



Python—Pandas类库使用培训课程

讲师 郭剑峰

Pandas概述

Pandas是Python下**最强大的数据分析和探索工具**。它包含高级的数据结构和精巧的工具，使得在Python中处理数据非常快速和简单。Pandas构建在Numpy之上，它使得以NumPy为中心的应用很容易使用。Pandas的名称来自于面板数据（PanelData）和Python数据分析（DataAnalysis），它最初被作为金融数据分析工具而开发出来，由AQR Capital Management公司于2008年4月开发出来，并于2009年底开源。Pandas的功能非常强大，支持类似于SQL的数据增、删、查、改，并且带有丰富的数据处理函数；支持时间序列分析功能；支持灵活处理缺失数据等。

Pandas安装

Anaconda 已自带Pandas库无需安装

Pycharm安装Pandas库

Window环境:

`pip install pandas pip -i https://pypi.douban.com/simple`

```
C:\Users\Administrator>pip install pandas pip -i https://pypi.douban.com/simple
Looking in indexes: https://pypi.douban.com/simple
Requirement already satisfied: pandas in c:\python38\lib\site-packages (1.0.2)
Requirement already satisfied: pip in c:\python38\lib\site-packages (20.0.2)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\python38\lib\site-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in c:\python38\lib\site-packages (from pandas) (2019.3)
Requirement already satisfied: numpy>=1.13.3 in c:\python38\lib\site-packages (from pandas) (1.18.1)
Requirement already satisfied: six>=1.5 in c:\python38\lib\site-packages (from python-dateutil>=2.6.1->pandas) (1.14.0)
WARNING: You are using pip version 20.0.2; however, version 20.1 is available.
You should consider upgrading via the 'c:\python38\python.exe -m pip install --upgrade pip' command.
```

Ubuntu & Debian环境:

`sudo apt-get install python-pandas`

CentOS/Fedora环境:

`sudo yum install python-pandas`

Mac环境:

`python -m pip install python-pandas`

Pandas数据结构

Pandas处理以下三个数据结构:

系列(Series)

数据帧(DataFrame)

面板(Panel)

DataFrame是Series的容器， Panel是DataFrame的容器。

系列是具有均匀数据的一维数组结构。例如，以下系列是整数：10,23,56, ...的集合。

10	23	56	17	52	61	73	90	26	72
----	----	----	----	----	----	----	----	----	----

关键点

- 1.均匀数据
- 2.尺寸大小不变
- 3.数据的值可变

Pandas数据结构

数据帧(DataFrame)是一个具有异构数据的二维数组

姓名	年龄	性别	等级
Maxsu	25	男	4.45
Katie	34	女	2.78
Vina	46	女	3.9
Lia	女	x女	4.6

数据帧中四列的数据类型

列	类型
姓名	字符串
年龄	整数
性别	字符串
等级	浮点型

关键点

- 1.异构数据
- 2.大小可变
- 3.数据可变

Pandas数据结构

面板是具有异构数据的三维数据结构。在图形表示中很难表示面板。但是一个面板可以说明为DataFrame的容器。

关键点

- 1.异构数据
- 2.大小可变
- 3.数据可变

Pandas对象操作

通过传递值列表来创建一个系列，让Pandas创建一个默认的整数索引

关键点

1. 异构数据
2. 大小可变
3. 数据可变

01_ObjectOperation.py演示

Pandas系列(Series)

系列(Series)是能够保存任何类型的数据(整数, 字符串, 浮点数, Python对象等)的一维标记数组。轴标签统称为索引。

`pandas.Series(data, index, dtype, copy)`

编号	参数	描述
1	<code>data</code>	数据采取各种形式, 如: <code>ndarray</code> , <code>list</code> , <code>constants</code>
2	<code>index</code>	索引值必须是唯一的和散列的, 与数据的长度相同。默认 <code>np.arange(n)</code> 如果没有索引被传递。
3	<code>dtype</code>	<code>dtype</code> 用于数据类型。如果没有, 将推断数据类型
4	<code>copy</code>	复制数据, 默认为 <code>false</code> 。

