Введение в Pandas

Берленко Татьяна Андреевна, МОЭВМ, 2020

Pandas. DataFrame

- Создание
  - с помощью numpy/списков Python

```
>> pd.DataFrame(np.arange(1, 2, 0.2))
    0 # имя столбца
0  1.0
1  1.2
2  1.4
3  1.6
4  1.8
>> pd.DataFrame(np.array([['a', 'b'], [10, 20]]))
   0   1
0  a   b
1  10  20
>> pd.DataFrame([['a', 'b'], [10, 20]])
   0   1
0  a   b
1  10  20
```

  - с помощью Series

```
>> pd.DataFrame([ pd.Series(['a', 'b', 'c']), pd.Series([10, 20])])
   0   1   2
0  a   b   c
1  10  20  NaN
# NaN появился, поскольку во второй Series меньше значений, произошло выравнивание
```

  - с помощью словарей

```
>>   pd.DataFrame({'symbols': pd.Series(['a', 'b', 'c']),
'numbers': [10, 20, 30]})
  symbols  numbers
0      a       10
1      b       20
2      c       30
# ключ словаря становится именем столбца
```

Pandas. DataFrame

- Характеристики DF
  - df_test = pd.DataFrame({'symbols': pd.Series(['a', 'b', 'c']), 'numbers': [10, 20, 30]})

	symbols	numbers
0	a	10
1	b	20
2	c	30

  - len(df_test)

    3 # количество строк
  - df_test.size

    6 # количество элементов
  - df_test.shape

    (3, 2) # кортеж: количество строк, количество столбцов
  - >>> df_test.columns

    Index(['symbols', 'numbers'], dtype='object')
  - >>> df_test.index

    RangeIndex(start=0, stop=3, step=1) # индекс в примере задан значениями по умолчанию

Pandas. DataFrame

- Добавление столбцов
  - df_test['new'] = [100, 200, 300]

```
  Символы  Числа  new
0       a     10     100
1       b     20     200
2       c     30     300
```

- Отбор столбцов
  - df_test['Символы']

```
0   a
1   b
2   c
Name: Символы, dtype: object
# тип - Series
```

  - df_test[['Символы', 'Числа']] # могут быть перечислены любые корректные столбцы

```
  Символы  Числа
0       a     10
1       b     20
2       c     30
# тип - DF
```

Pandas. DataFrame

- Добавление строк
  - df_test.loc[4] = ['d', 40, 400]

	Символы	Числа	new
0	a	10	100
1	b	20	200
2	c	30	300
4	d	40	400

- Отбор строк
  - df_test.loc[4]

```
Символы     d
Числа      40
new       400
Name: 4, dtype: object
# тип - Series
```

  - df_test.loc[[1, 2, 0]] # могут быть перечислены любые корректные порядковые номера строк

	Символы	Числа	new
1	b	20	200
2	c	30	300
0	a	10	100

тип - DF

Pandas. DataFrame

- Срезы
  - df_test.loc[1:, :'new'] # сначала указываем метку, потом столбец, могут быть как значения, так и срезы

	Символы	Числа
1	b	20
2	c	30
4	d	40

- Логический отбор строк
  - df_test[df_test['Символы'] > 'a']

	Символы	Числа	new
1	b	20	200
2	c	30	300
4	d	40	400

- Удаление столбцов
  - del df_test['new'], df_test.pop('new'), df_test.drop('new', axis=1)
- Удаление строк
  - df_test.drop(1, axis=0) # по умолчанию axis == 0

Pandas. DataFrame

- Индексы
  - Можно явно задать индекс при создании df

    pd.DataFrame([[1,2,3], [10, 20, 30])], columns=['t1', 't2', 't3'], **index**=[‘a’, ‘b’])
  - .set_index('имя_столбца') # Переносит столбец в индекс
  - .reset_index()

    # Сбрасывает индекс DF. Используется, когда нужно переместить содержимое индекса в столбец/столбцы.

Загрузка данных

Формат	Чтение данных	Сохранение данных
csv	pd.read_csv()	df.to_csv()
json	pd.read_json()	df.to_json()
excel	pd.read_excel()	df.to_excel()
sql	pd.read_sql()	df.to_sql()

Аргументы функции загрузки данных

- Указать индекс при чтении

  index_col = []
- Задать тип столбца

  dtype = {‘имя столбца’: тип}
- Имена столбцов

  header = <номер строки заголовка, обычно 0>

  names=[<список имен столбцов>]
- Загрузка некоторых столбцов

  usecols=[<список имен столбцов>]
- Разделитель

  sep = 'строка'
- Пропуск служебных строк

  skiprows=[0, 2, 3, 4] # какие из строк пропускаем

  skipfooter=10 # количество строк в футере таблицы, которое не собираемся читать. Нужно добавить engine=’python’

  nrows=10 # количество строк, которое хотим прочитать

Полезные материалы

- Документация https://pandas.pydata.org/docs/
- Онлайн-учебник https://coderlessons.com/tutorials/python-technologies/vyuchit-python-panda/uchebnik-po-python-pandas
- Онлайн-курс https://stepik.org/course/4852/syllabus
- Онлайн-курс https://www.coursera.org/learn/data-analysis-with-python#syllabus