Словари и pandas

Мотивация для словарей

Чтобы понять, почему словари полезны, взгляните на два списка, определенные в скрипте. страны содержит названия некоторых европейских «стран». `capitals` перечисляет соответствующие названия их столицы.

Инструкции

- Используйте метод ****index()**** для стран, чтобы найти индекс Германии. Сохраните этот индекс как `ind_ger`.
- Используйте `ind_ger` для доступа к «столице» Германии из списка столиц. Распечатай.

```
# Определение стран и их столиц
countries = ['spain', 'france', 'germany', 'norway']
capitals = ['madrid', 'paris', 'berlin', 'oslo']

# Получение индекса 'germany': ind_ger
ind_ger = countries.index('germany')

# Использование ind_ger для вывода столицы Германии
print(capitals[ind_ger])
```

```
berlin
```

Инструкции

Со строками в `countries` и `capitals` создайте словарь под названием `europe` с четырьмя парами ключ:значение. Остерегайтесь капитализации! Убедитесь, что вы везде используете строчные буквы. Распечатайте слово «Европа» и проверьте, соответствует ли результат вашим ожиданиям.

```
# Определение стран и их столиц
countries = ['spain', 'france', 'germany', 'norway']
capitals = ['madrid', 'paris', 'berlin', 'oslo']

# Получение индекса 'germany': ind_ger
ind_ger = countries.index('germany')

# Использование ind_ger для вывода столицы Германии
print(capitals[ind_ger])
```

```
berlin
```

Доступ к словарю

Если ключи словаря выбраны разумно, доступ к значениям в словаре будет простым и интуитивно понятным. Например, чтобы получить столицу Франции из Европы, вы можете использовать:

```
Европа['Франция']
```

Здесь `france` — это ключ, а `paris` — возвращаемое значение.

Инструкции

- Проверьте, какие ключи есть в `europe`, вызвав метод `keys()`. Выведите результат.
- Выведите значение, которое принадлежит ключу '`norway`'.

```
# Словарь словарей
europe = { 'spain': { 'capital':'madrid', 'population':46.77 },
           'france': { 'capital':'paris', 'population':66.03 },
           'germany': { 'capital':'berlin', 'population':80.62 },
           'norway': { 'capital':'oslo', 'population':5.084 } }

# Вывод столицы Франции
europe['france']['capital']

# Создание подсловаря data
data = {'capital':'rome','population':59.83}

# Добавление данных в europe под ключ 'italy'
europe['italy'] = data

# Вывод europe
print(europe)
```

```
{'spain': {'capital': 'madrid', 'population': 46.77}, 'france':
{'capital': 'paris', 'population': 66.03}, 'germany': {'capital':
'berlin', 'population': 80.62}, 'norway': {'capital': 'oslo',
'population': 5.084}, 'italy': {'capital': 'rome', 'population':
59.83}}
```

Управление словарями (1)

Если вы знаете, как получить доступ к словарю, вы также можете присвоить ему новое значение. Чтобы добавить новую пару ключ-значение в europe, вы можете использовать что-то вроде этого:

```
europe['iceland'] = 'reykjavik'
```

Инструкции

- Добавьте ключ '`italy`' со значением '`rome`' в словарь europe.

- Чтобы убедиться, что '`italy`' теперь является ключом в europe, выведите '`italy`' из `europe`.
- Добавьте еще одну пару ключ-значение в europe: ключ '`poland`', значение '`warsaw`'.
- Выведите `europe`.

```
# Определение словаря
europe = {'spain':'madrid', 'france':'paris', 'germany':'berlin',
'norway':'oslo' }

# Добавление Италии в Европу
europe['italy'] = 'rome'

# Вывод Италии в Европе
print('italy' in europe)

# Добавление Польши в Европу
europe['poland'] = 'warsaw'

# Вывод Европы
print(europe)
```

```
True
{'spain': 'madrid', 'france': 'paris', 'germany': 'berlin', 'norway':
'oslo', 'italy': 'rome', 'poland': 'warsaw'}
```

```
# Заранее определенные списки
names = ['Соединенные Штаты', 'Австралия', 'Япония', 'Индия',
'Россия', 'Марокко', 'Египет']
dr = [True, False, False, False, True, True, True]
cpc = [809, 731, 588, 18, 200, 70, 45]

# Импорт pandas как pd
import pandas as pd

# Создание словаря my_dict с тремя парами ключ:значение: my_dict
my_dict = {'country': names, 'drives_right': dr, 'cars_per_cap': cpc}

# Создание DataFrame cars из my_dict: cars
cars = pd.DataFrame(my_dict)

# Вывод cars
print(cars)
```

```
              country  drives_right  cars_per_cap
0  Соединенные Штаты          True           809
1          Австралия         False           731
2             Япония         False           588
3              Индия         False            18
4             Россия          True           200
5            Марокко          True            70
6             Египет          True            45
```

Изменение словаря (2)

- Кто-то решил пошутить и изменил ваш точно сгенерированный словарь. Адаптированная версия словаря `europe` доступна в скрипте.

