Shing Chi Leung's Learning Pandas from Zero

Chapter 1: Introduction

In this chapter we will briefly introduce the background of the Python package Pandas. We will go through the essential features of Pandas, its pros and cons, and some pre-requisite knowledge before our exploration in Pandas.

What is Pandas?

Pandas is the abbreviation of "Python Data Analysis" or "Panel Data". The connection between the three terms and the term "Pandas" is actually remote that Pandas is more like a term for memorization of this package. Panel data is a terminology often used in economics and statistics. It corresponds to a set of data taken at different time for one or more quantities observed. Panel data is usually presented in the form of a table. Below in Table 1, we show a small panel data example where the temperature of two places at different time is presented.

Table 1: A sample panel data (units omitted)

time	temperature 1	temperature 2
1	28	36
2	29	34
3	32	30

Pandas is developed by Wes McKinney from 2007 - 2010 for the analysis of financial data. It is developed on NumPy and Cython for benefitting the efficient manipulation of array and contains functionality of creating, modify and structuring tables. It also
contains a number of statstical and graph plotting tools for providing the first diagnosis of the data.

In the literature table is often referred as a relational database, which, in contrast to a non-relational database, all data consists of a unique key, followed by a range of attributes. Therefore when we examine a relational database, it can be always cast
in the form of a table where the first column corresponds to the key (or the name of the entry) and other columns are the attributes. In the above small example of Table 1, the time can be regarded as the key while temperature 1 and 2 are the attributes.

When do we use Pandas?

As its name Pandas (panel data) suggested, Pandas is a package specific for handling table type data. In the ecosphere of Python packages (Figure 1 below), Pandas belongs to the middle part of the system. The package is not targeting for advanced numerical processing such as machine learning (e.g., scikit-learn) or network analysis (network X), nor is it a fundamental package such as NumPy which aims at handling basic data structure (array in this case). It serves as a data processing tool which assists us in handling a large dataset and processes the dataset for further analysis by other numerical packages.

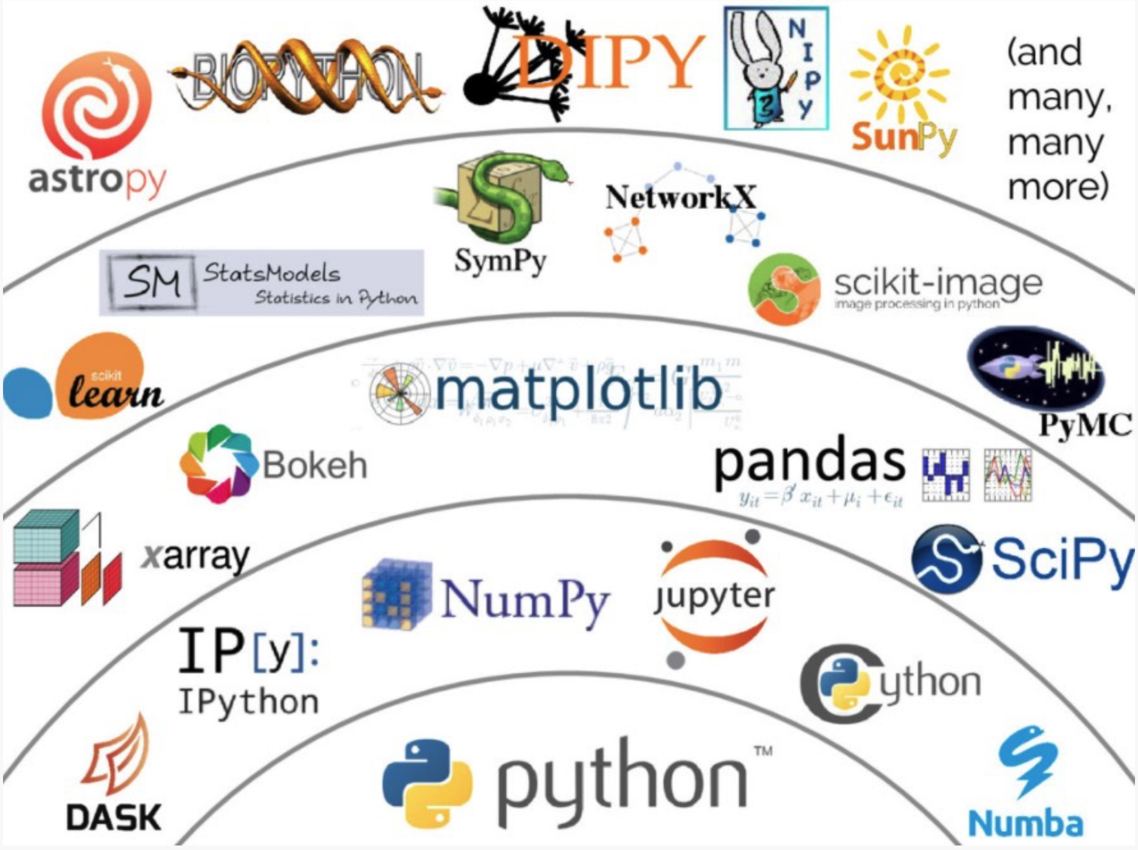

Figure 1: The illustration of the Python packages ecosphere, taken from here with reference to the presentation from VanderPlas

There are four scnearios where Pandas is highly specialized for its operation.

