第15章
数据分析与可视化

数据分析与可视化

主要内容

01. 数据分析概念

02. 相关技术介绍

03. 数据可视化

04. 综合案例

数据分析概念

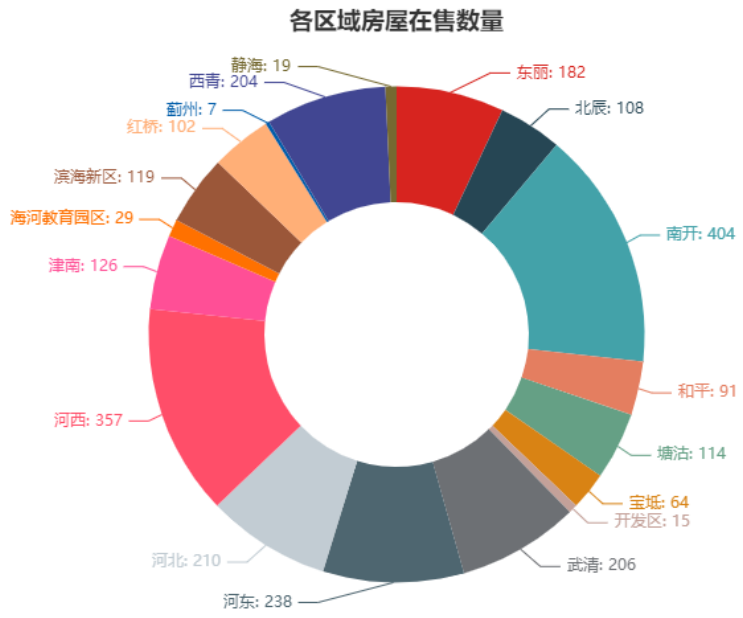
数据分析

- 数据分析是通过对收集到的大量数据进行适当分析，提取有用信息，最终形成相关结论
- 数据分析可以帮助用户对数据进行梳理，并根据根据结论做出相应处理

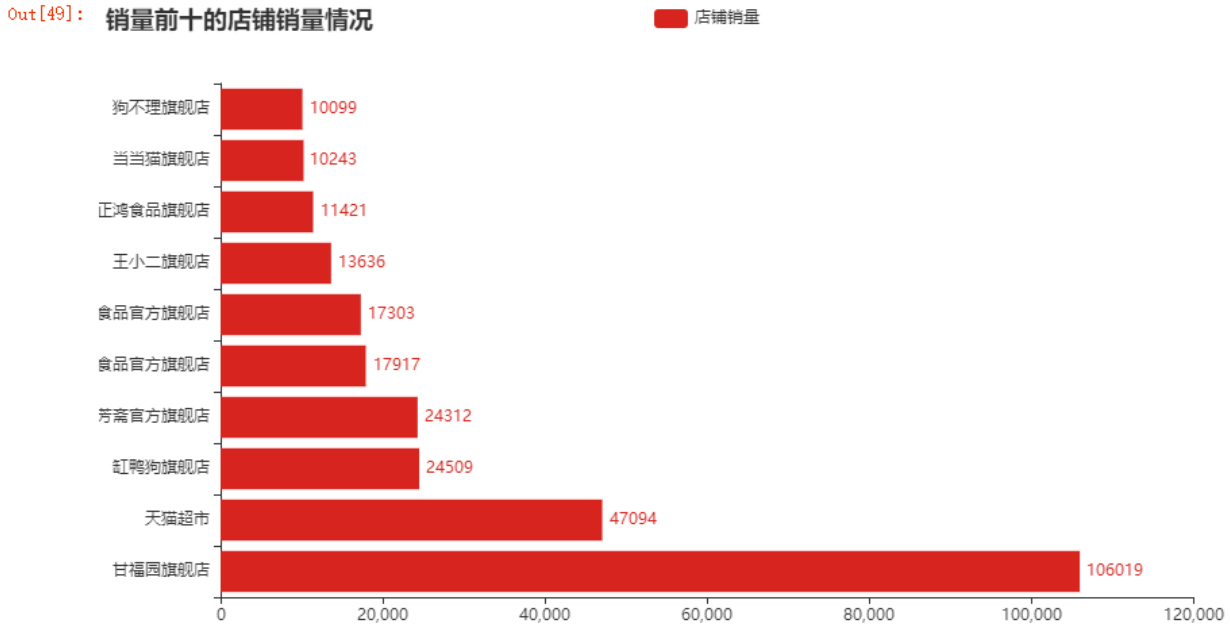
数据分析实例

- 二手房数据分析
- 淘宝销售数据分析

Out[16]:



