CS506 Programming for Computing

HOS06C– Getting High-Performance with pandas

06/06/2020 Created by Apiwat Chuaphan

3/28/2022 Revised by Cedric Huang

03/08/2023 Reviewed by Maram Elwakeel

School of Technology & Computing (STC) @ City University of Seattle (CityU)

Before You Start

- Version numbers may not match with the most current version at the time of writing. If given the option to choose between stable release (long-term support) or most recent, please select the stable release rather than the beta-testing version.
- There might be subtle discrepancies along with the steps. Please use your best judgment while going through this cookbook-style tutorial to complete each step.
- For your working directory, use your course number. This tutorial may use a different course number as an example.
- All the steps and concepts in this tutorial are from the textbook, so if you encounter problems in this tutorial, please try to read and compare the textbook to solve the problem. If you still can't solve the problem, please feel free to contact your course TA.
- Avoid copy-pasting code from the book or the GitHub repository. Instead, type out the code yourself. Resort to copy-pasting only when you are stuck and find that things are not working as expected.

Learning Outcomes

- Understand pandas object
- Learn the basic functionality

Resources

- Tutorialpoint
- Pandas User Guide: https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html
- McIntire, G., Martin, B., & Washington, L. Pandas A Complete Introduction. Retrieved from https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/

Section 1 - What is pandas?

1) pandas (derived from the word Panel Data – an Econometrics from Multidimensional data – Tutorialspoint) is a powerful, open-source Python library providing high-performance, easy-to-use data structures for data analysis, manipulation, and visualization.

2) Features of Pandas

   - Fast and efficient DataFrame object
   - Tools for loading data into in-memory data objects from different file formats.
   - Label-based slicing, indexing and subsetting of large data sets.
   - Columns from a data structure can be deleted or inserted.
   - Group by data for aggregation and transformations.
   - High performance merging and joining of data.

3) Install pandas.

   i. In Visual Studio Code, open the private repository generated when you accepted the HOS06 assignment (If you cannot find that repository in your machine, you might have not cloned the repo, if so, please do before proceeding).

   ii. Open terminal (Control + `) in VS Code, then execute the command:

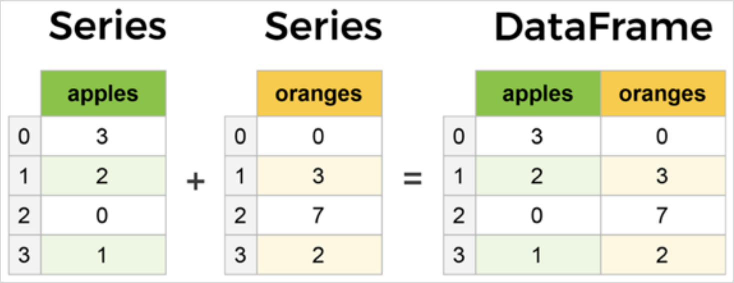
   iii.
```
pip install pandas
```

4) **Core Components -** The two primary data structures of pandas, Series and DataFrame.

   i. **Series** – 1D labeled homogeneously-typed array

   ii. **DataFrame** – General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed column

5) A Series is essentially a column, and a DataFrame is a multi-dimensional table made up of a collection of Series.



Series

	apples
0	3
1	2
2	0
3	1

+

Series

	oranges
0	0
1	3
2	7
3	2

=

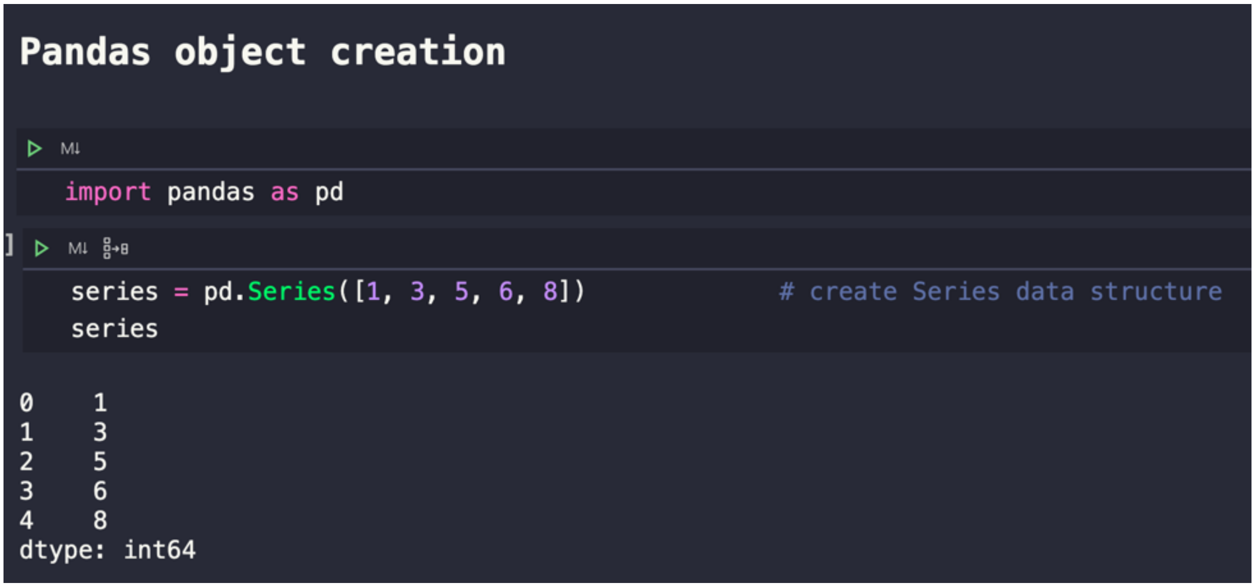
DataFrame

	apples	oranges
0	3	0
1	2	3
2	0	7
3	1	2

6) Open Jupyter Notebook:

   i. Under module folder, create a new file called `pandas_object.ipynb` and simply click on the file to open notebook.

   ii. Type the following into the file just created. Run selected cell to see each result.



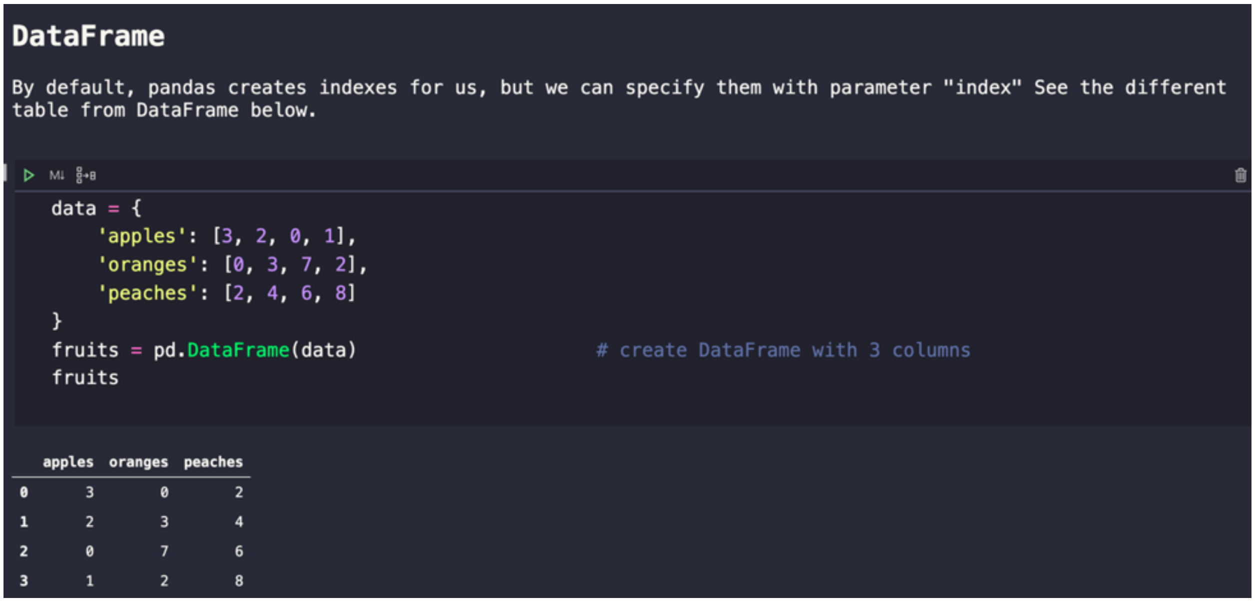
Pandas object creation

```
import pandas as pd
```

```
series = pd.Series([1, 3, 5, 6, 8])          # create Series data structure
series
```

```
0    1
1    3
2    5
3    6
4    8
dtype: int64
```

7) We just created Series object with pandas, next we will create DataFrame, which is a collection of Series.



DataFrame

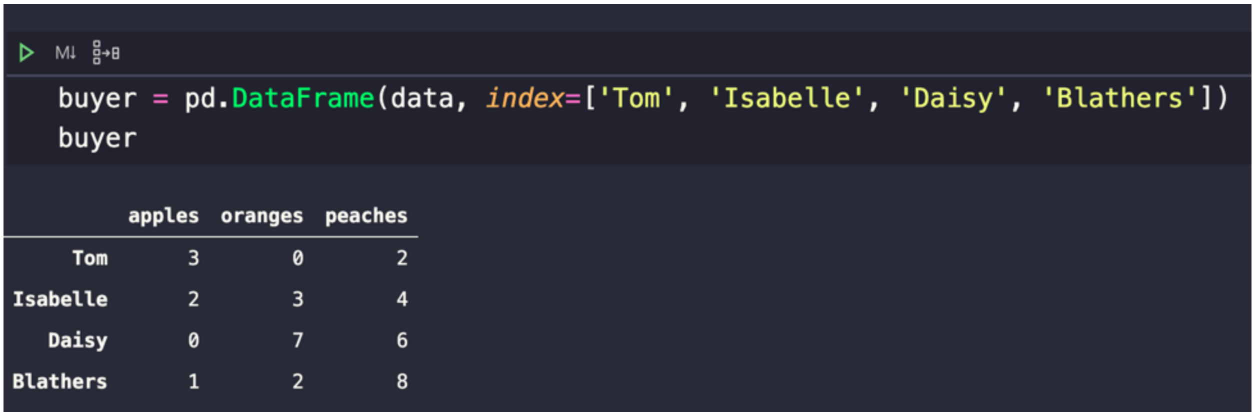
By default, pandas creates indexes for us, but we can specify them with parameter "index" See the different table from DataFrame below.

```
data = {
    'apples': [3, 2, 0, 1],
    'oranges': [0, 3, 7, 2],
    'peaches': [2, 4, 6, 8]
}
fruits = pd.DataFrame(data)                    # create DataFrame with 3 columns
fruits
```

```
   apples  oranges  peaches
0       3        0        2
1       2        3        4
2       0        7        6
3       1        2        8
```

Each (key, value) item in data corresponds to a column in the resulting DataFrame.

8) The previous DataFrame, index was given at the creation by default, but we can create our own labels as the following example.



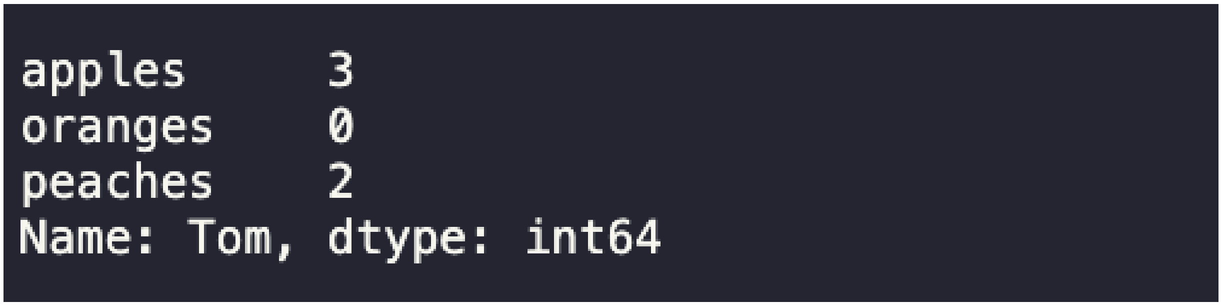
```
buyer = pd.DataFrame(data, index=['Tom', 'Isabelle', 'Daisy', 'Blathers'])
buyer
```

```
          apples  oranges  peaches
     Tom       3        0        2
Isabelle       2        3        4
   Daisy       0        7        6
Blathers       1        2        8
```

9) We gave string names as index replacement, so it is more convenient to locate the data by name. For example:

```
buyer.loc['Tom']
```

Output:



```
apples     3
oranges    0
peaches    2
Name: Tom, dtype: int64
```

10) That’s the basic pandas, there is a lot more from pandas that you can do. Learn more here:

    https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html

11) Save your Jupyter Notebook with all Output