

# Manipulating Financial Data in Python

From Quantitative Analysis Software Courses

## Contents

- 1 Lesson 1: Reading, slicing and plotting stock data
- 2 Lesson 2: Working with many stocks at once
- 3 Lesson 3: The power of Numpy
- 4 Lesson 4: Statistical analysis of time series
- 5 Lesson 5: Incomplete data
- 6 Lesson 6: Histograms and scatter plots
- 7 Lesson 7: Sharpe ratio & other portfolio statistics
- 8 Lesson 8: Optimizers: Building a parameterized model
- 9 Lesson 9: Optimizers: How to optimize a portfolio

## Lesson 1: Reading, slicing and plotting stock data

- Overview of the data we'll be working with (from Yahoo!)
- Introduction to our primary library: Pandas
- Reading CSV data into Pandas
- Filtering to specific dates
- Plotting

Reading: "Python for Finance", Chapter 6: Financial time series

## Lesson 2: Working with many stocks at once

- Our target data frame structure
- Address reverse order issue
- Reading data for multiple stocks into the structure
- Date slicing
- Symbol slicing
- Plotting
- Normalizing

## Lesson 3: The power of Numpy

- What is Numpy and how it relates to Pandas
- Why is Numpy powerful/important?
- Creating Numpy arrays
- Indexing and slicing Numpy arrays
- Important data processing on Numpy arrays
- Example use with pandas too

Reading: "Python for Finance", Chapter 4: Data types and structures

## Lesson 4: Statistical analysis of time series

- Gross statistics on dataframes
- Rolling statistics on dataframes
- Plotting a technical indicator (Bollinger Bands)

Reading: "Python for Finance", Chapter 6: Financial time series

## Lesson 5: Incomplete data

- How incomplete data arises in financial data
- Different approaches to dealing with it

## Lesson 6: Histograms and scatter plots

- Histogram of daily returns
- Compare SPY with XOM
- Scatter plots
- Correlation is not slope!
- Compare SPY vs XOM, with SPY vs GLD scatter plots

Reading: "Python for Finance", Chapter 5: Data Visualization

## Lesson 7: Sharpe ratio & other portfolio statistics

- Speed up reading data by memoizing
- Average daily return
- Volatility: stddev of daily return (don't count first day)
- Cumulative return
- Relationship between cumulative and daily
- Sharpe Ratio
- How to model a buy and hold portfolio

## Lesson 8: Optimizers: Building a parameterized model

- What does an optimizer do?
- Syntax of optimizer use
- Problem statement for an optimizer (inputs, outputs, assumptions)
- How to find  $X$  that minimizes  $f(X)$  with a minimizer
- How to build a parameterized polynomial model from real data using an optimizer

## Lesson 9: Optimizers: How to optimize a portfolio

- What does it mean to "optimize" a portfolio
- Framing the problem for an optimizer
- Constraints on  $X$  for an optimizer
- Ranges on  $X$  for an optimizer

Reading: "Python for Finance", Chapter 11: Statistics-Portfolio Optimization

Retrieved from "[http://quantsoftware.gatech.edu/index.php?title=Manipulating\\_Financial\\_Data\\_in\\_Python&oldid=1351](http://quantsoftware.gatech.edu/index.php?title=Manipulating_Financial_Data_in_Python&oldid=1351)"

---

- This page was last modified on 24 August 2016, at 19:10.
- This page has been accessed 67,784 times.