Можете его очистить? Не меняйте определение `europe`, а просто добавьте команды Python в скрипт для обновления и удаления пар ключ:значение.

Инструкции

- Столица Германии не '`Bonn`'; это '`Berlin`'. Обновите его значение.
- Австралия не входит в Европу, Австрия! Удалите ключ '`australia`' из `europe`.
- Напечатайте `europe`, чтобы увидеть, сработала ли ваша очистка.

```
# Определение словаря
europe = {'spain':'madrid', 'france':'paris', 'germany':'bonn',
          'norway':'oslo', 'italy':'rome', 'poland':'warsaw',
          'australia':'vienna' }

# Обновление столицы Германии
europe['germany'] = 'berlin'

# Удаление Австралии
del(europe['australia'])

# Вывод словаря europe
print(europe)
```

```
{'spain': 'madrid', 'france': 'paris', 'germany': 'berlin', 'norway':
'oslo', 'italy': 'rome', 'poland': 'warsaw'}
```

Словари в словарях

Помните списки? Они могли содержать всё, даже другие списки. Для словарей то же самое. Словари могут содержать пары ключ:значение, где значения снова являются словарями.

Для примера взгляните на скрипт, в котором создана другая версия europe - словаря, с которым вы работали. Ключи все еще - это названия стран, но значения представляют собой словари, которые содержат больше информации, чем просто столицы.

Совершенно возможно цеплять квадратные скобки для выбора элементов. Чтобы получить население Испании из europe, например, вам нужно:

```
europe['spain']['population']
```

Инструкции

- Используйте цепочку квадратных скобок, чтобы выбрать и вывести столицу Франции.
- Создайте словарь с именем `data`, с ключами '`capital`' и '`population`'. Установите их соответственно '`rome`' и 59.83.

- Добавьте новую пару ключ-значение в europe; ключ - '`italy`', а значение - `data`, словарь, который вы только что создали.

```
# Словарь из словарей
europe = { 'spain': { 'capital':'madrid', 'population':46.77 },
           'france': { 'capital':'paris', 'population':66.03 },
           'germany': { 'capital':'berlin', 'population':80.62 },
           'norway': { 'capital':'oslo', 'population':5.084 } }

# Вывести столицу Франции
europe['france']['capital']

# Создать подсловарь data
data = {'capital':'rome','population':59.83}

# Добавить data в словарь europe под ключом 'italy'
europe['italy'] = data

# Вывести europe
print(europe)
```

```
{'spain': {'capital': 'madrid', 'population': 46.77}, 'france':
{'capital': 'paris', 'population': 66.03}, 'germany': {'capital':
'berlin', 'population': 80.62}, 'norway': {'capital': 'oslo',
'population': 5.084}, 'italy': {'capital': 'rome', 'population':
59.83}}
```

Pandas

Словарь в DataFrame (1)

Pandas - это библиотека с открытым исходным кодом, предоставляющая высокопроизводительные и простые в использовании структуры данных и инструменты для анализа данных на Python. Звучит многообещающе!

DataFrame - одна из самых важных структур данных в Pandas. Это, по сути, способ хранить табличные данные, где вы можете помечать строки и столбцы. Один из способов создания DataFrame - из словаря.

В упражнениях, которые следуют, вы будете работать с данными о транспортных средствах из разных стран. Каждое наблюдение соответствует стране, а столбцы содержат информацию о количестве транспортных средств на душу населения, о том, ездят ли люди слева или справа и так далее.

Три списка определены в скрипте:

- `names`, содержащий имена стран, для которых доступны данные.
- `dr`, список с логическими значениями, определяющими, едут ли люди слева или справа в соответствующей стране.

- `cpc`, количество автотранспортных средств на 1000 человек в соответствующей стране.

Каждый ключ словаря - это метка столбца, а каждое значение - список, содержащий элементы столбца.

