

4.2 Pandas 的数据框

2023 年 9 月 8 日

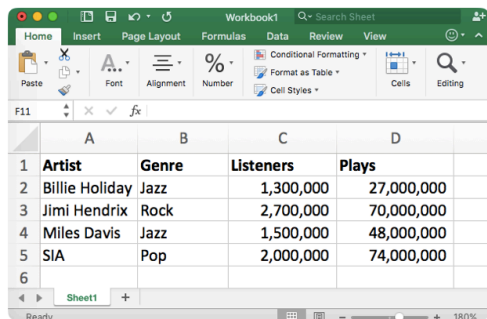
Pandas 的数据框

DataFrame，中文译为数据框，是一种二维带标签的数据结构。您可以将其视为电子表格或 SQL 表，或 Series 对象的字典。它通常是最常用的 pandas 对象。

1. 读取和创建数据框

获取数据框可以通过 2 种方式：读取数据文件、或者创建的方式。读取数据文件的方法如下：

music.csv



pandas.read_csv('music.csv')

	Artist	Genre	Listeners	Plays
0	Billie Holiday	Jazz	1,300,000	27,000,000
1	Jimi Hendrix	Rock	2,700,000	70,000,000
2	Miles Davis	Jazz	1,500,000	48,000,000
3	SIA	Pop	2,000,000	74,000,000

DataFrame 的对象 – 基本上是一个数值表，每一行和每一列都有一个标签。上面的 CSV 文件来自一个音乐流媒体服务的数据，包含 4 个艺术家及其曲风、粉丝数、播放量这四列。

练习

使用 `pd.read_csv()` 加载路径下的 CSV 文件：‘datasets/music.csv’

[]:

如果通过编程的方式创建数据框，那么方法如下：

```
[1]: import pandas as pd
df = pd.DataFrame({'Artist': ['Billie Holiday', 'Jimi Hendrix', 'Miles Davis', 'SIA'],
                  'Genre': ['Jazz', 'Rock', 'Jazz', 'Pop'],
                  'Listeners': [1300000, 2700000, 1500000, 2000000],
                  'Plays': [27000000, 70000000, 48000000, 74000000]})
```

df

	Artist	Genre	Listeners	Plays
0	Billie Holiday	Jazz	1,300,000	27,000,000
1	Jimi Hendrix	Rock	2,700,000	70,000,000
2	Miles Davis	Jazz	1,500,000	48,000,000
3	SIA	Pop	2,000,000	74,000,000

2. 选择

我们可以使用其标签选择任何一列：

```
df['Artists']
```

	Artist
0	Billie Holiday
1	Jimi Hendrix
2	Miles Davis
3	SIA

我们可以使用它们的编号（包括两个边界行的编号）选择一个或多个行：

`df[1:3]`

	Artist	Genre	Listeners	Plays
1	Jimi Hendrix	Rock	2,700,000	70,000,000
2	Miles Davis	Jazz	1,500,000	48,000,000

练习

选择第 0 行

[]:

我们可以用列标和行号用 `.loc` 来选择表格的任何片断（但这里会包括两个边界行号）：

`df.loc[1:3, 'Artist']`

	Artist
1	Jimi Hendrix
2	Miles Davis
3	SIA

练习

选择第 1 和第 2 行，列名为 “Artist” and “Plays” 的部分。

[]:

3. 过滤

我们可以很容易地使用特定行的值来过滤行。例如，这里是我们的爵士乐手：

```
df[df['Genre'] == 'Jazz']
```

	Artist	Genre	Listeners	Plays
0	Billie Holiday	Jazz	1,300,000	27,000,000
2	Miles Davis	Jazz	1,500,000	48,000,000

练习

选择 Genre 是 “Rock” 的行

```
[ ]:
```

以下是拥有超过 180 万名听众的艺术家：

```
df[df['Listeners'] > 1800000]
```

	Artist	Genre	Listeners	Plays
1	Jimi Hendrix	Rock	2,700,000	70,000,000
3	SIA	Pop	2,000,000	74,000,000

练习

选择 “Plays” 小于 50,000,000 的行

```
[ ]:
```

4. 处理缺失值

在你的数据科学旅程中，你要处理的许多数据集都有缺失值。比方说，我们的数据框有一个缺失值：

df

	Artist	Genre	Listeners	Plays
0	Billie Holiday	Jazz	1,300,000	27,000,000
1	Jimi Hendrix	Rock	2,700,000	NaN
2	Miles Davis	Jazz	1,500,000	48,000,000
3	SIA	Pop	2,000,000	74,000,000

Pandas 提供了多种方法来处理这个问题。最简单的是直接放弃有缺失值的行：

`df.dropna()`

	Artist	Genre	Listeners	Plays
0	Billie Holiday	Jazz	1,300,000	27,000,000
2	Miles Davis	Jazz	1,500,000	48,000,000
3	SIA	Pop	2,000,000	74,000,000

另一种方法是用 `fillna()` 填入缺失的值（例如用 0）。

5. 分组

当你开始用某些标准对行进行分组并汇总它们的数据时，事情就会变得非常有趣。例如，让我们按流派“Genre”对我们的数据集进行分组，看看每种流派有多少听众和播放次数：

`df.groupby('Genre').sum()`

	Listeners	Plays
Genre		
Jazz	2,800,000	75,000,000
Pop	2,000,000	74,000,000
Rock	2,700,000	70,000,000

Pandas 将两行“Jazz”爵士乐归为一行，由于我们使用了 `sum()` 聚合，它将两位爵士乐艺术家的听

众和播放次数加在一起，并将总和显示在合并的爵士乐列中。

这不仅有趣，而且是一种极其强大的数据分析方法。现在你知道了 `groupby()`，你就可以折叠数据集并从中发掘出洞察力。聚合是统计智慧的第一支柱，也是统计学的基础工具之一。

除了 `sum()`，pandas 还提供了多个聚合函数，包括计算平均值的 `mean()`，`min()`，`max()`，以及其他多个函数。更多关于 `groupby()` 的信息请参见 Group By 用户指南。

如果你充分使用 `groupby()`，并且不使用 pandas 中的其他东西，那么你就会把 pandas 发挥到极致。但是这个库仍然可以为你提供更多的东西。

练习

按“Genre”分组，使用 `mean()` 作为聚合函数

[]:

6. 从现有的列创建新的列

在数据分析过程中，我们经常发现自己需要从现有的列中创建新的列。Pandas 让这一切变得轻而易举。

```
df['Avg Plays'] = df['Plays']/df['Listeners']
```

	Artist	Genre	Listeners	Plays	Avg Plays
0	Billie Holiday	Jazz	1,300,000	27,000,000	20
1	Jimi Hendrix	Rock	2,700,000	70,000,000	25
2	Miles Davis	Jazz	1,500,000	48,000,000	32
3	SIA	Pop	2,000,000	74,000,000	37

通过告诉 Pandas 用一列除以另一列，它意识到我们要做的是分别除以各个数值（即每行的“Plays”值除以该行的“Listeners”值）。

参考

- 10 Minutes to pandas: <https://pandas.pydata.org/pandas-docs/stable/10min.html>
- A Gentle Visual Intro to Data Analysis in Python Using Pandas: <https://jalammar.github.io/gentle-visual-intro-to-data-analysis-python-pandas/>

[]: