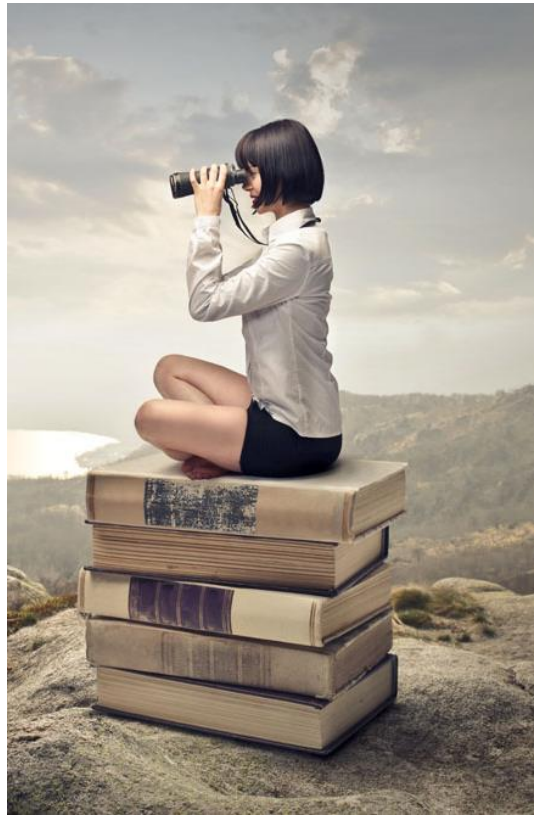
# Pandas数据分析

# 第3章 Pandas数据分析

目录
CONTENTE

# 本节要点

1. 了解Pandas的Series、DataFrame和Panel数据结构，掌握DataFrame数据结构的创建。
2. 熟悉DataFrame的基本功能，掌握DataFrame的行操作与列操作。
3. 熟悉Pandas操作外部数据的方法，掌握读取CVS、Excel和Sqlite数据库的方法。

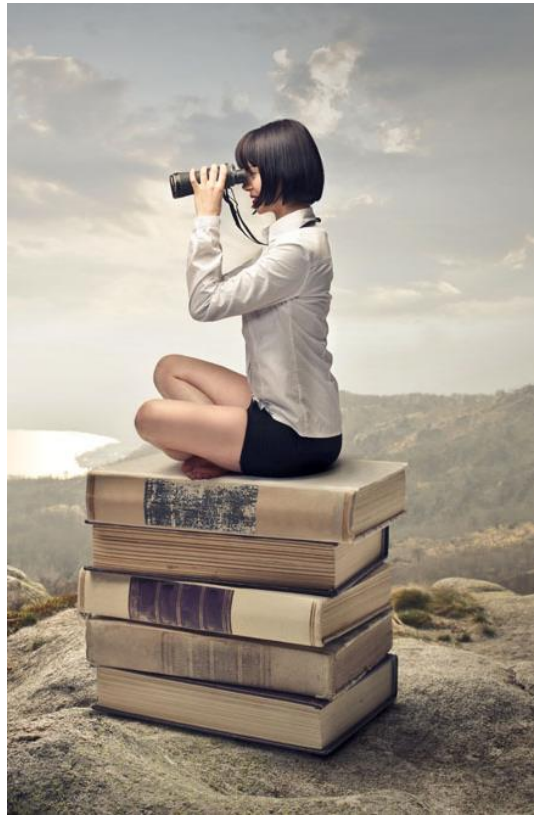


# 第四章 Pandas数据分析

## 引言

- Pandas 是 Python 中一个功能强大的数据分析库，它为数据处理提供了非常便捷的工具和数据结构。

- Pandas 补充了 NumPy 在数据结构灵活性、数据处理功能、数据输入输出、时间序列处理和数据可视化等方面的不足。



# 3.1 Pandas数据结构

Pandas有两种主要的数据结构：系列(Series)、数据帧(DataFrame)。

- **系列（Series）**是一种具有**索引**的类似于**一维数组**的数据结构。
- **数据帧（DataFrame）**是一种既有**行索引**又有**列索引**的类似于**二维**数组的数据结构。

# 3.1.1 Series数据结构

## （1） 创建Series

pandas.Series(data, index, dtype, name)