Out[49]:



2

相关技术介绍

数据分析

- NumPy库
- Pandas库
- Matplotlib库

numpy库

- 创建数组
- 常用属性
- 常用方法
- 元素访问
- 常用统计函数

numpy库

- numpy是Python语言的一个扩展程序库，支持大量的维度数组与矩阵运算，并且针对数组运算提供大量的数学函数库

- import numpy as np

numpy库

- numpy 中最重要的对象是多维数组（ndarray）
- ndarray：存储单一数据类型的N维数组对象，与list不同，其存储数据类型必须相同
- ndarray数组的索引下标从0开始

numpy库

- 创建ndarray数组
  - array(object, dtype = None)
    - object：数组或数列
    - dtype：数组元素的数据类型

data = np.array([1, 2, 3, 4, 5]) # 创建一个包含五个元素的一维数组

numpy库

- 【例15.1】创建数组对象

```
import numpy as np
data = np.array([1, 2, 3, 4, 5])    #创建一个包含 5 个元素的一维数组
print(data)
```

程序的运行结果如下。

```
[1 2 3 4 5]
```

numpy库

- 【例15.2】创建数组对象，指定元素的数据类型

```
import numpy as np
#创建一个包含2行5列的二维数组，并指定元素的数据类型
data = np.array(([1, 2, 3, 4, 5], [1, 2, 3, 4, 5]), dtype=np.float)
print(data)
```

程序的运行结果如下。可见，这里创建了二维数组，元素的数据类型为浮点型。

```
[[1. 2. 3. 4. 5.]
 [1. 2. 3. 4. 5.]]
```

numpy库

- ndarray对象常用属性

属性	说明
ndarray.ndim	秩，即轴的数量或维度的数量
ndarray.shape	数组的维度，对于矩阵，n 行 m 列
ndarray.size	数组元素的总个数，相当于 .shape 中 n*m 的值
ndarray.dtype	ndarray 对象的元素类型

numpy库

- 【例 15.3】ndarray 对象常用属性

```
import numpy as np
data = np.array(([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])) #创建一个包含 2 行 5 列的二维数组
print(data)
print("秩: " + str(data.ndim))
print("维度: " + str(data.shape))
print("元素总个数: " + str(data.size))
print("元素类型: " + str(data.dtype))
```

程序的运行结果如下。

```
[[1 2 3 4 5]
 [1 2 3 4 5]]
秩:2
维度:(2,5)
元素总个数:10
元素类型:int32
```

numpy库

- ndarray对象常用方法

方法	说明
arange	类似range，一定范围内固定间隔的数字
ones	值全为1的ndarray
zeros	值全为0的ndarray
empty	值未初始化的ndarray
asarray	从序列类型数据创建ndarray
reshape	重新分配维度，但不改变原有ndarray
resize	重新分配维度，且改变原有ndarray

numpy库

- 【例 15.4】ndarray 对象常用方法

```
import numpy as np
data_arange = np.arange(1, 12, 2)  #创建元素值为 1、3、5、7、9、11 的 ndarray
print(data_arange)
data_ones = np.ones((2, 4))        #创建 2 行 4 列值全为 1 的 ndarray
print(data_ones)
data_zeros = np.zeros((2, 4))      #创建 2 行 4 列值全为 0 的 ndarray
print(data_zeros)
data_empty = np.empty((2, 4))      #创建 2 行 4 列值全未初始化的 ndarray
print(data_empty)
data_list = [[2, 5, 6],[4, 6, 8]]  #通过 list 创建 ndarray
data_asarray = np.asarray(data_list)
print(data_asarray)
data_arange.reshape((2,3))         #重新分配维度，原有数组元素不会发生变化
print(data_arange)
data_reshape = data_arange.reshape(-1, 1)    #根据第二个参数（列数）重新调整数组
print(data_reshape)
data_reshape = data_arange.reshape(1, -1)    #根据第一个参数（行数）重新调整数组
print(data_reshape)
data_arange.resize((2,3))          #重新分配维度，原有数组元素会发生变化
print(data_arange)
```

