IQIC - Quantitative Tools

#12: Useful Libraries

Pandas - Working with Financial Data

Pandas is a Python library for organizing, cleaning, and analyzing structured data.

In quantitative investing, Pandas is commonly used to:

- Load financial datasets
- Organize prices, returns, and other variables
- Filter and sort data
- Prepare data for analysis and backtesting

Think of Pandas as: Your spreadsheet for python trading

Visual - Code

```python
import pandas as pd

# Create data dictionary
data = {
    "date": ["jan 2", "jan 3", "jan 4", "jan 5"],
    "ticker": ["aapl", "aapl", "aapl", "aapl"],
    "price": ["$185.64", "$184.25", "$186.21", "$188.29"],
    "return": ["", "0.75", "1.06", "1.12"],
}

# Create DataFrame
df = pd.DataFrame(data)

print(df)

```

Visual - Output

	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

NumPy - Numerical Computing

Numpy is a Python library designed for fast numerical calculations

In quantitative investing, it can help you:

- Perform mathematical calculations on large sets of numbers
- Work with arrays of financial data
- Calculate statistics and mathematical functions
- Build the numerical calculations behind a strategy

Think of NumPy as: The mathematical engine behind many quantitative programs

Visual - Code & Output

```python
import numpy as np

returns = np.array([0.02, -0.01, 0.03, 0.01])

average_return = np.mean(returns)
volatility = np.std(returns)

volatility = round(volatility, 4) #rounds to 2 decimals

print(f'{average_return * 100}%')
print(f'{volatility * 100}%')
```

```
1.25%
1.48%
```

Matplotlib - Visualizing Your Results

Matplotlib is a Python library designed for creating charts and graphs

In quantitative investing, it can help you visualize:

- Stock prices
- Portfolio performance
- Strategy vs. benchmark
- Drawdowns
- Other patterns in your data

Think of Matplotlib as: The visualization layer of your quantitative investing workdlow

Visual - Code

```python
import matplotlib.pyplot as plt

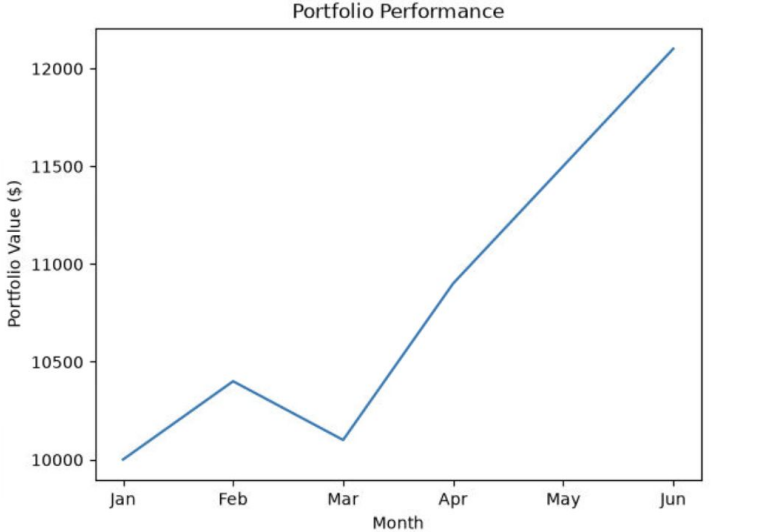
# Hypothetical portfolio values over 6 months
dates = ["Jan", "Feb", "Mar", "Apr", "May", "Jun"]
portfolio_values = [10000, 10400, 10100, 10900, 11500, 12100]

# Create the chart
plt.plot(dates, portfolio_values)

# Add labels
plt.title("Portfolio Performance")
plt.xlabel("Month")
plt.ylabel("Portfolio Value ($)")

# Display the chart
plt.show()
```

Visual - Output



How the libraries work together

The quantitative workflow usually looks like:

1. Get Data

   ↓

2. Organize & Clean (Pandas)

   ↓

3. Calculate & Analyze (NumPy)

   ↓

4. Visualize (Matplotlib)

   ↓

5. **Evaluate Your Strategy**

Additional Resources

- Python Data Science Handbook - Jake VanderPlas: Python Data Science Handbook
  - Explores how Pandas, NumPy, and Matplotlib work together to analyze datasets, perform calculations, and visualize results
    - PRACTICAL EXAMPLES INCLUDED