

pandas 数据处理

1. 在 python 中引入 pandas 模块的方法:

```
import pandas as pd
```

2. **Series**: Series 是一种一维的数据结构, 包含一个数组的数据(values)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。



(1) **创建 Series 对象**: 列表、字典等可以用来创建 Series 数据结构, 与列表不同的是, Series 的索引可以指定, 类型可以为字符串型。

(2) **修改 Series 对象中的数据**: 通过索引可以选取 Series 对象中的值, 通过赋值语句可以修改 Series 对象中的值。

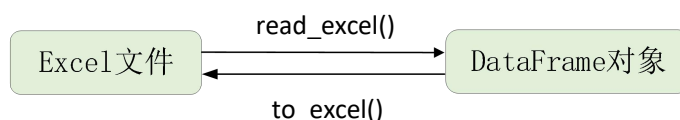
(3) **程序案例**

1. 创建 Series 对象		2. 输出 Series 对象中指定索引的数据行	
<pre>import pandas as pd # 使用列表创建 Series 对象 s1=pd.Series([166,178,180]) # 使用字典创建 Series 对象 s2=pd.Series({"s01":166, "s02":178, "s03":180}) # 创建 Series 对象时，自定义索引值 s3=pd.Series([166,178,180], index=["s01", "s02", "s03"])</pre> S1 输出: 0 166 1 178 2 180 dtype: int64 S2, S3 输出: s01 166 s02 178 s03 180 dtype: int64		<pre>s1=pd.Series([166,178,180]) print(s1[2])</pre> 输出: 180 <pre>s2=pd.Series({"s01":166, "s02":178, "s03":180}) print(s2["s03"])</pre> 输出: 180	
3. 查看 Series 对象常用属性			
<pre># 查看 Series 对象的 index s1=pd.Series([166,178,180]) # 枚举全部索引 for i in s1.index: print(i)</pre> 输出: 0 1 2	<pre># 查看 Series 对象的 values s1=pd.Series([166,178,180]) # 枚举 s1 中的值 for i in s1.values: print(i)</pre> 输出: 166 178 180	<pre>s1=pd.Series([166,178,180]) # 枚举 s1，默认为枚举 s1 中的值 for i in s1: print(i)</pre> 输出: 166 178 180	

3.DataFrame: Dataframe 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。

	学号	姓名	性别	借阅册数	来源
0	20190101	王静怡	女	15	校内图书馆
1	20190101	王静怡	女	8	校外图书馆
2	20190101	王静怡	女	5	其他

(1) **创建 DataFrame 对象:** 1. 通常用一个相等长度的列表或字典来创建。2. 也可利用函数直接读取二维数据文件创建 DataFrame 对象。



(2) **查看 DataFrame 对象:** 通过 index、columns、values 属性可以查看 Dataframe 对象的行索引、列索引及数据。

(3) **获取和修改 DataFrame 对象中的行列数据**

	姓名	性别	借阅册数
0	王静怡	女	28
1	张佳妮	女	56
2	李臣武	男	37

(4) **DataFrame 对象中数据进行统计与计算**

	姓名	性别	借阅册数
0	王静怡	女	28
1	张佳妮	女	56
2	李臣武	男	37

(5) **DataFrame 对象中数据的排序**

按索引排序可以使用 `sort_index()` 函数; 按值排序可以使用 `sort_values()` 函数。

通过选项 `axis=0/1` 确定排序的轴向, `axis` 默认值为 0, 纵向排序。通过选项 `ascending=True/False` 确定升/降序, `ascending` 默认值为 `True`, 升序排序; 排序结果返回一个新 DataFrame 对象。

(6)程序案例

1. 创建 DataFrame 对象

使用字典创建 DataFrame 对象

```
import pandas as pd
data={'姓名':['王静怡','张佳妮','李臣武'],'性别':['女','女','男'],'借阅册数':['28','56','37']}
df=pd.DataFrame(data,columns=["姓名","性别","借阅册数"])
print(df)
```

输出:

	姓名	性别	借阅册数
0	王静怡	女	28
1	张佳妮	女	56
2	李臣武	男	37

使用 excel 文件创建 DataFrame 对象

	A	B	C
1	姓名	性别	借阅册数
2	王静怡	女	28
3	张佳妮	女	56
4	李臣武	男	37

```
import pandas as pd
df=pd.read_excel('图书借阅.xlsx')
print(df)
```

