Data Science

Intro to pandas

CS202 09.21.22

Pandas

What is Pandas?

- Pandas is a Python library used for working with data sets.
- It has functions for analyzing, cleaning, exploring, and manipulating data.
- The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
- Pandas allows us to analyze big data and make conclusions based on statistical theories.
- Pandas can clean messy data sets, and make them readable and relevant.

Pandas

Getting started

Just the way we installed numpy we would need to install the pandas library. We do this by typing “pip install pandas” into our terminal or we can go to our jupyter notebook and type “pip! install pandas”.

Once installation is done we will need to import it before using:

```
import pandas
```

Just like numpy we can also write for short:

```
import pandas as pd
```

Pandas

Series

A Pandas Series is like a column in a table.

It is a one-dimensional array holding data of any type.

```
import pandas as pd

a = [1, 7, 2]

myvar = pd.Series(a)

print(myvar)
```

If nothing else is specified, the values are labeled with their index number. First value has index 0, second value has index 1 etc.

This label can be used to access a specified value.

```
print(myvar[0])
```

Pandas

Series

With the `index` argument, you can name your own labels.

```
import pandas as pd

a = [1, 7, 2]

myvar = pd.Series(a, index = ["x", "y", "z"])

print(myvar)
```

When you have created labels, you can access an item by referring to the label.

```
print(myvar["y"])
```

Pandas

Series

You can also use a key/value object, like a dictionary, when creating a Series.

```
import pandas as pd

calories = {"day1": 420, "day2": 380, "day3": 390}

myvar = pd.Series(calories)

print(myvar)
```

To select only some of the items in the dictionary, use the `index` argument and specify only the items you want to include in the Series.

```
import pandas as pd

calories = {"day1": 420, "day2": 380, "day3": 390}

myvar = pd.Series(calories, index = ["day1", "day2"])

print(myvar)
```

Pandas

Data Frames

We will be mostly using what are called data frames.

Data sets in Pandas are usually multi-dimensional tables, called DataFrames.

Series is like a column, a DataFrame is the whole table.

```
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

myvar = pd.DataFrame(data)

print(myvar)
```

Pandas

Data Frames

A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

```
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

Pandas

Data Frames

Locating items:

As you can see from the result above, the DataFrame is like a table with rows and columns.

Pandas use the loc attribute to return one or more specified row(s)

Return row 0:

```
#refer to the row index:
print(df.loc[0])
```

Return row 0 and 1:

```
#use a list of indexes:
print(df.loc[[0, 1]])
```

Pandas

Data Frames

With the index argument, you can name your own indexes.

```
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

df = pd.DataFrame(data, index = ["day1", "day2", "day3"])

print(df)
```

Use the named index in the loc attribute to return the specified row(s).

```
#refer to the named index:
print(df.loc[“day2"])
```

If your data sets are stored in a file, Pandas can load them into a DataFrame.

```
import pandas as pd

df = pd.read_csv('data.csv')

print(df)
```