Ph.D. Program in Consumers and Markets

Curriculum: *Finance, Markets, and Regulation*

PYTHON FOR FINANCE

PhD Course

Given by Kevyn Stefanelli | Professor in Finance at Università degli Studi della Tuscia – Dipartimento di Economia, Ingegneria, Società e Impresa (DEIM)

Total workload 24 hours (8 lectures of 3 hours each)

Time	Date	Day
15:00 – 18:00	**23 February**	Monday
15:00 – 18:00	**2 March**	Monday
15:00 – 18:00	**9 March**	Monday
15:00 – 18:00	**16 March**	Monday
15:00 – 18:00	**23 March**	Monday
15:00 – 18:00	**13 April**	Monday
15:00 – 18:00	**20 April**	Monday
15:00 – 18:00	**27 April**	Monday

Location Room 13B (first floor),
Department of Business Studies – Roma Tre University

Address Via Silvio D'Amico, 77, 00145 Rome, Italy – Google Maps

General Information

- **Teaching format:** hands-on lectures based on Jupyter Notebooks, combining live coding sessions, guided exercises, and interactive discussions. Each topic is developed through practical examples drawn from financial applications, with continuous feedback on implementation choices and results.
- **Expected output:** fully reproducible notebooks featuring clean, modular, and well-documented code. Students are expected to produce clearly commented results, including visualizations and quantitative performance metrics, with an emphasis on interpretability and best practices for research-oriented programming.

Learning Objectives

Upon successful completion of the course, PhD students will be able to:

- write solid and well-structured Python code, including data types, core data structures, functions, and control flow;
- effectively use the scientific Python stack (`NumPy`, `Pandas`, `Matplotlib`) for data manipulation, visualization, and empirical analysis;
- acquire, clean, and process financial time series (prices to returns, descriptive statistics, graphical analysis);
- apply core empirical tools for financial data analysis, including autocorrelation analysis, dependence diagnostics, and basic statistical testing;
- implement foundational computational tools for fixed income pricing, simulation-based risk measurement, and portfolio optimization.

Prerequisites

- Basic knowledge of probability and statistics, as well as introductory linear algebra.
- No prior experience in Python is required, although familiarity with programming concepts is beneficial.

Software and Libraries

- **Environment:** Python 3.x; Jupyter Notebook or JupyterLab (Anaconda distribution recommended).
- **Core libraries:** `numpy`, `pandas`, `matplotlib`, `yfinance`, `scipy`, `statsmodels`.

Assessment

- **Short exercises / problem sets (coding + commentary):** focused assignments designed to reinforce lecture material and promote good coding practices.
- **Final project (notebook + short report):** an applied empirical project involving data acquisition, model implementation, and critical interpretation of results.

Course Program (8 lectures × 3h)

Lecture 1 (3h) – Python essentials & Jupyter workflow

Contents

- Python language fundamentals: `print()`, variables and data types; logical and comparison operators;
- core objects and data structures (variables, lists, tuples, `DataFrame`);
- Jupyter Notebook workflow: cell execution order, experiment reproducibility, and workspace management;
- function definition and lambda functions;
- control flow:
  - conditional structures `if/elif/else`, boolean logic, composite and nested conditions;
  - loops (`for`, `while`) and use of `break`;
- introductory data visualization.

Learning outcomes

- correctly define and manipulate Python data types, objects, and operators;
- work in Jupyter environments in a structured, transparent, and reproducible way;
- write functions with clearly defined inputs and outputs;
- implement robust decision rules and control logical flows;
- visualize and interpret empirical results.

Lecture 2 (3h) – Structure of a scientific paper and stylized facts of financial returns

Contents

- overview of the structure of an empirical research paper, with discussion of major academic journals and data sources;
- analysis of the empirical properties of financial assets based on *Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. Quantitative Finance, 1(2), 223.*;
- guided replication in Python of the main stylized facts on financial time series.

Learning outcomes

- improve empirical research methodology in finance;
- understand, analyze, and replicate a scientific article;
- download and process financial time series using Python;
- export tables and results for professional academic output.