Инструкции

Импортировать `pandas` как `pd`

Использовать предопределенные списки для создания словаря под названием `my_dict`. Должно быть три пары ключ-значение:

- ключ '`country`' и значение `names`.
- ключ `drives_right` и значение `dr`.
- ключ `cars_per_cap` и значение `cpc`.
- my_dict = {'country': names, 'drives_right': dr, 'cars_per_cap': cpc}

Использовать `pd.DataFrame()`, чтобы преобразовать ваш словарь в `DataFrame` под названием `cars`.

```
cars = pd.DataFrame(my_dict)
```

Напечатать `cars` и посмотреть, какой он красивый.

```
print(cars)

# Предопределенные списки
names = ['Соединенные Штаты', 'Австралия', 'Япония', 'Индия',
'Россия', 'Марокко', 'Египет']
dr = [True, False, False, False, True, True, True]
cpc = [809, 731, 588, 18, 200, 70, 45]

# Импорт pandas как pd
import pandas as pd

# Создать словарь my_dict с тремя парами ключ:значение: my_dict
my_dict = {'country': names, 'drives_right': dr, 'cars_per_cap': cpc}

# Создать DataFrame cars из my_dict: cars
cars = pd.DataFrame(my_dict)

# Напечатать cars
print(cars)
```

```
              country  drives_right  cars_per_cap
0  Соединенные Штаты          True           809
1          Австралия         False           731
2             Япония         False           588
3              Индия         False            18
```

```
4              Россия          True           200
5             Марокко          True            70
6              Египет          True            45
```

Словарь в DataFrame (2)

В скрипте уже представлен код на Python, который решает предыдущее упражнение. Заметили ли вы, что метки строк (т.е. метки для различных наблюдений) автоматически установлены как целые числа от 0 до 6?

Для решения этой проблемы был создан список `row_labels`. Вы можете использовать его для указания меток строк в DataFrame `cars`. Вы это делаете, устанавливая атрибут `index` для `cars`, к которому можно обратиться как `cars.index`.

Инструкции

- Укажите метки строк, установив `cars.index` равным `row_labels`.
- Снова выведите `cars` на печать и проверьте, правильно ли установлены метки строк.

```
# Предопределенные списки
names = ['Соединенные Штаты', 'Австралия', 'Япония', 'Индия',
'Россия', 'Марокко', 'Египет']
dr = [True, False, False, False, True, True, True]
cpc = [809, 731, 588, 18, 200, 70, 45]

# Импорт pandas как pd
import pandas as pd

# Создать словарь my_dict с тремя парами ключ:значение: my_dict
my_dict = {'country': names, 'drives_right': dr, 'cars_per_cap': cpc}

# Создать DataFrame cars из my_dict: cars
cars = pd.DataFrame(my_dict)

# Создать список меток строк
row_labels = ['US', 'AUS', 'JAP', 'IN', 'RU', 'MOR', 'EG']

# Установить метки строк для cars
cars.index = row_labels

# Напечатать cars
print(cars)
```

```
               country  drives_right  cars_per_cap
US   Соединенные Штаты          True           809
AUS          Австралия         False           731
JAP             Япония         False           588
IN               Индия         False            18
RU              Россия          True           200
```

```
MOR            Марокко          True            70
EG              Египет          True            45
```

CSV в DataFrame

Перенос данных в словарь, а затем построение DataFrame работает, но это не очень эффективно. Что если у вас миллионы наблюдений? В таких случаях данные обычно доступны в файлах с регулярной структурой. Один из таких типов файлов - это CSV-файл, что означает "значения, разделенные запятыми".

Чтобы импортировать CSV-данные в Python как DataFrame Pandas, можно использовать `read_csv()`.

Давайте изучим эту функцию с теми же данными об автомобилях из предыдущих упражнений. На этот раз, однако, данные доступны в CSV-файле под названием `cars.csv`. Он доступен в вашей текущей рабочей директории, поэтому путь к файлу просто `cars.csv`.

Инструкции

- Для импорта файлов CSV вам все еще нужен пакет `pandas`: импортируйте его как `pd`.
- Используйте `pd.read_csv()`, чтобы импортировать данные из `cars.csv` в виде DataFrame. Сохраните этот DataFrame как `cars`.
- Выведите `cars` на экран. Все выглядит нормально?

```
# Импортировать pandas как pd
import pandas as pd

# Импортировать данные из cars.csv: cars
cars = pd.DataFrame(pd.read_csv('datasets/cars.csv'))
# Вывести cars
print(cars)
```

```
  Unnamed: 0  cars_per_cap        country  drives_right
0         US           809  United States          True
1        AUS           731      Australia         False
2        JAP           588          Japan         False
3         IN            18          India         False
4         RU           200         Russia          True
5        MOR            70        Morocco          True
6         EG            45          Egypt          True
```

CSV в DataFrame (2)

Ваш вызов `read_csv()` для импорта данных CSV не вызвал ошибку, но вывод не совсем соответствует нашим ожиданиям. Метки строк были импортированы как еще один столбец без имени.