- 参数说明：
  - ✓ data：Series 的数据部分，可以是列表、数组、字典、标量值等。如果不提供此参数，则创建一个空的 Series。
  - ✓ index：Series 的索引部分，用于对数据进行标记。可以是列表、数组、索引对象等。如果不提供此参数，则创建一个默认的整数索引。
  - ✓ dtype：指定 Series 的数据类型。可以是 NumPy 的数据类型，例如 np.int64、np.float64 等。如果不提供此参数，则根据数据自动推断数据类型。
  - ✓ name：Series 的名称，用于标识 Series 对象。如果提供了此参数，则创建的 Series 对象将具有指定的名称。

## [示例] 利用列表创建series

```
import pandas as pd


height = [171,168,182,163]
s1 = pd.Series(height)
print('创建默认索引的Series：\n', s1)


names = ['吴明', '王毅', '陈辉', '魏云']
s2 = pd.Series(height, index=names)
print('创建指定索引的Series：\n', s2)
```

默认索引从0开始，依次增1。

```
创建默认索引的Series：
0    171
1    168
2    182
3    163
dtype: int32

创建指定索引的Series：
吴明    171
王毅    168
陈辉    182
魏云    163
dtype: int32
```

index参数指定的索引，长度必须与数据一致。

## [示例] 利用字典创建series

```
import pandas as pd


info = {'name':'李宁','age':'12','gender':'男'}
s1 = pd.Series(info)
print('创建键为索引的Series：\n', s1, sep='')
```

```
创建键为索引的Series：
name      李宁
age       12
gender    男
dtype: object
```

## [示例] 利用numpy数组创建series

```
import pandas as pd
import numpy as np

height = np.array([171,168,182,163])   # 创建一个数组

s1 = pd.Series(height)  # 用数组做Series的参数
print('创建默认索引的Series：\n', s1, sep='')
```

```
创建默认索引的Series：
0    171
1    168
2    182
3    163
dtype: int64
```

# 1.2 获取Series的数据和索引

- series.values

  Series中的数据（属性类型：numpy数组）

- series.index

  Series的索引（属性类型：Index对象）

- series.items()

  （索引，值）项（返回值类型：zip对象）

## [示例] 分别取得Series的数据和索引

```
import pandas as pd
import numpy as np

height = np.array([171,168,182,163])
names = ['吴明', '王毅', '陈辉', '魏云']
s2 = pd.Series(height,  index=names)
print( s2.values )
print( s2.index )
print( list(s2.items()) )
```

```
[171 168 182 163]
Index(['吴明', '王毅', '陈辉', '魏云'], dtype='object')
[('吴明', 171), ('王毅', 168), ('陈辉', 182), ('魏云', 163)]
```

# 1.3 通过Series的索引取值

- 位置索引

  通过0 ~ n-1进行索引

- 名称索引

  通过传入指定的index名称来进行索引

- 点索引

  通过"series.index名称"的形式进行索引

  (注意：index类型为非数值类型才可以使用)

- 布尔索引

  通过series[布尔表达式]取数

## [示例] 取Series中的单个数据

```
import pandas as pd
import numpy as np
height = np.array([171,168,182,163])
names = ['吴明', '王毅', '陈辉', '魏云']
s = pd.Series(height, index=names)
a = s[0]    # 通过位置索引取单个数据
print('第一个人的身高：',a)
b = s['王毅']   # 通过名称索引取单个数据
print('王毅的身高：', b)
c = s.陈辉   # 通过点索引取单个数据（注意索引不要加引号）
print('陈辉的身高：', c)
d = s[height>170]  # 通过条件索引取取数，也可以s[s>170]
print('身高大于170的人：\n',d)
```

```
第一个人的身高：  171

王毅的身高：  168

陈辉的身高：  182

身高大于170的人：
吴明    171
陈辉    182
dtype: int32.
```

## [示例] 列表作为索引，取离散多个数据

```
import pandas as pd
import numpy as np

height = np.array([171,168,182,163])
names = ['吴明', '王毅', '陈辉', '魏云']
s = pd.Series(height, index=names)

a = s[['吴明','陈辉']]
print('吴明和陈辉的身高：\n',a)

b = s[[0,2]]
print('第1个人和第三个人的身高：\n',b)
```

```
吴明和陈辉的身高：
吴明    171
陈辉    182
dtype: int32

第1个人和第三个人的身高：
吴明    171
陈辉    182
dtype: int32
```

