

Finance Theory and Data-Driven Finance

1142AIFQA05

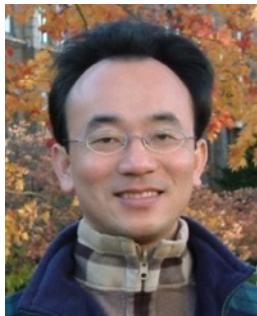
MBA, IM, NTPU (M5147) (Spring 2026)
Tue 5, 6, 7 (13:10-16:00) (B3F17)



<https://meet.google.com/miy-fbif-max>

 **NVIDIA**
University Ambassador
Certified Instructor

 **aws educate** | Cloud Ambassador
2020 Cohort



Min-Yuh Day, Ph.D,
Professor and Director

Institute of Information Management, National Taipei University

<https://web.ntpu.edu.tw/~myday>



Syllabus

Week	Date	Subject/Topics
1	2026/02/24	Introduction to Artificial Intelligence in Finance and Quantitative Analysis
2	2026/03/03	AI in FinTech: Metaverse, Web3, DeFi, NFT, Generative AI and Agentic AI for Financial Innovation Applications
3	2026/03/10	Investing Psychology and Behavioral Finance
4	2026/03/17	Event Studies in Finance
5	2026/03/24	Case Study on AI in Finance and Quantitative Analysis I
6	2026/03/31	Finance Theory and Data-Driven Finance

Syllabus

Week Date Subject/Topics

7 2026/04/07 Make-up holiday for NTPU Sports Day (No Classes)

8 2026/04/14 Midterm Project Report

9 2026/04/21 Financial Econometrics

10 2026/04/28 AI-First Finance

**11 2026/05/05 Industry Practices of AI in Finance and
Quantitative Analysis**

12 2026/05/12 Case Study on AI in Finance and Quantitative Analysis II

Syllabus

Week	Date	Subject/Topics
13	2026/05/19	Deep Learning in Finance; Reinforcement Learning in Finance; Generative AI and Agentic AI in Finance
14	2026/05/26	Algorithmic Trading; Risk Management; Trading Bot and Event-Based Backtesting
15	2026/06/02	Final Project Report I
16	2026/06/09	Final Project Report II

Financial Theories and Data-Driven Finance

Financial Theories and Data-Driven Finance

- **Financial Theories**
 - **Uncertainty and Risk**
 - **Expected Utility Theory (EUT)**
 - **Mean-Variance Portfolio Theory (MVPT)**
 - **Capital Asset Pricing Model (CAPM)**
 - **Arbitrage Pricing Theory (APT)**
- **Data-Driven Finance**
 - **Scientific Method**
 - **Financial Econometrics and Regression**
 - **Data Availability**

AI Agents in Financial Markets

Financial AI: from Rule-based Execution to Agentic Financial Systems



A Four-layer Architecture for AI Agents in Financial Markets

Layer 1: Data Perception

Market data; filings; news; social and macro signals; blockchain state; internal risk, compliance, and portfolio data.



Layer 2: Reasoning Engine

Domain LLMs, retrieval systems, forecasting models, optimisation modules, memory, ranking, and scenario analysis.



Layer 3: Strategy Generation

Trade ideas, allocation proposals, anomaly alerts, hedging recommendations, compliance flags, and explanatory narratives.



Layer 4: Execution and Control

OMS/EMS systems, APIs, smart contracts, approval workflows, limit frameworks, audit logs, monitoring, and emergency stop mechanisms.

Conceptual Structure of the Agentic Financial Market Model (AFMM)



Application Map for AI Agents in Finance

Use case	Typical agent tasks	Likely autonomy level	Main constraints
Autonomous trading	Signal synthesis, news interpretation, order preparation, post-trade analysis	Medium in narrow mandates	Latency, slippage, best execution, manipulation risk
Portfolio management	Macro synthesis, memo generation, allocation suggestions, scenario analysis	Low to medium	Fiduciary duty, mandate constraints, explainability
Risk and compliance	Alerting, anomaly detection, policy mapping, transaction surveillance	Medium for triage	False positives, accountability, due process
DeFi intelligence	Wallet tracing, governance monitoring, liquidity mapping, contract-event screening	Medium	Data noise, oracle risk, fragmented venues

AI Agents in Financial Markets

Data Structure for the Empirical Application

Construct	Illustrative proxy	Public or low-cost source
Market reaction	Daily adjusted returns, abnormal volume, turnover	Yahoo Finance or Alpha Vantage; CRSP, Refinitiv, or Bloomberg if available
Firm exposure	Keyword-based legacy-modernisation index from 10-K/10-Q filings	SEC EDGAR submissions and filing text
Sector classification	Banking, insurance, payments, software, infrastructure, consulting dummies	SEC metadata, company filings, exchange classifications
Labour exposure	Occupational exposure or task exposure linked to firm business lines	Anthropic observed-exposure framework, O*NET, BLS occupational data
Macro controls	VIX, Treasury yields, credit spreads, financial-conditions indices	FRED

Financial Theories

Financial Theories

- **Uncertainty and Risk**
- **Expected Utility Theory (EUT)**
- **Mean-Variance Portfolio Theory (MVPT)**
- **Capital Asset Pricing Model (CAPM)**
- **Arbitrage Pricing Theory (APT)**

Major Normative Financial Theories and Models

- Normative Theory

- Based on **assumptions (mathematically, axioms)** and **derives insights, results**, and more from **the set of relevant assumptions**.

- Positive theory

- Based on **observation, experiments, data, relationships**, and the like and **describes phenomena given the insights** gained from the available information and the derived results.

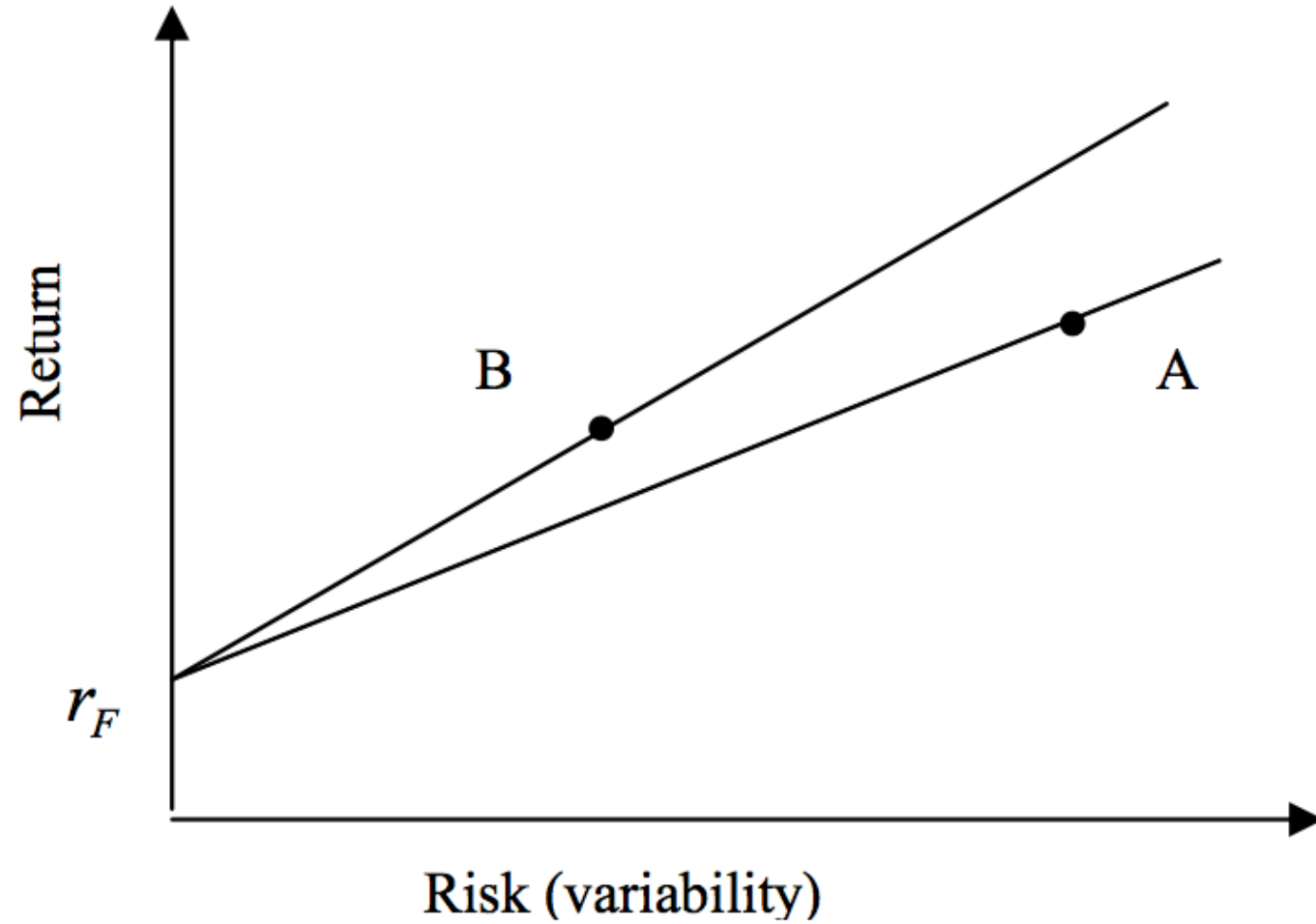
Normative Finance

- The **CAPM** is based on many unrealistic assumptions.
 - The assumption that investors care only about the mean and variance of one-period portfolio returns is extreme.
- (Eugene Fama and Kenneth French, 2004)**

Uncertainty and Risk

- Financial theory deals with investment, trading, and valuation in the presence of **uncertainty** and **risk**.
- The focus is on fundamental concepts from **probability theory** that build the backbone of **quantitative finance**.

Risk and Return



Sharpe Ratio

$$\text{Sharpe Ratio} = \frac{\text{Portfolio Return} - \text{Risk Free Return}}{\text{Portfolio Risk}}$$

Sharpe Ratio

$$\text{Sharpe Ratio } SR = \frac{r_P - r_F}{\sigma_P}$$

Where

r_P = portfolio return

r_F = risk free rate

σ_P = portfolio risk

(variability, standard deviation of return)