1. **Time series**

When we need to handle a set of time series data, which can be a collection of temperature of different places at a range of time, or a collection of the closing prices of many stocks at different days, Pandas contains a number of useful tools for rapidly digesting the data within a few lines of code.

2. **Table**

As described above, the major aim of Pandas is to process a dataframe. In Pandas, a table is referred as a dataframe. When we need to handle a large table such as the academic results of all students in a school, or the transactional record of a company within a year, the C-binding feature of pandas will largely help the user to process the data efficiently.

3. **Data analysis**

Pandas contains a number of statistical tools such as average, kurtosis, variance and so on which helps user to extract some preliminary statistical features in the data. It also has the backend of matplotlib where users can generate sophisticated visualization within a few lines of codes. Furthermore, it remains the flexibility of fine-tuning the figures for presentation in different scenarios.

4. **Data processing**

As a tool for panel data or relational database, Pandas allows users to make query about certain entries or a subset of the table for further processing. It also allows users to merge multiple tables or to split a table into multiple tables easily. On top of that, there are options such as feeding data through a set of pipeline procedure or generating pivot table from the dataframe.

Features of Pandas

- Efficiency

The C-binding of Pandas makes the iteration and process of a large number of data very fast. This feature is particularly notable compared to other standard spreadsheet software, when the table size reaches above 10000 lines or above. One may need to scroll up and down in order to select or filter necessary data because of the interface. On the other hand, the command-line approach in Python, and hence Pandas, makes it much faster to go through essential items in the table.

- Convenience

The integrated environment (data process, analysis and visualization) allows users to achieve a number of tasks by only using Pandas. The notation, which is similar to MySQL in data query, provides a direct approach to access, select and filter useful part from a large dataset.

- Simplicity

One of the main benefits of using Pandas is its easy-to-learn commands. Most commands are written
in a pythonic way, and therefore the codes can be presented in a human-readable way. This makes the
details of the data process very clear. This feature is particularly clear when we use workbook-like platform such as Jupyter notebook, where we can examine the intermediate results so that we can
present and design a clean procedure for others to understand the logic of the data process.

Other packages

Even though this ebook is fully on Pandas, I should also point out that there are other numerical packages in Python which is used in handling a large dataset. One of which is Datatable. The syntax in Datatable is very similar to MySQL such as the use of **SELECT**, **WHERE**, **ORDER BY** and so on. However, Datatable does not fully replicate all functionality from SQL-type software. And it does not contain statistical analysis tools and visualization backend as in Pandas. Thus Datatable is more specialized in processing large datasets.

Before using Pandas

Pandas is a package of Python and naturally a Python interpreter is indispensable for generating results presented in this book. For readers interested in trying all the code by hands on their own python interpretor (or in some integrated environment such as Visual Studio Code or Jupyter Notebook), here are the setting of my Python interpretor and packages.

- Python 3 (version 3.9.2)
- NumPy (version 1.18.5)
- Pandas (version 1.0.3
- MatPlotLib (version 3.3.2)

In case when the above packages are not yet installed in your machine, one can simply type

```
!pip install pandas
```

to install the library accordingly through the Jupyter Notebook interface, or

```
pip install pandas
```

in the Anacornda Prompt. For both cases, the installation only needs to be done once before the first time we use the library. Other libraries can be installed in the same manner.

Nowadays software is updating in an unprecedented rate that almost everyday there are some new patches for some software package available. It is very possible that the exact version of your package when you read this book, the version is already outdated. As long as the functionality described is available in your version of package, it will be fine for the practice purpose.

The Best Way to Learn

Just like all other textbooks in programming you may have encountered, the best way to make the data processing skills using Pandas will be to practice while you read this book! I encourage all readers to follow the short code examples and type them on your own machine for practice. Besides the code examples, I also included a number of coding exercises at the end of each chapter. Certainly, I strongly encourage all readers to try solving one or two to check that the new ideas are firmly grasped in the process.

Have fun in the exploration of Pandas!