IQIC - Quantitative Tools

#9: Python Basics for Finance

Why Python for Quantitative Investing

STEMise

- Quantitative Trading involves working with large amounts of financial data
- Python helps investors:
  - Analyze historical prices
  - Calculate returns & risk
  - Test strategies
  - Automate repetitive calculations
  - Visualize portfolio performance
- Essentially, Python gives you the tools to **analyze evidence systematically**

Minimal Python Concepts you need

1. Variables - Store Information
2. Lists - Store multiple values
3. Functions - Perform Operations
4. Libraries (at a beginner level) - Give Python additional capabilities
5. Basic Logic - Useful for strategy rules

** We will go deeper into python libraries in a deeper slide

Pandas & Financial DataFrames

- Financial analysis involves thousands or millions of observations (price, returns, trading volumes, dates, etc)
  - Working with this information individually would be inefficient
- **Pandas** is a python library designed to help organize and analyze structured data.
- Its main structure, the dataframe, organizes information into rows and columns, similar to a spreadsheet
  - Rows can represent individual observations
  - Columns can represent variables such as price, return, or volume

Dataframe Visual

	date	ticker	price	return
0	jan 2	aapl	$185.64	
1	jan 3	aapl	$184.25	0.75
2	jan 4	aapl	$186.21	1.06
3	jan 5	aapl	$188.29	1.12

** In lesson 12, you will see how to extract this dataframe!

Simple Financial Analysis Example

Week	Stock X	Stock Y
1	+2.0%	+1.0%
2	-1.0%	+2.0%
3	+3.0%	-1.0%
4	+2.0%	+2.0%

How does the average weekly return of stock X and stock Y compare?

Simple Financial Analysis Example

```python
# Weekly returns for each stock as a decimal
stock_x = [0.02, -0.01, 0.03, 0.02]
stock_y = [0.01, 0.02, -0.01, 0.02]

"""
Calculate the average weekly return using the formula:
mean = (sum of all values) / (total number of values)
"""

avg_x = sum(stock_x) / len(stock_x)
avg_y = sum(stock_y) / len(stock_y)

# Compare the two stocks
if avg_x > avg_y:
    print("Stock X had the higher average weekly return.")
else:
    print("Stock Y had the higher average weekly return.")

# Print out the average return for both stocks
print("Stock X average weekly return:", avg_x)
print("Stock Y average weekly return:", avg_y)
```

```
Stock X had the higher average weekly return.
Stock X average weekly return: 0.015
Stock Y average weekly return: 0.01
```

Additional Resources

This is just the beginning! Here are the next topics you can learn to properly backtest your strategies

Additional optional resources:

1. https://www.quantconnect.com/learning/articles/introduction-to-financial-python
   a. Short articles that connects quant concepts to python code
   b. Return, variance, risk, regression, etc
2. https://www.datacamp.com/courses/introduction-to-python-for-finance?utm
   a. Free 4 hour intro to python for finance course
   b. Python syntax, lists, arrays (numpy), visualization (matplotlib
3. https://www.datacamp.com/courses/intermediate-python-for-finance?utm
   a. Free 4 hour intermediate intro python for finance course
   b. Datetime, Logic, Dataframe (Pandas), Extracting stock data

You don't need to become a software engineer. You just need enough Python to turn data into evidence.