02_Series.py演示

Pandas系列(Series)

02_Series.py演示

系列基本功能

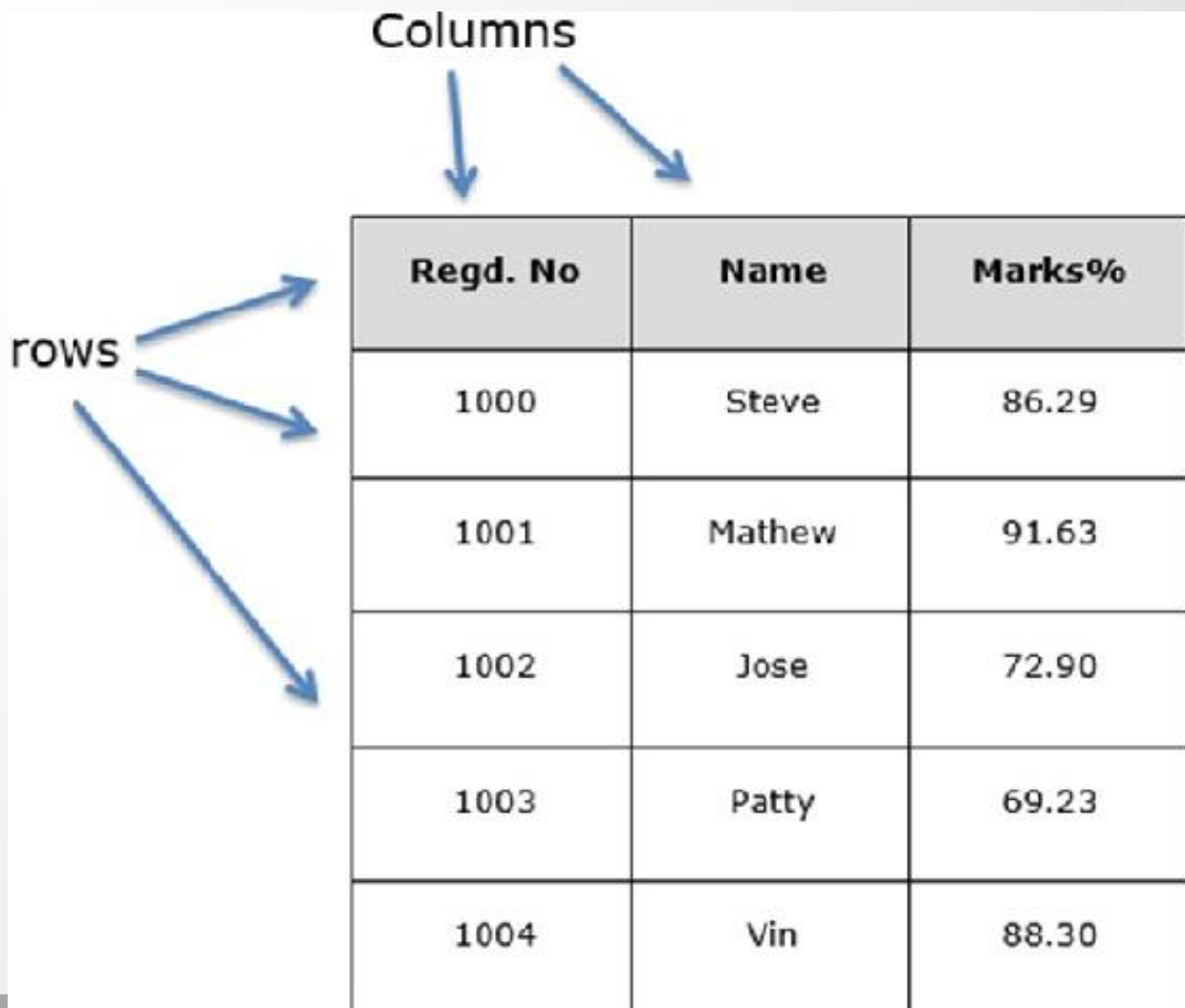
编号	属性或方法	描述
1	<code>axes</code>	返回行轴标签列表。
2	<code>dtype</code>	返回对象的数据类型(<code>dtype</code>)。
3	<code>empty</code>	如果系列为空，则返回 <code>True</code> 。
4	<code>ndim</code>	返回底层数据的维数，默认定义: <code>1</code> 。
5	<code>size</code>	返回基础数据中的元素数。
6	<code>values</code>	将系列作为 <code>ndarray</code> 返回。
7	<code>head()</code>	返回前 <code>n</code> 行。
8	<code>tail()</code>	返回最后 <code>n</code> 行。

