Pandas

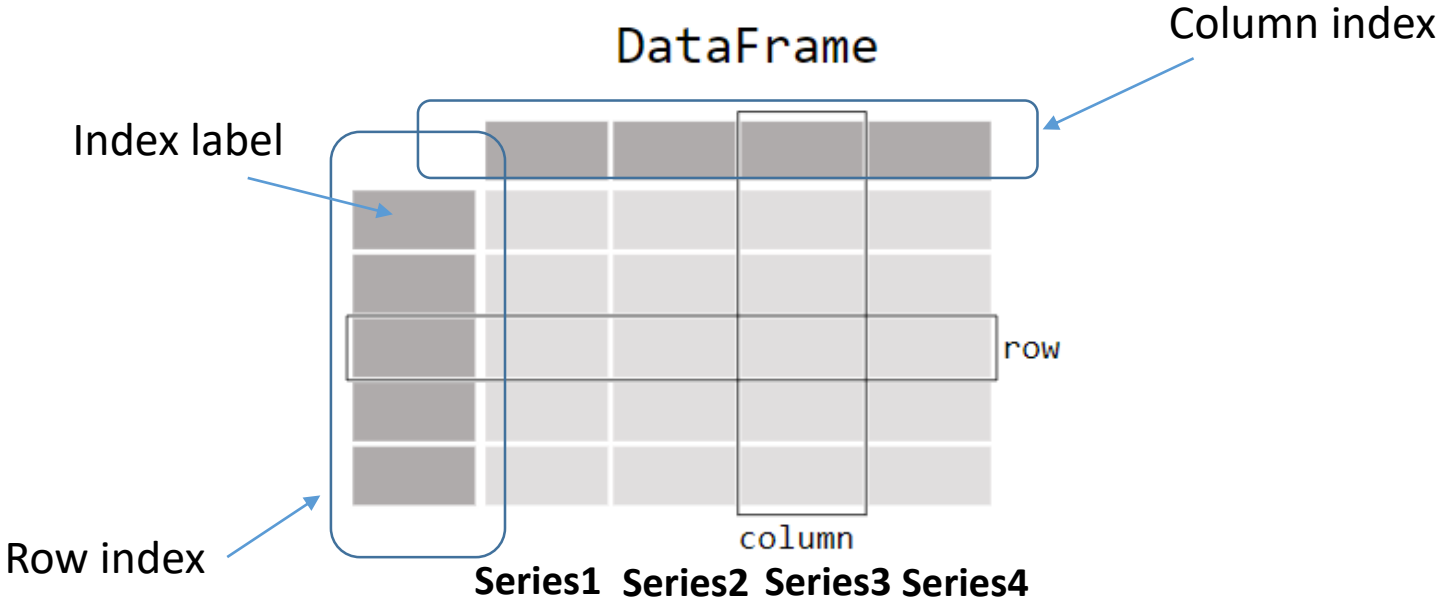
In this first lecture: Series, DataFrame, Different functions

Pandas

- Pandas is for "Python and data analysis" and "panel data".
- Pandas is a Python package used for working with data sets and analyze them.
- Pandas is built on top of Numpy, that is Numpy is required by pandas.

Pandas

- Pandas provides special data structures called Series and DataFrames, and functions for analyzing, cleaning, processing data in them.
- *Series:* a one-dimensional labeled array that is capable of holding data of any type (integer, string, float, etc).
- *DataFrame:* a two-dimensional labeled data structure like a table with rows and columns.



