

# Why **Python**, Some Libraries and Functionalities in Spyder

Arturo Aguilar, Diego Hernandez & Mariano Herrera

# Why Python?

- Free open-source cross-platform language
- Rich library for almost any imaginable task
- Very easy to learn
- Efficient handling of data structures
- Many APIs to access data



# Points to consider...

- Treated as an object... so variables are almost 3 times the size of other languages. Need to optimize data sets to avoid low execution
- Understand third party libraries, to avoid losing speed and clarity
- Like in any other language, always comment your codes
- Learn how to combine platforms, i.e. Python + julia

# Other available options

- C++ - high speed, efficient memory management / difficult to pick up
- C# & Java – good for backtesting, native GUI & several numerical analysis libraries
- Matlab & R excellent for advance statistical analysis / R low speed unless operations are vectorized & Matlab licensing cost
- julia high speed, simple to type / not so many APIs, users or examples yet

\*\*Speed comparison: [https://github.com/astrojhgu/adaptrapezoid\\_benchmark/blob/master/Result.md](https://github.com/astrojhgu/adaptrapezoid_benchmark/blob/master/Result.md)

# Libraries

## 15 Python Libraries for Data Science You Should Know



February 5, 2020



Python is one of the most popular languages used by data scientists and software developers alike for data science tasks. It can be used to predict outcomes, automate tasks, streamline processes, and offer business intelligence insights.

It's possible to work with data in vanilla Python, but there are quite a few open-source libraries that make Python data tasks much, much easier.

You've certainly heard of some of these, but is there a helpful library you might be missing? Here's a line-up of the most important Python libraries for data science tasks, covering areas such as data processing, modeling, and visualization.



Learn by  
~~watching videos~~  
coding!

Try it now >>

Source: <https://www.dataquest.io/blog/15-python-libraries-for-data-science/>

# What do I need to know to use Python?

- Variables
- Operators
- Conditionals
- Loops
- Functions
- IDE

# Variables

Python has the following data types built-in by default, in these categories:

Text Type -	str
Numeric Types -	int, float, complex
Sequence Types -	list, tuple, range
Mapping Type -	dict
Set Types -	set, frozenset
Boolean Type -	bool
Binary Types -	bytes, bytearray, memoryview

Source: [https://www.w3schools.com/python/python\\_datatypes.asp](https://www.w3schools.com/python/python_datatypes.asp)

# Objects

Objects in Python have three main characteristics:

- `id()` function returns identity (unique integer) of an object
- `type()` function either returns the type of the object or returns a new type object based on the arguments passed
- value

Source: <https://www.programiz.com/python-programming/methods/built-in>



# Main operators

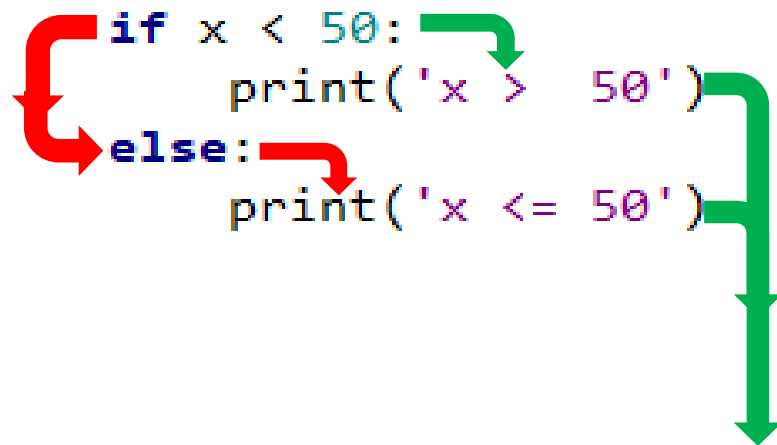
Example	Meaning
$a + b$	Sum of a and b
$a - b$	B subtracted from a
$a * b$	Product of a and b
$a / b$	Quotient when a is divided by b
$a \% b$	Remainder when a is divided by b
$a // b$	Quotient when a is divided by b
$a ** b$	a raised to the power of b

Example	Result
$a == b$	True if a is equal to b
$a != b$	True if a is not equal to b
$a < b$	True if a is less than b
$a <= b$	True if a less/equal to b
$a > b$	True if a is greater than b
$a >= b$	True if a greater/equal to b
$a \text{ or } b$	True if either a or b is True
$a \text{ and } b$	True if either a and b are True

