

7 lesson 05

Data Wrangling and Visualization

Python for Financial Analysis
Rajah Chacko

|elvtr|

Syllabus Review

1 Introduction to Python: Python in Finance

2 Python Basic Syntax: Importing Libraries

3 Working with Pandas

4 Pandas Underneath the Hood: Working with NumPy

5 Data Wrangling and Visualization

6 Extracting Financial Insights from Charts and Graphs

7 Financial Calculations with Python: Part 1

8 Financial Calculations with Python: Part 2

9 CAPM and Portfolio Management

10 Linear Regression

11 Time Series Analysis

12 Algorithmic Trading



Bonus Class: Cryptocurrency Beyond the Basics with a Fintech Guest Speaker

Class agenda

- Best data sources
- Data cleansing and normalization
- Libraries for data visualization
- Types of charts/graphs and how to build them
- Pythonic: List indexes, appending, inserting, removing
- Pythonic: Iterating over lists; what happens if the list is empty?

Best data sources

- Data cleansing
 - A. Trimmed Mean inflation PCE:
 - B. Case-Shiller home index
 - C. Real M2
- Fama-French model
 - A. http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html
 - B. F-F three-factor model
 - C. Portfolios
- Hunt on GitHub

Data cleansing and normalization

- Federal Reserve: <https://fred.stlouisfed.org/>
 - A. Drop N/A
 - B. Drop by index
 - C. Coerce to datetime
 - D. Drop columns
 - E. Set Index
 - F. (optional) Fill in missing values with mean
 - G. (optional) remove outliers
- Normalization
 - A. (optional reading)
<https://towardsdatascience.com/normalization-techniques-in-python-using-numpy-b998aa81d754>
 - B. (optional) puts values between $[0,1]$ or $[-1,1]$

Libraries for data visualization

- Matplotlib and pyplot
 - A. Lines and bar charts
- Seaborn
 - A. Categorical
 - B. Box & whisker
 - C. Regression

Pythonic: Lists

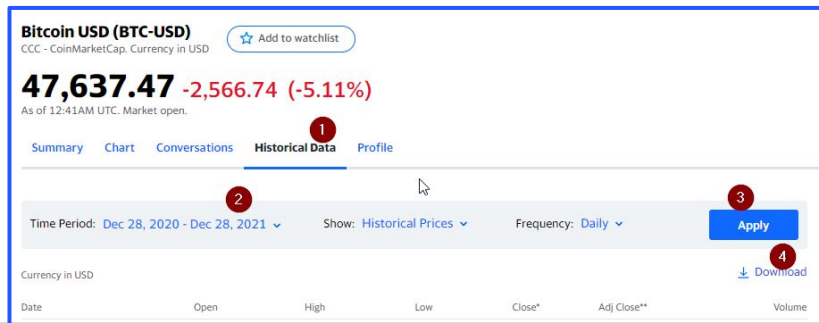
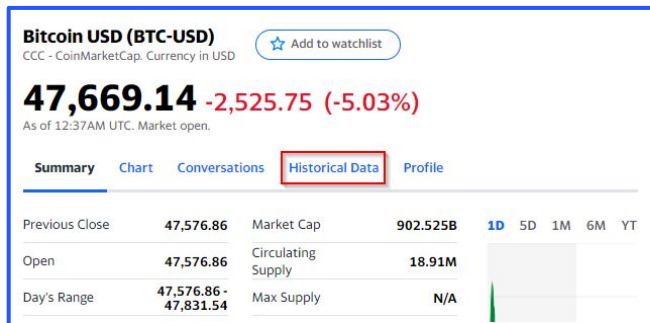
- Indexes
- Slicing
- Extend — adds a single thing. If adding a list, add one element at a time.
- Append — adds a single thing. If adding a list, add it as a list.
- Insert — adds a single thing at a specified list position
- Removing

Pythonic: Iterating over lists

- For loop
 - A. By index
 - B. By element
 - C. Using enumerate
- Handling empty lists

Importing stock data

- Let's download a Comma-Separated Variable (CSV) file.
 - A. Start at finance.yahoo.com
 - B. Search bar: enter a favorite (e.g., AAPL, TSLA, AMZN, BTC-USD, ^SPX)
 - C. Select the Historical Data tab, change time to 1Y, Click Apply,, and then Download
 - D. Save the CSV file to the directory where you're working. Feel free to peek!