Sortino Ratio

$$\text{Sortino Ratio} = \frac{r_P - r_T}{\sigma_D}$$

Where

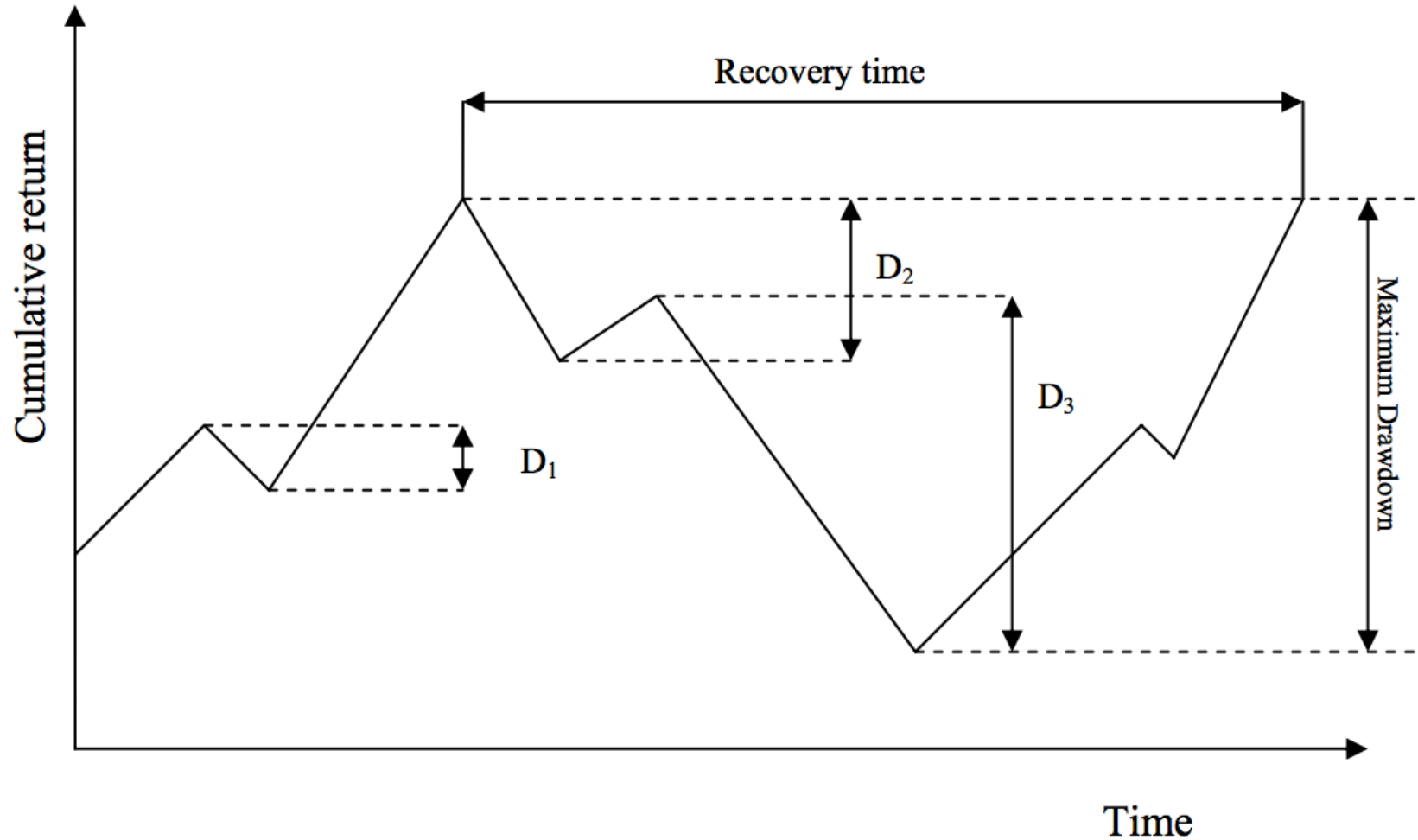
r_P = portfolio return

r_T = Minimum Target Return

σ_D = Downside Risk

$$\text{Downside Risk } \sigma_D = \sqrt{\frac{\sum_{i=1}^n \min[(r_i - r_T), 0]^2}{n}}$$

Max Drawdown



Traded Assets

- In the economy, **two assets** are traded.
 - The first is a **risky asset**, the **stock**, with a certain price today of $S_0 = 10$ and an uncertain payoff tomorrow.
 - The second is a **risk-less asset**, the **bond**, with a certain price today of $B_0 = 10$ and a certain payoff tomorrow.

Arbitrage Pricing

- Deriving the fair value of a European call option on the stock with a strike price of $K = 14.5$
- **Arbitrage pricing theory** can be considered one of the strongest financial theories with some of the most robust mathematical results, such as the **fundamental theorem of asset pricing (FTAP)**.

Expected Utility Theory (EUT)

- **Expected utility theory (EUT)**
 - **1940s**
 - **cornerstone of financial theory**
 - **One of the central paradigms for modeling decision making under uncertainty**

Expected Utility Theory

- **Expected utility theory (EUT)**
 - **EUT is an axiomatic theory**
 - von Neumann and Morgenstern (1944)
 - **Axiomatic**
 - Major results of the theory can be deduced from a small number of axioms only
- **Axioms and normative theory**
 - **An axiom is a proposition regarded as self-evidently true without proof.**

Preferences of an Agent

- Assume an agent with preferences \succeq is faced with the problem of investing in the two traded assets of the model economy M^2 .
- One possible set of axioms leading to EUT
 - Completeness
 - Transitivity
 - Continuity
 - Independence
 - Dominance

Utility functions

- A **utility function** is a way to represent the **preferences \succeq of an agent** in a mathematical and numerical way in that such a function assigns a numerical value to a certain payoff.

Expected Utility Functions

- **Von Neumann and Morgenstern (1944) show that if the preferences of an agent \succeq satisfy the preceding five axioms, then there exists an expected utility function.**

Risk aversion

- In finance, the concept of **risk aversion** is important.
- The most commonly used measure of risk aversion is the Arrow-Pratt measure of **absolute risk aversion (ARA)** (Pratt, 1964).
 - $ARA(x) > 0$, risk-averse
 - $ARA(x) = 0$, risk-neutral
 - $ARA(x) < 0$, risk-loving

Mean-Variance Portfolio Theory (MVPT)

- **Mean-variance portfolio (MVP) theory**
 - Markowitz (1952)
 - cornerstone in financial theory
- One of the **first theories of investment under uncertainty** that focused on statistical measures only for the construction of stock **investment portfolios**.
- MVP completely abstracts from fundamentals of a company that might drive its stock performance or assumptions about the future competitiveness of a company that might be important for the growth prospects of a company.

Mean-Variance Portfolio Theory (MVPT)

- The only input data that counts is the time series of share prices and statistics derived therefrom, such as the (historical) **annualized mean return** and the (historical) **annualized variance of the returns**.
- The central assumption of MVP, according to Markowitz (1952), is that investors only care about **expected returns** and the **variance of these returns**.

Mean-Variance Portfolio Theory (MVPT)

- **Portfolio statistics**
 - **returns vector**
 - **expected return**
 - **vector of expected returns**
 - **expected return of the portfolio**
 - **covariance matrix**
 - **expected variance of the portfolio**
 - **expected volatility of the portfolio**

Sharpe ratio

- Sharpe (1966) introduces a measure to judge the risk-adjusted performance of mutual funds and other portfolios, or even single risky assets.
- It relates the (expected, realized) return of a portfolio to its (expected, realized) volatility.

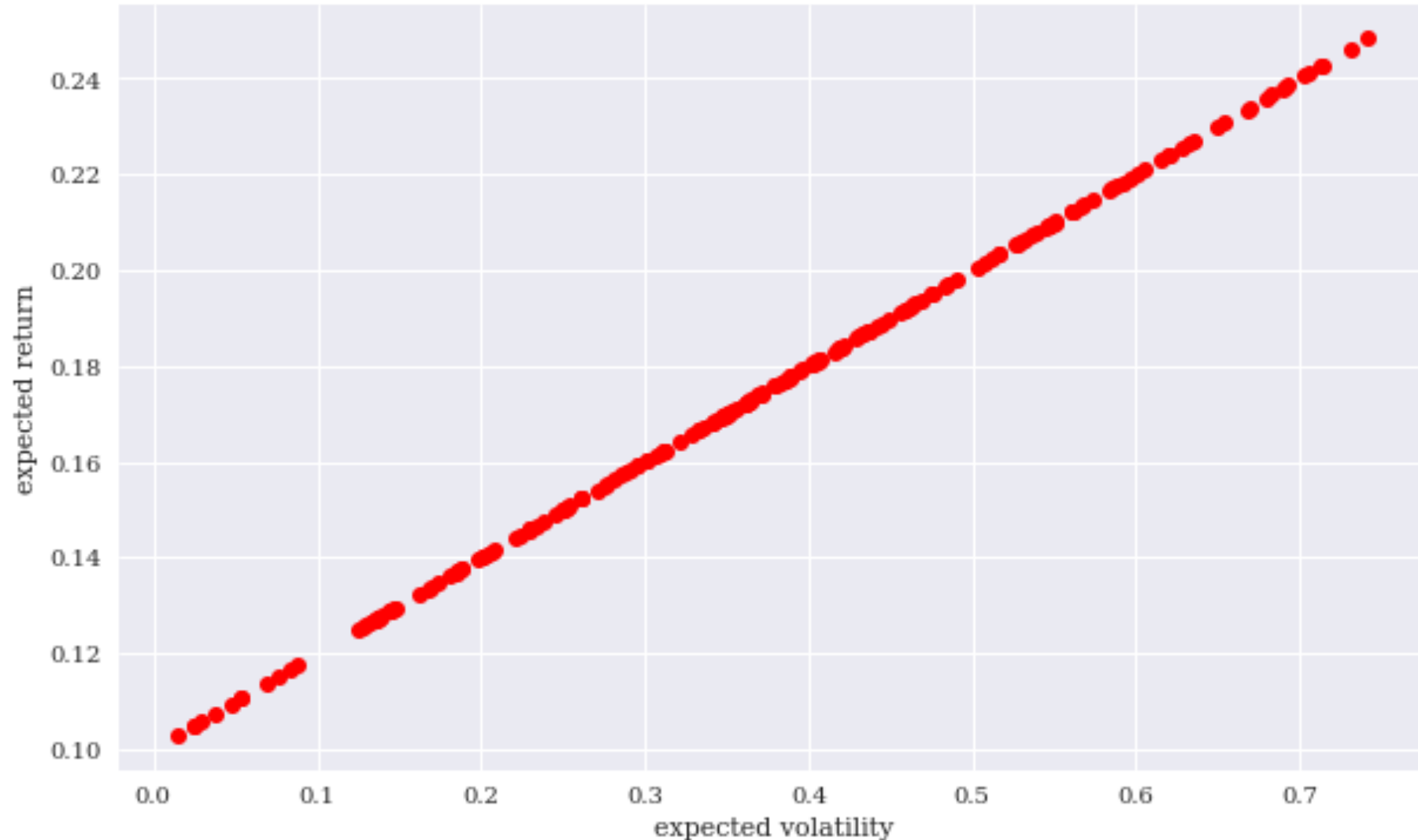
- Sharpe ratio $\pi = \frac{\mu}{\sigma}$

- If r represents the risk-less short rate, the **risk premium** or **excess return** of a portfolio ϕ over a risk-free alternative is defined by $\mu^{\phi} - r$