Помните о `index_col`, аргументе `read_csv()`, который вы можете использовать для указания столбца в файле CSV, который следует использовать в качестве меток строк? Именно это вам и нужно здесь!

Код на Python, который решает предыдущее упражнение, уже включен; можете ли вы внести соответствующие изменения, чтобы исправить импорт данных?

Инструкции

- Запустите код с помощью кнопки "Запустить код" и убедитесь, что первый столбец должен фактически использоваться в качестве меток строк.
- Укажите аргумент `index_col` внутри `pd.read_csv()`: установите его на 0, чтобы первый столбец был использован в качестве меток строк.
- Улучшился ли вывод `cars` сейчас?

```
# Импорт pandas как pd
import pandas as pd

# Исправление импорта с помощью включения index_col
cars = pd.read_csv('datasets/cars.csv', index_col=0)

# Вывод cars
print(cars)
```

```
     cars_per_cap        country  drives_right
US            809  United States          True
AUS           731      Australia         False
JAP           588          Japan         False
IN             18          India         False
RU            200         Russia          True
MOR            70        Morocco          True
EG             45          Egypt          True
```

Квадратные скобки (1)

В видео вы видели, что вы можете индексировать и выбирать данные в Pandas DataFrame разными способами. Самый простой, но не самый мощный способ - использовать квадратные скобки.

В примере кода те же данные об автомобилях импортируются из файлов CSV в виде Pandas DataFrame. Чтобы выбрать только столбец `cars_per_cap` из `cars`, вы можете использовать:

```
cars['cars_per_cap']
cars[['cars_per_cap']]
```

Одиночная скобка создает объект Pandas Series, двойная скобка создает объект Pandas DataFrame.

Инструкции

- Используйте одинарные квадратные скобки для вывода столбца `country` из `cars` как объекта Pandas Series.
- Используйте двойные квадратные скобки для вывода столбца `country` из `cars` как объекта Pandas DataFrame.
- Используйте двойные квадратные скобки для вывода DataFrame с колонками `country` и `drives_right` из `cars`, в данном порядке.

```
# Вывод столбца 'country' в виде объекта Pandas Series
print(cars['country'])

# Вывод столбца 'country' в виде объекта Pandas DataFrame
print(cars[['country']])

# Вывод DataFrame с столбцами 'country' и 'drives_right'
print(cars[['country','drives_right']])
```

```
US     United States
AUS        Australia
JAP            Japan
IN             India
RU            Russia
MOR          Morocco
EG             Egypt
Name: country, dtype: object
          country
US   United States
AUS      Australia
JAP          Japan
IN           India
RU          Russia
MOR        Morocco
EG           Egypt
          country  drives_right
US   United States          True
AUS      Australia        False
JAP          Japan        False
IN           India        False
RU          Russia         True
MOR        Morocco         True
EG           Egypt          True
```

Квадратные скобки (2)

Квадратные скобки могут делать не только выбор столбцов, но и получать строки или наблюдения из DataFrame. Следующий вызов выбирает первые пять строк из DataFrame cars:

```
cars[0:5]
```

Результатом будет еще один DataFrame, содержащий только указанные вами строки.

Обратите внимание: вы можете выбирать строки с помощью квадратных скобок, только если указываете срез, например, 0:4. Кроме того, вы используете целочисленные индексы строк, а не метки строк!

Инструкции

- Выберите первые 3 наблюдения из `cars` и выведите их.
- Выберите четвертое, пятое и шестое наблюдения, соответствующие индексам строк 3, 4 и 5, и выведите их.

```
# Импорт данных по автомобилям
cars = pd.read_csv('datasets/cars.csv', index_col=0)

# Вывод первых 3 наблюдений
print(cars[0:3])

# Вывод четвертого, пятого и шестого наблюдений
print(cars[3:6])
```

```
     cars_per_cap        country  drives_right
US            809  United States          True
AUS           731      Australia         False
JAP           588          Japan         False
     cars_per_cap  country  drives_right
IN             18    India         False
RU            200   Russia          True
MOR            70  Morocco          True
```

loc и iloc (1)

С помощью `loc` и `iloc` можно выполнять практически любую операцию выбора данных в DataFrame, о которой вы можете подумать. loc основан на метках, что означает, что вы должны указать строки и столбцы на основе их меток строк и столбцов. `iloc` основан на целочисленных индексах, поэтому вам нужно указывать строки и столбцы по их целочисленным индексам, как вы делали в предыдущем упражнении.

