Introduction to pandas: Takeaways

Syntax

PANDAS DATAFRAME BASICS

- Reading a file into a dataframe:

```
f500 = pd.read_csv('f500.csv',index_col=0)
```

- Returning a dataframe's data types:

```
col_types = f500.dtypes
```

- Returning the dimensions of a dataframe:

```
dims = f500.shape
```

SELECTING VALUES FROM A DATAFRAME

- Selecting a single column:

```
f500["rank"]
```

- Selecting multiple columns:

```
f500[["country", "rank"]]
```

- Selecting the first n rows:

```
first_five = f500.head(5)
```

- Selecting rows from a dataframe by label:

```
drink_companies = f500.loc[["Anheuser-Busch InBev", "Coca-Cola", "Heineken Holding"]]

big_movers = f500.loc[["Aviva", "HP", "JD.com", "BHP Billiton"],

["rank","previous_rank"]]

middle_companies = f500.loc["Tata Motors":"Nationwide", "rank":"country"]
```

Concepts

- NumPy provides fundamental structures and tools that make working with data easier, but there are several things that limit its usefulness as a single tool when working with data:
    - The lack of support for column names forces us to frame the questions we want to answer as multi-dimensional array operations.
    - Support for only one data type per ndarray makes it more difficult to work with data that contains both numeric and string data.

- The **pandas** library provides solutions to all of these pain points and more. Pandas is not so much a replacement for NumPy as an *extension* of NumPy. The underlying code for pandas uses the NumPy library extensively. The main objects in pandas are **Series** and **Dataframes**. Series is equivalent to a 1D Ndarray while a dataframe is equivalent to a 2D Ndarray.
- Different label selection methods:

Select by Label	**Explicit Syntax**	**Shorthand Convention**
Single column from dataframe	df.loc[:,"col1"]	df["col1"]
List of columns from dataframe	df.loc[:,["col1","col7"]]	df[["col1","col7"]]
Slice of columns from dataframe	df.loc[:,"col1":"col4"]	
Single row from dataframe	df.loc["row4"]	
List of rows from dataframe	df.loc[["row1", "row8"]]	
Slice of rows from dataframe	df.loc["row3":"row5"]	df["row3":"row5"]
Single item from series	s.loc["item8"]	s["item8"]
List of items from series	s.loc[["item1","item7"]]	s[["item1","item7"]]
Slice of items from series	s.loc["item2":"item4"]	s["item2":"item4"]

There are lots of low level methods — however, there are many common analysis patterns that don't have pre-built methods.

Resources

- Dataframe.loc[]

- Indexing and Selecting Data