- Sharpe ratio $\pi = \frac{\mu - r}{\sigma}$

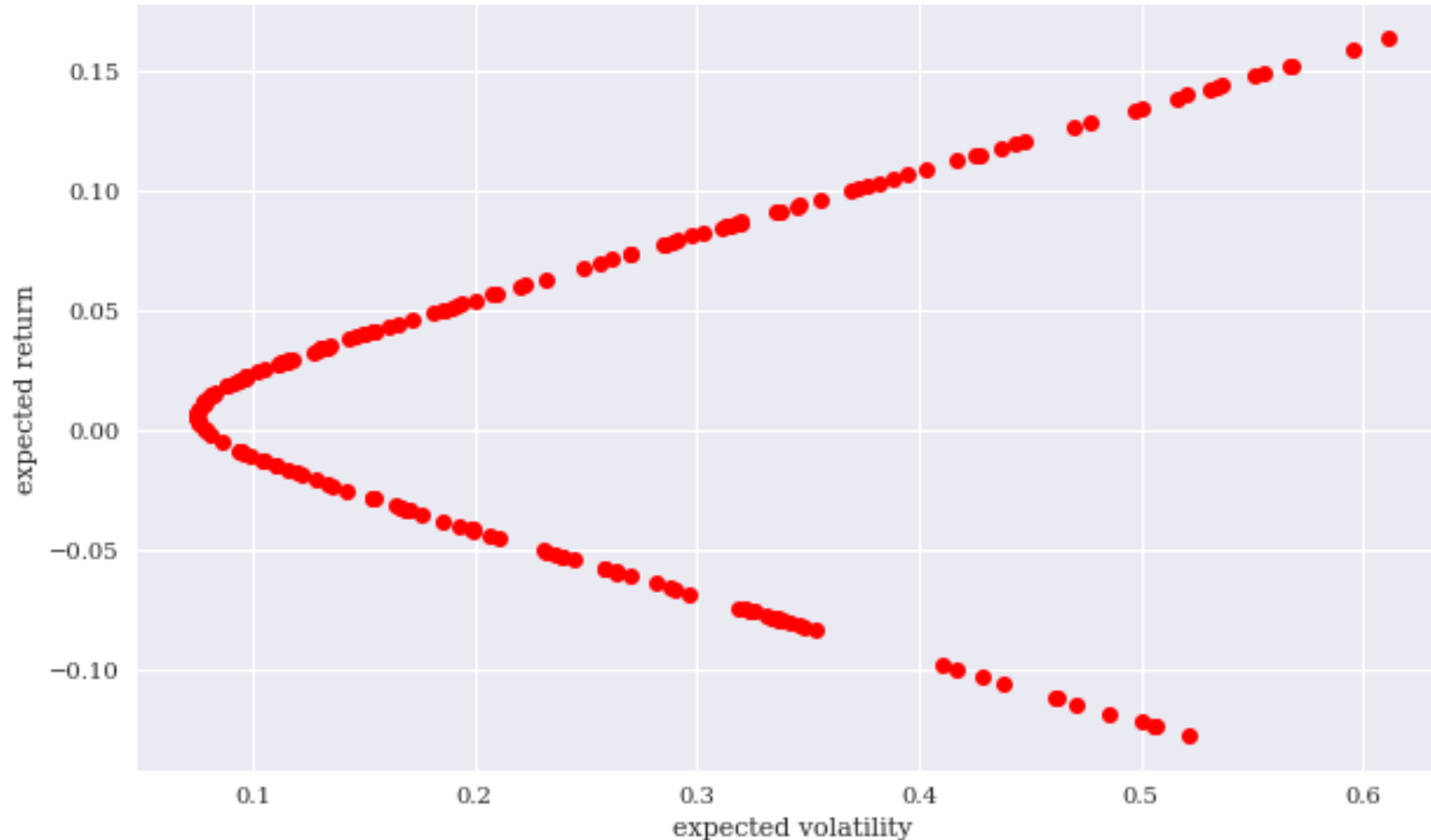
Investment Opportunity Set

Simulated expected portfolio volatility and return (one risky asset)

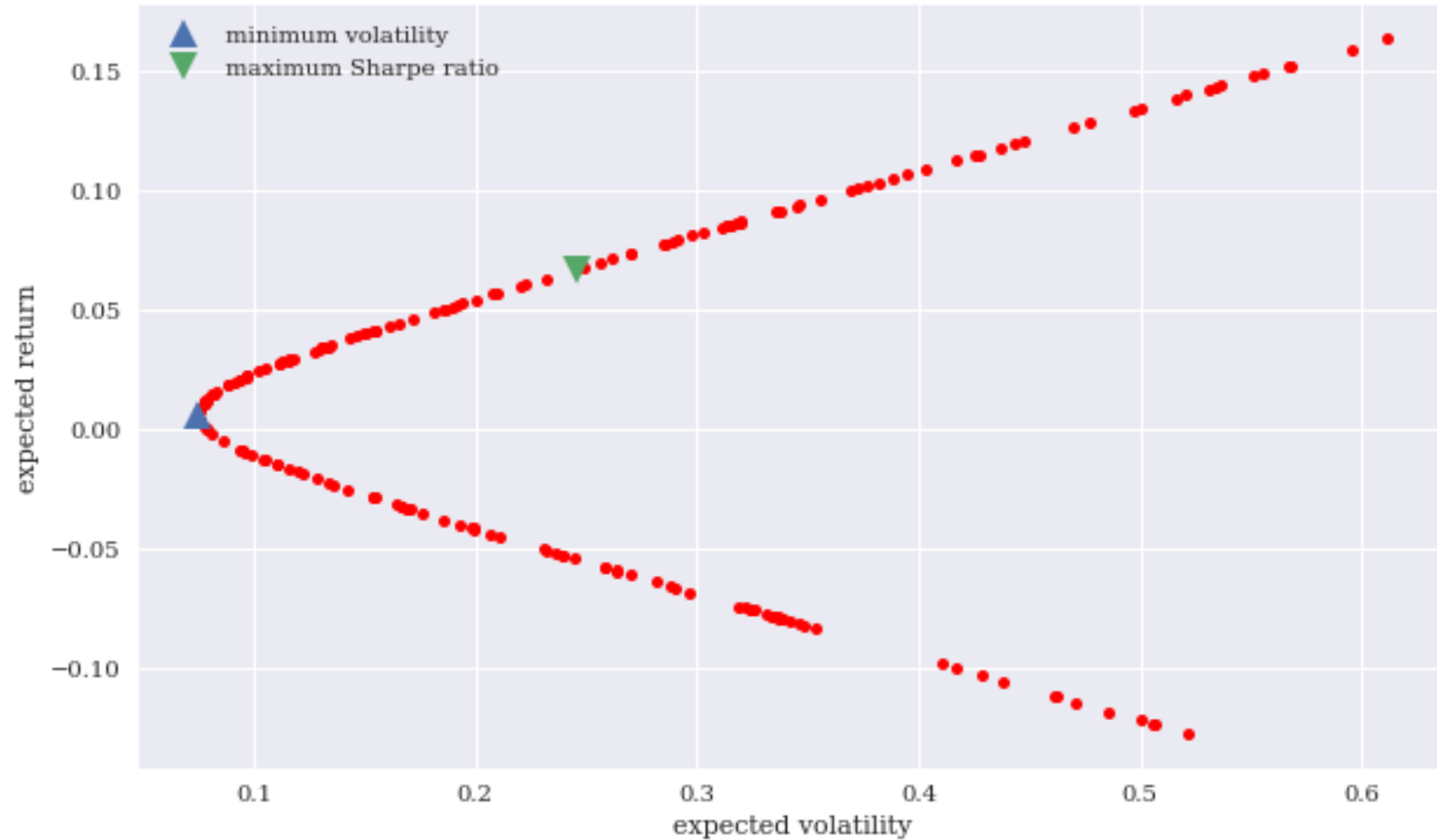


Investment Opportunity Set

Simulated expected portfolio volatility and return (two risky assets)



Minimum volatility and maximum Sharpe ratio portfolios

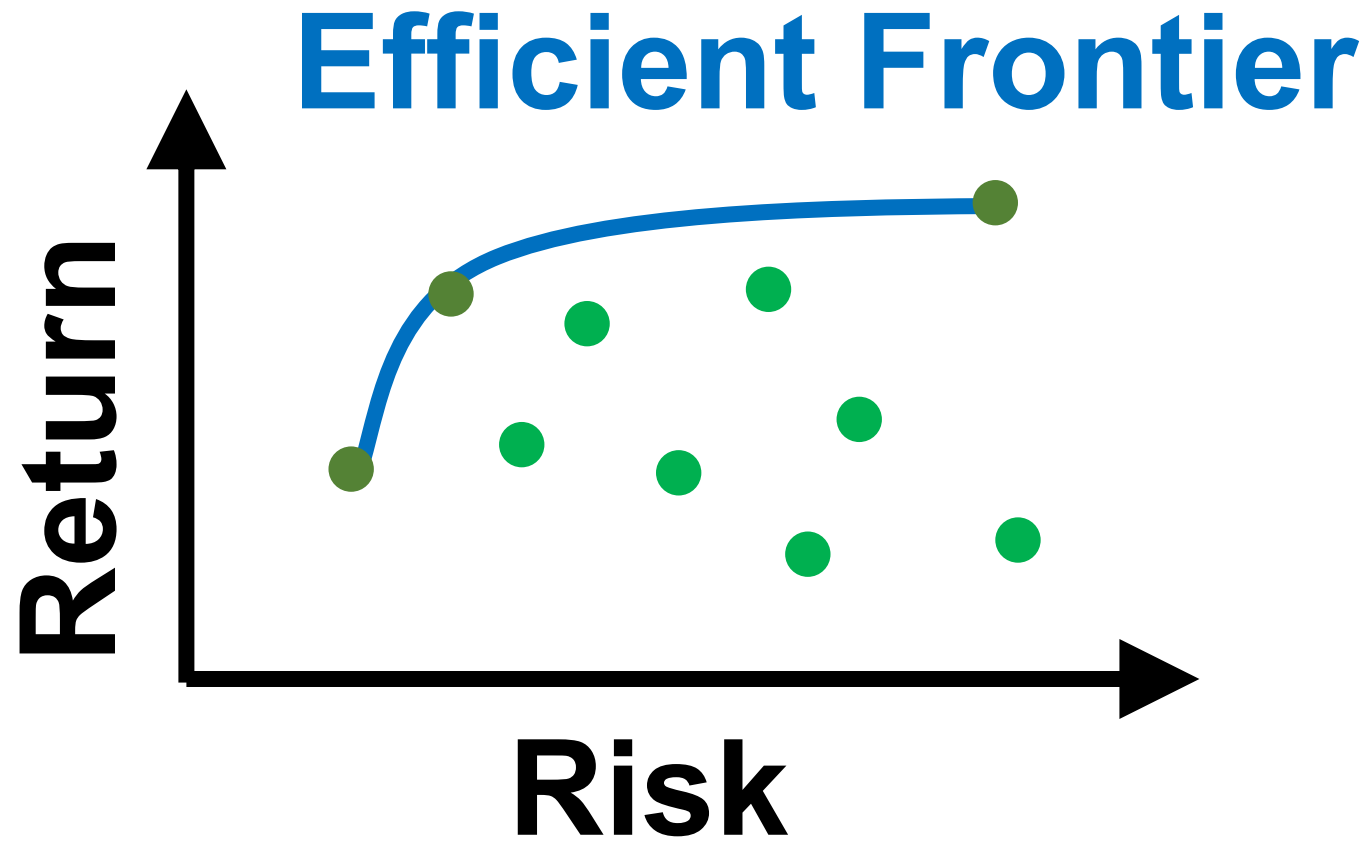


Efficient Frontier

- An **efficient portfolio**
 - has a maximum expected return (risk) given its expected risk (return)
- All those portfolios that have a **lower expected return** than the **minimum risk portfolio** are **inefficient**.
- **Efficient frontier**
 - The set of **all efficient portfolios**
 - Agents will only choose **a portfolio** that lies on the efficient frontier