Pandas数据帧(DataFrame)

数据帧(DataFrame)是二维数据结构，即数据以行和列的表格方式排列。

数据帧(DataFrame)的功能特点：

- 1.潜在的列是不同的类型
- 2.大小可变
- 3.标记轴(行和列)
- 4.可以对行和列执行算术运算



The diagram illustrates a DataFrame as a table. The word 'Columns' is positioned above the table with two arrows pointing to the 'Regd. No' and 'Name' headers. The word 'rows' is positioned to the left of the table with three arrows pointing to the first three rows of data.

Regd. No	Name	Marks%
1000	Steve	86.29
1001	Mathew	91.63
1002	Jose	72.90
1003	Patty	69.23
1004	Vin	88.30

Pandas数据帧(DataFrame)

`pandas.DataFrame(data, index, columns, dtype, copy)`

编号	参数	描述
1	<code>data</code>	数据采取各种形式, 如: <code>ndarray</code> , <code>series</code> , <code>map</code> , <code>lists</code> , <code>dict</code> , <code>constant</code> 和另一个 <code>DataFrame</code> 。
2	<code>index</code>	对于行标签, 要用于结果帧的索引是可选缺省值 <code>np.arange(n)</code> , 如果没有传递索引值。
3	<code>columns</code>	对于列标签, 可选的默认语法是 - <code>np.arange(n)</code> 。 这只有在没有索引传递的情况下才是这样。
4	<code>dtype</code>	每列的数据类型。
5	<code>copy</code>	如果默认值为 <code>False</code> , 则此命令(或任何它)用于复制数据。

03_DataFrame.py演示

Pandas数据帧(DataFrame)

DateFrame基本功能

编号	属性或方法	描述
1	<code>T</code>	转置行和列。
2	<code>axes</code>	返回一个列，行轴标签和列轴标签作为唯一的成员。
3	<code>dtypes</code>	返回此对象中的数据类型(<code>dtypes</code>)。
4	<code>empty</code>	如果 <code>NDFrame</code> 完全为空[无项目]，则返回为 <code>True</code> ；如果任何轴的长度为 <code>0</code> 。
5	<code>ndim</code>	轴/数组维度大小。
6	<code>shape</code>	返回表示 <code>DataFrame</code> 的维度的元组。
7	<code>size</code>	<code>NDFrame</code> 中的元素数。
8	<code>values</code>	NDFrame的Numpy表示。
9	<code>head()</code>	返回开头前 <code>n</code> 行。
10	<code>tail()</code>	返回最后 <code>n</code> 行。

Pandas&xarray替代Panel面板

Pandas0.24版本中对Panel的解释:

面板(Panel)是3D容器的数据。面板数据一词来源于计量经济学，部分源于名称：Pandas - pan(el)-da(ta)-s。

3轴(axis)这个名称旨在给出描述涉及面板数据的操作的一些语义。

items - axis 0, 每个项目对应于内部包含的数据帧(DataFrame)。

major_axis - axis 1, 它是每个数据帧(DataFrame)的索引(行)。

minor_axis - axis 2, 它是每个数据帧(DataFrame)的列。

Panel在pandas的0.25版本以后被废除，取而代之的是结合第三方类库xarray完成Panel的工作

Pandas&xarray替代Panel面板

windows安装xarray-0.15.1

pip install --upgrade pip -i https://pypi.douban.com/simple

pip install xarray pip -i https://pypi.douban.com/simple

```
C:\Users\Administrator>pip install --upgrade pip -i https://pypi.douban.com/simple
Looking in indexes: https://pypi.douban.com/simple
Requirement already up-to-date: pip in c:\python38\lib\site-packages (20.1)
Could not build wheels for pip, since package 'wheel' is not installed.

C:\Users\Administrator>pip install xarray pip -i https://pypi.douban.com/simple
Looking in indexes: https://pypi.douban.com/simple
Collecting xarray
  Downloading https://pypi.doubanio.com/packages/ee/11/fb2a8a6015e3de4ff19a4870bb0d11f48ebdd997062557d24cd076b3088f/xarray-0.15.1-py3-none-any.whl (668 kB)
    | 668 kB 939 kB/s
Requirement already satisfied: pip in c:\python38\lib\site-packages (20.1)
Requirement already satisfied: setuptools>=41.2 in c:\python38\lib\site-packages (from xarray) (41.2.0)
Requirement already satisfied: numpy>=1.15 in c:\python38\lib\site-packages (from xarray) (1.18.1)
Requirement already satisfied: pandas>=0.25 in c:\python38\lib\site-packages (from xarray) (1.0.2)
Requirement already satisfied: pytz>=2017.2 in c:\python38\lib\site-packages (from pandas>=0.25->xarray) (2019.3)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\python38\lib\site-packages (from pandas>=0.25->xarray) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\python38\lib\site-packages (from python-dateutil>=2.6.1->pandas>=0.25->xarray) (1.14.0)
Could not build wheels for pip, since package 'wheel' is not installed.
Could not build wheels for setuptools, since package 'wheel' is not installed.
```

Xarray概述

dataArray: 具有标注或命名维度的多维数组。 DataArray对象将元数据（例如维名称，坐标和属性（如下定义））添加到基础的“未标记”数据结构（例如numpy和Dask数组）中。如果设置了其可选的name属性，则它是一个命名的DataArray。

Dataset: 尺寸一致的，类似于dict的DataArray对象集合。因此，可以在单个DataArray的维度上执行的大多数操作都可以在数据集上执行。数据集具有数据变量，尺寸，坐标和属性。

Variable: 由维，数据和描述单个数组的属性组成。变量和numpy数组之间的主要功能区别在于，对变量的数字运算通过维名称实现数组广播。每个DataArray都有一个基础变量，可以通过arr.variable访问。

Xarray数据结构(DataArray)

`xarray.DataArray`是xarray标记的多维数组的实现。它具有几个关键属性：

values：一个保存数组值的numpy.ndarray

dims：每个轴的尺寸名称（例如 ('x', 'y', 'z')

coords：类似dict的数组（坐标）容器，用于标记每个点（例如，数字，日期时间对象或字符串的一维数组）

attrs：存放任意元数据（属性）的字典

xarray使用dims和coords启用其核心元数据感知操作。维度提供xarray使用的名称，而不是许多numpy函数中使用的axis参数。坐标基于pandas的DataFrame或Series上的索引功能，可实现基于标签的快速索引和对齐。