Working with Pandas DataFrames

- The import statement
 - A. `import pandas as pd`
- Using Pandas to read the csv into a dataframe
 - A. `tsla = pd.read_csv('TSLA.csv')`
 - B. Reads the first line as a header.
- Printing the head, the tail, or the whole thing
 - A. With `head()`, `tail()`, or just the df name.

But will it work for Excel?

- You bet.
- Excel files also have worksheets. You need to tell it which worksheet to import.
- Note that it'll read .xls and .xlsx files even if you don't have MS Excel installed.
- Can also read from OpenOffice (odf, ods and odt files).

More fun with pandas

- Coercing data types
- `len(df)` tells us how many rows
- `Df.columns` tells us the column names
- Powerful filters and indexes
 - A. `Df[df['col_name'] == x]`
 - B. `Df.col_name` as an alternative
- Indexing
 - A. Often one level.
 - B. Indexing by datetime

Functions

- Building Functions
 - A. Encapsulation
 - a. Well-defined parameters and return values
 - b. You've got one job
 - c. What happens in functions stays in functions
 - d. Avoids cut & paste (and code smell)
 - B. Build simple functions greet and double
 - C. Three bond prices: does this look easier?
- Calling functions
 - D. We've already called functions with like `sqrt()`, `head()`, and `set_index()`
 - E. We set parameters on the way in and get a return value out

Classes, and Subclasses

- Classes are factories for creating Python objects
 - A. The objects get initialized
 - B. They have state (like a name and an age)
 - C. They have methods (the functions we just learned about)
 - D. Self keyword
 - E. Change of perspective
 - a. The original syntax for a function call, `print_time(current_time)`, suggests that the function is the active agent. It says something like, “Hey, `print_time`! Here’s an object for you to print.”
 - b. In object-oriented programming, the objects are considered the active agents. An invocation like `current_time.print_time()` says “Hey `current_time`! Please print yourself!”
- Subclasses inherit from their parent class
 - F. They inherit methods and state (and can add or change their own)
 - G. They help us add new items without breaking existing ones
 - H. (Example: a `ConvertibleBond` inherits from `Bond`)

Assignment #5

Download two macroeconomic variables that you think are related from FRED into Excel spreadsheets. Load them into Pandas dataframes and plot them in a way that tells a story. You can plot them as simply or as fancy as you'd like.

Python take-home (optional): Implement both a FIFO (First In, First Out) queue and LIFO (Last In, First Out) stack. (Can you implement these as classes?) Show that you can enqueue and dequeue in FIFO order and can handle an empty queue (without throwing a Python error). Show that you can push and pop in LIFO order and can handle an empty stack.



Resources

- Federal Reserve: <https://fred.stlouisfed.org/>

- Fama-French model

http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.htm

- Data cleansing

Drop N/A

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html>

Drop by index

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html>

Coerce to datetime

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.to_datetime.html

Set Index:

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.set_index.html

Resources

for Data Visualization (part 1)

- Matplotlib

Tutorial:

<https://matplotlib.org/stable/tutorials/introductory/usage.html>

Reference:

<https://matplotlib.org/stable/api/index.html>

Pyplot:

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html

Bar chart Reference:

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html

Bar chart example:

<https://pythonspot.com/matplotlib-bar-chart/>

Resources

for Data Visualization (part 2)

- Seaborn

Reference:

<https://seaborn.pydata.org/tutorial.html>

Gallery:

<https://seaborn.pydata.org/examples/index.html>

Official intro:

<https://seaborn.pydata.org/introduction.html>

Tutorial:

<https://www.geeksforgeeks.org/python-seaborn-tutorial/>

Q&A