Portfolio Optimization

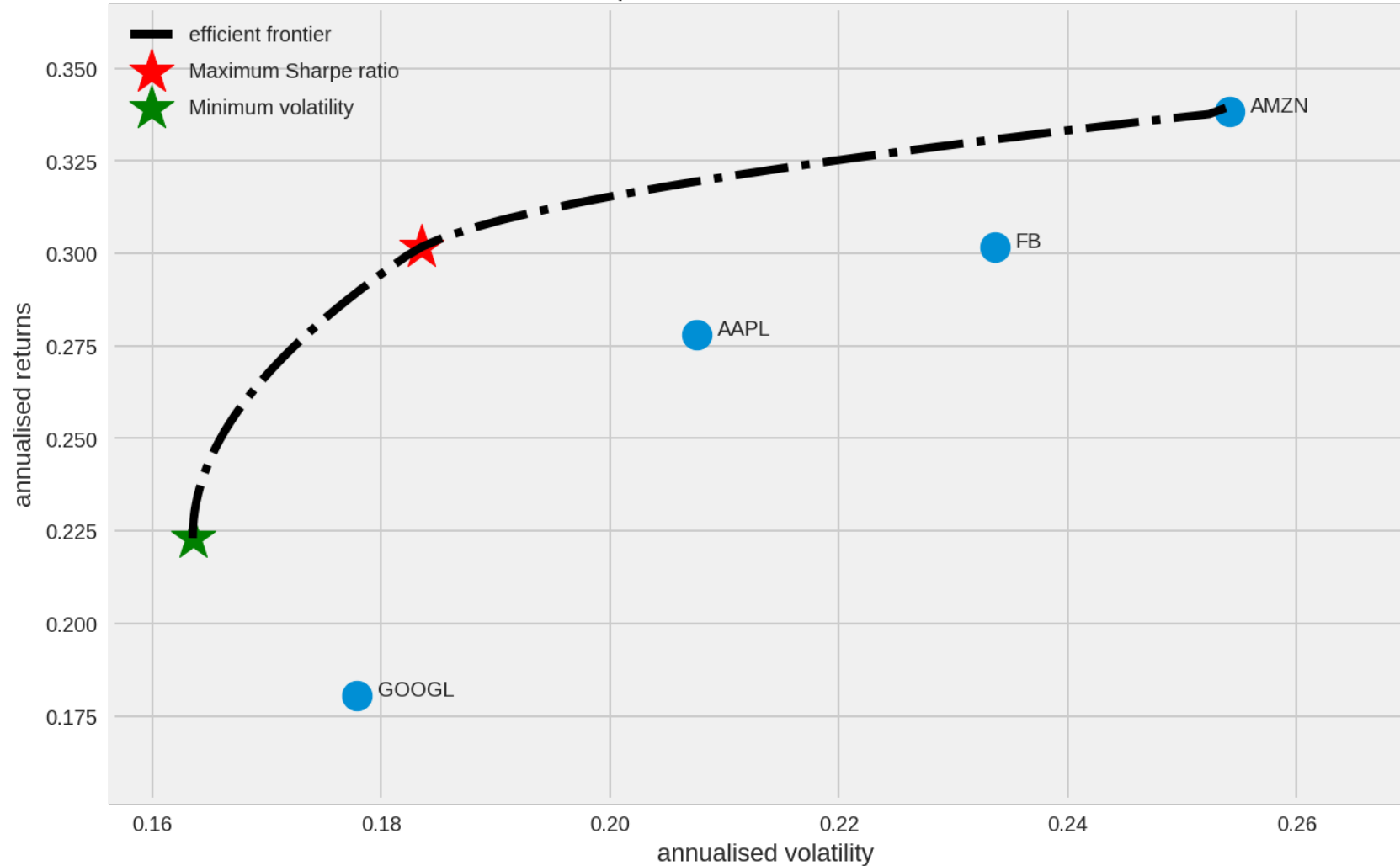
Efficient Frontier



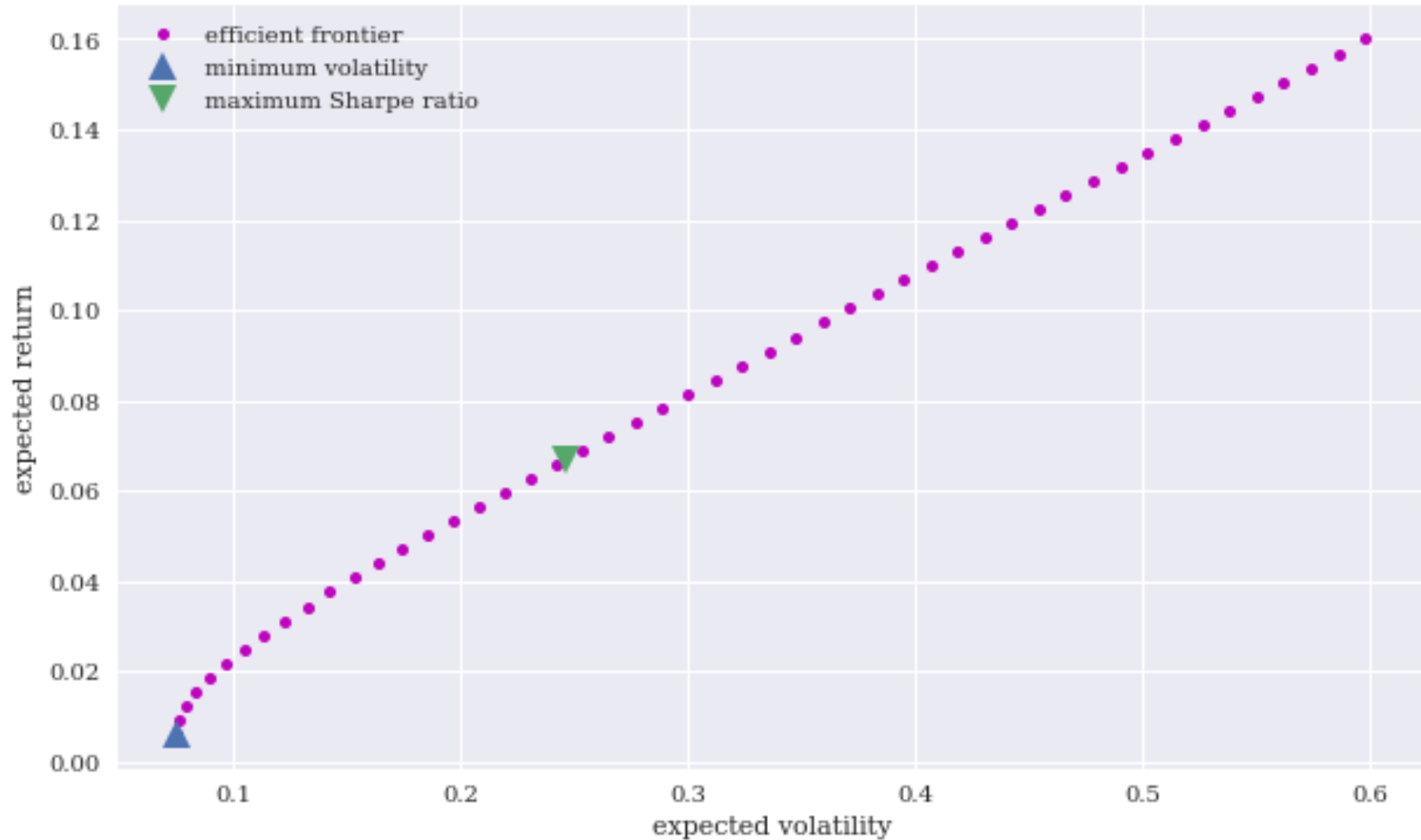
Portfolio Optimization

Efficient Frontier

Portfolio Optimization with Individual Stocks



Efficient Frontier



Portfolio Optimization and Algorithmic Trading

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

RAM Disk Editing

Table of contents

- Python101
- Python File Input / Output
- OS, IO, files, and Google Drive
- Python Programming
- Python String and Text
- Python Numpy
- Python Pandas
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python**
 - Investment Portfolio Optimization
- Text Analytics and Natural Language Processing (NLP)
 - Python for Natural Language Processing
 - spaCy Chinese Model
 - Open Chinese Convert (OpenCC, 開放中文轉換)
 - Jieba 結巴中文分詞
 - Natural Language Toolkit (NLTK)
 - Stanza: A Python NLP Library for Many Human Languages

```
Annualised Return: 0.19
Annualised Volatility: 0.18

      AAPL  AMZN  FB  GOOGL
allocation 44.67 29.05 26.28 0.0
```

Minimum Volatility Portfolio Allocation

```
Annualised Return: 0.22
Annualised Volatility: 0.16

      AAPL  AMZN  FB  GOOGL
allocation 34.02 0.73 6.98 58.26
```

Calculated Portfolio Optimization based on Efficient Frontier

Legend:

- efficient frontier
- Maximum Sharpe ratio
- Minimum volatility

<https://tinyurl.com/aintpupython101>

Capital Asset Pricing Model (CAPM)

- Capital Asset Pricing Model (CAPM)
 - One of the most widely documented and applied models in finance
 - It relates in linear fashion the expected return for **a single stock** to the expected return of the **market portfolio**, usually approximated by a broad stock index such as the S&P 500.
 - Sharpe (1964) and Lintner (1965)

Capital Asset Pricing Model (CAPM)

[Capital asset prices: A theory of market equilibrium under conditions of risk](#)
[WF Sharpe - The journal of finance, 1964 - Wiley Online Library](#)

ONE OF THE PROBLEMS which has plagued those attempting to predict the behavior of capital markets is the absence of a body of positive microeconomic theory dealing with conditions of risk. Although many useful insights can be obtained from the traditional models of investment under conditions of certainty, the pervasive influence of risk in financial transactions has forced those working in this area to adopt models of price behavior which are little more than assertions. A typical classroom explanation of the determination of ...

[Cite](#) [Cited by 30490](#) [Related articles](#) [All 26 versions](#)

Capital Asset Pricing Model (CAPM)

The Journal of FINANCE

VOL. XIX

SEPTEMBER 1964

No. 3

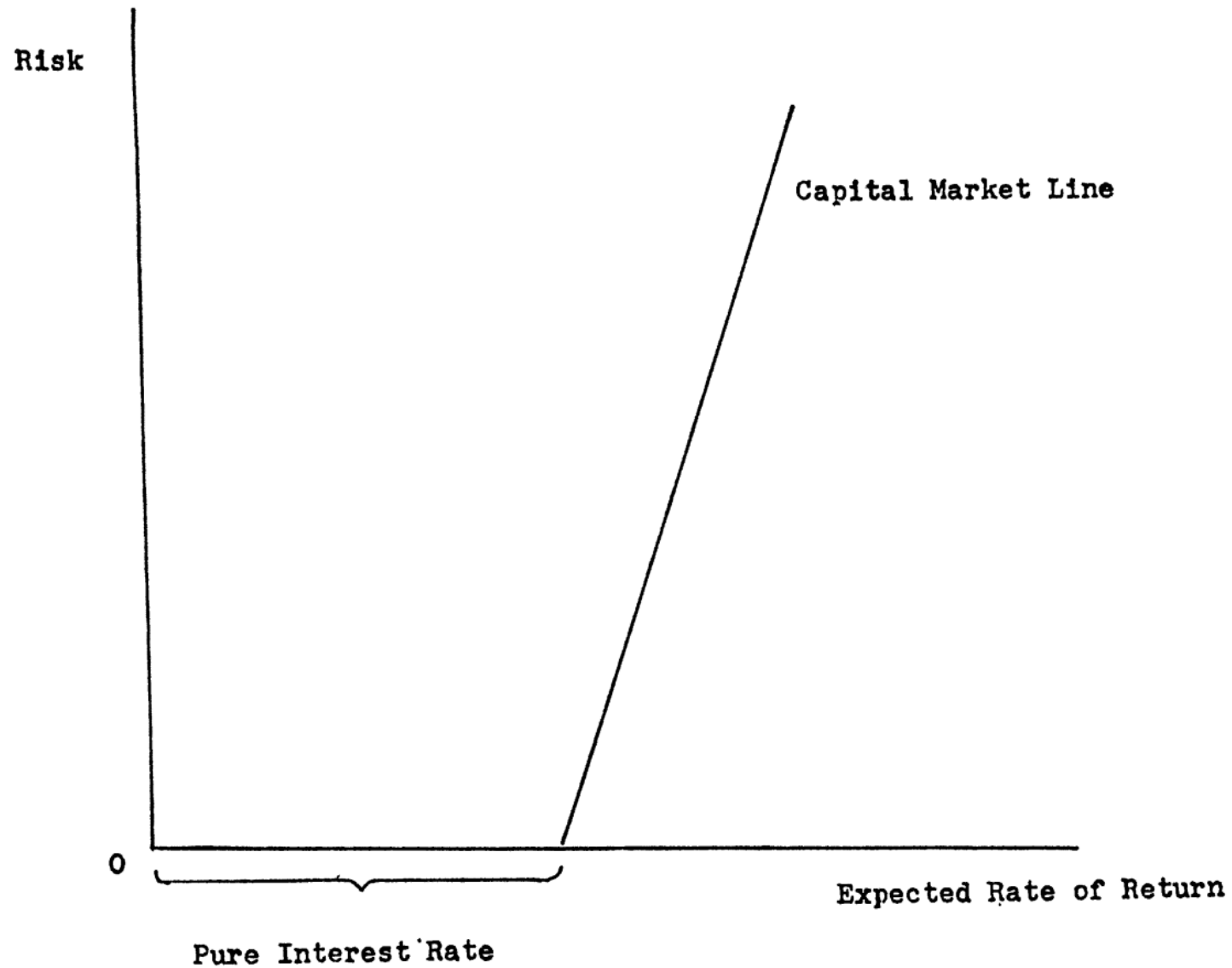
CAPITAL ASSET PRICES: A THEORY OF MARKET EQUILIBRIUM UNDER CONDITIONS OF RISK*

WILLIAM F. SHARPE†

I. INTRODUCTION

ONE OF THE PROBLEMS which has plagued those attempting to predict the behavior of capital markets is the absence of a body of positive micro-economic theory dealing with conditions of risk. Although many useful insights can be obtained from the traditional models of investment under conditions of certainty, the pervasive influence of risk in financial transactions has forced those working in this area to adopt models of price behavior which are little more than assertions. A typical classroom explanation of the determination of capital asset prices, for example, usually begins with a careful and relatively rigorous description of the process through which individual preferences and physical relationships interact to determine an equilibrium pure interest rate. This is generally followed by the assertion that somehow a market risk-premium is also determined, with the prices of assets adjusting accordingly to account for differences in their risk.