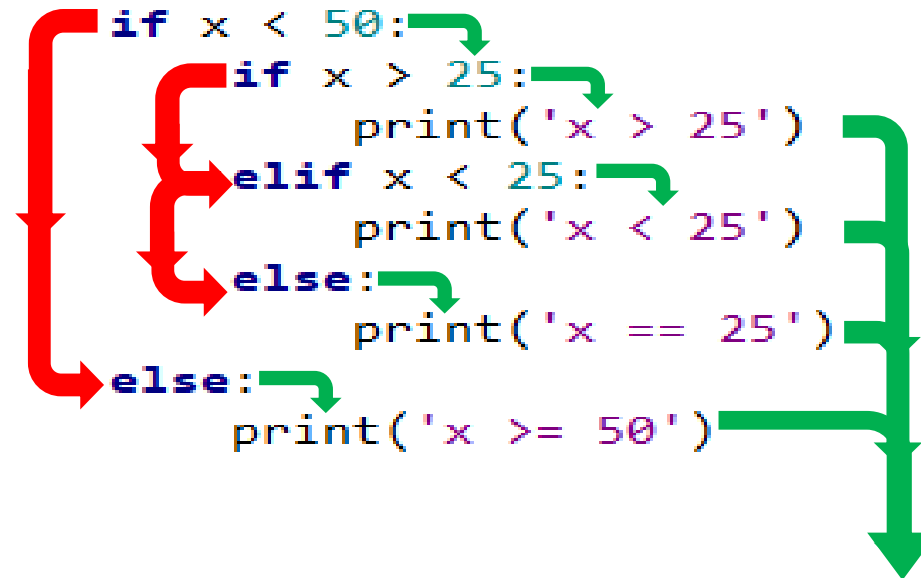
Source: <https://www.programiz.com/python-programming/operators>

# Conditionals

x = 120



x = 30



# Loops

Remember...

In Python you start from 0

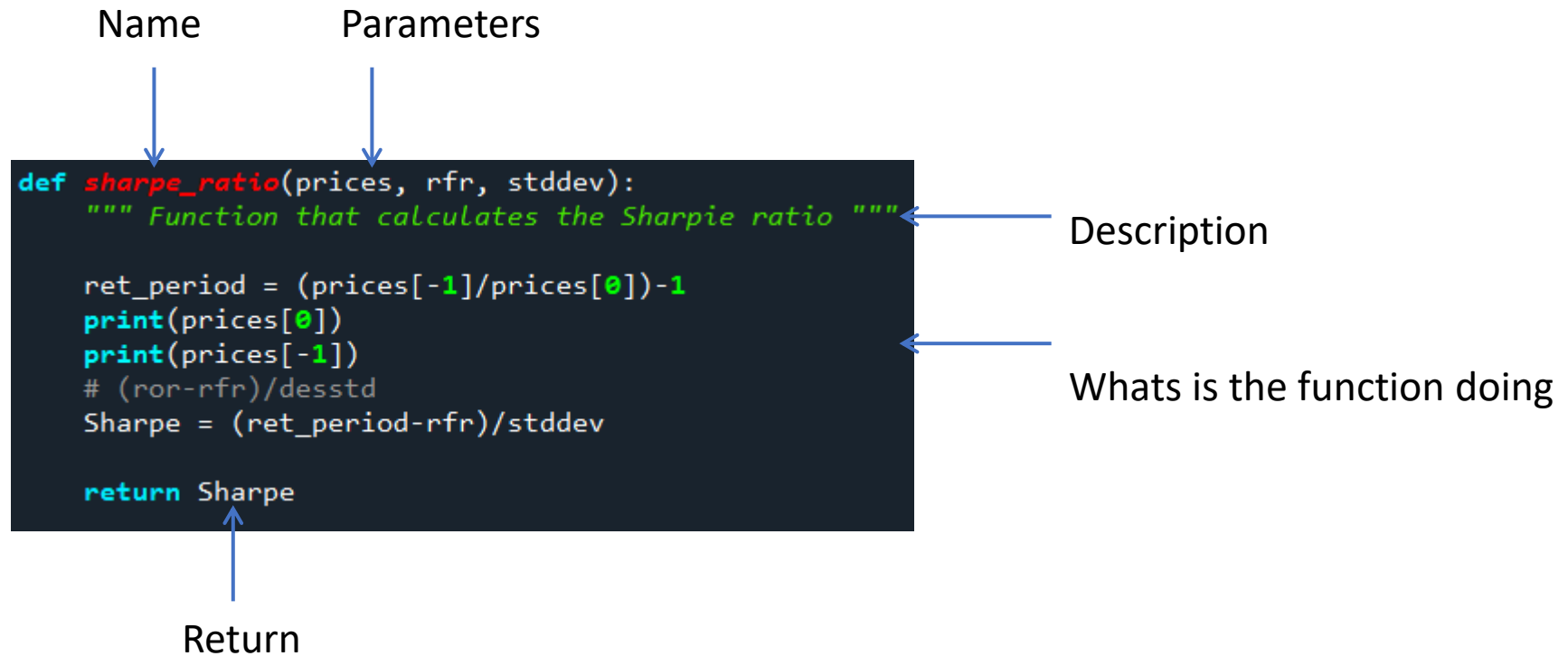
```
#           0       1       2       3
elements = ['op1', 'op2', 'op3', 'op4']
#          -4       -3       -2       -1
```

```
for element in elements:
    print(element)
```



```
op1
op2
op3
op4
```