04_xarray_DataArray.py演示

Xarray DataSet

05_xarray_Dataset.py演示

xarray.Dataset是xarray的DataFrame的多维等效项。它是类似dict的具有对齐尺寸的标记数组（DataArray对象）的容器。

除了数据集本身的类似dict的界面（可用于访问数据集中的任何变量）之外，数据集还具有四个关键属性：

dims：从维度名称到每个维度的固定长度的字典（例如{'x': 6, 'y': 6, 'time': 8}）

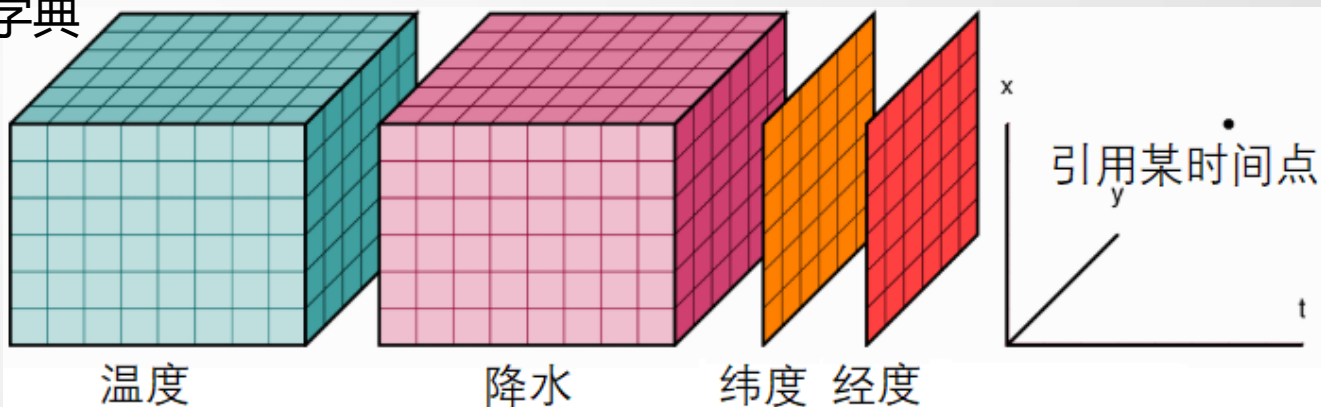
data_vars：类似于dict的DataArrays容器，对应于变量

coords：另一个类似Array的类似dict的容器，用于标记data_vars中使用的点（例如，数字数组，日期时间对象或字符串）

[更多用法参见](http://xarray.pydata.org/en/stable/index.html)

<http://xarray.pydata.org/en/stable/index.html>

attrs：存放任意元数据的字典



在此示例中，将温度和降水称为“数据变量”，将所有其他数组称为“坐标变量”，它们沿维度标注了点。

Pandas描述性统计

06_descriptionStat.py演示

编号	函数	描述
1	<code>count()</code>	非空观测数量
2	<code>sum()</code>	所有值之和
3	<code>mean()</code>	所有值的平均值
4	<code>median()</code>	所有值的中位数
5	<code>mode()</code>	值的模值
6	<code>std()</code>	值的标准偏差
7	<code>min()</code>	所有值中的最小值
8	<code>max()</code>	所有值中的最大值
9	<code>abs()</code>	绝对值
10	<code>prod()</code>	数组元素的乘积
11	<code>cumsum()</code>	累计总和
12	<code>cumprod()</code>	累计乘积

Pandas函数应用

07_func.py

表合理函数应用:

`pipe()`

行或列函数应用:

`apply()`

元素函数应用:

`applymap()`



Pandas重建索引

重新索引会更改DataFrame的行标签和列标签。重新索引意味着符合数据以匹配特定轴上的一组给定的标签。

可以通过索引来实现多个操作:

- 1.重新排序现有数据以匹配一组新的标签。
- 2.在没有标签数据的标签位置插入缺失值(NA)标记。

08_RebuildIndex.py演示

Pandas迭代

09_iter.py

Pandas对象之间的基本迭代的行为取决于类型。
当迭代一个系列时，它被视为数组式，基本迭代产生这些值。



Pandas排序

10_sort.py

Pandas有两种排序方式:

按标签

按实际值



Pandas字符串和文本数据

编号	函数	描述	编号	函数	描述
1	<code>lower()</code>	将 <code>Series/Index</code> 中的字符串转换为小写。	10	<code>repeat(value)</code>	重复每个元素指定的次数。
2	<code>upper()</code>	将 <code>Series/Index</code> 中的字符串转换为大写。	11	<code>count(pattern)</code>	返回模式中每个元素的出现总数。
3	<code>len()</code>	计算字符串长度。	12	<code>startswith(pattern)</code>	如果系列/索引中的元素以模式开始, 则返回 <code>true</code> 。
4	<code>strip()</code>	帮助从两侧的系列/索引中的每个字符串中删除空格(包括换行符)。	13	<code>endswith(pattern)</code>	如果系列/索引中的元素以模式结束, 则返回 <code>true</code> 。
5	<code>split(' ')</code>	用给定的模式拆分每个字符串。	14	<code>find(pattern)</code>	返回模式第一次出现的位置。
6	<code>cat(sep=' ')</code>	使用给定的分隔符连接系列/索引元素。	15	<code>findall(pattern)</code>	返回模式的所有出现的列表。
7	<code>get_dummies()</code>	返回具有单热编码值的数据帧(DataFrame)。	16	<code>swapcase</code>	变换字母大小写。
8	<code>contains(pattern)</code>	如果元素中包含子字符串, 则返回每个元素的布尔值 <code>True</code> , 否则为 <code>False</code> 。	17	<code>islower()</code>	检查系列/索引中每个字符串中的所有字符是否小写, 返回布尔值
9	<code>replace(a,b)</code>	将值 <code>a</code> 替换为值 <code>b</code> 。	18	<code>isupper()</code>	检查系列/索引中每个字符串中的所有字符是否大写, 返回布尔值
			19	<code>isnumeric()</code>	检查系列/索引中每个字符串中的所有字符是否为数字, 返回布尔值。

11_text.py演示

Pandas选项和自定义

12_option.py

get_option()

set_option()

reset_option()

describe_option()

option_context()



Pandas统计函数

13_index&chooseData.py

.loc() 基于标签

.iloc() 基于整数



Pandas统计函数

14_stat.py

pct_change()元素百分比变化

cov()协方差

corr()相关性

rank()数据排名



Pandas窗口函数

窗口函数主要用于通过平滑曲线来以图形方式查找数据内的趋势。如果日常数据中有很多变化，并且有很多数据点可用，那么采样和绘图就是一种方法，应用窗口计算并在结果上绘制图形是另一种方法。通过这些方法，可以平滑曲线或趋势。

15_window.py

为了处理数字数据，Pandas提供了几个变体，如滚动，展开和指数移动窗口统计的权重。其中包括总和，均值，中位数，方差，协方差，相关性等。



Pandas聚合

16_aggregate.py

当有了滚动，扩展和ewm对象创建了以后，就有几种方法可以对数据执行聚合



Pandas缺失数据

数据丢失(缺失)在现实生活中总是一个问题。机器学习和数据挖掘等领域由于数据缺失导致的数据质量差，在模型预测的准确性上面临着严重的问题。在这些领域，缺失值处理是使模型更加准确和有效的重点。

何时以及为什么数据丢失？

想象一下有一个产品的在线调查。很多时候，人们不会分享与他们有关的所有信息。很少有人分享他们的经验，但不是他们使用产品多久；很少有人分享使用产品的时间，经验，但不是他们的个人联系信息。因此，以某种方式或其他方式，总会有一部分数据总是会丢失，这是非常常见的现象。

17_missData.py演示

Pandas分组(GroupBy)

任何分组(groupby)操作都涉及原始对象的以下操作之一:

分割对象

应用一个函数

结合的结果

在许多情况下，我们将数据分成多个集合，并在每个子集上应用一些函数。在应用函数中，可以执行以下操作:

聚合 - 计算汇总统计

转换 - 执行一些特定于组的操作

过滤 - 在某些情况下丢弃数据

18_groupby.py演示

Pandas合并/连接

Pandas具有功能全面的高性能内存中连接操作，与SQL等关系数据库非常相似。

Pandas提供了一个单独的merge()函数，作为DataFrame对象之间所有标准数据库连接操作的入口：

```
pd.merge(left, right, how='inner', on=None, left_on=None, right_on=None, left_index=False, right_index=False, sort=True)
```