numpy库

- ndarray对象元素访问
  - 一维数组访问方式与list相同
  - 多维数组访问方式：每个维度索引之间用逗号分割

numpy库

- 【例 15.5】访问 ndarray 对象中的元素

```
import numpy as np
data_one = np.array([1, 2, 3, 4, 5]) #创建一个包含 5 个元素的一维数组
data_two = np.array(([10, 20, 30, 40, 50], [11, 12, 13, 14, 15]))
#创建一个包含 2 行 5 列的二维数组
print("第 2 个元素: " + str(data_one[1]))#一维数组的访问方式
print("第 2 行第 2 列: " + str(data_two[1, 1]))#二维数组的访问方式
```

程序的运行结果如下。

```
第 2 个元素:2
第 2 行第 2 列:12
```

numpy库

- numpy常用统计函数

函数	说明
sum	求和
mean	求平均值
min	求最小值
max	求最大值
std	求标准差
var	求方差

numpy库

- 【例 15.6】常用统计函数

```
import numpy as np
#创建 50 个元素的一维数组，每个元素都是 10 到 99 的随机数
data = np.random.randint(10, 100, (50))
print("sum: " + str(data.sum()))
print("mean: " + str(data.mean()))
print("min: " + str(data.min()))
print("max: " + str(data.max()))
print("std: " + str(data.std()))
print("var: " + str(data.var()))
```

程序的运行结果如下。

```
sum:2931
mean:58.62
min:10
max:98
std:26.215178809231876
var:687.2356
```

pandas库

- Series结构和DataFrame结构
- 读取和保存CSV文件
- 数据基本操作

pandas库

- pandas是Python语言的一个扩展程序库，用于数据分析
- pandas是基于numpy的工具
- pandas提供了大量函数和方法可用于数据分析

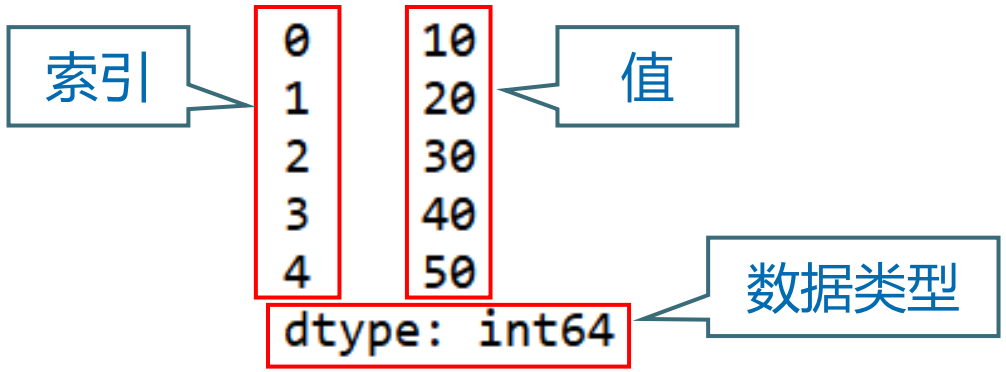
- import pandas as pd

pandas库

- pandas提供了两种数据结构：
  - Series结构（一维数据）
  - DataFrame结构（二维数据）

pandas库

- Series结构
  - 也称 Series 序列，类似于一维数组，由索引和数据值组成



pandas库

- 创建Series
  - Series(data, index, dtype)
    - data：输入的数据，可以是列表、常量、ndarray 数组
    - index：索引标签值，数字或字符，默认为从0开始的整数
    - dtype：数据类型，默认系统会根据获得的值自动指定

pandas库

- 【例 15.7】创建 Series，不指定索引标签值

```
import pandas as pd
data_list = [10, 20, 30, 40, 50]
series = pd.Series(data_list)
print(series)
```

