04-numpy复习和pandas入门

1. 数组的创建

a. 一维数组

案例：假设你是一位健身爱好者，正在跟踪你一周内每天的步数。你想将这些数据存储在NumPy数组中以便分析。

Python

```
每天的步数情况： [10000, 12000, 9000, 11000, 13000, 8000, 10500]
```

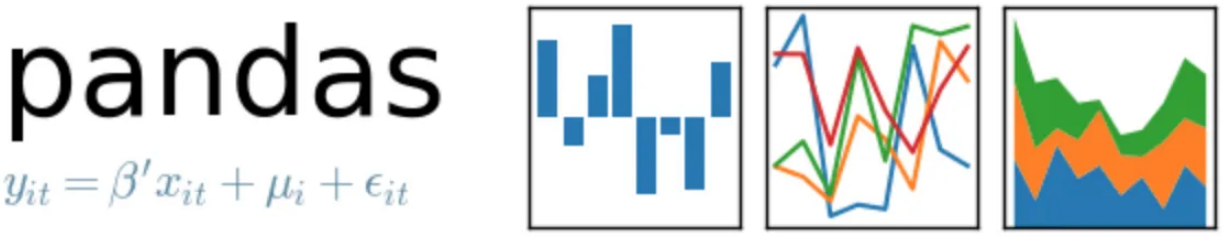
b. 二维数组

案例：你是一名学生，正在记录一个学期内每个月的各科成绩。你想将这些数据存储在NumPy的二维数组中，其中每一行代表一个月，每一列代表一门课程。

```python
数学 英语 物理
[85, 92, 78],  # 第一个月的成绩
[90, 88, 85],  # 第二个月的成绩
[88, 95, 82],  # 第三个月的成绩
[92, 89, 90]  # 第四个月的成绩
```

2. 思考：为什么要学习numpy?

一、pandas 简介



Pandas诞生于2008年，它的开发者是Wes McKinne，一个量化金融分析工程师。因为疲于应付繁杂的财务数据，Wes McKinney便自学Python，并开发了Pandas。大神就是这么任性，没有，就创造。为什么叫作Pandas，其实这是“Python data analysis”的简写，同时也衍生自计量经济学术语“panel data”（面板数据）。所以说Pandas的诞生是为了分析金融财务数据，当然现在它已经应用在各个领域了。

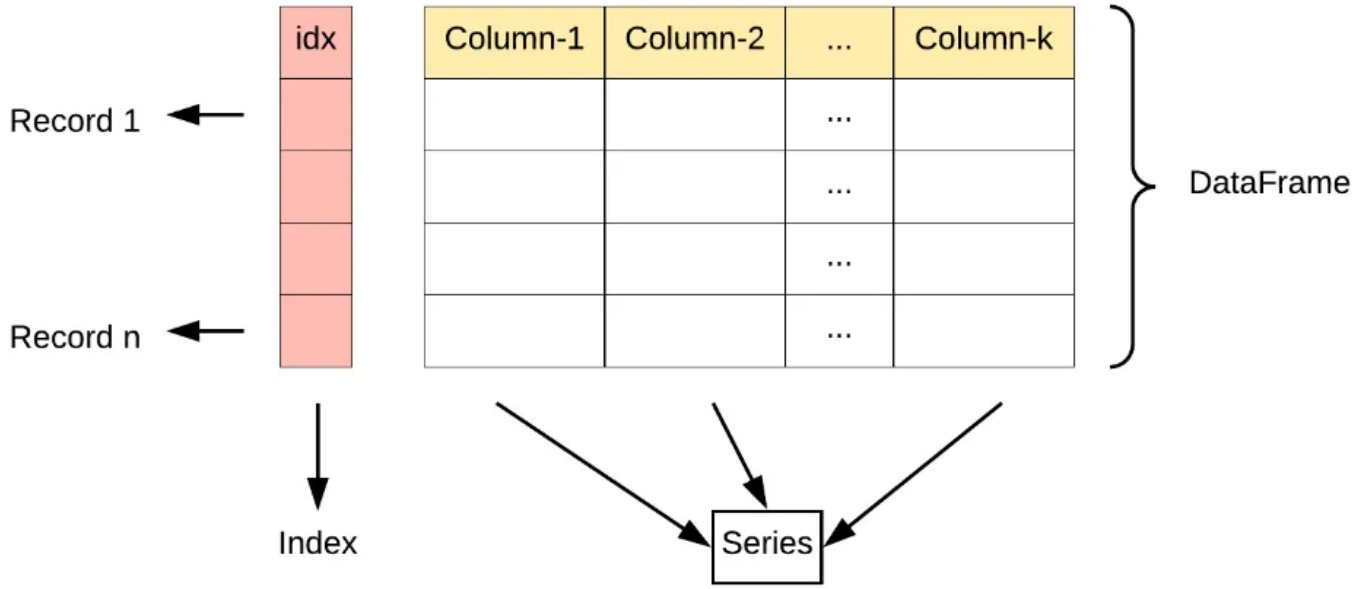
pandas即 Python Data Analysis Library 是基于NumPy 的一种工具，该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具。pandas提供了大量能使我们快速便捷地处理数据的函数和方法。你很快就会发现，它是使Python成为强大而高效的数据分析环境的重要因素之一。

