PANDAS

Python Bootcamp #7

Max Shestov

Pandas

- **Pan**el **Da**ta **S**ystem
- Powerful and productive Python data analysis and management library
- Rich data structures and functions to make working with structured data fast, easy, and expressive
- flexible data manipulation capabilities of spreadsheets and relational databases
- Sophisticated indexing functionality
- Pandas introduced 2 data structures: Series and DataFrames

Series Data Structure

- One-dimensional array-like object

```
>>> s = pd.Series((1,2,3,4,5))
```

- Contains an array of data: strings, floats, integers, etc

```
>>> s.values
```

- Has an associated array of data labels, the index (Default index from 0 to N - 1)

```
>>> s.index
```

Series Data Structure

```
>>> import numpy as np
>>> randn = np.random.randn
>>> s = pd.Series(randn(3),('a','b','c’))
a    0.435930
b   -1.666787
c    1.872852
dtype: float64
>>> s.mean()
-0.34125097685130107
>>> s[0]
0.4359304267021378 3
```

Series to/from dict

- Series to Python dict - **No more explicit order**

```
>>> dict(s)
{'a': 0.4359304267021378 3,
'b': -1.6667869134072428,
'c': 1.8728516418594083}
```

- Back to a Series with a **new Index from sorted dictionary keys**

```
>>> Series(dict(s))
a    0.435930
b   -1.666787
c    1.872852
dtype: float64
```

Reindexing labels

```
>>> s
a    0.435930
b   -1.666787
c    1.872852
dtype: float64
>>> s.index
Index(['a', 'b', 'c'], dtype='object’)
>>> s.reindex(['c','b','a’])
c    1.872852
b   -1.666787
a    0.435930
dtype: float64
```

Vectorization

```
>>> s + s
a    0.871861
b   -3.333574
c    3.745703
dtype: float64
```

- Series work with Numpy package

```
>>> numpy.mean(s)
0.21399838505143443
```

DataFrame

- 2-dimensional table like data structure
- Data manipulation with integrated indexing
- Support heterogeneous columns
- Homogeneous columns

DataFrame

```
>>> d = {'ONE': s*s, 'TWO': s+s}
>>> df = pd.DataFrame(d)
>>> df
```

	ONE	TWO
a	0.190035	0.871861
b	2.778179	-3.333574
c	3.507573	3.745703

```
>> df.index
Index(['a', 'b', 'c'], dtype='object’)
>> df.columns
Index(['ONE', 'TWO'], dtype='object')
```

Dataframe add column

- Add a third column

```
>>> df['THREE'] = s * 3
```

- It will share the existing index

```
>>> df
```

	ONE	TWO	THREE
a	0.190035	0.871861	1.307791
b	2.778179	-3.333574	-5.000361
c	3.507573	3.745703	5.618555

Access to columns

- Access by attribute – can be converted to list via list(df.ONE) function

```
>>> df.ONE
a    0.190035
b    2.778179
c    3.507573
Name: ONE, dtype: float64
```

- Access by dictionary like notation - can be converted to list via list(df[‘ONE’]) function

```
>>> df[‘ONE’]
a    0.190035
b    2.778179
c    3.507573
Name: ONE, dtype: float64
```

Reindexing

```
>>> df.reindex(['c','b','a’])
>>> df
```

	ONE	TWO	THREE
c	3.507573	3.745703	5.618555
b	2.778179	-3.333574	-5.000361
a	0.190035	0.871861	1.307791

Drop entries from an axis

>>> df.drop(‘c’)

	ONE	TWO	THREE
a	0.190035	0.871861	1.307791
b	2.778179	-3.333574	-5.000361

>>> df.drop(['b','a’])

	ONE	TWO	THREE
c	3.507573	3.745703	5.618555

DataFrame column sorting

```
>>> df
```

	ONE	TWO	THREE
a	0.190035	0.871861	1.307791
b	2.778179	-3.333574	-5.000361
c	3.507573	3.745703	5.618555

```
>>> df.sort_values(['TWO'],axis=0)
```

	ONE	TWO	THREE
b	2.778179	-3.333574	-5.000361
a	0.190035	0.871861	1.307791
c	3.507573	3.745703	5.618555

Descriptive statistics

```
>>> df.mean()
ONE      2.158596
TWO      0.427997
THREE    0.641995
dtype: float64
```

- Also: count, sum, median, min, max, abs, prod, std, var, skew, kurt, quantile, cumsum, cumprod, cummax, cummin

DataFrame Joins

	student_id	first_name	last_name
0	111	Alex	Anderson
1	222	Amy	Brown
2	333	Allen	Green
3	444	Alice	White
4	555	Anna	Smith

	sid	course_name
0	111	Calculus
1	111	Chemistry
2	111	Biology
3	333	Physics
4	333	Statistics
5	444	Piano
6	666	Guitar

```
pd.merge(df_a, df_b, left_on='student_id', right_on='sid', how='outer')
```

DataFrame Joins

	student_id	first_name	last_name	sid	course_name
0	111.0	Alex	Anderson	111.0	Calculus
1	111.0	Alex	Anderson	111.0	Chemistry
2	111.0	Alex	Anderson	111.0	Biology
3	222.0	Amy	Brown	NaN	NaN
4	333.0	Allen	Green	333.0	Physics
5	333.0	Allen	Green	333.0	Statistics
6	444.0	Alice	White	444.0	Piano
7	555.0	Anna	Smith	NaN	NaN
8	NaN	NaN	NaN	666.0	Guitar

Thank you!