参数说明:

left - 一个DataFrame对象。

right - 另一个DataFrame对象。

on - 列(名称)连接，必须在左和右DataFrame对象中存在(找到)。

left_on - 左侧DataFrame中的列用作键，可以是列名或长度等于DataFrame长度的数组。

right_on - 来自右的DataFrame的列作为键，可以是列名或长度等于DataFrame长度的数组。

left_index - 如果为True，则使用左侧DataFrame中的索引(行标签)作为其连接键。在具有MultiIndex(分层)的DataFrame的情况下，级别的数量必须与来自右DataFrame的连接键的数量相匹配。

right_index - 与右DataFrame的left_index具有相同的用法。

how - 它是left, right, outer以及inner之中的一个，默认为inner。下面将介绍每种方法的用法。

sort - 按照字典顺序通过连接键对结果DataFrame进行排序。默认为True，设置为False时，在很多情况下大大提高性能。

19_merge.py演示

Pandas级联

20_concat.py演示

Pandas提供了各种工具(功能), 可以轻松地将Series, DataFrame对象组合在一起。

```
pd.concat(objs,axis=0,join='outer',join_axes=None,ignore_index=False)
```

参数说明:

objs - 这是Series, DataFrame或Panel对象的序列或映射。

axis - {0, 1, ...}, 默认为0, 这是连接的轴。

join - {'inner', 'outer'}, 默认inner。如何处理其他轴上的索引。联合的外部 and 交叉的内部。

join_axes - 这是Index对象的列表。用于其他(n-1)轴的特定索引, 而不是执行内部/外部集逻辑。

ignore_index - 布尔值, 默认为False。如果指定为True, 则不要使用连接轴上的索引值。

结果轴将被标记为: 0, ..., n-1。

Pandas日期功能

日期功能扩展了时间序列，在财务数据分析中起主要作用。在处理日期数据的同时：

- 1.生成日期序列
- 2.日期序列转换为不同的频率

大量的字符串别名被赋予常用的时间序列频率，称为偏移别名

21_date.py演示

别名	描述说明	别名	描述说明
B	工作日频率	H	小时频率
BQS	商务季度开始频率	MS	月起始频率
D	日历/自然日频率	T, min	分钟的频率
A	年度(年)结束频率	SMS	SMS半开始频率
W	每周频率	S	秒频率
BA	商务年底结束	BMS	商务月开始频率
M	月结束频率	L, ms	毫秒
BAS	商务年度开始频率	Q	季度结束频率
SM	半月结束频率	U, us	微秒
BH	商务时间频率	BQ	商务季度结束频率
SM	半月结束频率	N	纳秒
BH	商务时间频率	BQ	商务季度结束频率
BM	商务月结束频率	QS	季度开始频率

Pandas时间差(Timedelta)

22_timedelta.py演示
时间差(Timedelta)是时间上的差异，以不同的单位来表示。例如：日，小时，分钟，秒。它们可以是正值，也可以是负值。



Pandas分类数据

通常实时的数据包括重复的文本列。例如：性别，国家和代码等特征总是重复的。这些是分类数据的例子。

分类变量只能采用有限的数量，而且通常是固定的数量。除了固定长度，分类数据可能有顺序，但不能执行数字操作。分类是Pandas数据类型。

分类数据类型在以下情况下非常有用：

1. 一个字符串变量，只包含几个不同的值。将这样的字符串变量转换为分类变量将会节省一些内存。
2. 变量的词汇顺序与逻辑顺序("one", "two", "three")不同。通过转换为分类并指定类别上的顺序，排序和最小/最大将使用逻辑顺序，而不是词法顺序。作为其他python库的一个信号，这个列应该被当作一个分类变量(例如，使用合适的统计方法或plot类型)。

23_Categories.py演示

Pandas可视化

24_plot.py演示

Series和DataFrame上的这个功能只是使用matplotlib库的plot()方法的简单包装实现



Pandas IO工具

25_io.py演示

读取文本文件(或平面文件)的两个主要功能是`read_csv()`和`read_table()`。它们都使用相同的解析代码来智能地将表格数据转换为DataFrame对象





感谢您的聆听！