二、Pandas安装

因为pandas是python的第三方库所以使用前需要安装一下，直接使用pip install pandas 就会自动安装pandas以及相关组件。（anaconda在promt窗口安装，pycharm建议在pycharm软件里安装，cmd可能装了不能识别到）

```python
pip install pandas
import pandas as pd
```

三、数据结构



1、Series

基本属性及其代码实现：

1. values: 返回底层的numpy数组
2. index: 返回索引对象
3. dtype: 返回数据的元素类型
4. shape: 返回数据的形状

5. size: 返回数据的元素个数
6. nbytes: 返回数据的字节大小
7. ndim: 返回数据的维度
8. name: 返回或设置Series的名称

Python

```python
import pandas as pd

# 创建一个pandas Series对象
s = pd.Series([1, 2, 3, 4, 5], name='my_series')
# 访问基本属性
print("Values:\n", s.values)
print("Index:\n", s.index)
print("Data type:\n", s.dtype)
print("Shape:\n", s.shape)
print("Size:\n", s.size)
print("Bytes:\n", s.nbytes)
print("Dimensions:\n", s.ndim)
print("Name:\n", s.name)
```

Series的创建

```python
import pandas as pd
import numpy as np

# 从列表创建Series
s1 = pd.Series([1, 2, 3, 4, 5])
print(s1)
# 从字典创建Series
s2 = pd.Series({'a': 1, 'b': 2, 'c': 3})
print(s2)
# 从Numpy数组创建Series
s3 = pd.Series(np.array([1, 2, 3, 4, 5]))
print(s3)
# 从字典和标签列表创建Series
s4 = pd.Series({'a': 1, 'b': 2, 'c': 3}, index=['a', 'b', 'c'])
print(s4)
```

Series取值

```python
import pandas as pd

# 创建一个Series对象
s = pd.Series([1, 2, 3, 4, 5], index=['a', 'b', 'c', 'd', 'e'])
# 通过索引取值
value = s['b']
print(value)  # 输出: 2
# 通过切片取值
slice_values = s['a':'c']
print(slice_values)
# 取第二行
row_value = s.iloc[1]
print(row_value)
# 创建一个DataFrame对象
df = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
# 取列
column_values = df['A']
print(column_values)
```

2、DataFrame

DataFrame的创建

```python
import pandas as pd
import numpy as np
# 创建一个字典
data = {'name': ['Tom', 'Nick', 'John', 'Tom'], 'Age': [20, 21, 19, 18]}
# 从字典创建DataFrame
df = pd.DataFrame(data)
print(df)
# 创建一个列表
data = [['Tom', 20], ['Nick', 21], ['John', 19], ['Tom', 18]]
# 从列表创建DataFrame，指定列为'name'和'Age'
df = pd.DataFrame(data, columns=['name', 'Age'])
print(df)
# 创建一个二维数组
data = np.array([['Tom', 20], ['Nick', 21], ['John', 19], ['Tom', 18]])
# 从二维数组创建DataFrame，指定列为'name'和'Age'
df = pd.DataFrame(data, columns=['name', 'Age'])
print(df)
# 创建一个DataFrame，然后使用该DataFrame创建另一个DataFrame
df1 = pd.DataFrame({'name': ['Tom', 'Nick', 'John'], 'Age': [20, 21, 19]})
df2 = pd.DataFrame(df1['name'], columns=['name'])  # 使用df1的一个列创建新的DataFrame
print(df2)
```

DataFrame的属性

```python
import pandas as pd

# 创建一个 DataFrame 对象
data = {'name': ['Tom', 'Nick', 'John'], 'Age': [20, 21, 19]}
df = pd.DataFrame(data)
# 查看 DataFrame 的列名
print(df.columns)
# 查看 DataFrame 的索引
print(df.index)
# 查看 DataFrame 的数据
print(df.values)
# 查看 DataFrame 的形状
print(df.shape)
# 查看 DataFrame 的数据类型
print(df.dtypes)
```

DataFrame取值

1. 通过列标签取值

你可以使用列标签来选取DataFrame中的一列或多列。

```python
import pandas as pd

# 创建一个示例DataFrame
data = {'Name': ['Tom', 'Nick', 'John', 'Peter'],
        'Age': [20, 21, 19, 18],
        'Score': [85, 80, 90, 88]}
df = pd.DataFrame(data)

# 选取单列
print(df['Name'])

# 选取多列
print(df[['Name', 'Age']])

```

2. 通过标签索引取值

如果你的DataFrame有标签索引（也称为索引），你可以使用这些标签来选取行。

```python
# 设置行标签并选取行
df.index = ['Student1', 'Student2', 'Student3', 'Student4']
print(df.loc['Student1'])  # 使用loc按标签选取行
```

3. 通过位置取值

你还可以使用基于整数位置的方法来选取数据。

```python
# 选取第一行（索引为0）
print(df.iloc[0])  # 使用iloc按位置选取行
```

4. 使用布尔索引取值

你可以使用布尔索引来根据条件选取数据。

```python
# 选取年龄大于20的行
print(df[df['Age'] > 20])
```