## [示例] 索引切片，取连续多个数据

```
import pandas as pd
import numpy as np

height = np.array([171,168,182,163])
names = ['吴明', '王毅', '陈辉', '魏云']
s = pd.Series(height, index=names)

# 非数字切片，全闭区间
b = s['吴明':'陈辉']
print('吴明到陈辉(包含)所有人的身高：\n', b)
# 数字索引切片，前闭后开
c = s[0:2]
print('吴明到陈辉(不包含)所有人的身高：\n', c)
```

```
吴明到陈辉(包含)所有人的身高:
吴明    171
王毅    168
陈辉    182
dtype: int32

吴明到陈辉(不包含)所有人的身高:
吴明    171
王毅    168
dtype: int32
```

# 2 DataFrame数据帧

数据帧（DataFrame）是一种**既有行索引又有列索引的二维数组。**

## 2.1 创建DataFrame

Pandas.DataFrame()

| 主要参数 | 描述 |
| --- | --- |
| data | 可为ndarray、series、lists、dict或另一个DataFrame |
| index | 行标签，默认0~n-1（n为data的行数），也可以设置 |
| columns | 列标签，默认0~m-1（m为data的列数），也可以设置 |
| dtype | 每列的数据类型。 |

## [示例] 基于数组创建DataFrame对象

```
import numpy as np
from pandas import DataFrame

d = np.arange(12).reshape(3,4)
print('d=\n', d, sep='')

df = DataFrame(d)
print('\ndf=\n', df, sep='')
```

```
d=
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]

df=
   0  1   2   3
0  0  1   2   3
1  4  5   6   7
2  8  9  10  11
```

默认**列索引**从0开始，依次增1。

默认**行索引**从0开始，依次增1。

## [示例] 基于列表创建DataFrame

```
import pandas as pd


# 列表作为df数据，每个元素为df中的一行
data = [['小明','男',23], ['小花', '女', 22]]
df = pd.DataFrame(data)
print('df=\n', df)
```

```
df=
     0   1   2
0  小明  男  23
1  小花  女  22
```

## [示例] 创建DataFrame时指定行索引和列索引

```
import pandas as pd
import numpy as np

data = [['小明','男',23], ['小花', '女', 22]]

df = pd.DataFrame(data, index=[1,2], columns=['name','gender','age'])
print('df=\n', df)
```

```
df=
   name gender  age
1   小明      男   23
2   小花      女   22
```

## [示例] 基于字典创建DataFrame

```
import pandas as pd

# 字典作为df数据，每个“键值对”对应df的一列，键自动作为列索引data = {'name':['小明','小花','小兰','小胜'],
                'gender':['男','女','女','男']}
df = pd.DataFrame(data)
print('df=\n', df, sep='')
```

```
df=
   name  gender
0   小明      男
1   小花      女
2   小兰      女
3   小胜      男
```

## [示例] 基于Series创建DataFrame对象

```
import pandas as pd

# Series作为df数据，对应df的一列
s1 = pd.Series(['a','b','c'],index=[1,2,3])
print(s1)
print()

df1 = pd.DataFrame(s1)
print(df1)
```

```
1    a
2    b
3    c
dtype: object


   0
1  a
2  b
3  c
```

# 2.2 获取DataFrame的数据和索引

- dataframe.values

  dataframe的数据（二维数组类型）

- dataframe.index

  dataframe的行索引

- dataframe.columns

  dataframe的列索引

## [示例] 获取DataFrame的数据、行索引和列索引

```
import pandas as pd

data = {'name':['小明','小花','小兰','小胜'], 'gender':['男','女','女','男']}
df = pd.DataFrame(data)
print('df.values=\n', df.values, sep='')
print('df.index=', df.index, sep='')
print('df.columns=', df.columns, sep='')
```

```
[['小明' '男']
 ['小花' '女']
 ['小兰' '女']
 ['小胜' '男']]

RangeIndex(start=0, stop=4, step=1)

Index(['name', 'gender'], dtype='object')
```

# 第3章 Pandas数据分析

Python

# 3.2 DataFrame基本功能

# 4.2 DataFrame基本功能

数据帧（DataFrame）的基本功能包括了数据帧的重要属性和方法，如表所示。

| 属性或方法 | 描述 |
| --- | --- |
| T | 转置行和列 |
| axes | 轴序列 |
| dtypes | DataFrame中的数据的类型(dtypes) |
| empty | 如果DataFrame完全为空[]，则返回为True。<br>**注意：若 Series/DataFrame 只包含 NaN，它不被认为是空的** |
| ndim | 轴/数组维度大小 |
| shape | 返回表示DataFrame的维度的元组 |
| size | DataFrame中的元素数 |
| head(n) | 返回开头前n行,默认行数为5 |
| tail(n) | 返回最后n行,默认行数为5 |