# Functions



# What are IDEs and Code Editors?

Whether you are new to this game or you are a veteran player, you need an IDE (Integrated Development Environment) or a code editor to showcase your coding skills and talent. An IDE is a software that consists of common developer tools into a single user-friendly GUI (Graphical User interface). An IDE majorly consists of a source code editor for writing software code, local build automation for creating a local build of the software like compiling computer source code. Lastly, it has a debugger, a program for testing other programs. An IDE can have many more features apart from these & those vary for each IDE.

# Spyder



Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\artur\Desktop\UDLAP - Python and Machine Learning for Finance\Understanding Python

...ktop\UDLAP - Python and Machine Learning for Finance\Understanding Python\Understanding Python S3.py

Update Portfolio and Metrics.py X Portfolio Optimisation.py X Morning Shot US tk.py X Understanding Python S3.py X

```
1
2 # Understanding Python
3
4 'The objective of this code is undestand data types, objects, operators'
5
6 # Libraries
7
8 'Always add libraries on the top and comments to your codes'
9
10 from datetime import date
11 import numpy as np
12 import pandas as pd
13 import pandas_datareader as dr
14 import matplotlib.pyplot as plt
15
16 # This function calculates the RSI
17
18 def rsi(prices, n=14):
19
20     deltas = np.diff(prices)
21     seed = deltas[:n]
22
23     up = seed[seed>=0].sum()/n
24     down = -seed[seed<0].sum()/n
25
26     rs = up/down
27     rsi = np.zeros_like(prices)
28     rsi[:n+1] = 100. - 100./(1.+rs)
29
30     for i in range(n+1, len(prices)):
31         delta = deltas[i-1]
32
33         if delta>0:
34             upval = delta
35             downval = 0.
36         else:
37             upval = 0.
38             downval = -delta
39
40         up = (up*(n-1) + upval)/n
41         down = (down*(n-1) + downval)/n
42
43         rs = up/down
44         rsi[i] = 100. - 100./(1.+rs)
45
46 return rsi
```

Name	Type	Size	Value
df	DataFrame	(755, 15)	Column names: High, Low, Open, Close, Volume, Adj Close, Date, 1D Retu ...
i	int	1	76
Names	DataFrame	(77, 2)	Column names: Label, Ticker
prices	DataFrame	(755, 15)	Column names: High, Low, Open, Close, Volume, Adj Close, Date, 1D Retu ...
stock_ticker	str	1	FXI
today	date	1	2020-12-28

