Toolbox

Bhaswar Chakma

2021-07-08

Contents

Chapter 1

Python

1.1 Pandas

NumPy creates ndarrays that must contain values that are of the same data type. Pandas creates dataframes. Each column in a dataframe is an ndarray. This allows us to have traditional tables of data where each column can be a different data type.

Important References:

- Series: https://pandas.pydata.org/pandas-docs/stable/reference/series.html
- DataFrame: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

```
import numpy as np
import pandas as pd
```

1.1.1 Series

The basic data structure in pandas is the series. You can construct it in a similar fashion to making a numpy array. The command to make a Series object is `pd.Series(data, index=index)`. Note that the `index` argument is optional.

```
data = pd.Series([0.25, 0.5, 0.75, 1.0])
print(data)
```

```
## 0    0.25
## 1    0.50
## 2    0.75
## 3    1.00
## dtype: float64
```

```
print(type(data)) # data type
```

```
## <class 'pandas.core.series.Series'>
```

```
print(data.values) # data values
```

```
## [0.25 0.5  0.75 1.  ]
```

```
print(type(data.values)) # The values attribute of the series is a numpy array.
```

```
## <class 'numpy.ndarray'>
```

```
print(data.index)
```

```
## RangeIndex(start=0, stop=4, step=1)
```

```
print(type(data.index)) # the row names are known as the index
```

```
## <class 'pandas.core.indexes.range.RangeIndex'>
```

You can subset a pandas series like other python objects

```
print(data) # example data
```

```
## 0    0.25
## 1    0.50
## 2    0.75
## 3    1.00
## dtype: float64
```

```
print(data[1]) # select the 2nd value
```

```
## 0.5
```

```
print(type(data[1])) # when you select only one value, it simplifies the object
```

```
## <class 'numpy.float64'>
```

```
print(data[1:3])
```

```
## 1    0.50
## 2    0.75
## dtype: float64
```

```
print(type(data[1:3])) # slicing / selecting multiple values returns a series
```

```
## <class 'pandas.core.series.Series'>
```

You can also do fancy indexing by subsetting w/a numpy array e.g. repeat observations.

```
print(data[np.array([1, 0, 1, 2])])
```

```
## 1    0.50
## 0    0.25
## 1    0.50
## 2    0.75
## dtype: float64
```

Pandas uses a 0-based index by default. You may also specify the index values.

```
data = pd.Series([0.25, 0.5, 0.75, 1.0],
index = ['a', 'b', 'c', 'd'])
print(data)
```

```
## a    0.25
## b    0.50
## c    0.75
## d    1.00
## dtype: float64
```

```
data.values
```

```
## array([0.25, 0.5 , 0.75, 1.  ])
```

```
data.index
```

```
## Index(['a', 'b', 'c', 'd'], dtype='object')
```

- subset with index position

```
data[1]
```

```
## 0.5
```

- subset with index name

```
data["a"]
```

```
## 0.25
```

Chapter 2

R

Chapter 3

SQL

Coming!