Capital Asset Pricing Model (CAPM)



Capital Asset Pricing Model (CAPM)

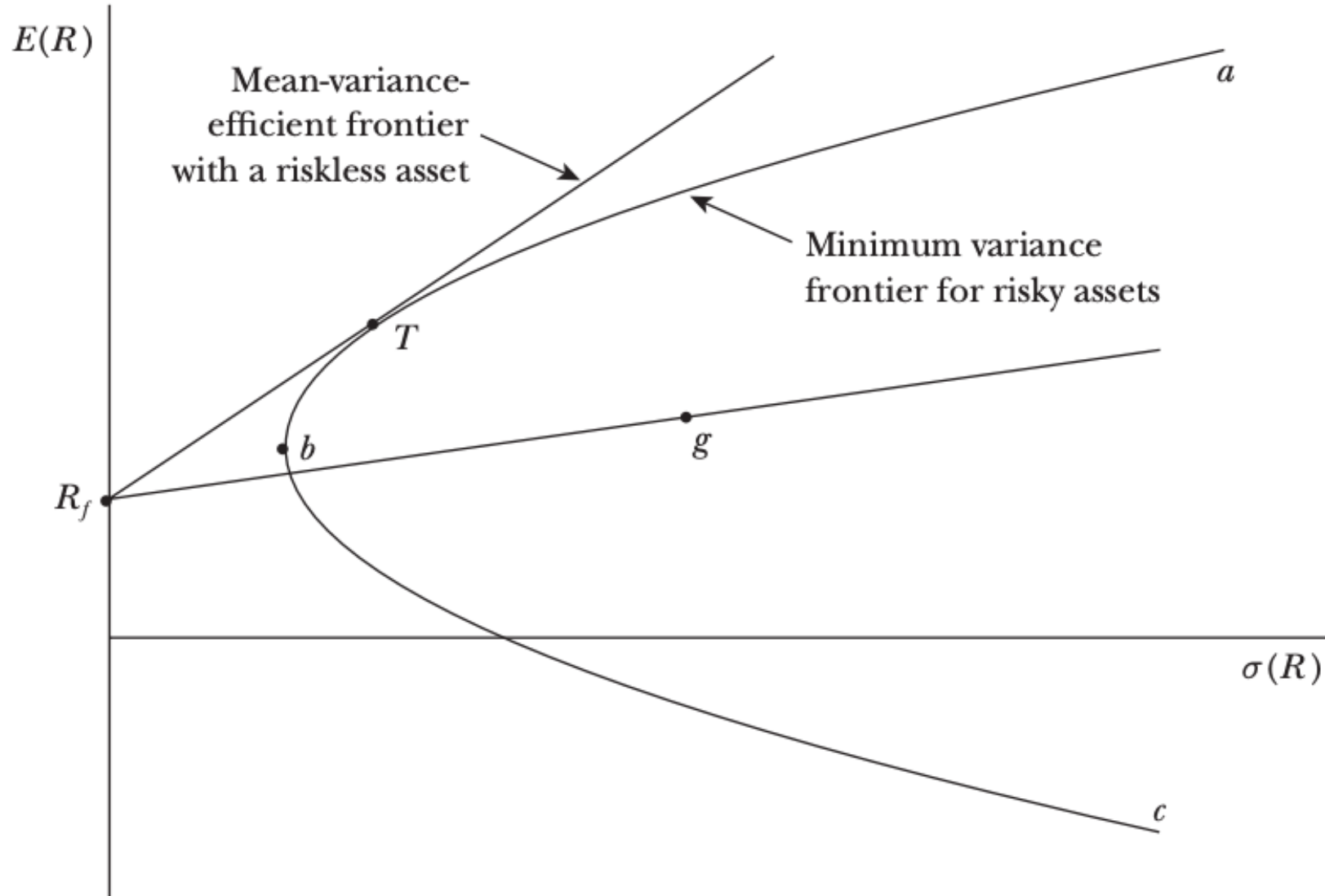
Journal of Economic Perspectives—Volume 18, Number 3—Summer 2004—Pages 25–46

The Capital Asset Pricing Model: Theory and Evidence

Eugene F. Fama and Kenneth R. French

The capital asset pricing model (CAPM) of William Sharpe (1964) and John Lintner (1965) marks the birth of asset pricing theory (resulting in a Nobel Prize for Sharpe in 1990). Four decades later, the CAPM is still widely used in applications, such as estimating the cost of capital for firms and evaluating the performance of managed portfolios. It is the centerpiece of MBA investment courses. Indeed, it is often the only asset pricing model taught in these courses.¹

Investment Opportunities

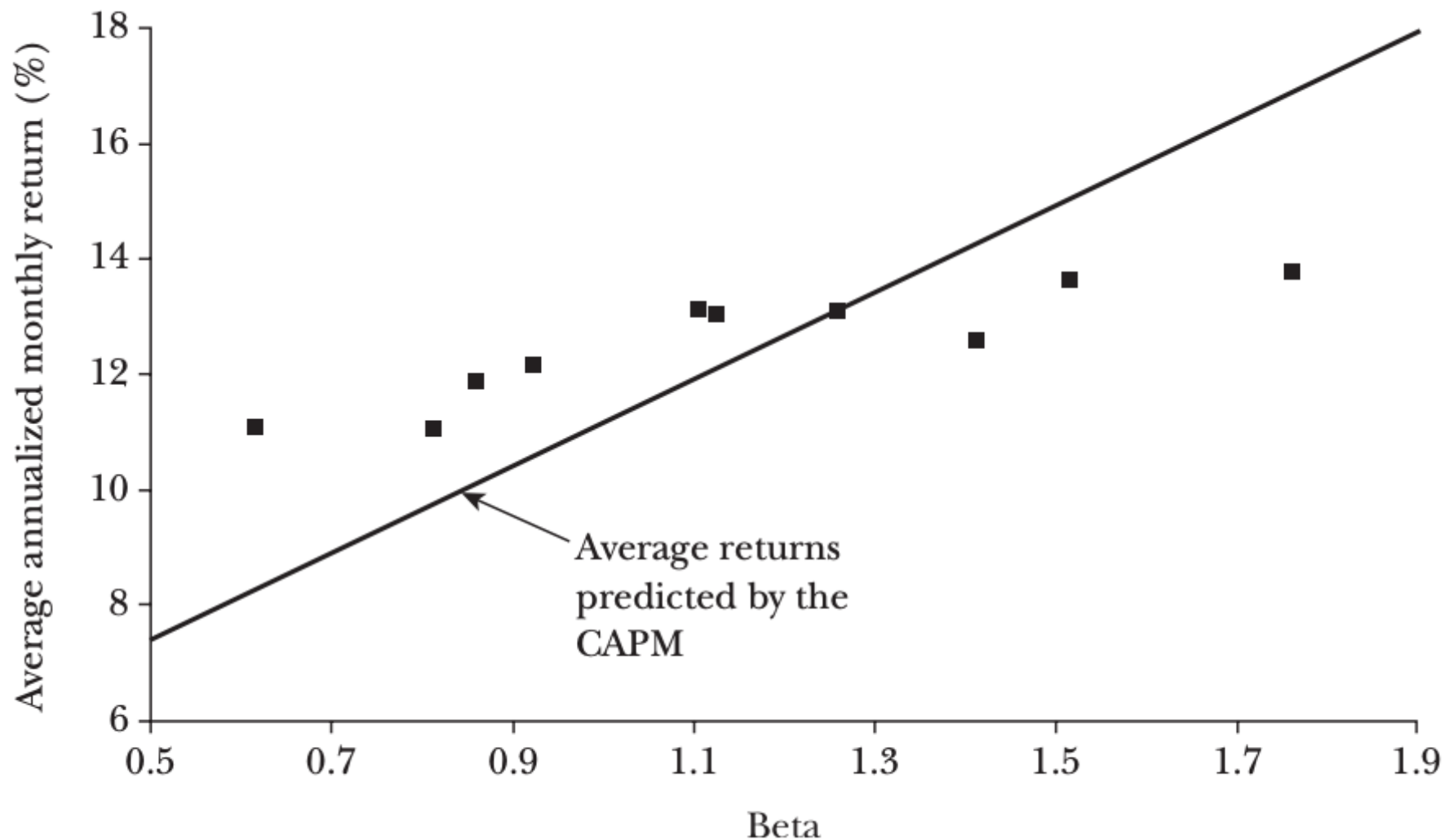


Capital Asset Pricing Model (CAPM)

$$E(R_i) = R_f + \beta_{iM} [E(R_M) - R_f]$$

The expected return on any asset i is the risk-free interest rate, R_f , plus a risk premium, which is the asset's market beta, β_{iM} , times the premium per unit of beta risk, $E(R_M) - R_f$.

Capital Asset Pricing Model (CAPM)



Average Annualized Monthly Return versus Beta for Value Weight Portfolios Formed on Prior Beta, 1928–2003

Capital Asset Pricing Model (CAPM)

- **Capital market theory** is a **positive theory** in that it hypothesizes how investors do behave rather than how investors should behave, as in the case of **modern portfolio theory (MVP)**
 - It is reasonable to view capital market theory as an extension of portfolio theory, but it is important to understand that MVP is not based on the **validity**, or lack thereof, of capital market theory.