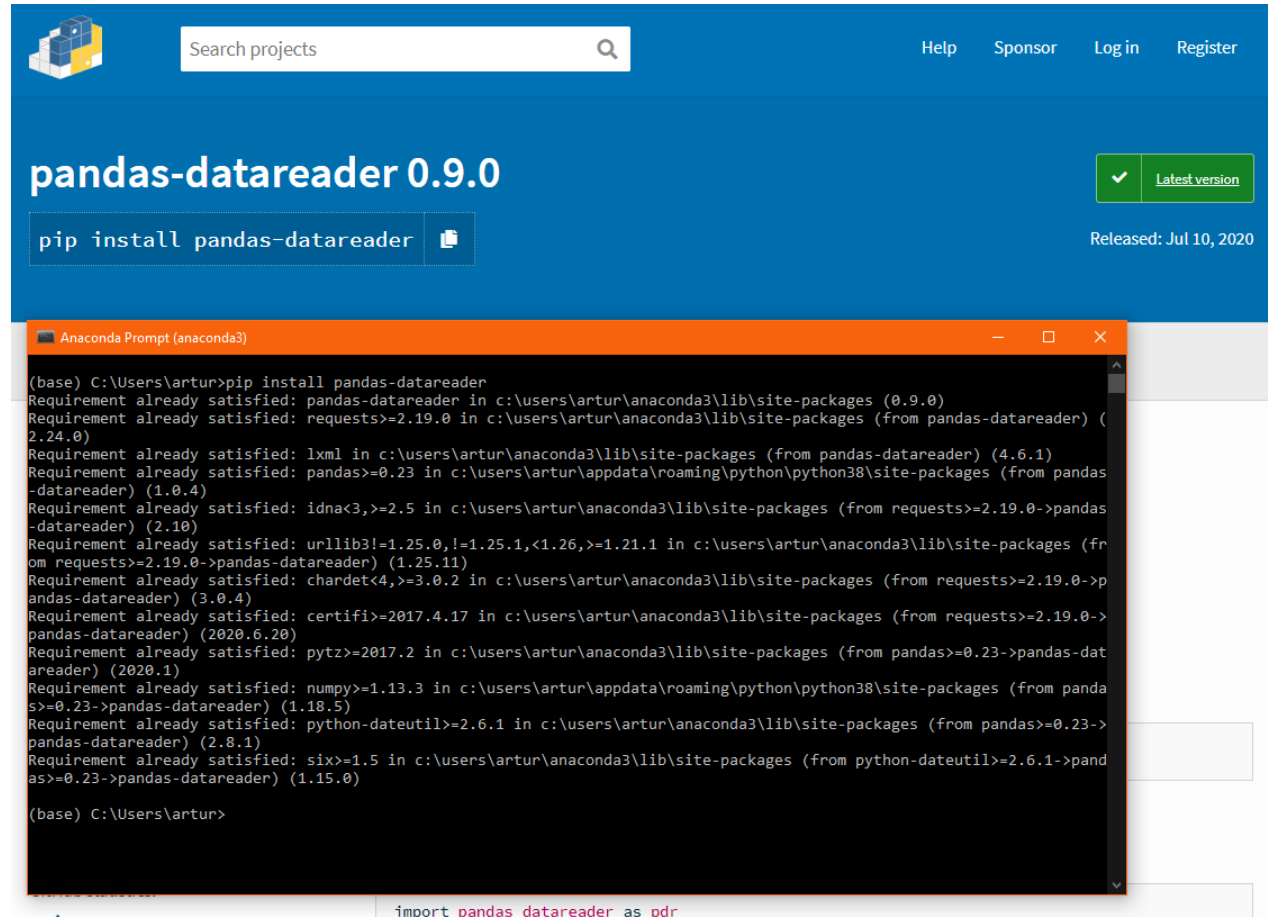
Variable explorer Plots Help Profiler Code Analysis

Console 1/A X

In [5]:

LSP Python: ready conda: base (Python 3.8.5) Line 120, Col 1 UTF-8 CRLF RW Mem 52%

# Anaconda Prompt



The screenshot displays the Anaconda Prompt interface. At the top, there is a blue header bar with the Anaconda logo on the left, a search bar labeled "Search projects" in the center, and links for "Help", "Sponsor", "Log in", and "Register" on the right. Below the header, the main content area has a blue background. It features the text "pandas-datareader 0.9.0" in large white font. To the right of this text is a green button with a white checkmark and the text "Latest version". Below the main text, there is a white box containing the command "pip install pandas-datareader" and a copy icon. To the right of this box, it says "Released: Jul 10, 2020".

