

Data Analysis with pandas

Sirasit Lochanachit, PhD

Why Python for Data Analysis?

- Free and open source
 - Communities, docs, and tutorials
- Powerful libraries & flexible
- Steep learning curve but simple to learn

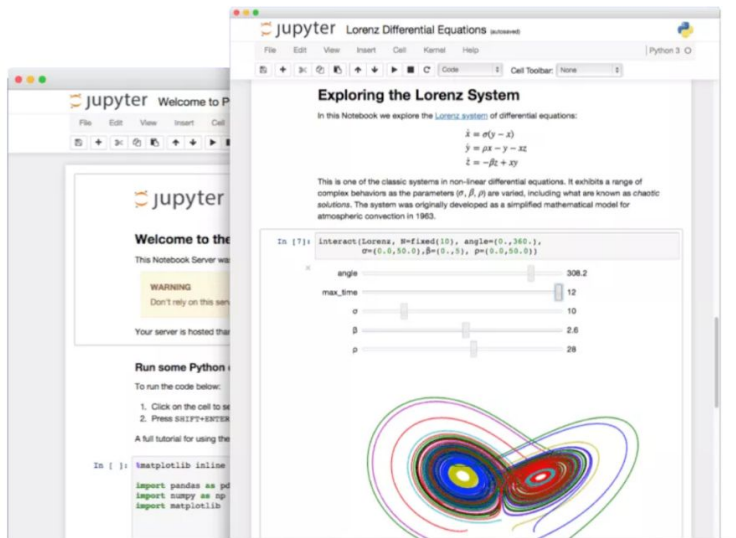
Outline (This Week)

- Jupyter Notebook
- NumPy
- pandas

Jupyter Notebook



- **interactive computing environment** that enables users to author notebook documents that include: Live code, plots, narrative text, equations, images, dashboards and other media.



Jupyter Notebook: The Classic Notebook Interface

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

[Try it in your browser](#)

[Install the Notebook](#)

Jupyter Notebook



- These documents provide a **complete** and **self-contained record** of a computation.
 - Sequence of multiple **cells**
- Documents can be
 - Converted to various formats (html/pdf)
 - Shared with others using email, Dropbox, version control systems (like git/GitHub) or nbviewer.jupyter.org

What is pandas?



What is pandas?

- Open source library in Python
- For working with data (e.g. Data manipulation and analysis)
- Fast, powerful, flexible and easy to use
- Widely used by Data Analysts and Data Scientists
- Under active development
- Works well with other packages
 - Built on top of Numpy
 - scikit-learn

How to install pandas?

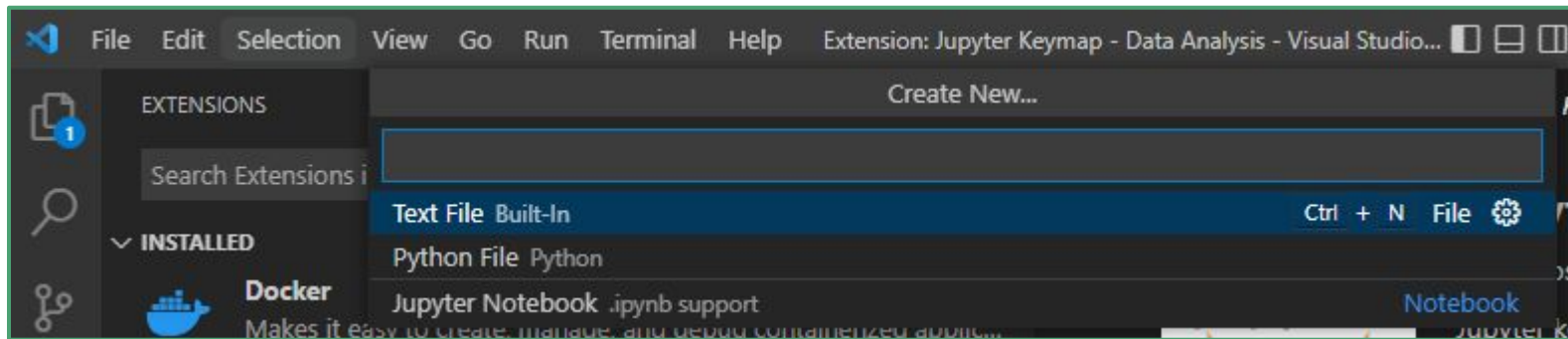
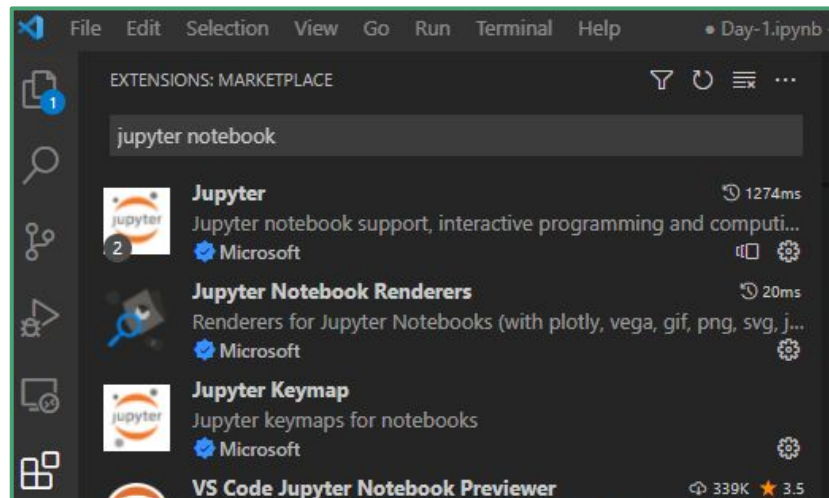
- Easiest way is by downloading the Anaconda distribution of Python
 - <https://docs.continuum.io/anaconda/>
- `pip install pandas`
 - Try this in VS Code or other IDEs