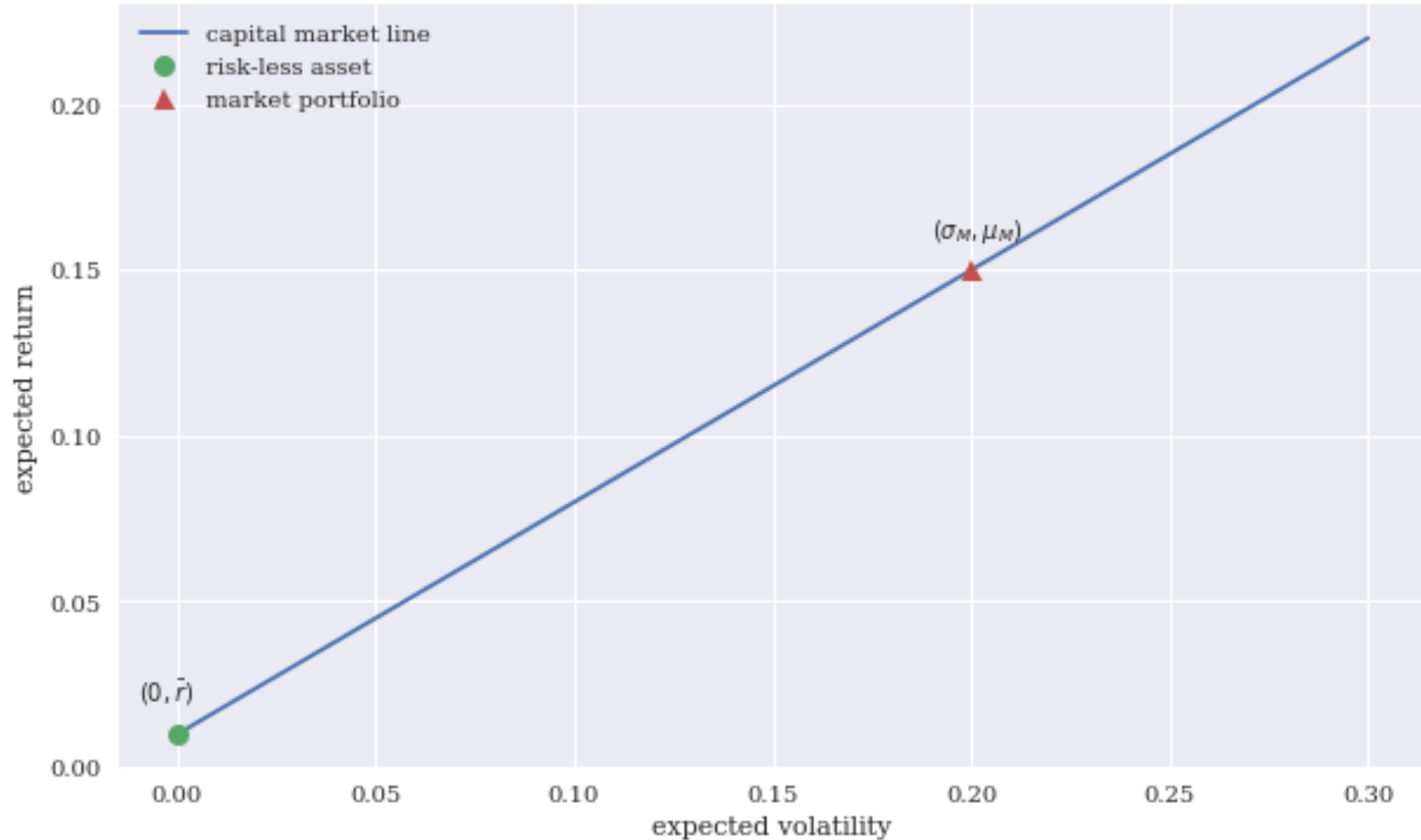
Capital Asset Pricing Model (CAPM)

- **The specific equilibrium model** of interest to many investors is known as the **capital asset pricing model**, typically referred to as the **CAPM**.
 - It allows us to **assess the relevant risk of an individual security** as well as to **assess the relationship between risk and the returns** expected from investing.
 - The CAPM is attractive as an **equilibrium model** because of its simplicity and its implications.

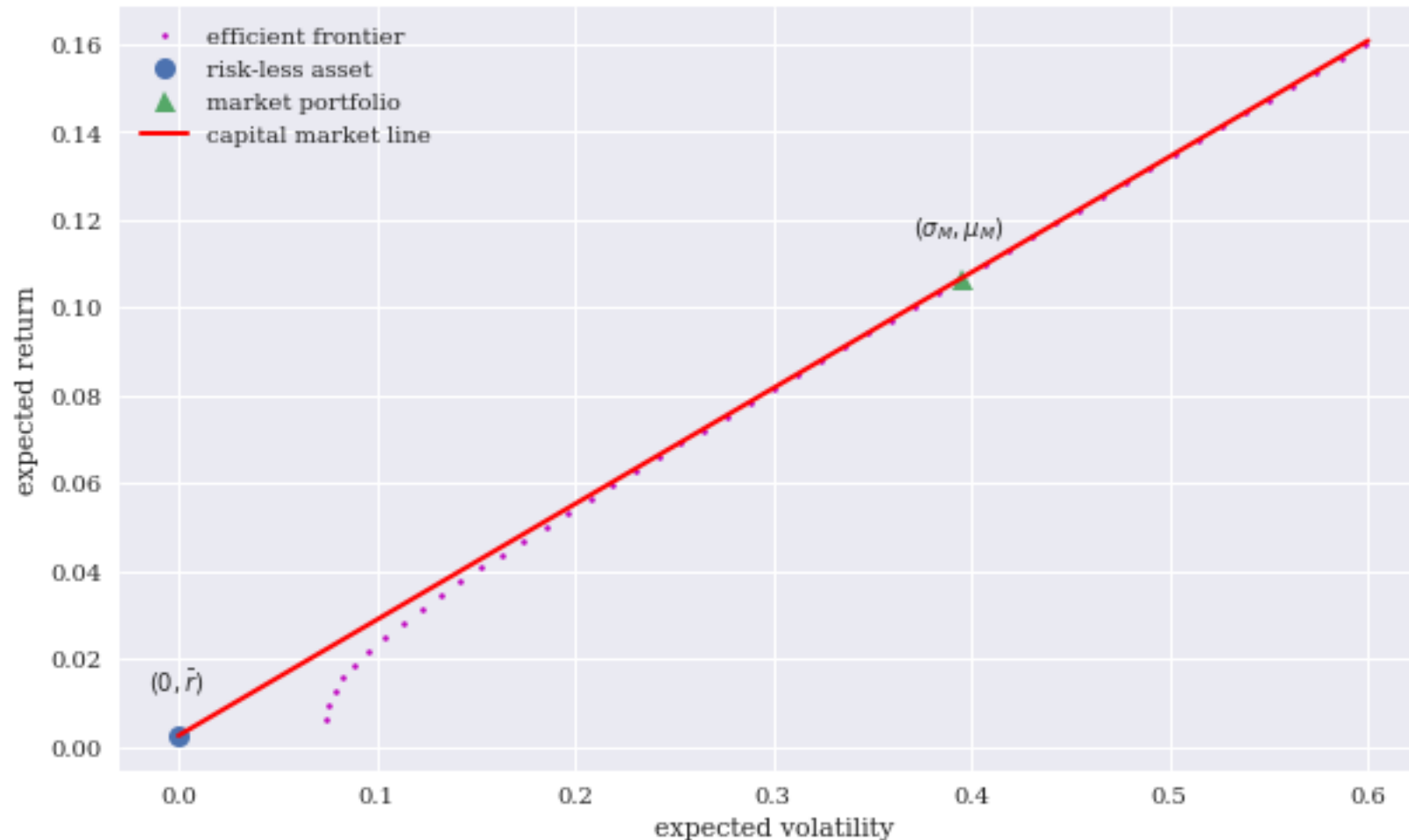
Capital Asset Pricing Model (CAPM)

- In the CAPM, agents are assumed to invest according to MVP, caring only about the **risk** and **return** statistics of risky assets over one period.
- In a **capital market equilibrium**, all available assets are held by all agents and the markets clear.
- **Market portfolio (set of tradable assets)** must lie on the **efficient frontier**.
- Two fund separation theorem
 - Every agent will hold a combination of the market portfolio and the risk-free asset in equilibrium.
 - The set of all such portfolios is called the **Capital Market Line (CML)**.

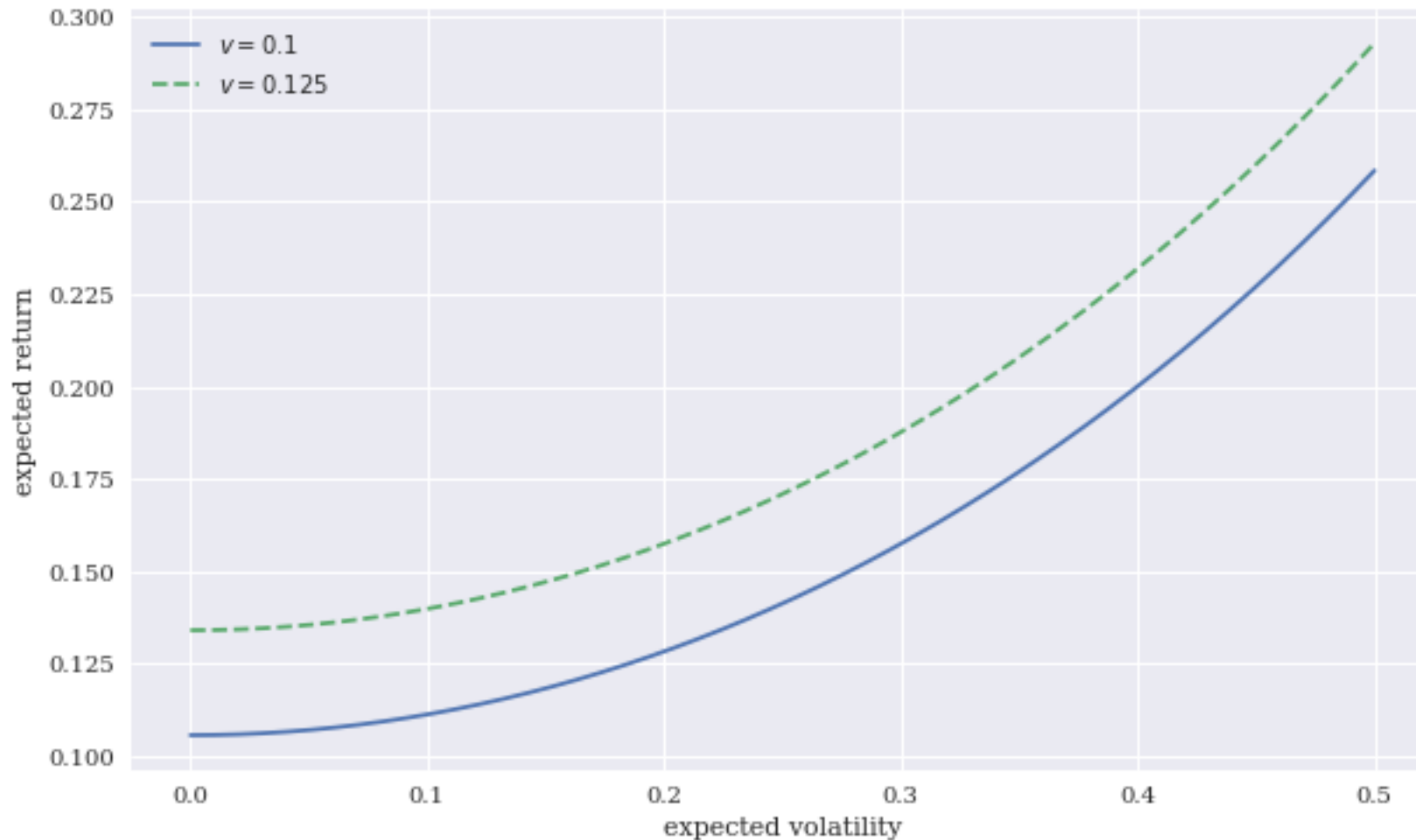
Capital Market Line (CML)