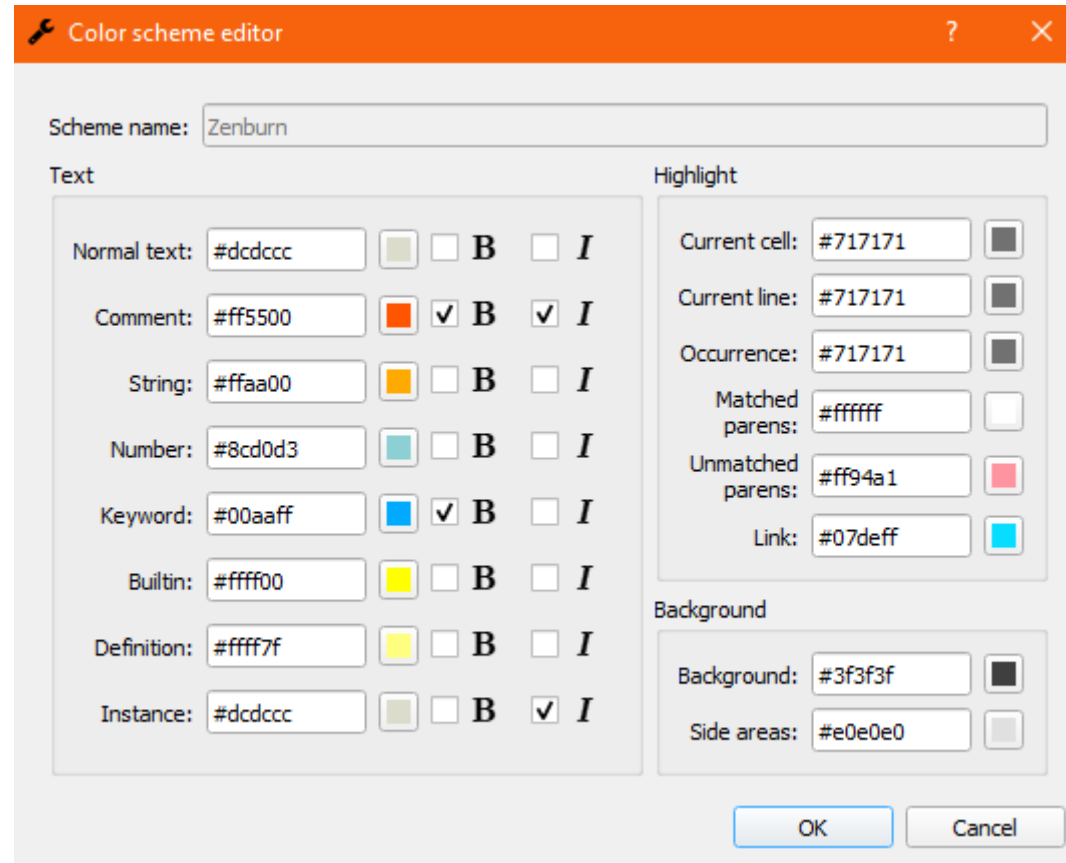
Below the main content area, there is a black terminal window titled "Anaconda Prompt (anaconda3)". The terminal shows the output of the command "pip install pandas-datareader". The output indicates that the requirement for pandas-datareader is already satisfied in the current environment (0.9.0). It also lists several other requirements that are already satisfied, including requests, lxml, pandas, idna, urllib3, chardet, certifi, pytz, numpy, python-dateutil, and six, along with their respective versions and the environments they are installed in.

```
(base) C:\Users\artur>pip install pandas-datareader
Requirement already satisfied: pandas-datareader in c:\users\artur\anaconda3\lib\site-packages (0.9.0)
Requirement already satisfied: requests>=2.19.0 in c:\users\artur\anaconda3\lib\site-packages (from pandas-datareader) (2.24.0)
Requirement already satisfied: lxml in c:\users\artur\anaconda3\lib\site-packages (from pandas-datareader) (4.6.1)
Requirement already satisfied: pandas>=0.23 in c:\users\artur\appdata\roaming\python\python38\site-packages (from pandas-datareader) (1.0.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\artur\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in c:\users\artur\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (1.25.11)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\artur\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\artur\anaconda3\lib\site-packages (from requests>=2.19.0->pandas-datareader) (2020.6.20)
Requirement already satisfied: pytz>=2017.2 in c:\users\artur\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (2020.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\artur\appdata\roaming\python\python38\site-packages (from pandas>=0.23->pandas-datareader) (1.18.5)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\artur\anaconda3\lib\site-packages (from pandas>=0.23->pandas-datareader) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\users\artur\anaconda3\lib\site-packages (from python-dateutil>=2.6.1->pandas>=0.23->pandas-datareader) (1.15.0)

(base) C:\Users\artur>
```

At the bottom of the terminal window, there is a line of code: `import pandas_datareader as pdr`.

# Spyder suggested color scheme





# Coding... + Libraries

















# Standard Library

The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming.

# NumPy

NumPy brings the computational power of languages like C and Fortran to Python, a language much easier to learn and use. With this power comes simplicity: a solution in NumPy is often clear and elegant.