Lecture 3 (3h) – Asset pricing and factor models

Contents

- introduction to equity markets: indices, funds, and ETFs;
- Capital Asset Pricing Model (CAPM);
- factor models, with particular reference to:
  - *Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. Journal of Financial Economics, 33(1), 3–56.*
  - *Fama, E. F., & French, K. R. (2015). A five-factor asset pricing model. Journal of Financial Economics, 116(1), 1–22.*
  - *Carhart, M. M. (1997). On persistence in mutual fund performance. The Journal of Finance, 52(1), 57–82.*
  - *Frazzini, A., & Pedersen, L. H. (2014). Betting against beta. Journal of Financial Economics, 111(1), 1–25.*
  - *Asness, C. S., Frazzini, A., & Pedersen, L. H. (2019). Quality minus junk. Review of Accounting Studies, 24(1), 34–112.*
- factor data download, model estimation, and interpretation of results;
- guided extensions of the core material and critical discussion.

Learning outcomes

- build financial datasets from public sources and document the data pipeline;
- estimate asset pricing models and interpret results from an economic and financial perspective.

Lecture 4 (3h) – Volatility modelling

Contents

- GARCH models and their economic interpretation;
- parameter estimation via maximum likelihood;
- empirical replication in Python of:
  - *Moreira, A., & Muir, T. (2017). Volatility-managed portfolios. The Journal of Finance, 72(4), 1611–1644.*
  - *DeMiguel, V., Martín-Utrera, A., & Uppal, R. (2024). A Multifactor Perspective on Volatility-Managed Portfolios. The Journal of Finance, 79(6), 3859–3891.*
- in-class hands-on work in Python.

Learning outcomes

- model and interpret financial return volatility;
- replicate and extend empirical contributions from the literature;
- gain familiarity with econometric modelling in Python.

Lecture 5 (3h) – Fixed income: discounting, bond pricing, YTM, duration

Contents

- nominal and real interest rates; simple and compound interest; discounting and present value;
- pricing of zero-coupon and coupon bonds;
- bond characteristics (YTM, coupons, duration, embedded options);
- bond matching and discussion of *Zerbib, Olivier David. "The effect of pro-environmental preferences on bond prices: Evidence from green bonds." Handbook of Quantitative Sustainable Finance. Chapman and Hall/CRC, 2025. 410–448.*;
- term structure of interest rates: Cox–Ingersoll–Ross (CIR), Vasicek, Nelson–Siegel (NS), and Nelson–Siegel–Svensson (NSS) models;
- analysis of fixed-income datasets using Python.

Learning outcomes

- implement present value calculations and bond pricing in Python;
- estimate yield to maturity using numerical methods and assess its stability;
- implement and interpret mean-reverting interest rate models.

Lecture 6 (3h) – Simulation and risk: bootstrap, (m)VaR, backtesting concepts

Contents

- data-driven and model-driven simulation; Monte Carlo and Bootstrap;
- option pricing via simulation techniques;
- risk measures: Value at Risk (VaR) and Cornish–Fisher modified VaR;
- backtesting concepts (coverage) and interpretation of results;
- asymptotic statistical tests (*Ledoit, O., & Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. Journal of Empirical Finance, 15(5), 850–859.*)

Learning outcomes

- apply bootstrap techniques to estimate uncertainty and empirical distributions;
- implement VaR and mVaR measures and discuss their assumptions and limitations;
- design empirical validation exercises and interpret results correctly.

Lecture 7 (3h) – Portfolio optimization: EWP, GMVP, Markowitz, rolling analysis

Contents

- baseline strategies: Equally Weighted Portfolio (EWP);
- risk and diversification: covariance matrices and the Global Minimum Variance Portfolio (GMVP);
- mean–variance optimization à la Markowitz (risk aversion parameter γ);
- realistic constraints and numerical optimization (`scipy.optimize.minimize`);
- rolling-window comparison (in-sample / out-of-sample) and graphical reporting.

Learning outcomes

- construct and interpret EWP, GMVP, and mean–variance portfolios, analyzing the risk–return trade-off;
- solve constrained optimization problems in Python;
- evaluate investment strategies in a dynamic framework and present results clearly.

Lecture 8 (3h) – TBD

Contents

- the topic of the final lecture is intentionally left open to students' suggestions, in order to maximize the relevance and scientific usefulness of the course;
- proposed themes may include:
  - textual analysis and sentiment analysis;
  - networks and dynamic systems;
  - applications of Machine Learning;
  - other agreed-upon topics.

Note. The course structure and content may be adjusted depending on the background of the audience and specific requests for further in-depth discussion.