程序的运行结果如下。

```
0    10
1    20
2    30
3    40
4    50
dtype:int64
```

pandas库

- 【例 15.8】创建 Series，指定索引标签值

```
import pandas as pd
data_list = [10, 20, 30, 40, 50]
series = pd.Series(data_list, index = [1, 2, 3, 4, 5])
print(series)
```

程序的运行结果如下。

```
1    10
2    20
3    30
4    40
5    50
dtype:int64
```

pandas库

- 获取Series的索引值和元素值
  - index属性：索引标签值
  - values属性：元素值

pandas库

- 【例 15.9】获取 Series 的索引标签值和元素值

```
import pandas as pd
data_list = [10, 20, 30, 40, 50]
series = pd.Series(data_list, index = [1, 2, 3, 4, 5])
print(series.index)   #索引值
print(series.values)  #元素值
```

程序的运行结果如下。

```
Int64Index([1, 2, 3, 4, 5],dtype='int64')
[10 20 30 40 50]
```

pandas库

- 访问Series元素
  - 下标访问
  - 索引标签值访问

pandas库

- 访问Series元素--下标访问
  - 给出下标或切片值

pandas库

- 【例 15.10】通过下标访问 Series 中元素

```
import pandas as pd
data_list = [10, 20, 30, 40, 50]
series = pd.Series(data_list, index = ['a', 'b', 'c', 'd', 'e'])
print(series[3])    #下标访问单个元素
print(series[1:4]) #下标访问多个元素
```

程序的运行结果如下。

```
40
b    20
c    30
d    40
dtype:int64
```

pandas库

- 访问Series元素--索引标签值访问
  - 给出索引标签值或索引标签值列表

pandas库

- 【例 15.11】通过索引标签访问 Series 中元素

```
import pandas as pd
data_list = [10, 20, 30, 40, 50]
series = pd.Series(data_list, index = ['a', 'b', 'c', 'd', 'e'])
print(series['b'])   #索引标签访问单个元素
print(series[['b', 'd']]) #索引标签访问多个元素
```