<b>Quantum Computing</b>  QuTiP PyQuil Qiskit	<b>Statistical Computing</b>  Pandas statsmodels Seaborn	<b>Signal Processing</b>  SciPy PyWavelets	<b>Image Processing</b>  Scikit-image OpenCV	<b>Symbolic Computing</b>  SymPy	<b>Astronomy Processes</b>  AstroPy SunPy SpacePy	<b>Cognitive Psychology</b>  PsychoPy
<b>Bioinformatics</b>  BioPython Scikit-Bio PyEnsembl	<b>Bayesian Inference</b>  PyStan PyMC3	<b>Mathematical Analysis</b>  SciPy SymPy cvxpy FEniCS	<b>Simulation Modeling</b>  PyDSTool	<b>Multi-variate Analysis</b>  PyChem	<b>Geographic Processing</b>  Shapely GeoPandas Folium	<b>Interactive Computing</b>  Jupyter IPython Binder

Source: <https://docs.python.org/3/library/index.html>

# pandas

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Additionally, it has the broader goal of becoming **the most powerful and flexible open source data analysis/manipulation tool available in any language**. It is already well on its way toward this goal.

Source: [https://pandas.pydata.org/docs/getting\\_started/overview.html](https://pandas.pydata.org/docs/getting_started/overview.html)

# datareader

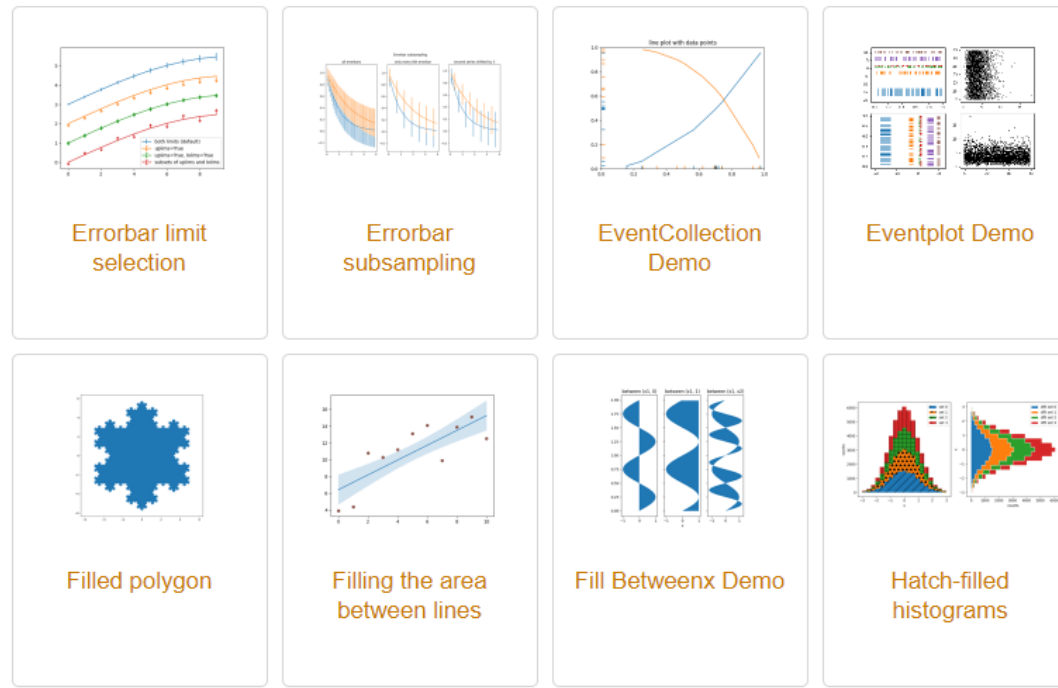
Functions from `pandas_datareader.data` and `pandas_datareader.wb` extract data from various Internet sources into a pandas DataFrame. Currently the following sources are supported:

Alpha Vantage	Eurostat
Enigma	MOEX
IEX	Nasdaq Trader symbol definitions
Kenneth French's data library	Naver Finance
Quandl	OECD
St.Louis FED (FRED)	Stooq
Tiingo	Thrift Savings Plan
World Bank	

Source: [https://pandas-datareader.readthedocs.io/en/latest/remote\\_data.html](https://pandas-datareader.readthedocs.io/en/latest/remote_data.html)

# Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.



Source: <https://matplotlib.org/gallery/index.html>

# Requisites

Watch these videos:

<https://www.youtube.com/watch?v=eXBD2bB9-RA&list=PLQVvvaa0QuDeAams7fkdcwOGBpGdHpXln>

Download Anaconda:

<https://www.anaconda.com/products/individual>

# References

<https://www.python.org/>

<https://docs.python.org/3/>

<https://realpython.com/>



# Disclaimer

The opinions expressed in this presentation and on the previous slides are solely those of the presenters.