## [示例] DataFrame的转置

```
import pandas as pd
import numpy as np

d={'name':pd.Series(['小明','小花','小兰','小胜']),
   'gender':pd.Series(['男','女','女','男']),
   'age':pd.Series([20,22,19,23]),
   'clazz': pd.Series(['1班','1班','2班','1班'])}

df = pd.DataFrame(d)
print(df)


print(df.T)
```

| | name | gender | age | clazz |
|---|---|---|---|---|
| 0 | 小明 | 男 | 20 | 1班 |
| 1 | 小花 | 女 | 22 | 1班 |
| 2 | 小兰 | 女 | 19 | 2班 |
| 3 | 小胜 | 男 | 23 | 1班 |

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| name | 小明 | 小花 | 小兰 | 小胜 |
| gender | 男 | 女 | 女 | 男 |
| age | 20 | 22 | 19 | 23 |
| clazz | 1班 | 1班 | 2班 | 1班 |

## [示例] DataFrame基本属性

```
import pandas as pd
import numpy as np
d = {'name':pd.Series(['小明','小花','小兰','小胜']),
     'gender':pd.Series(['男','女','女','男']),
     'age':pd.Series([20,22,19,23]),
     'clazz': pd.Series(['1班','1班','2班','1班'])}
df = pd.DataFrame(d)
print(df)

print ('数据类型：')
print(df.dtypes)
print ('是否为空',df.empty)
print ('维度:',df.ndim)
print ('形状：',df.shape)
print ('元素数量：',df.size)
```

```
  name gender  age clazz
0   小明      男   20    1班
1   小花      女   22    1班
2   小兰      女   19    2班
3   小胜      男   23    1班
数据类型：
name      object
gender    object
age        int64
clazz     object
dtype: object
是否为空 False
维度： 2
形状： (4, 4)
元素数量： 16
```

## [示例] DataFrame基本功能

```
import pandas as pd
import numpy as np

d = {'name':pd.Series(['小明','小花','小兰','小胜']),
   'gender':pd.Series(['男','女','女','男']),
     'age':pd.Series([20,22,19,23]),
    'clazz': pd.Series(['1班','1班','2班','1班'])}
df = pd.DataFrame(d)
print(df)

print ('前3行数据：')
print(df.head(3))

print ('后两行数据：')
print(df.tail(2))
```

```
  name gender  age clazz
0   小明      男   20    1班
1   小花      女   22    1班
2   小兰      女   19    2班
3   小胜      男   23    1班

前3行数据:
  name gender  age clazz
0   小明      男   20    1班
1   小花      女   22    1班
2   小兰      女   19    2班

后两行数据:
  name gender  age clazz
2   小兰      女   19    2班
3   小胜      男   23    1班
```

# 第3章 Pandas数据分析

Python

## 3.3 读取外部数据

- 读取外部数据分为读取 **文件**、**数据库**和**网络**中的数据。
  - 保存数据的文件主要有CSV、Excel、txt和 json，本节主要介绍使用较多的CSV和Excel文件，txt文件和json的使用与CSV和Excel的使用相似。
  - 数据库数据读取分为两部分：建立连接、执行SQL语句。本部分介绍如何读取Sqlite数据库。
  - 网络数据的读取使用最多的是网络爬虫，Pandas提供了read_htlm函数读取网页数据（read_html() 函数是最简单的爬虫，可以爬取静态网页表格数据）。

## 3.3.1 读写文件的方法

**（1）read_csv**

CSV（Comma-Separated Values）格式的文件是指以纯文本形式存储的表格数据，巨量的数据常使用CSV格式。

**（2）read_table**

函数read_table与read_csv大同小异，不同处是read_table默认分隔符为制表符，而read_csv默认的分隔符为英文逗号。

**（4）read_Excel**

**（3）to_csv**

把DataFrame数据保存数据到CSV文件。

**（5）to_excel**

[示例]

（1）将数据帧写入CSV文件

```
import pandas as pd

d = {'name':pd.Series(['小明','小花','小兰','小胜']),
     'gender':pd.Series(['男','女','女','男']),
     'age':pd.Series([20,22,19,23]),
     'clazz': pd.Series(['1班','1班','2班','1班'])}
df = pd.DataFrame(d)
df.to_csv('e:\stu.csv', index=False, encoding='gbk')
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | name | gender | age | clazz |
| 2 | 小明 | 男 | 20 | 1班 |
| 3 | 小花 | 女 | 22 | 1班 |
| 4 | 小兰 | 女 | 19 | 2班 |
| 5 | 小胜 | 男 | 23 | 1班 |

## （2）从CSV文件中读取数据帧

```
df1 = pd.read_csv('e:/stu.csv',encoding='gbk')
print(df1)    # 自动加0~n-1行索引
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | name | gender | age | clazz |
| 2 | 小明 | 男 | 20 | 1班 |
| 3 | 小花 | 女 | 22 | 1班 |
| 4 | 小兰 | 女 | 19 | 2班 |
| 5 | 小胜 | 男 | 23 | 1班 |

```
   name gender  age clazz
0    小明      男   20    1班
1    小花      女   22    1班
2    小兰      女   19    2班
3    小胜      男   23    1班
```

## （3）读取文件时指定某列为数据帧的index

**参数index_col：指定索引列（列名或列号）**

```
print('name列设为行索引：')
df1 = pd.read_csv('e:/stu.csv',encoding='gbk', index_col='name')
print(df1)
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | name | gender | age | clazz |
| 2 | 小明 | 男 | 20 | 1班 |
| 3 | 小花 | 女 | 22 | 1班 |
| 4 | 小兰 | 女 | 19 | 2班 |
| 5 | 小胜 | 男 | 23 | 1班 |

```
name列设为行索引：
      gender  age clazz
name
小明          男   20    1班
小花          女   22    1班
小兰          女   19    2班
小胜          男   23    1班
```

## （4）读取CSV文件时忽略读取指定行，最上一行自动成为列索引

**参数skiprows：跳过指定行号/行数**

```
print('忽略第一行：')
df2 = pd.read_csv('e:/stu.csv',encoding='gbk', skiprows=[0])
print(df2)
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | name | gender | age | clazz |
| 2 | 小明 | 男 | 20 | 1班 |
| 3 | 小花 | 女 | 22 | 1班 |
| 4 | 小兰 | 女 | 19 | 2班 |
| 5 | 小胜 | 男 | 23 | 1班 |

```
忽略第一行：
   小明  男  20  1班
0  小花  女  22  1班
1  小兰  女  19  2班
2  小胜  男  23  1班
```

## (4) 读取CSV文件时忽略指定的行数，最上一行自动成为列索引

```
df2 = pd.read_csv('e:/stu.csv',encoding='gbk', skiprows=2)
print(df2)
```

| | A | B | C | D |
|---|---|---|---|---|
| 1 | name | gender | age | clazz |
| 2 | 小明 | 男 | 20 | 1班 |
| 3 | 小花 | 女 | 22 | 1班 |
| 4 | 小兰 | 女 | 19 | 2班 |
| 5 | 小胜 | 男 | 23 | 1班 |

```
   小花  女  22  1班
0  小兰  女  19  2班
1  小胜  男  23  1班
```

## （5）从CSV文件中读取若干行数据

**参数nrows：仅读取前 n 行**

```
from pandas import read_csv


df1 = pd.read_csv('e:/stu.csv',encoding='gbk', nrows=2)
print('取前两行：')
print(df1)
```

```
取前两行：
   name gender  age clazz
0    小明      男   20    1班
1    小花      女   22    1班
```

## （6）从CSV文件中读取指定列数据

**参数usecols： 选择读取的列　　usecols=[0, 2] 或 usecols=['列名']**

```
df2 = read_csv('e:/stu.csv',encoding='gbk', usecols=['name','age'])
print('取name和age列：')
print(df2)
```

```
取name和age列：
   name  age
0    小明   20
1    小花   22
2    小兰   19
3    小胜   23
```

# 参数names：自定义列名（列表）

```
import pandas as pd

d = {'name':pd.Series(['小明','小花','小兰','小胜']),
     'gender':pd.Series(['男','女','女','男']),
     'age':pd.Series([20,22,19,23]),
     'clazz': pd.Series(['1班','1班','2班','1班'])}
df = pd.DataFrame(d)
df.to_csv('e:/stu.csv', encoding='gbk')

n = pd.Series(['姓名','性别','年龄','班级'])
df4 = pd.read_csv('e:/stu.csv',encoding='gbk', skiprows=[0], names=n)
print('忽略第一行，并指定结果的列名：')
print(df4)
```

忽略第一行，并指定结果的列名：

| | 姓名 | 性别 | 年龄 | 班级 |
|---|---|---|---|---|
| 0 | 小明 | 男 | 20 | 1班 |
| 1 | 小花 | 女 | 22 | 1班 |
| 2 | 小兰 | 女 | 19 | 2班 |
| 3 | 小胜 | 男 | 23 | 1班 |

# END