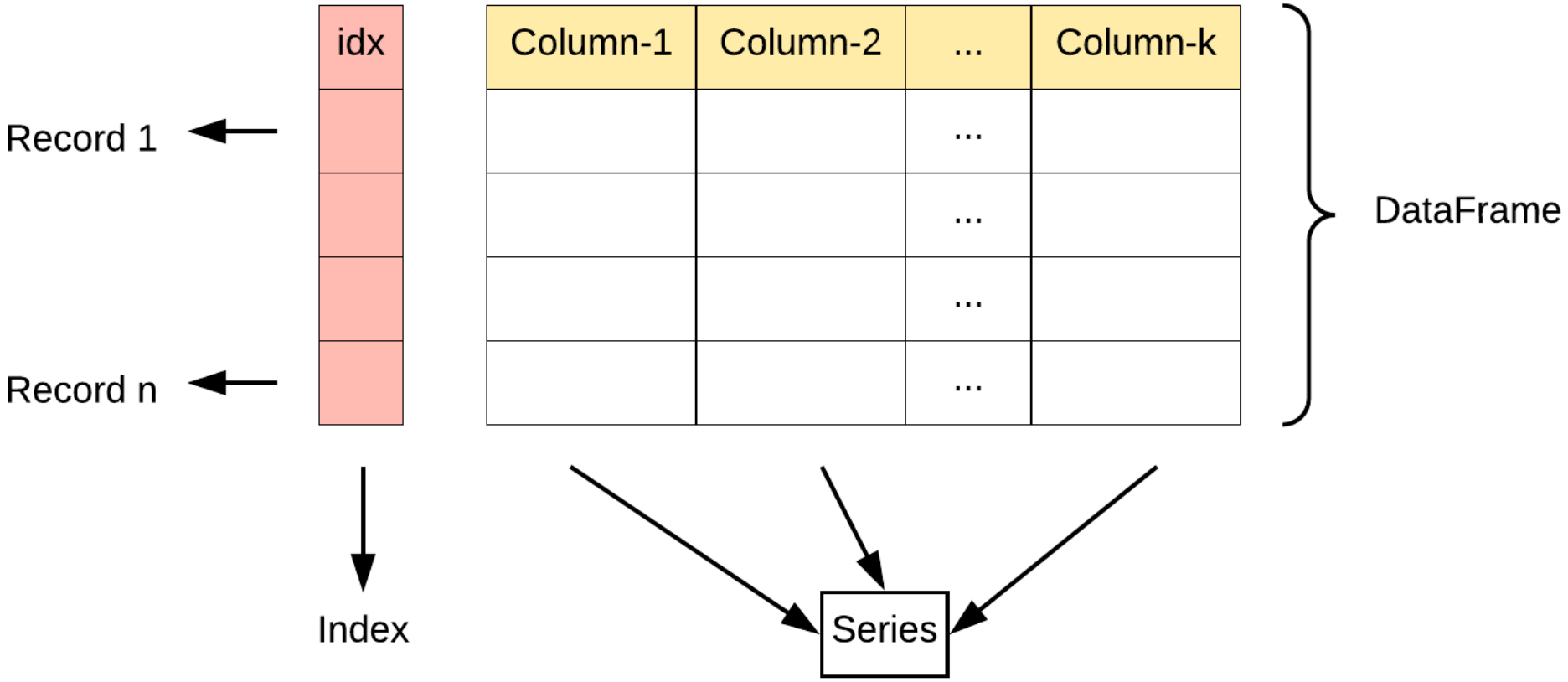
程序的运行结果如下。

```
20
b   20
d   40
dtype:int64
```

pandas库

- DataFrame结构
  - 是一个表格型的数据结构，它含有一组有序的列，每列可以是不同的值类型（数值、字符串、布尔型值）
  - DataFrame 既有行索引也有列索引，可以看作是多个Series

pandas库


idx
Column-1
Column-2
...
Column-k
Record 1
Record n
DataFrame
Index
Series

pandas库

- 创建DataFrame
  - DataFrame(data, index, columns, dtype)
    - data：输入的数据，可以是列表、常量、ndarray 数组、字典
    - index：行索引值，列表数据，默认为从0开始的整数
    - columns：列索引值，列表数据，默认为从0开始的整数
    - dtype：数据类型，默认系统会根据获得的值自动指定

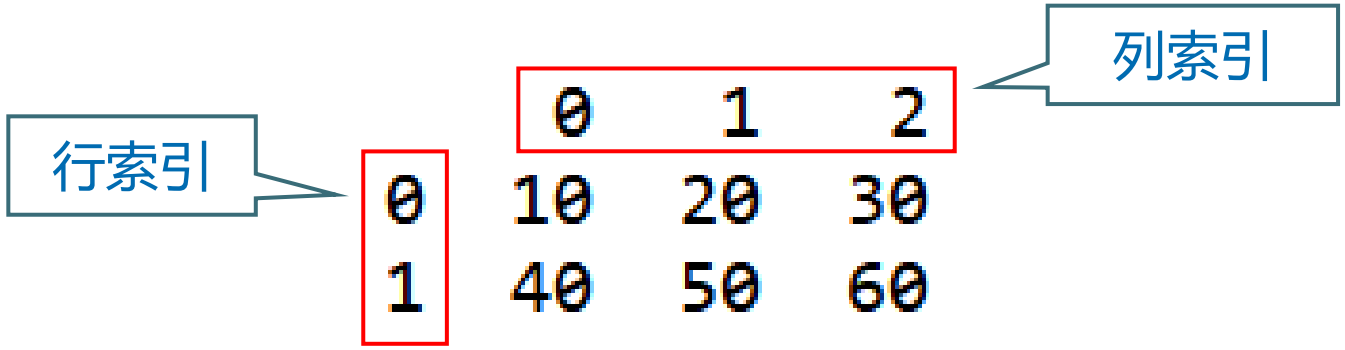
pandas模块--da02

- 从list数据创建，不指定行列索引

```
data_list = [[10, 20, 30],[40, 50, 60]]

data_dataframe = pd.DataFrame(data_list)

print(data_dataframe)
```



pandas库

- 【例 15.12】创建 DataFrame，不指定索引值

```
import pandas as pd
data_list = [[10, 20, 30],[40, 50, 60]]
data_dataframe = pd.DataFrame(data_list)
print(data_dataframe)
```

