

Introduction to Pandas

September 22, 2022

1 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

2 Why Use Pandas?

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.

Relevant data is very important in data science.

3 Installation of Pandas

```
[32]: # !pip install pandas # To install Pandas library
```

4 How to import Python Pandas

```
[33]: import pandas
      #or
      import pandas as pd
```

```
Input In [33]
  or
  ^
IndentationError: unexpected indent
```

5 Pandas Version

```
[ ]: print(pd.__version__)
```

```
[ ]:
```

6 Data Set

attachment:image-2.png

7

```
[ ]: DataSet = {
    'cars': ["BMW", "Volvo", "Ford"],
    'passings': [3, 7, 2]
}

# myvar = pandas.DataFrame(DataSet)
myvar = pd.DataFrame(DataSet)

print(myvar)
```

8 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 = [10, 20, 30]

myvar = pd.Series(a)

print(myvar)
```

9 Labels

This label can be used to access a specified value.

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

10 Create Labels

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

```
[ ]: a = [10, 20, 30]      # List of all the Elements

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

print(myvar)
```

```
[ ]: print(myvar['x'])    # Can Access by labels
```

11 Key/Value Objects as Series

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

```
[ ]: calories = {"day1": 420, "day2": 380, "day3": 390}    # Dictionary

myvar = pd.Series(calories)

print(myvar)
```

12 The keys of the dictionary become the labels.

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.

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

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

print(myvar)
```

13 DataFrames

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]
}

df = pd.DataFrame(data)
#myvar = pd.DataFrame(data, index=["Row-1", "Row-2", "Row-3"])    # With User-Defined Labels

print(df)
```

14 Locate Row

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)

Note: When using `[]`, the result is a Pandas DataFrame.

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

      #print("\n")      # Print New Line
      #refer to the row index:
      #print(df.loc[[0,1,2]])
```

15 Named Indexes

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)

[ ]: #refer to the named index:
      print(df.loc["day3"])
```