输出:

	姓名	性别	借阅册数
0	王静怡	女	28
1	张佳妮	女	56
2	李臣武	男	37

2. 查看 DataFrame 对象常用属性

```
for i in df.index:
    print(i)
```

输出:

0
1
2

```
for i in df.columns:
    print(i)
```

输出:

姓名
性别
借阅册数

```
for i in df.columns:
    print(i)
```

输出:

['王静怡' '女' '28']
['张佳妮' '女' '56']
['李臣武' '男' '37']

3. 转置 DataFrame 对象的列、行

```
print(df.T)
```

输出:

	0	1	2
姓名	王静怡	张佳妮	李臣武
性别	女	女	男
借阅册数	28	56	37

4. 获取 DataFrame 对象中的数据

获取列

```
# 获取“姓名”列
# ①通过属性检索列
print(df.姓名)
# ②通过字典法检索列
print(df["姓名"])
```

输出均为:

0 王静怡
1 张佳妮
2 李臣武

根据行列索引获取数据

```
# ①获取“借阅册数”列、第3行的数据
print(df.at[2,"借阅册数"])
# ②通过字典的方式获取
print(df["借阅册数"][2])
```

输出:
37

有条件的获取行

```
#输出 df 中“性别”列数据为“女”的记录--> 通过布尔型数据实现“筛选”
print(df[df["性别"]=="女"])
```

输出:

	姓名	性别	借阅册数
0	王静怡	女	28
1	张佳妮	女	56

5. 追加、删除数据

append(): 在结尾增加一行数据

```
df_add=df.append({"姓名":"张晓明","性别":"女",  
借阅册数":29},ignore_index=True)
```

```
print(df_add)
```

输出:

	姓名	性别	借阅册数
0	王静怡	女	28
1	张佳妮	女	56
2	李臣武	男	37
3	张晓明	女	29

drop(): 删除数据, 通过 axis=0/1 确定行列

```
df_delr=df.drop(0,axis=0) # 删除第 1 行的数据  
print(df_delr)
```

输出:

	姓名	性别	借阅册数
1	张佳妮	女	56
2	李臣武	男	37

6.groupby(): 数据分组、mean(): 计算平均值

根据“性别”分类统计借阅图书的平均册数

按性别进行分组

```
g=df.groupby("性别",as_index=False)# as_index=False 表示显示行索引
```

计算平均册数

```
print(g.mean())
```

也可以写作 df.groupby("性别",as_index=False).mean()

输出:

	性别	借阅册数
0	女	42.0
1	男	37.0

***其他常用函数: max()、min(): 返回最大值、最小值, sum()、mean(): 求和、求平均值**

统计借阅册数中的最大值

```
print(df["借阅册数"].max())
```

输出:

56

统计借阅册数总和

```
print(df["借阅册数"].sum())
```

输出:

121

7 sort()_values: 数据排序

输出借阅册数降序排序的记录

```
df_sort=df.sort_values("借阅册数",ascending=False)# ascending=False 降序排序
```

```
print(df_sort)
```

输出:

	姓名	性别	借阅册数
1	张佳妮	女	56
2	李臣武	男	37
0	王静怡	女	28

4. 利用 matplotlib 模块绘图

(1) **matplotlib 模块:** 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。

(2) **在 Python 中引入 matplotlib 的 pyplot 子库的方法**

```
import matplotlib.pyplot as plt
```

(3) **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
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