How to use pandas without installing?

- Google Colab provides pandas from the start (pre-install)
 - <https://colab.research.google.com/>

Extra: How to use Jupyter Notebook in VS Code

- Install Jupyter Notebook Extension



How do I read a tabular data file into pandas?

What is tabular data?

- Data that looks like a table
- Data with rows and columns
- Looks like excel spreadsheet

Common formats of tabular data:

- csv (comma, separated value)

```
import pandas as pd

pd.read_table(filename or URL)

pd.read_csv(filename or URL)


head()

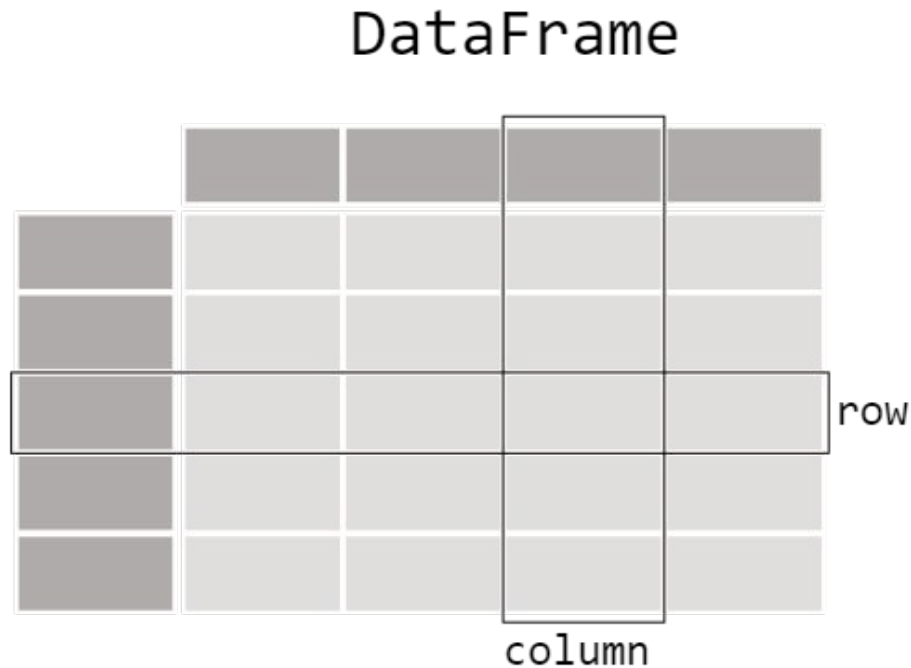
tail()
```

What kind of data does pandas handle?

A **DataFrame** is a 2-dimensional data structure that can store data of different types (including characters, integers, floating point values, categorical data and more) in columns.

It is similar to a spreadsheet (i.e. Excel)

- Basically, it looks like a table.



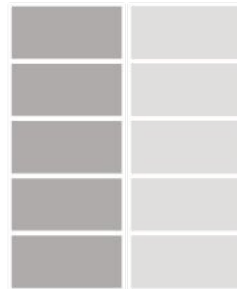
How to **select** a pandas Series from a DataFrame?