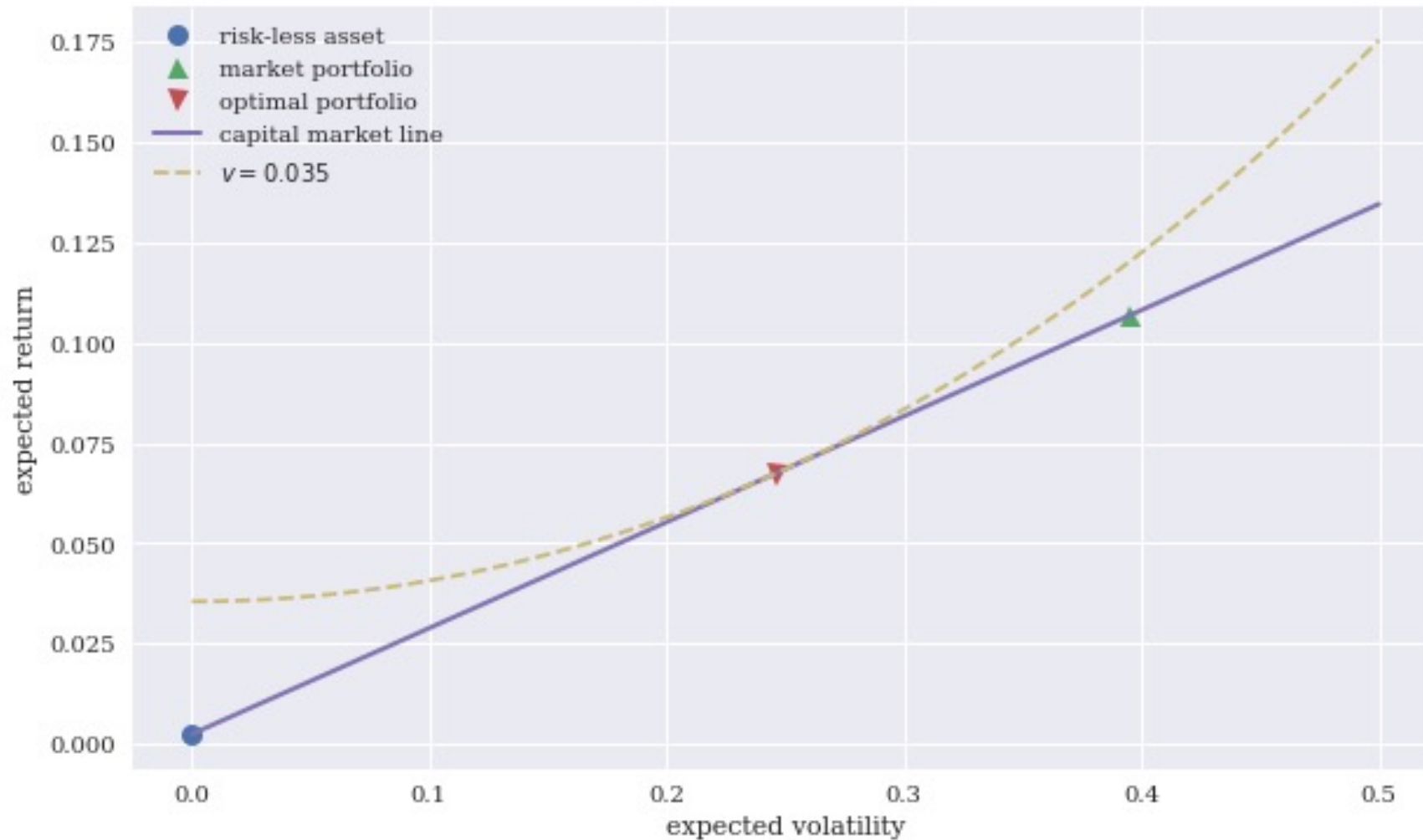
Capital Market Line with Two Risky Assets



Indifference curves in risk-return space



Optimal Portfolio on the Capital Market Line (CML)



Arbitrage Pricing Theory (APT)

- **Arbitrage Pricing Theory (APT)**
 - One of the **major generalizations of the Capital Asset Pricing Model (CAPM)**
 - **Ross (1971) and Ross (1976)**
 - **The purpose of this paper is to examine rigorously the **arbitrage model** of capital asset pricing developed in Ross (1971).**
 - The **arbitrage model** was proposed as an alternative to the **mean variance capital asset pricing model**, introduced by Sharpe, Lintner, and Treynor, that has become the major analytic tool for explaining phenomena observed in capital markets for risky assets.

Arbitrage Pricing Theory (APT)

JOURNAL OF ECONOMIC THEORY 13, 341–360 (1976)

The Arbitrage Theory of Capital Asset Pricing

STEPHEN A. ROSS*

*Departments of Economics and Finance, University of Pennsylvania,
The Wharton School, Philadelphia, Pennsylvania 19174*

Received March 19, 1973; revised May 19, 1976

The purpose of this paper is to examine rigorously the arbitrage model of capital asset pricing developed in Ross [13, 14]. The arbitrage model was proposed as an alternative to the mean variance capital asset pricing model, introduced by Sharpe, Lintner, and Treynor, that has become the major analytic tool for explaining phenomena observed in capital markets for risky assets. The principal relation that emerges from the mean variance model holds that for any asset, i , its (ex ante) expected return

$$E_i = \rho + \lambda b_i, \quad (1)$$

Arbitrage Pricing Theory (APT)

- The APT is a **generalization** of the CAPM to **multiple risk factors**.
- APT does not assume that the **market portfolio** is the **only relevant risk factor**
 - There are rather multiple types of risk that together are assumed to drive the performance (expected returns) of a stock.
 - Such risk factors might include **size**, **volatility**, **value**, and **momentum**.

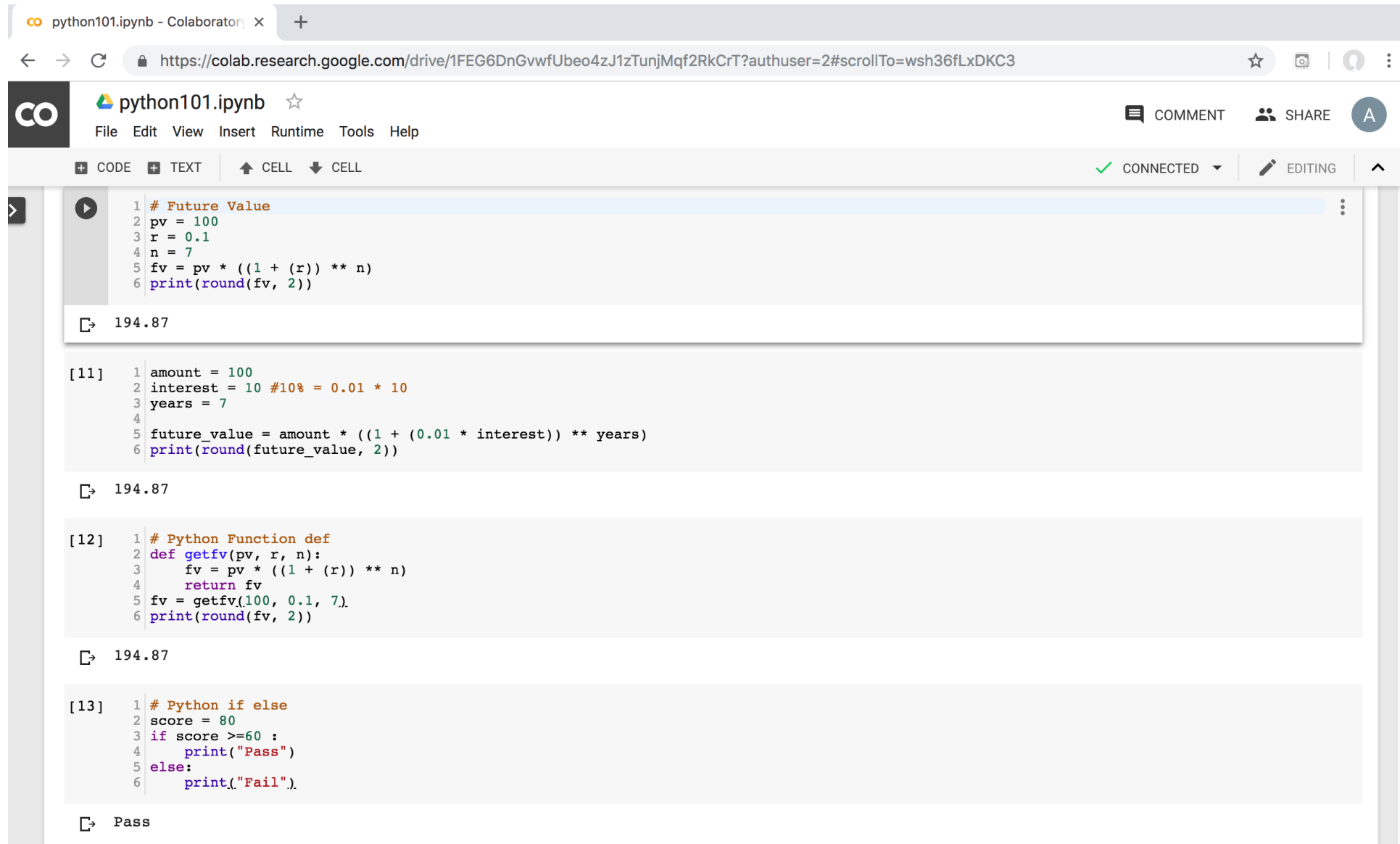
Capital Asset Pricing Model (CAPM)

Arbitrage Pricing Theory (APT)

- **Capital Asset Pricing Model (CAPM)**
 - **univariate ordinary least-squares (OLS) regression**
- **Arbitrage Pricing Theory (APT)**
 - **multivariate ordinary least-squares (OLS) regression**

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: <https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT?authuser=2#scrollTo=wsh36fLxDKC3>. The notebook title is "python101.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with options for CODE, TEXT, CELL, and a status indicator showing "CONNECTED" and "EDITING".

The notebook contains four code cells, each followed by its output:

```
1 # Future Value
2 pv = 100
3 r = 0.1
4 n = 7
5 fv = pv * ((1 + (r)) ** n)
6 print(round(fv, 2))
```

Output: 194.87

```
[11] 1 amount = 100
2 interest = 10 #10% = 0.01 * 10
3 years = 7
4
5 future_value = amount * ((1 + (0.01 * interest)) ** years)
6 print(round(future_value, 2))
```

Output: 194.87

```
[12] 1 # Python Function def
2 def getfv(pv, r, n):
3     fv = pv * ((1 + (r)) ** n)
4     return fv
5 fv = getfv(100, 0.1, 7)
6 print(round(fv, 2))
```

Output: 194.87

```
[13] 1 # Python if else
2 score = 80
3 if score >=60 :
4     print("Pass")
5 else:
6     print("Fail").
```

Output: Pass

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

The screenshot shows a Google Colab notebook titled "python101.ipynb". The interface includes a top navigation bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help" menus, along with "Comment", "Share", and "Settings" icons. A "Table of contents" sidebar on the left lists various topics, with "Uncertainty and Risk" selected. The main content area displays a table of contents with expandable sections: "AI in Finance", "Normative Finance and Financial Theories", and "Uncertainty and Risk". Below the table of contents, a code cell is visible, containing Python code that imports numpy and defines variables for stock and bond prices and market price vectors.

python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings A

RAM Disk Editing

Table of contents

- AI in Finance
 - Normative Finance and Financial Theories
 - Uncertainty and Risk**
 - Expected Utility Theory (EUT)
 - Mean-Variance Portfolio Theory (MVPT)
 - Capital Asset Pricing Model (CAPM)
 - Arbitrage Pricing Theory (APT)
 - Deep Learning for Financial Time Series Forecasting
 - Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python
 - Investment Portfolio Optimization

AI in Finance

- Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.
- Github: <https://github.com/yhilpisch/aiif/>

Normative Finance and Financial Theories

Uncertainty and Risk

```
1 import numpy as np
2
3 #The prices of the stock and bond today.
4 S0 = 10
5 B0 = 10
6 print('S0', S0)
7 print('B0', B0)
8
9 #The uncertain payoff of the stock and bond tomorrow.
10 S1 = np.array((20, 5))
11 B1 = np.array((11, 11))
12 print('S1', S1)
13 print('B1', B1)
14
15 #The market price vector
16 M0 = np.array((S0, B0))
```

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

The screenshot shows a Google Colab notebook interface. At the top, the notebook is titled "python101.ipynb" and has a star icon. The top toolbar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", with a status "All changes saved". On the right side of the top toolbar, there are icons for "Comment", "Share", "Settings", and a user profile icon labeled "A". Below the top toolbar, there are indicators for "RAM" and "Disk" usage, and an "Editing" mode indicator.