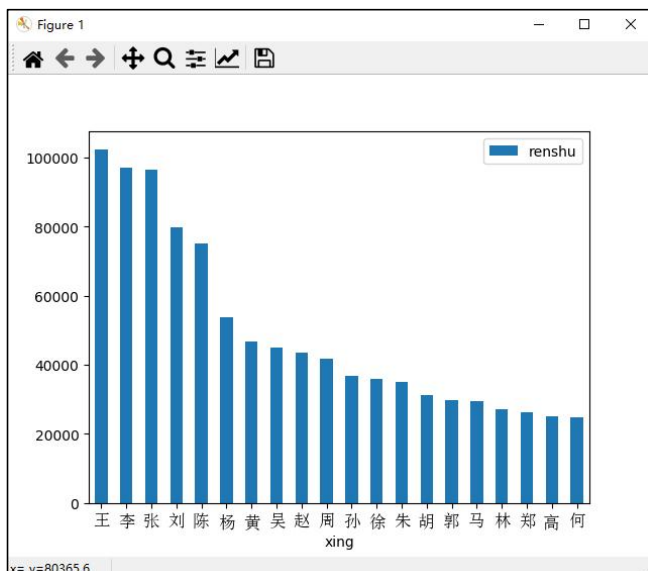
(4) **程序案例**

```
# 必修一 P125 页 “绘制正弦曲线图” 案例
import matplotlib.pyplot as plt
import numpy as np
# 生成 1000 个 [0, 10] 范围内的等差数列
x=np.linspace(0, 10, 1000)
y1=np.sin(x);y2=np.sin(x**2)
plt.figure(figsize=(8,4)) # 创建图表对象，figsize 指定图表的宽度和高度
plt.title("sin(x) and sin(x**2)") # 设置图表标题文字
# 绘制线型图，color 指定线条的颜色，linewidth 指定线条的宽度
plt.plot(x,y1,label="sin(x)", color="r", linewidth=2)
# 绘制散点图，label 给线条指定标签名称，该名称显示在图例中
plt.scatter(x,y2,label="sin(x**2)")
plt.ylim(-1.5,1.5) # 设置 y 坐标轴的取值范围
plt.xlim(0,10) # 设置 x 坐标轴的取值范围
plt.legend() # 显示图例
plt.show() # 显示图像
```

```

# 必修一 P126 页 “身边的百家姓” 案例
import pandas as pd
import matplotlib.pyplot as plt
#处理中文 utf8 编码
import codecs
# 显示中文字体处理需要
from matplotlib.font_manager import FontProperties
file=codecs.open('names.csv','r','utf-8') # 打开文件
# 定义复姓
fx=['欧燕','太史','端木','上官','司马','东方','独孤','南宫','万俟','闻人','夏侯','诸葛',
'尉迟','公羊','赫连','澹台','皇甫','宗政','濮阳','公冶','太叔','申屠','公孙','慕容',
'仲孙','钟离','长孙','宇文','司徒','鲜于','司空','闾丘','子车','元官','司寇','巫马',
'公西','颀孙','壤驷','公良','漆雕','乐正','宰父','谷梁','拓跋','夹谷','轩辕','令狐',
'段干','百里','呼延','东郭','南门','羊舌','微生','公户','公玉','公议','公议','梁丘',
'公仲','公上','公门','公山','公坚','左丘','公伯','西门','公祖','第五','公乘']
xing=[]
# 按行取文本，分解取姓和名
for line in file:
    if line[0:2] in fx: # 复姓处理
        xing.append(line[0:2])
    else: # 单姓处理
        xing.append(line[0:1])
# 构造 DataFrame 数据结构
data={'xing':xing,'renshu':0}
df=pd.DataFrame(data)
s=df.groupby('xing').count()
s=s.sort_values('renshu',ascending=False)
ax=s[0:20].plot(kind='bar',rot=0) # 前 20 绘图，rot 表示横坐标显示的旋转度数
# 显示中文标签
font=FontProperties(fname=r"c:\windows\fonts\simsun.ttc",size=12)
for label in ax.get_xticklabels():
    label.set_fontproperties(font)
plt.show() # 显示图形

```



利用 pandas 模块处理数据

姓名：_____ 教学班：_____ 座位号：_____

1. _____ 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。Series 对象常用的属性有 _____ (Series 的下标索引，其值默认是从 0 起递增的整数)，_____ (存放 Series 值的一个数组)。

2. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
import pandas as pd s1=_____ print(s1)	0 166 1 178 2 180 dtype: int64

3. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
import pandas as pd s2=pd.Series([166,178,180], _____) print(s2)	s01 166 s02 178 s03 180 dtype: int64

4. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
_____ s3=pd.Series({"s01":166, "s02":178, "s03":180}) print(s3)	s01 166 s02 178 s03 180 dtype: int64

5. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
import pandas as pd s=pd.Series([166,178,180],index=["S1","S2","S3"]) _____ print(s)	S1 200 S2 178 S3 180 dtype: int64

6. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
import pandas as pd s1=pd.Series([166,178,180]) print(s1) for i in _____: #枚举 s1 中的值 print(i)	0 166 1 178 2 180 dtype: int64 166 178 180