Series

- Pandas Series is a one-dimensional array (like a column in a table).
- We can think of it as a data structure with two arrays: one array contains the index (labels), and the other array contains the actual values.

```
import pandas as pd
a = [5, 4, 3]
ser = pd.Series(a)
print(ser)
```

- Labels: if labels are not specified explicitly, then the values are labeled with their index number. That is, the first value will have label 0, second value will have label 1, etc.

```
print(ser.index)        # prints the range of the row index
print(ser.values)       # prints all the rows
```

Series

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

```
import pandas as pd
a = [5, 4, 3]
ser = pd.Series(a, index = [“a”, “b”, “c”])  # index = [25, 35, 45]
print(ser)
```

- Then we can access an item by referring to the label.

```
print(ser[“a”])
```

Series

- We can also use a dictionary to make a Series.

```
import pandas as pd
person = {"Name": "Turan", "Age": 30, "City": "Astana"}
ser = pd.Series(person)
print(ser)
```

- The keys of the dictionary become the labels. We can include only some of the items using the *index* argument.

```
import pandas as pd
person = {"Name": "Turan", "Age": 30, "City": "Astana"}
ser = pd.Series(person, index = ["Name", "Age"])
print(ser)
```

Accessing data from Series

```
import pandas as pd
ser = pd.Series([1,2,3,4],index=[‘a’, ‘b’, ‘c’, ‘d’])
print(ser[0])    #print using position
print(ser[:3])
print(ser[2:4])
print(ser[“c”])  #print using label
```

DataFrames

- A DataFrame is a two-dimensional data structure, like a two-dimensional array, or a table with rows and columns.
- Series is like a column, a DataFrame is the whole table.

```
import pandas as pd
person = {
            "Name": ["Turan“,”Block”,”Seven”],
              "Age": [30,25, 20],
              "City": ["Astana“, ”Bishkek”, ”Tashkent”]
          }
df = pd.DataFrame(person)
print(df)
```

DataFrames

- We can also make a DataFrame from a list

```
import pandas as pd
data = [1,2,3,4,5]
df = pd.DataFrame(data)
print(df)


data = [[“Alice”,10],[“Bob”,12],[“Trudy”,13]]
df = pd.DataFrame(data, index=[“a”,”b”,”c”], columns=[“Name”,”Age”])
print (df)
```

DataFrame selection

- One of the attributes that is used to select one or more specified row(s) is the loc attribute.

```
print(df.loc[0])        # returns one row
print(df.loc[[0, 1]]) # returns two rows
print(df.loc[“a"])
```

DataFrame selection

Function	Description
DataFrame.head(n)	It is used to select top ‘n’ rows in DataFrame.
DataFrame.tail(n)	It is used to select bottom ‘n’ rows in DataFrame.
DataFrame.at	It is used to get and set the particular value of DataFrame using row and column labels.
DataFrame.iat	It is used to get and set the particular value of DataFrame using row and column index positions.
DataFrame.get(key)	It is used to get the value of a key in DataFrame where Key is the column name.
DataFrame.loc()	It is used to select a group of data based on the row and column labels. It is used for slicing and filtering of the DataFrame.
DataFrame.iloc()	It is used to select a group of data based on the row and column index position. Use it for slicing and filtering the DataFrame.

https://pynative.com/python-pandas-dataframe/#h-dataframe-options

Loading data from File into DataFrame

- Pandas provides different functions to read different types of files. Here are some of them:
  - read_csv() # to_csv() to write to a csv file
  - read_excel() # to_excel() to write to an excel file
  - read_json() …
  - read_html()
  - read_sql()
  - read_pickle()

```
import pandas as pd
df = pd.read_csv('data.csv')
print(df)
print(df.to_string())  # prints all rows in the dataframe; the above statement prints only a
                       #few from the top and a few from the bottom
```

https://www.w3schools.com/python/pandas/pandas_csv.asp

DataFrame functions

- DataFrame’s info() function gives metadata of DataFrame. Which includes, information such as number of rows, number of columns, data type of column, count of the total number of non-null values in the column, memory usage by the DataFrame, etc.

- DataFrame’s describe() function gives mathematical statistics of the data in DataFrame. It applies only to the columns with numeric values.

DataFrame functions

- DataFrame’s head(n) function shows first n rows of DataFrame
- DataFrame’s tail(n) function shows last n rows of DataFrame
- By default, n = 5.

```
import pandas as pd
df = pd.read_csv('data.csv')
print(df.head(3))   # prints the first 3 rows in of the dataframe
print(df.tail(8))      #  prints the last 8 rows in of the dataframe
```