Bracket notation

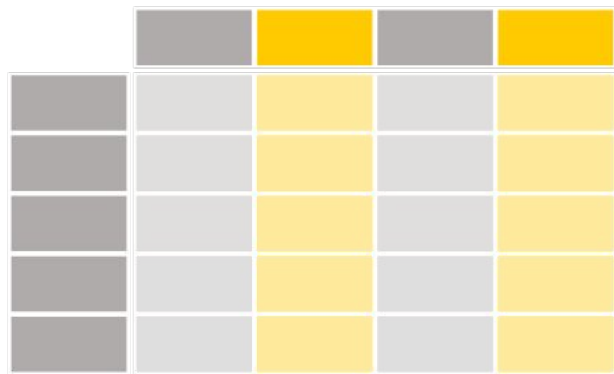
- Can select multiple columns

Dot notation

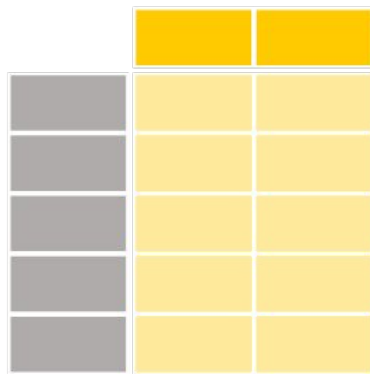
Series



A diagram representing a pandas Series as a single column of six cells. The first four cells are dark gray, and the last two cells are light gray.



A diagram representing a pandas DataFrame as a grid of 20 cells (4 rows by 5 columns). The first column has 4 dark gray cells. The second column has 4 light gray cells. The third and fifth columns have 4 yellow cells. The fourth column has 4 light gray cells.



A diagram representing a pandas Series as a single column of six cells. The first two cells are yellow, and the last four cells are light gray.

Basic DataFrame **Exploration**

shape

mean()

dtypes

std()

info()

min()

describe()

max()

value_counts()

count()

unique()

nunique()

How to **rename** pandas DataFrame's columns?

1. `rename()`
2. re-assigned by list
3. `read_csv()`

How to **remove** columns from a pandas DataFrame?

1. `drop()`
2. Select all other columns instead

How to **filter rows** of a pandas DataFrame by column value?

- Need a Series of boolean values
- Use & and | to add filter criterias

How to **sort** a pandas DataFrame or Series?

- `sort_values()`

String methods in pandas

- `str.upper()`
- `str.contains('abc')`

Changing data type of a pandas Series

- `astype()`

How to select specific **rows** and **columns** from a DataFrame?

- `loc[row]`
- `loc[row, column]`
- `iloc[row_number, column_number]`



When to use a **groupby** command?

- `groupby()`

How to handle **missing values**?

Check missing values in a DataFrame

- `isnull()` or `isna()`
- `notnull()` or `notna()`

1. Drop missing values with `dropna()`

Or

2. Fill missing values with `fillna()`

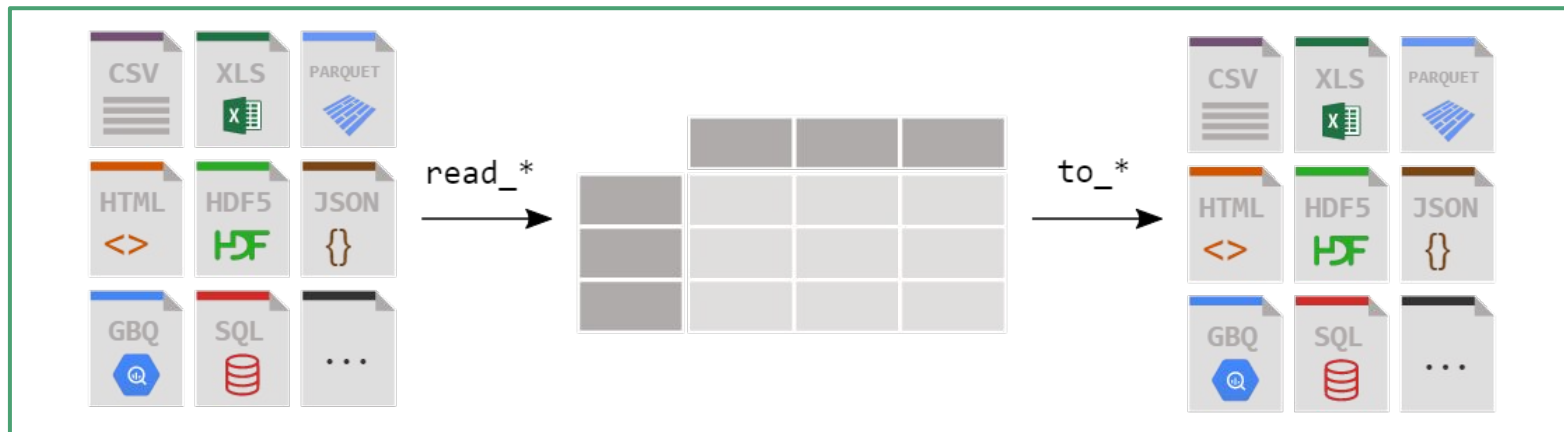
How to **export** a DataFrame?

CSV

- `to_csv()`

Excel

- `to_excel()`



Exercise: SF Salaries

See SF Salaries Exercise.ipynb