7.执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>import pandas as pd s1=pd.Series([166,178,180]) print(s1) for i in _____: #枚举 s1 中的索引 print(i)</pre>	<pre>0 166 1 178 2 180 dtype: int64 0 1 2</pre>

8. _____ 是一种二维的数据结构，由一个索引列和若干个数据列组成，每个数据列可以是不同的类型。DataFrame 对象常用属性有： _____ (DataFrame 的行索引、 _____ (存放各列的列标题)、 _____ (存放值的二维数据)、 _____ (行列转置)。

9.执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>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= _____ print(df1)</pre>	<pre> 姓名 借阅次数 性别 0 王静怡 28 女 1 张佳妮 56 女 2 李臣武 37 男</pre>

10.执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>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','性别','借阅次数']) print(df1) print(_____)#输出姓名列的值 print(_____)#输出按"性别"分类统计借阅图书的总册数 print(_____)#输出性别是“女”的记录 print(_____)#输出第 3 行、姓名列的值 print(_____)#输出借阅册数升序排序的记录</pre>	<pre> 姓名 性别 借阅次数 0 王静怡 女 28 1 张佳妮 女 56 2 李臣武 男 37 0 王静怡 1 张佳妮 2 李臣武 Name: 姓名, dtype: object 性别 借阅册数 0 女 84.0 1 男 37.0 姓名 性别 借阅次数 0 王静怡 女 28 1 张佳妮 女 56 2 李臣武 男 37 姓名 性别 借阅册数 0 王静怡 女 28 1 张佳妮 女 56 2 李臣武 男 37</pre>

利用 pandas 模块处理数据

1. **Series** 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。**Series** 对象常用的属性有 **index**（**Series** 的下标索引，其值默认是从 0 起递增的整数），**values**（存放 **Series** 值的一个数组）。

2. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>import pandas as pd s1=pd.Series([166,178,180]) print(s1)</pre>	<pre>0 166 1 178 2 180 dtype: int64</pre>

3. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>import pandas as pd s2=pd.Series([166,178,180], index=["s01", "s02", "s03"]) print(s2)</pre>	<pre>s01 166 s02 178 s03 180 dtype: int64</pre>

4. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>import pandas as pd s3=pd.Series({"s01":166, "s02":178, "s03":180}) print(s3)</pre>	<pre>s01 166 s02 178 s03 180 dtype: int64</pre>

5. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>import pandas as pd s=pd.Series([166,178,180],index=["S1","S2","S3"]) s[0]=200 print(s)</pre>	<pre>S1 200 S2 178 S3 180 dtype: int64</pre>

6. 执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>import pandas as pd s1=pd.Series([166,178,180]) print(s1) for i in s1: #枚举 s1 中的值 print(i)</pre>	<pre>0 166 1 178 2 180 dtype: int64 166 178 180</pre>

7.执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>import pandas as pd s1=pd.Series([166,178,180]) print(s1) for i in s1.index: #枚举 s1 中的索引 print(i)</pre>	<pre>0 166 1 178 2 180 dtype: int64 0 1 2</pre>

8.DataFrame 是一种二维的数据结构，由一个索引列和若干个数据列组成，每个数据列可以是不同的类型。DataFrame 对象常用属性有：**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)。

9.执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>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']) print(df1)</pre>	<pre>姓名 借阅次数 性别 0 王静怡 28 女 1 张佳妮 56 女 2 李臣武 37 男</pre>

10.执行 Python 程序，结果如下，请完善划线处的代码。

代码	输出结果
<pre>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','性别','借阅次数']) print(df1) print(df1.姓名)#输出姓名列的值 print(df1.groupby("性别",as_index=False).sum()) #输出按"性别"分类统计借阅图书的总册数 print(df1[df1["性别"]=="女"])#输出性别是“女”的记录 print(df1.at[2,"姓名"])#输出第 3 行、姓名列的值 print(df1.sort_values("借阅册数")) #输出借阅册数升序排序的记录</pre>	<pre>姓名 性别 借阅次数 0 王静怡 女 28 1 张佳妮 女 56 2 李臣武 男 37 0 王静怡 1 张佳妮 2 李臣武 Name: 姓名, dtype: object 性别 借阅册数 0 女 84.0 1 男 37.0 姓名 性别 借阅次数 0 王静怡 女 28 1 张佳妮 女 56 李臣武 姓名 性别 借阅册数 0 王静怡 女 28 2 李臣武 男 37 1 张佳妮 女 56</pre>