# Манипулирование данными с помощью Pandas

## Нарезка и индексирование фреймов данных

### Установка и удаление индексов

Библиотека pandas позволяет назначать столбцы в качестве индекса. Это облегчает написание более чистого кода при работе с подмножествами данных (а также обеспечивает более эффективный поиск в некоторых случаях).

В этой главе вы будете изучать температуры - это DataFrame средних температур в городах по всему миру. Библиотека pandas загружена как pd.

#### Инструкции

* Посмотрите на temperatures.
* Установите индекс temperatures на "city", присвоив результат переменной temperatures\_ind.
* Посмотрите на temperatures\_ind. В чем разница между temperatures и temperatures\_ind?
* Сбросьте индекс temperatures\_ind, сохраняя его содержимое.
* Сбросьте индекс temperatures\_ind, удалив его содержимое.

import pandas as pd  
temperatures = pd.read\_csv('datasets/temperatures.csv')  
  
# Посмотреть на данные в переменной temperatures  
print()  
  
# Установить индекс для данных в переменной temperatures по городам  
temperatures\_ind =   
  
# Посмотреть на данные в переменной temperatures\_ind  
print()  
  
# Сбросить индекс для данных в переменной temperatures\_ind, сохраняя их содержимое  
print()  
  
# Сбросить индекс для данных в переменной temperatures\_ind, удалив его содержимое  
print()

Unnamed: 0 date city country avg\_temp\_c  
0 0 2000-01-01 Abidjan Côte D'Ivoire 27.293  
1 1 2000-02-01 Abidjan Côte D'Ivoire 27.685  
2 2 2000-03-01 Abidjan Côte D'Ivoire 29.061  
3 3 2000-04-01 Abidjan Côte D'Ivoire 28.162  
4 4 2000-05-01 Abidjan Côte D'Ivoire 27.547  
... ... ... ... ... ...  
16495 16495 2013-05-01 Xian China 18.979  
16496 16496 2013-06-01 Xian China 23.522  
16497 16497 2013-07-01 Xian China 25.251  
16498 16498 2013-08-01 Xian China 24.528  
16499 16499 2013-09-01 Xian China NaN  
  
[16500 rows x 5 columns]  
 Unnamed: 0 date country avg\_temp\_c  
city   
Abidjan 0 2000-01-01 Côte D'Ivoire 27.293  
Abidjan 1 2000-02-01 Côte D'Ivoire 27.685  
Abidjan 2 2000-03-01 Côte D'Ivoire 29.061  
Abidjan 3 2000-04-01 Côte D'Ivoire 28.162  
Abidjan 4 2000-05-01 Côte D'Ivoire 27.547  
... ... ... ... ...  
Xian 16495 2013-05-01 China 18.979  
Xian 16496 2013-06-01 China 23.522  
Xian 16497 2013-07-01 China 25.251  
Xian 16498 2013-08-01 China 24.528  
Xian 16499 2013-09-01 China NaN  
  
[16500 rows x 4 columns]  
 city Unnamed: 0 date country avg\_temp\_c  
0 Abidjan 0 2000-01-01 Côte D'Ivoire 27.293  
1 Abidjan 1 2000-02-01 Côte D'Ivoire 27.685  
2 Abidjan 2 2000-03-01 Côte D'Ivoire 29.061  
3 Abidjan 3 2000-04-01 Côte D'Ivoire 28.162  
4 Abidjan 4 2000-05-01 Côte D'Ivoire 27.547  
... ... ... ... ... ...  
16495 Xian 16495 2013-05-01 China 18.979  
16496 Xian 16496 2013-06-01 China 23.522  
16497 Xian 16497 2013-07-01 China 25.251  
16498 Xian 16498 2013-08-01 China 24.528  
16499 Xian 16499 2013-09-01 China NaN  
  
[16500 rows x 5 columns]  
 Unnamed: 0 date country avg\_temp\_c  
0 0 2000-01-01 Côte D'Ivoire 27.293  
1 1 2000-02-01 Côte D'Ivoire 27.685  
2 2 2000-03-01 Côte D'Ivoire 29.061  
3 3 2000-04-01 Côte D'Ivoire 28.162  
4 4 2000-05-01 Côte D'Ivoire 27.547  
... ... ... ... ...  
16495 16495 2013-05-01 China 18.979  
16496 16496 2013-06-01 China 23.522  
16497 16497 2013-07-01 China 25.251  
16498 16498 2013-08-01 China 24.528  
16499 16499 2013-09-01 China NaN  
  
[16500 rows x 4 columns]

### Выделение подмножества с помощью .loc[]

Одной из ключевых возможностей индексов является метод .loc[], предназначенный для выделения подмножества данных на основе значений индекса. Если передать ему один аргумент, он выделит соответствующее подмножество строк.

Код для выделения подмножества с использованием .loc[] может быть более читаемым, чем стандартное выделение с помощью квадратных скобок, что делает ваш код менее трудоемким при поддержке.

Библиотека pandas загружена как pd. Доступны temperatures и temperatures\_ind; последний индексирован по city.

#### Инструкции

* Создайте список под названием cities, содержащий "Москва" и "Санкт-Петербург".
* Используйте [] для фильтрации temperatures по строкам, где столбец city принимает значение из списка cities.
* Используйте .loc[] для фильтрации temperatures\_ind по строкам, где город находится в списке cities.

# Создание списка городов для выделения подмножества  
cities = ["\_\_\_\_", "\_\_\_\_"]  
  
# Выделение подмножества в переменной temperatures с использованием квадратных скобок  
print(temperatures[\_\_\_\_])  
  
# Выделение подмножества в переменной temperatures\_ind с использованием .loc[]  
print(temperatures\_ind.loc[\_\_\_\_])

Unnamed: 0 date city country avg\_temp\_c  
10725 10725 2000-01-01 Moscow Russia -7.313  
10726 10726 2000-02-01 Moscow Russia -3.551  
10727 10727 2000-03-01 Moscow Russia -1.661  
10728 10728 2000-04-01 Moscow Russia 10.096  
10729 10729 2000-05-01 Moscow Russia 10.357  
... ... ... ... ... ...  
13360 13360 2013-05-01 Saint Petersburg Russia 12.355  
13361 13361 2013-06-01 Saint Petersburg Russia 17.185  
13362 13362 2013-07-01 Saint Petersburg Russia 17.234  
13363 13363 2013-08-01 Saint Petersburg Russia 17.153  
13364 13364 2013-09-01 Saint Petersburg Russia NaN  
  
[330 rows x 5 columns]  
 Unnamed: 0 date country avg\_temp\_c  
city   
Moscow 10725 2000-01-01 Russia -7.313  
Moscow 10726 2000-02-01 Russia -3.551  
Moscow 10727 2000-03-01 Russia -1.661  
Moscow 10728 2000-04-01 Russia 10.096  
Moscow 10729 2000-05-01 Russia 10.357  
... ... ... ... ...  
Saint Petersburg 13360 2013-05-01 Russia 12.355  
Saint Petersburg 13361 2013-06-01 Russia 17.185  
Saint Petersburg 13362 2013-07-01 Russia 17.234  
Saint Petersburg 13363 2013-08-01 Russia 17.153  
Saint Petersburg 13364 2013-09-01 Russia NaN  
  
[330 rows x 4 columns]

### Установка многоуровневых индексов

Индексы также могут состоять из нескольких столбцов, формируя многоуровневый индекс (иногда называемый иерархическим индексом). Это имеет свои особенности.

Преимущество заключается в том, что многоуровневые индексы делают более естественным рассуждение о вложенных категориальных переменных. Например, в клиническом исследовании у вас может быть контрольная и терапевтическая группы. Затем каждый испытуемый принадлежит к одной из этих групп, и мы можем сказать, что испытуемый вложен в одну из групп. Точно так же, в наборе данных о температуре города находятся в стране, и мы можем сказать, что город вложен в страну.

Основной недостаток заключается в том, что код для работы с индексами отличается от кода для работы со столбцами, поэтому вам приходится изучать два синтаксиса и следить за тем, как представлены ваши данные.

Библиотека pandas загружена как pd. Доступна переменная temperatures.

#### Инструкции

* Установите индекс для temperatures по столбцам "country" и "city", присвоив это temperatures\_ind.
* Укажите две пары страна/город для сохранения: "Бразилия"/"Рио-де-Жанейро" и "Пакистан"/"Лахор", присвоив переменной rows\_to\_keep.
* Выведите на печать и выделите подмножество temperatures\_ind для rows\_to\_keep с использованием .loc[].

# Индексирование температур по стране и городу  
temperatures\_ind = \_\_\_\_  
  
# Список кортежей: Бразилия, Рио-де-Жанейро и Пакистан, Лахор  
rows\_to\_keep = [\_\_\_\_]  
  
# Выделение подмножества для указанных строк  
print(temperatures\_ind.\_\_\_\_)

Unnamed: 0 date avg\_temp\_c  
country city   
Brazil Rio De Janeiro 12540 2000-01-01 25.974  
 Rio De Janeiro 12541 2000-02-01 26.699  
 Rio De Janeiro 12542 2000-03-01 26.270  
 Rio De Janeiro 12543 2000-04-01 25.750  
 Rio De Janeiro 12544 2000-05-01 24.356  
... ... ... ...  
Pakistan Lahore 8575 2013-05-01 33.457  
 Lahore 8576 2013-06-01 34.456  
 Lahore 8577 2013-07-01 33.279  
 Lahore 8578 2013-08-01 31.511  
 Lahore 8579 2013-09-01 NaN  
  
[330 rows x 3 columns]

### Сортировка по значениям индекса

Ранее вы изменяли порядок строк в DataFrame, вызывая метод .sort\_values(). Также полезно иметь возможность сортировать по элементам индекса. Для этого нужно использовать метод .sort\_index().

Библиотека pandas загружена как pd. temperatures\_ind имеет многоуровневый индекс с категориями country и city и доступен для использования.

#### Инструкции

* Отсортируйте temperatures\_ind по значениям индекса.
* Отсортируйте temperatures\_ind по значениям индекса на уровне "city".
* Отсортируйте temperatures\_ind по возрастанию страны, а затем по убыванию города.

# Сортировка temperatures\_ind по значениям индекса  
print(\_\_\_\_)  
  
# Сортировка temperatures\_ind по значениям индекса на уровне города  
print(\_\_\_\_)  
  
# Сортировка temperatures\_ind сначала по стране, затем по убыванию города  
print(\_\_\_\_)

Unnamed: 0 date avg\_temp\_c  
country city   
Afghanistan Kabul 7260 2000-01-01 3.326  
 Kabul 7261 2000-02-01 3.454  
 Kabul 7262 2000-03-01 9.612  
 Kabul 7263 2000-04-01 17.925  
 Kabul 7264 2000-05-01 24.658  
... ... ... ...  
Zimbabwe Harare 5605 2013-05-01 18.298  
 Harare 5606 2013-06-01 17.020  
 Harare 5607 2013-07-01 16.299  
 Harare 5608 2013-08-01 19.232  
 Harare 5609 2013-09-01 NaN  
  
[16500 rows x 3 columns]  
 Unnamed: 0 date avg\_temp\_c  
country city   
Côte D'Ivoire Abidjan 0 2000-01-01 27.293  
 Abidjan 1 2000-02-01 27.685  
 Abidjan 2 2000-03-01 29.061  
 Abidjan 3 2000-04-01 28.162  
 Abidjan 4 2000-05-01 27.547  
... ... ... ...  
China Xian 16495 2013-05-01 18.979  
 Xian 16496 2013-06-01 23.522  
 Xian 16497 2013-07-01 25.251  
 Xian 16498 2013-08-01 24.528  
 Xian 16499 2013-09-01 NaN  
  
[16500 rows x 3 columns]  
 Unnamed: 0 date avg\_temp\_c  
country city   
Afghanistan Kabul 7260 2000-01-01 3.326  
 Kabul 7261 2000-02-01 3.454  
 Kabul 7262 2000-03-01 9.612  
 Kabul 7263 2000-04-01 17.925  
 Kabul 7264 2000-05-01 24.658  
... ... ... ...  
Zimbabwe Harare 5605 2013-05-01 18.298  
 Harare 5606 2013-06-01 17.020  
 Harare 5607 2013-07-01 16.299  
 Harare 5608 2013-08-01 19.232  
 Harare 5609 2013-09-01 NaN  
  
[16500 rows x 3 columns]

### Выделение срезов значений индекса

Срезы позволяют выбирать последовательные элементы объекта, используя синтаксис первый:последний. DataFrame можно нарезать по значениям индекса или по номеру строки/столбца; мы начнем с первого случая. Это включает выделение срезов внутри метода .loc[].

По сравнению с нарезкой списков, есть несколько моментов, которые следует запомнить.

* Вы можете делать срез индекса только если индекс отсортирован (с использованием .sort\_index()).
* Для выделения среза на внешнем уровне первый и последний могут быть строками.
* Для выделения среза на внутренних уровнях первый и последний должны быть кортежами.
* Если вы передаете единичный срез в .loc[], он будет выделять строки.
* Библиотека pandas загружена как pd. temperatures\_ind содержит в индексе страну и город и доступен для использования.

#### Инструкции

* Отсортируйте индекс в temperatures\_ind.
* Используйте нарезку с помощью .loc[], чтобы получить следующие подмножества:
  + от Пакистана до России.
  + от Лахора до Москвы. (Это вернет бессмыслицу.)
  + от Пакистана, Лахора до России, Москвы.

# Сортировка индекса в temperatures\_ind  
temperatures\_srt = \_\_\_\_  
  
# Выделение строк от Пакистана до России  
print(\_\_\_\_)  
  
# Попытка выделить строки от Лахора до Москвы  
print(\_\_\_\_)  
  
# Выделение строк от Пакистана, Лахора до России, Москвы  
print(\_\_\_\_)

Unnamed: 0 date avg\_temp\_c  
country city   
Pakistan Faisalabad 4785 2000-01-01 12.792  
 Faisalabad 4786 2000-02-01 14.339  
 Faisalabad 4787 2000-03-01 20.309  
 Faisalabad 4788 2000-04-01 29.072  
 Faisalabad 4789 2000-05-01 34.845  
... ... ... ...  
Russia Saint Petersburg 13360 2013-05-01 12.355  
 Saint Petersburg 13361 2013-06-01 17.185  
 Saint Petersburg 13362 2013-07-01 17.234  
 Saint Petersburg 13363 2013-08-01 17.153  
 Saint Petersburg 13364 2013-09-01 NaN  
  
[1155 rows x 3 columns]  
 Unnamed: 0 date avg\_temp\_c  
country city   
Mexico Mexico 10230 2000-01-01 12.694  
 Mexico 10231 2000-02-01 14.677  
 Mexico 10232 2000-03-01 17.376  
 Mexico 10233 2000-04-01 18.294  
 Mexico 10234 2000-05-01 18.562  
... ... ... ...  
Morocco Casablanca 3130 2013-05-01 19.217  
 Casablanca 3131 2013-06-01 23.649  
 Casablanca 3132 2013-07-01 27.488  
 Casablanca 3133 2013-08-01 27.952  
 Casablanca 3134 2013-09-01 NaN  
  
[330 rows x 3 columns]  
 Unnamed: 0 date avg\_temp\_c  
country city   
Pakistan Lahore 8415 2000-01-01 12.792  
 Lahore 8416 2000-02-01 14.339  
 Lahore 8417 2000-03-01 20.309  
 Lahore 8418 2000-04-01 29.072  
 Lahore 8419 2000-05-01 34.845  
... ... ... ...  
Russia Moscow 10885 2013-05-01 16.152  
 Moscow 10886 2013-06-01 18.718  
 Moscow 10887 2013-07-01 18.136  
 Moscow 10888 2013-08-01 17.485  
 Moscow 10889 2013-09-01 NaN  
  
[660 rows x 3 columns]

### Срезка в обе стороны

Вы уже видели, как можно нарезать DataFrame по строкам и по столбцам, но поскольку DataFrame - это двумерный объект, часто естественно выделять сразу оба измерения. То есть, передавая два аргумента в .loc[], вы можете выделять как строки, так и столбцы одновременно.

Библиотека pandas загружена как pd. temperatures\_srt проиндексирован по стране и городу, имеет отсортированный индекс и доступен для использования.

#### Инструкции

Используйте срез с помощью .loc[], чтобы выделить строки от Индии, Хайдарабада до Ирака, Багдада. Используйте срез с помощью .loc[], чтобы выделить столбцы от date до avg\_temp\_c. Выделите срез одновременно и по строкам (от Хайдарабада до Багдада) и по столбцам (от date до avg\_temp\_c) с помощью .loc[].

# Выделение строк от Индии, Хайдарабада до Ирака, Багдада  
print(\_\_\_\_)  
  
# Выделение столбцов от date до avg\_temp\_c  
print(\_\_\_\_)  
  
# Выделение одновременно и по строкам (от Индии, Хайдарабада до Ирака, Багдада) и по столбцам (от date до avg\_temp\_c)  
print(\_\_\_\_)

Unnamed: 0 date avg\_temp\_c  
country city   
India Hyderabad 5940 2000-01-01 23.779  
 Hyderabad 5941 2000-02-01 25.826  
 Hyderabad 5942 2000-03-01 28.821  
 Hyderabad 5943 2000-04-01 32.698  
 Hyderabad 5944 2000-05-01 32.438  
... ... ... ...  
Iraq Baghdad 1150 2013-05-01 28.673  
 Baghdad 1151 2013-06-01 33.803  
 Baghdad 1152 2013-07-01 36.392  
 Baghdad 1153 2013-08-01 35.463  
 Baghdad 1154 2013-09-01 NaN  
  
[2145 rows x 3 columns]  
 date avg\_temp\_c  
country city   
Afghanistan Kabul 2000-01-01 3.326  
 Kabul 2000-02-01 3.454  
 Kabul 2000-03-01 9.612  
 Kabul 2000-04-01 17.925  
 Kabul 2000-05-01 24.658  
... ... ...  
Zimbabwe Harare 2013-05-01 18.298  
 Harare 2013-06-01 17.020  
 Harare 2013-07-01 16.299  
 Harare 2013-08-01 19.232  
 Harare 2013-09-01 NaN  
  
[16500 rows x 2 columns]  
 date avg\_temp\_c  
country city   
India Hyderabad 2000-01-01 23.779  
 Hyderabad 2000-02-01 25.826  
 Hyderabad 2000-03-01 28.821  
 Hyderabad 2000-04-01 32.698  
 Hyderabad 2000-05-01 32.438  
... ... ...  
Iraq Baghdad 2013-05-01 28.673  
 Baghdad 2013-06-01 33.803  
 Baghdad 2013-07-01 36.392  
 Baghdad 2013-08-01 35.463  
 Baghdad 2013-09-01 NaN  
  
[2145 rows x 2 columns]

### Нарезка временных рядов

Нарезка особенно полезна для временных рядов, поскольку часто бывает необходимо отфильтровать данные в определенном диапазоне дат. Добавьте столбец с датой в индекс, затем используйте .loc[] для выделения подмножества. Важно помнить о хранении дат в формате ISO 8601, то есть "гггг-мм-дд" для год-месяц-день, "гггг-мм" для год-месяц и "гггг" для года.

Напомним из главы 1, что вы можете объединять несколько булевых условий, используя логические операторы, такие как &. Чтобы сделать это в одной строке кода, вам нужно будет добавить круглые скобки () вокруг каждого условия.

Библиотека pandas загружена как pd, и temperatures, без индекса, доступны.

#### Инструкции

* Используйте логические условия, а не .isin() или .loc[], и полные даты "гггг-мм-дд", чтобы выделить подмножество температур для строк за 2010 и 2011 годы, и выведите результаты.
* Установите индекс для temperatures в столбец с датой и отсортируйте его.
* Используйте .loc[] для выделения подмножества temperatures\_ind для строк за 2010 и 2011 годы.
* Используйте .loc[] для выделения подмножества temperatures\_ind для строк с августа 2010 года по февраль 2011 года.

# Используйте логические условия для выделения строк температур за 2010 и 2011 годы  
temperatures\_bool = \_\_\_\_[(\_\_\_\_ >= \_\_\_\_) & (\_\_\_\_ <= \_\_\_\_)]  
print(temperatures\_bool)  
  
# Установите дату в качестве индекса и отсортируйте индекс  
temperatures\_ind = temperatures.\_\_\_\_.\_\_\_\_  
  
# Используйте .loc[] для выделения подмножества temperatures\_ind для строк за 2010 и 2011 годы  
print(\_\_\_\_)  
  
# Используйте .loc[] для выделения подмножества temperatures\_ind для строк с августа 2010 по февраль 2011 года  
print(\_\_\_\_)

Unnamed: 0 date city country avg\_temp\_c  
120 120 2010-01-01 Abidjan Côte D'Ivoire 28.270  
121 121 2010-02-01 Abidjan Côte D'Ivoire 29.262  
122 122 2010-03-01 Abidjan Côte D'Ivoire 29.596  
123 123 2010-04-01 Abidjan Côte D'Ivoire 29.068  
124 124 2010-05-01 Abidjan Côte D'Ivoire 28.258  
... ... ... ... ... ...  
16474 16474 2011-08-01 Xian China 23.069  
16475 16475 2011-09-01 Xian China 16.775  
16476 16476 2011-10-01 Xian China 12.587  
16477 16477 2011-11-01 Xian China 7.543  
16478 16478 2011-12-01 Xian China -0.490  
  
[2400 rows x 5 columns]  
 Unnamed: 0 city country avg\_temp\_c  
date   
2010-01-01 4905 Faisalabad Pakistan 11.810  
2010-01-01 10185 Melbourne Australia 20.016  
2010-01-01 3750 Chongqing China 7.921  
2010-01-01 13155 São Paulo Brazil 23.738  
2010-01-01 5400 Guangzhou China 14.136  
... ... ... ... ...  
2010-12-01 6896 Jakarta Indonesia 26.602  
2010-12-01 5246 Gizeh Egypt 16.530  
2010-12-01 11186 Nagpur India 19.120  
2010-12-01 14981 Sydney Australia 19.559  
2010-12-01 13496 Salvador Brazil 26.265  
  
[1200 rows x 4 columns]  
 Unnamed: 0 city country avg\_temp\_c  
date   
2010-08-01 2602 Calcutta India 30.226  
2010-08-01 12337 Pune India 24.941  
2010-08-01 6562 Izmir Turkey 28.352  
2010-08-01 15637 Tianjin China 25.543  
2010-08-01 9862 Manila Philippines 27.101  
... ... ... ... ...  
2011-01-01 4257 Dar Es Salaam Tanzania 28.541  
2011-01-01 11352 Nairobi Kenya 17.768  
2011-01-01 297 Addis Abeba Ethiopia 17.708  
2011-01-01 11517 Nanjing China 0.144  
2011-01-01 11847 New York United States -4.463  
  
[600 rows x 4 columns]

### Выделение по номеру строки/столбца

Самые распространенные способы выделения строк - это те, о которых мы ранее говорили: использование булевого условия или меток индекса. Однако иногда полезно также использовать номера строк.

Это делается с помощью .iloc[], и, как и .loc[], он может принимать два аргумента для выделения строк и столбцов.

Библиотека pandas загружена как pd. temperatures (без индекса) доступны.

#### Инструкции

* Используйте .iloc[] на объекте temperatures для выделения подмножества.
* Получите 23-ю строку, 2-й столбец (индексные позиции 22 и 1).
* Получите первые 5 строк (индексные позиции с 0 по 4).
* Получите все строки, столбцы 3 и 4 (индексные позиции с 2 по 4).
* Получите первые 5 строк, столбцы 3 и 4.

# Получить 23-ю строку, 2-й столбец (индекс 22, 1)  
print(\_\_\_\_)  
  
# Использовать нарезку для получения первых 5 строк  
print(\_\_\_\_)  
  
# Использовать нарезку для получения столбцов с 3 по 4  
print(\_\_\_\_)  
  
# Использовать нарезку в обе стороны одновременно  
print(\_\_\_\_)

2001-11-01  
 Unnamed: 0 date city country avg\_temp\_c  
0 0 2000-01-01 Abidjan Côte D'Ivoire 27.293  
1 1 2000-02-01 Abidjan Côte D'Ivoire 27.685  
2 2 2000-03-01 Abidjan Côte D'Ivoire 29.061  
3 3 2000-04-01 Abidjan Côte D'Ivoire 28.162  
4 4 2000-05-01 Abidjan Côte D'Ivoire 27.547  
 city country  
0 Abidjan Côte D'Ivoire  
1 Abidjan Côte D'Ivoire  
2 Abidjan Côte D'Ivoire  
3 Abidjan Côte D'Ivoire  
4 Abidjan Côte D'Ivoire  
... ... ...  
16495 Xian China  
16496 Xian China  
16497 Xian China  
16498 Xian China  
16499 Xian China  
  
[16500 rows x 2 columns]  
 city country  
0 Abidjan Côte D'Ivoire  
1 Abidjan Côte D'Ivoire  
2 Abidjan Côte D'Ivoire  
3 Abidjan Côte D'Ivoire  
4 Abidjan Côte D'Ivoire

### Поворот температуры по городу и году

Интересно посмотреть, как меняются температуры для каждого города со временем - рассматривать каждый месяц приводит к большой таблице, что может быть сложно для анализа. Вместо этого давайте посмотрим, как меняются температуры по годам.

Вы можете получить доступ к компонентам даты (год, месяц и день) с помощью кода вида dataframe["столбец"].dt.component. Например, компонент месяца - dataframe["столбец"].dt.month, а компонент года - dataframe["столбец"].dt.year.

После получения столбца с годом вы можете создать сводную таблицу с данными, сгруппированными по городу и году, что вы и будете исследовать в предстоящих упражнениях.

Библиотека pandas загружена как pd. temperatures доступны.

#### Инструкции

* Добавьте столбец с годом в данные о температуре из компонента года столбца с датой.
* Создайте сводную таблицу для столбца avg\_temp\_c, где строки будут представлять собой страну и город, а столбцы - год. Присвойте переменной temp\_by\_country\_city\_vs\_year и посмотрите на результат.

# Добавить столбец с годом в данные о температуре  
\_\_\_\_\_  
  
# Создать сводную таблицу для avg\_temp\_c по стране и городу по году  
temp\_by\_country\_city\_vs\_year = \_\_\_\_  
  
# Посмотреть результат  
print(temp\_by\_country\_city\_vs\_year)

year 2000 2001 2002 2003 \  
country city   
Afghanistan Kabul 15.822667 15.847917 15.714583 15.132583   
Angola Luanda 24.410333 24.427083 24.790917 24.867167   
Australia Melbourne 14.320083 14.180000 14.075833 13.985583   
 Sydney 17.567417 17.854500 17.733833 17.592333   
Bangladesh Dhaka 25.905250 25.931250 26.095000 25.927417   
... ... ... ... ...   
United States Chicago 11.089667 11.703083 11.532083 10.481583   
 Los Angeles 16.643333 16.466250 16.430250 16.944667   
 New York 9.969083 10.931000 11.252167 9.836000   
Vietnam Ho Chi Minh City 27.588917 27.831750 28.064750 27.827667   
Zimbabwe Harare 20.283667 20.861000 21.079333 20.889167   
  
year 2004 2005 2006 2007 \  
country city   
Afghanistan Kabul 16.128417 14.847500 15.798500 15.518000   
Angola Luanda 24.216167 24.414583 24.138417 24.241583   
Australia Melbourne 13.742083 14.378500 13.991083 14.991833   
 Sydney 17.869667 18.028083 17.749500 18.020833   
Bangladesh Dhaka 26.136083 26.193333 26.440417 25.951333   
... ... ... ... ...   
United States Chicago 10.943417 11.583833 11.870500 11.448333   
 Los Angeles 16.552833 16.431417 16.623083 16.699917   
 New York 10.389500 10.681417 11.519250 10.627333   
Vietnam Ho Chi Minh City 27.686583 27.884000 28.044000 27.866667   
Zimbabwe Harare 20.307667 21.487417 20.699750 20.746250   
  
year 2008 2009 2010 2011 \  
country city   
Afghanistan Kabul 15.479250 15.093333 15.676000 15.812167   
Angola Luanda 24.266333 24.325083 24.440250 24.150750   
Australia Melbourne 14.110583 14.647417 14.231667 14.190917   
 Sydney 17.321083 18.175833 17.999000 17.713333   
Bangladesh Dhaka 26.004500 26.535583 26.648167 25.803250   
... ... ... ... ...   
United States Chicago 10.242417 10.298333 11.815917 11.214250   
 Los Angeles 17.014750 16.677000 15.887000 15.874833   
 New York 10.641667 10.141833 11.357583 11.272250   
Vietnam Ho Chi Minh City 27.611417 27.853333 28.281750 27.675417   
Zimbabwe Harare 20.680500 20.523833 21.165833 20.781750   
  
year 2012 2013   
country city   
Afghanistan Kabul 14.510333 16.206125   
Angola Luanda 24.240083 24.553875   
Australia Melbourne 14.268667 14.741500   
 Sydney 17.474333 18.089750   
Bangladesh Dhaka 26.283583 26.587000   
... ... ...   
United States Chicago 12.821250 11.586889   
 Los Angeles 17.089583 18.120667   
 New York 11.971500 12.163889   
Vietnam Ho Chi Minh City 28.248750 28.455000   
Zimbabwe Harare 20.523333 19.756500   
  
[100 rows x 14 columns]

### Выделение подмножества сводных таблиц

Сводная таблица - это просто DataFrame с отсортированными индексами, поэтому техники, которые вы уже изучили, могут быть использованы для их выделения. В частности, комбинация .loc[] + нарезки часто бывает полезной.

Библиотека pandas загружена как pd. temp\_by\_country\_city\_vs\_year доступна.

#### Инструкции

Используйте .loc[] для выделения подмножества из temp\_by\_country\_city\_vs\_year.

* От Египта до Индии.
* От Египта, Каира до Индии, Дели.
* От Египта, Каира до Индии, Дели и с 2005 по 2010 год.

# Выделение для Египта до Индии  
\_\_\_\_\_  
  
# Выделение для Египта, Каира до Индии, Дели  
\_\_\_\_\_  
  
# Выделение для Египта, Каира до Индии, Дели, и с 2005 по 2010 год  
\_\_\_\_\_  
  
print(temp\_Egypt\_Cairo\_2005\_India\_Delhi\_2010)

year 2005 2006 2007 2008 2009 \  
country city   
Egypt Cairo 22.006500 22.050000 22.361000 22.644500 22.625000   
 Gizeh 22.006500 22.050000 22.361000 22.644500 22.625000   
Ethiopia Addis Abeba 18.312833 18.427083 18.142583 18.165000 18.765333   
France Paris 11.552917 11.788500 11.750833 11.278250 11.464083   
Germany Berlin 9.919083 10.545333 10.883167 10.657750 10.062500   
India Ahmadabad 26.828083 27.282833 27.511167 27.048500 28.095833   
 Bangalore 25.476500 25.418250 25.464333 25.352583 25.725750   
 Bombay 27.035750 27.381500 27.634667 27.177750 27.844500   
 Calcutta 26.729167 26.986250 26.584583 26.522333 27.153250   
 Delhi 25.716083 26.365917 26.145667 25.675000 26.554250   
  
year 2010   
country city   
Egypt Cairo 23.718250   
 Gizeh 23.718250   
Ethiopia Addis Abeba 18.298250   
France Paris 10.409833   
Germany Berlin 8.606833   
India Ahmadabad 28.017833   
 Bangalore 25.705250   
 Bombay 27.765417   
 Calcutta 27.288833   
 Delhi 26.520250

### Расчеты на сводной таблице

Сводные таблицы заполнены сводными статистиками, но это только первый шаг к нахождению информативных данных. Часто вам нужно выполнять дополнительные расчеты с ними. Часто нужно найти строки или столбцы, где значение максимально или минимально.

Напомним, что вы легко можете выделить строки интереса в Series или DataFrame, используя логическое условие в квадратных скобках. Например: series[series > value].

Библиотека pandas загружена как pd, и DataFrame temp\_by\_country\_city\_vs\_year доступен.

#### Инструкции

* Вычислите среднюю температуру для каждого года и присвойте результат переменной mean\_temp\_by\_year.
* Отфильтруйте mean\_temp\_by\_year для года с наибольшей средней температурой.
* Вычислите среднюю температуру для каждого города (по столбцам) и присвойте результат переменной mean\_temp\_by\_city.
* Отфильтруйте mean\_temp\_by\_city для города с наименьшей средней температурой.

# temp\_by\_country\_city\_vs\_year  
  
# Получение среднегодовой температуры по всему миру  
mean\_temp\_by\_year = temp\_by\_country\_city\_vs\_year.\_\_\_\_  
  
# Отфильтровать для года с наибольшей среднегодовой температурой  
print(mean\_temp\_by\_year[\_\_\_\_])  
  
# year\_with\_max\_temp = mean\_temp\_by\_year[mean\_temp\_by\_year == mean\_temp\_by\_year.max()].index  
  
  
# Получение средней температуры по городам  
mean\_temp\_by\_city = temp\_by\_country\_city\_vs\_year.\_\_\_\_  
  
# Отфильтровать для города с наименьшей средней температурой  
print(mean\_temp\_by\_city[\_\_\_\_])

year  
2013 20.312285  
dtype: float64  
country city   
China Harbin 4.876551  
dtype: float64