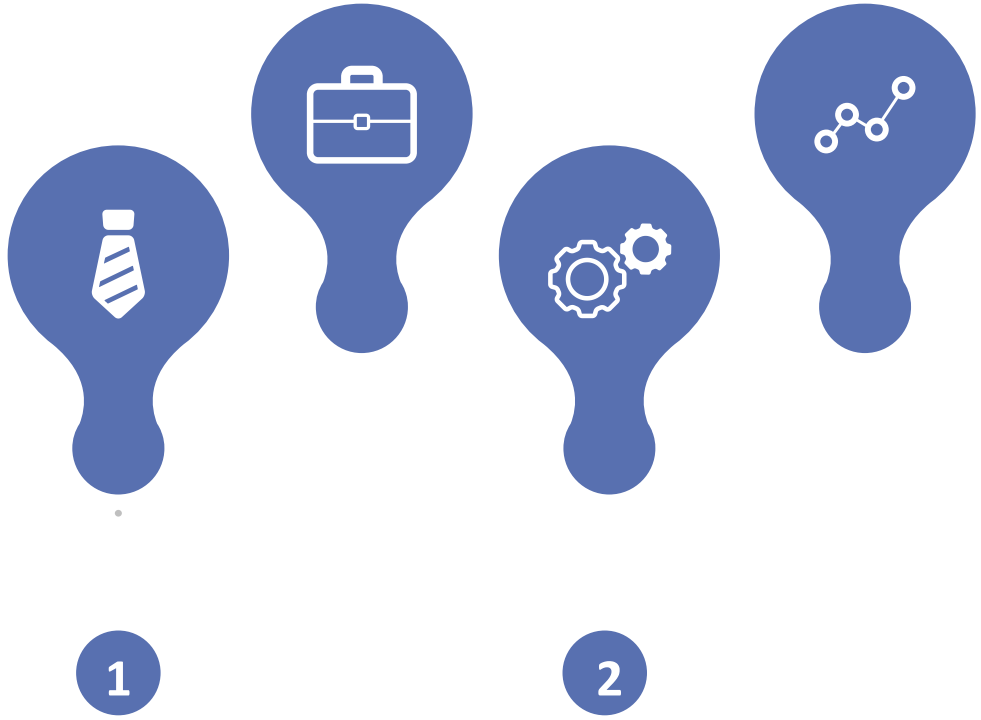
在这个练习中，我们会读取好莱坞电影信息的csv文件，并统计出数据集中共有多少个导演。

练习二

分析过程

1. 电影信息存储在csv中，路径为/data/course_data/data_analysis/movie_data.csv

2. 获取导演名字信息并算出一共多少个导演。



练习二

代码实现


```
In [1]:
```

```python
import pandas as pd
import random
# 1. 读取数据
movie = pd.read_csv('/data/course_data/data_analysis/movie_data.csv')
# 2. 了解数据的基本信息
print(movie.head())
# 3. 获取导演列信息,并转成list
directors = movie['director_name'].tolist()
# 4. 去重后获取个数
num = set(directors)
print(len(num))
```

练习二

```
   Unnamed: 0  color      director_name  num_critic_for_reviews  duration  \
0           0  Color      James Cameron                   723.0     178.0
1           1  Color     Gore Verbinski                   302.0     169.0
2           2  Color         Sam Mendes                   602.0     148.0
3           3  Color  Christopher Nolan                   813.0     164.0
4           5  Color     Andrew Stanton                   462.0     132.0

   director_facebook_likes  actor_3_facebook_likes      actor_2_name  \
0                      0.0                   855.0  Joel David Moore
1                    563.0                  1000.0     Orlando Bloom
2                      0.0                   161.0      Rory Kinnear
3                  22000.0                 23000.0    Christian Bale
4                    475.0                   530.0   Samantha Morton

   actor_1_facebook_likes        gross  ... num_user_for_reviews language  \
0                  1000.0  760505847.0  ...               3054.0  English
1                 40000.0  309404152.0  ...               1238.0  English
2                 11000.0  200074175.0  ...                994.0  English
3                 27000.0  448130642.0  ...               2701.0  English
4                   640.0   73058679.0  ...                738.0  English

  country  content_rating       budget  title_year  actor_2_facebook_likes  \
0     USA           PG-13  237000000.0      2009.0                   936.0
1     USA           PG-13  300000000.0      2007.0                  5000.0
2      UK           PG-13  245000000.0      2015.0                   393.0
3     USA           PG-13  250000000.0      2012.0                 23000.0
4     USA           PG-13  263700000.0      2012.0                   632.0

  imdb_score  aspect_ratio  movie_facebook_likes
0        7.9          1.78                 33000
1        7.1          2.35                     0
2        6.8          2.35                 85000
3        8.5          2.35                164000
4        6.6          2.35                 24000

[5 rows x 29 columns]
1659
```

运行结果


I Love You
THANK YOU