程序的运行结果如下。

```
    0   1   2
0  10  20  30
1  40  50  60
```

pandas库

- 【例 15.13】创建 DataFrame，指定索引值

```
import pandas as pd
data_list = [['Male', 38, 10, 48000],
             ['Male', 25, 2, 40000],
             ['Female', 32, 6, 60000],
             ['Male', 34, 8, 50000],
             ['Female', 43, 15, 64000],
             ['Male', 47, 20, 70000],
             ['Female', 44, 16, 56000],
             ['Male', 45, 10, 49000]]
columns = ['性别', '年龄', '工作年限', '年收入']
index = ['No.1', 'No.2', 'No.3', 'No.4', 'No.5', 'No.6', 'No.7', 'No.8']
data_dataframe = pd.DataFrame(data_list, index = index, columns = columns)
print(data_dataframe)
```

程序的运行结果如下。

```
        性别  年龄  工作年限  年收入
No.1    Male  38    10  48000
No.2    Male  25     2  40000
No.3  Female  32     6  60000
No.4    Male  34     8  50000
No.5  Female  43    15  64000
No.6    Male  47    20  70000
No.7  Female  44    16  56000
No.8    Male  45    10  49000
```

pandas库

- 【例 15.14】使用字典创建 DataFrame

```
import pandas as pd
data_dict = {'性别': ['Male', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male'],
             '年龄': [38, 25, 32, 34, 43, 47, 44, 45],
             '工作年限': [10, 2, 6, 8, 15, 20, 16, 10],
             '年收入': [48000, 40000, 60000, 50000, 64000, 70000, 56000, 49000]}
index = ['No.1', 'No.2', 'No.3', 'No.4', 'No.5', 'No.6', 'No.7', 'No.8']
data_dataframe = pd.DataFrame(data_dict, index = index)
print(data_dataframe)
```

程序的运行结果如下。

```
        性别  年龄  工作年限  年收入
No.1    Male  38    10  48000
No.2    Male  25     2  40000
No.3  Female  32     6  60000
No.4    Male  34     8  50000
No.5  Female  43    15  64000
No.6    Male  47    20  70000
No.7  Female  44    16  56000
No.8    Male  45    10  49000
```

pandas库

- 查看DataFrame数据
  - shape：查看各维度大小
  - index：行索引
  - columns：查看列名
  - head()：默认查看前5行，可设定查看行数
  - tail()：默认查看最后5行，可设定查看行数

pandas库

- 【例 15.15】查看 DataFrame 基本信息

```
import pandas as pd
data_dict = {'性别': ['Male', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male'],
             '年龄': [38, 25, 32, 34, 43, 47, 44, 45],
             '工作年限': [10, 2, 6, 8, 15, 20, 16, 10],
             '年收入': [48000, 40000, 60000, 50000, 64000, 70000, 56000, 49000]}
index = ['No.1', 'No.2', 'No.3', 'No.4', 'No.5', 'No.6', 'No.7', 'No.8']
data_dataframe = pd.DataFrame(data_dict, index = index)  #查看各维度大小
print(data_dataframe.shape[0])  #查看行数
print(data_dataframe.shape[1])  #查看列数
print(data_dataframe.index)     #查看行索引值
print(data_dataframe.columns)   #查看列名
print(data_dataframe.head(6))   #查看前 6 行
print(data_dataframe.tail(6))   #查看最后 6 行
```

pandas库

- 重建DataFrame行索引
  - 修改index属性，赋值为新的行索引列表