Попробуйте выполнить следующие команды в IPython Shell, чтобы экспериментировать с `loc` и `iloc` для выбора наблюдений. Каждая пара команд здесь дает одинаковый результат.

```
cars.loc['RU']
cars.iloc[4]

cars.loc[['RU']]
cars.iloc[[4]]

cars.loc[['RU', 'AUS']]
cars.iloc[[4, 1]]
```

Как и раньше, включен код, который импортирует данные об автомобилях как DataFrame в Pandas.

Инструкции

- Используйте `loc` или `iloc`, чтобы выбрать наблюдение, соответствующее Японии, в виде Series. Метка этой строки - `JAP`, индекс - 2. Обязательно выведите полученный Series.
- Используйте `loc` или `iloc`, чтобы выбрать наблюдения для Австралии и Египта в виде DataFrame. Вы можете узнать метки/индексы этих строк, осмотрев `cars` в IPython Shell. Обязательно выведите полученный DataFrame.

```
# Импорт данных по автомобилям
cars = pd.read_csv('datasets/cars.csv', index_col=0)

# Вывод наблюдения для Японии
print(cars.loc["JAP"])

# Вывод наблюдений для Австралии и Египта
print(cars.iloc[[1, -1]])
```

```
cars_per_cap      588
country         Japan
drives_right    False
Name: JAP, dtype: object
     cars_per_cap    country  drives_right
AUS           731  Australia         False
EG             45      Egypt          True
```

loc и iloc (2)

`loc` и `iloc` также позволяют выбирать как строки, так и столбцы из DataFrame. Чтобы провести эксперимент, попробуйте следующие команды в IPython Shell. Опять же, парные команды дают одинаковый результат.

```
cars.loc['IN', 'cars_per_cap']
cars.iloc[3, 0]

cars.loc[['IN', 'RU'], 'cars_per_cap']
cars.iloc[[3, 4], 0]

cars.loc[['IN', 'RU'], ['cars_per_cap', 'country']]
cars.iloc[[3, 4], [0, 1]]
```

Инструкции

- Выведите значение `drives_right` для строки, соответствующей `Марокко` (метка строки - `MOR`).
- Выведите под-DataFrame, содержащий наблюдения для России и Марокко, а также столбцы `country` и `drives_right`.

```
# Вывести значение drives_right для Марокко
print(cars.loc['MOR', 'drives_right'])

# Вывести под-DataFrame
print(cars.loc[['RU','MOR'],['country','drives_right']])
```

```
True
     country  drives_right
RU    Russia          True
MOR  Morocco          True
```

loc и iloc (3)

Возможно также выбирать только столбцы с помощью `loc` и `iloc`. В обоих случаях достаточно указать срез от начала до конца перед запятой:

```
cars.loc[:, 'country']
cars.iloc[:, 1]

cars.loc[:, ['country','drives_right']]
cars.iloc[:, [1, 2]]
```

Инструкции

- Выведите столбец `drives_right` в виде Series, используя `loc` или `iloc`.
- Выведите столбец `drives_right` в виде DataFrame, используя `loc` или `iloc`.
- Выведите как столбец `cars_per_cap`, так и driv`es_right в виде DataFrame, используя`locили`iloc`.

```
# Вывести столбец drives_right как Series
print(cars.loc[:, 'drives_right'])

# Вывести столбец drives_right как DataFrame
print(cars.loc[:, ['drives_right']])

# Вывести cars_per_cap и drives_right как DataFrame
print(cars.loc[:, ['cars_per_cap', 'drives_right']])
```

```
US      True
AUS    False
JAP    False
IN     False
RU      True
MOR     True
EG      True
Name: drives_right, dtype: bool
     drives_right
US           True
AUS         False
JAP         False
```

```
IN          False
RU           True
MOR          True
EG           True
     cars_per_cap  drives_right
US            809          True
AUS           731         False
JAP           588         False
IN             18         False
RU            200          True
MOR            70          True
EG             45          True
```