The left sidebar contains a "Table of contents" with a search icon and a list of topics:

- Python101
- Python File Input / Output
- OS, IO, files, and Google Drive
- Python Programming
- Pythong String and Text
- Python Numpy
- Python Pandas
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading**
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python
 - Investment Portfolio Optimization
- Text Analytics and Natural Language Processing (NLP)
 - Python for Natural Language Processing
 - spaCy Chinese Model
 - Open Chinese Convert (OpenCC, 開放中文轉換)
 - Jieba 結巴中文分詞
 - Natural Language Toolkit (NLTK)
 - Stanza: A Python NLP Library for Many Human Languages

```
1 ! pip install pandas_datareader
2 import pandas as pd
3 import pandas_datareader.data as web
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 import datetime as dt
7 %matplotlib inline
8
9 #Read Stock Data from Yahoo Finance
10 end = dt.datetime.now()
11 #start = dt.datetime(end.year-2, end.month, end.day)
12 start = dt.datetime(2010, 1, 1)
13 df = web.DataReader("AAPL", 'yahoo', start, end)
14 df.to_csv('AAPL.csv')
15 #df = pd.read_csv('AAPL.csv')
16 print(df.head())
17 print(df.tail())
18 print(df.describe())
19
20 df['Adj Close'].plot(legend=True, figsize=(12, 8), title='AAPL', label='Adj Close')
21 plt.figure(figsize=(12,9))
22 top = plt.subplot2grid((12,9), (0, 0), rowspan=10, colspan=9)
23 bottom = plt.subplot2grid((12,9), (10,0), rowspan=2, colspan=9)
24 top.plot(df.index, df['Adj Close'], color='blue') #df.index gives the dates
25 bottom.bar(df.index, df['Volume'])
26
27 # set the labels
28 top.axes.get_xaxis().set_visible(False)
29 top.set_title('AAPL')
30 top.set_ylabel('Adj Close')
31 bottom.set_ylabel('Volume')
32
33 plt.figure(figsize=(12,9))
```

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

CO python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings A


RAM Disk Editing

Table of contents

- Python101
- Python File Input / Output
- OS, IO, files, and Google Drive
- Python Programming
- Python String and Text
- Python Numpy
- Python Pandas
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading**
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python
 - Investment Portfolio Optimization
- Text Analytics and Natural Language Processing (NLP)
 - Python for Natural Language Processing
 - spaCy Chinese Model
 - Open Chinese Convert (OpenCC, 開放中文轉換)
 - Jieba 結巴中文分詞
 - Natural Language Toolkit (NLTK)
 - Stanza: A Python NLP Library for Many Human Languages

```
2 !pip install plotly
3 import plotly.graph_objects as go
4
5 import pandas as pd
6 from datetime import datetime
7 df = pd.read_csv('AAPL.csv')
8 fig = go.Figure(data=[go.Candlestick(x=df['Date'],
9                                     open=df['Open'],
10                                    high=df['High'],
11                                    low=df['Low'],
12                                    close=df['Close'])])
13
14 fig.show()
```

Requirement already satisfied: plotly in /usr/local/lib/python3.6/dist-packages (4.4.1)
Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.6/dist-packages (from plotly) (1.3.3)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from plotly) (1.12.0)



<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings A

RAM Disk Editing

Table of contents

- Python101
- Python File Input / Output
- OS, IO, files, and Google Drive
- Python Programming
- Python String and Text
- Python Numpy
- Python Pandas
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading

Investment Portfolio Optimisation with Python

- Efficient Frontier Portfolio Optimisation in Python
- Investment Portfolio Optimization
- Text Analytics and Natural Language Processing (NLP)
- Python for Natural Language Processing
 - spaCy Chinese Model
- Open Chinese Convert (OpenCC, 開放中文轉換)
- Jieba 結巴中文分詞
- Natural Language Toolkit (NLTK)
- Stanza: A Python NLP Library for Many Human Languages

```
51 max_sharpe_port = results_frame.iloc[results_frame['sharpe'].idxmax()]
52 #locate position of portfolio with minimum standard deviation
53 min_vol_port = results_frame.iloc[results_frame['stdev'].idxmin()]
54
55 #create scatter plot coloured by Sharpe Ratio
56 plt.figure(figsize=(10,6))
57 plt.scatter(results_frame.stdev,results_frame.ret,c=results_frame.sharpe,cmap='RdYlBu')
58 plt.xlabel('Volatility')
59 plt.ylabel('Returns')
60 plt.colorbar()
61 #plot red star to highlight position of portfolio with highest Sharpe Ratio
62 plt.scatter(max_sharpe_port[1],max_sharpe_port[0],marker=(5,1,0),color='r',s=1000)
63 #plot green star to highlight position of minimum variance portfolio
64 plt.scatter(min_vol_port[1],min_vol_port[0],marker=(5,1,0),color='g',s=500)
```

<matplotlib.collections.PathCollection at 0x7f13132a01d0>



The figure is a scatter plot showing the relationship between Volatility (x-axis, ranging from 0.24 to 0.34) and Returns (y-axis, ranging from 0.16 to 0.30). The data points are colored based on the Sharpe Ratio, with a color bar on the right indicating values from 0.6 (red) to 1.1 (blue). A red star highlights the portfolio with the highest Sharpe Ratio, located at approximately (0.25, 0.28). A green star highlights the minimum variance portfolio, located at approximately (0.24, 0.23).

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings A

RAM Disk Editing

Table of contents

- Python101
- Python File Input / Output
- OS, IO, files, and Google Drive
- Python Programming
- Python String and Text
- Python Numpy
- Python Pandas
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python**
 - Investment Portfolio Optimization
- Text Analytics and Natural Language Processing (NLP)
 - Python for Natural Language Processing
 - spaCy Chinese Model
 - Open Chinese Convert (OpenCC, 開放中文轉換)
 - Jieba 結巴中文分詞
 - Natural Language Toolkit (NLTK)
 - Stanza: A Python NLP Library for Many Human Languages

```
Annualised Return: 0.18
Annualised Volatility: 0.18

      AAPL  AMZN  FB  GOOGL
allocation 44.67 29.05 26.28 0.0
-----
Minimum Volatility Portfolio Allocation

Annualised Return: 0.22
Annualised Volatility: 0.16

      AAPL  AMZN  FB  GOOGL
allocation 34.02 0.73 6.98 58.26
```

Calculated Portfolio Optimization based on Efficient Frontier

Legend: efficient frontier (black line), Maximum Sharpe ratio (red star), Minimum volatility (green star). Color scale for volatility ranges from 1.0 to 1.5.

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

Portfolio Optimization

Efficient Frontier Portfolio Optimization

python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings A

RAM Disk Editing ^

Table of contents

- Python101
- Python File Input / Output
- OS, IO, files, and Google Drive
- Python Programming
- Pythong String and Text
- Python Numpy
- Python Pandas
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python**
 - Investment Portfolio Optimization
- Text Analytics and Natural Language Processing (NLP)
 - Python for Natural Language Processing
 - spaCy Chinese Model
 - Open Chinese Convert (OpenCC, 開放中文轉換)
 - Jieba 結巴中文分詞
 - Natural Language Toolkit (NLTK)
 - Stanza: A Python NLP Library for Many Human Languages

```
Annualised Return: 0.22
Annualised Volatility: 0.16

      AAPL  AMZN   FB  GOOGL
allocation 34.02  0.73  6.98  58.26
-----
Individual Stock Returns and Volatility

AAPL : annuaised return 0.28 , annualised volatility: 0.21
AMZN : annuaised return 0.34 , annualised volatility: 0.25
FB : annuaised return 0.3 , annualised volatility: 0.23
GOOGL : annuaised return 0.18 , annualised volatility: 0.18
-----
```

Portfolio Optimization with Individual Stocks

Stock	Allocation (%)	Annualised Return	Annualised Volatility
AAPL	34.02	0.28	0.21
AMZN	0.73	0.34	0.25
FB	6.98	0.30	0.23
GOOGL	58.26	0.18	0.18

<https://tinyurl.com/aintpupython101>

Summary

- **Uncertainty and Risk**
- **Expected Utility Theory (EUT)**
- **Mean-Variance Portfolio Theory (MVPT)**
- **Capital Asset Pricing Model (CAPM)**
- **Arbitrage Pricing Theory (APT)**

Data-Driven Finance

Data-Driven Finance

- **Scientific Method**
- **Financial Econometrics and Regression**
- **Data Availability**
- **Normative Theories Revisited**
- **Debunking Central Assumptions in Finance**

Data-driven finance

- **Financial context (theory, model, application)** that is primarily driven by and based on **insights** gained from **data**.

Data-driven finance

Robin Wigglesworth (2019)

- Nowadays, analysts sift through **non-traditional information** such as **satellite imagery** and **credit card data**, or use **artificial intelligence** techniques such as **machine learning** and **natural language processing** to glean fresh **insights** from **traditional sources** such as **economic data** and **earnings-call transcripts**.

Scientific Method

- Generally accepted principles that should guide scientific effort
- The **scientific method** is an **empirical method** of acquiring **knowledge** that has characterized the **development of science**
- It involves careful **observation**, applying **rigorous skepticism** about what is observed, given that cognitive **assumptions** can distort how one **interprets** the observation.

Scientific Method

- It involves **formulating hypotheses**,
via **induction**, based on such **observations**;
experimental and measurement-based testing of
deductions drawn from the hypotheses;
and **refinement** (or elimination) of the hypotheses
based on the **experimental findings**

Normative Finance and Scientific Method

- **Normative financial theories** mostly rely on assumptions and axioms in combination with deduction as the major analytical method to arrive at their central results.
 - **Expected utility theory (EUT)** assumes that agents have the same utility function no matter what state of the world unfolds and that they maximize expected utility under conditions of uncertainty.
 - **Mean-variance portfolio (MVP)** theory describes how investors should invest under conditions of uncertainty assuming that only the expected return and the expected volatility of a portfolio over one period count.

Normative Finance and Scientific Method

- The **capital asset pricing model (CAPM)** assumes that only the nondiversifiable market risk explains the expected return and the expected volatility of a stock over one period.
- **Arbitrage pricing theory (APT)** assumes that a number of identifiable risk factors explains the expected return and the expected volatility of a stock over time; admittedly, compared to the other theories, the formulation of APT is rather broad and allows for wide-ranging interpretations.

Financial Econometrics and Regression

- **[Financial] econometrics** is the **quantitative application of statistical and mathematical models** using **[financial] data** to develop financial theories or test existing hypotheses in finance and to forecast future trends from historical data.
- It subjects real-world **[financial] data** to statistical trials and then compares and contrasts the results against the **[financial] theory or theories** being tested.

Financial Econometrics and Regression

- One of the major tools in **financial econometrics** is **regression**, in both its univariate and multivariate forms
- **Regression** is also a central tool in **statistical learning** in general

Data Availability

- **Types of (financial) data**
 - Financial econometrics is driven by statistical methods, such as regression, and the availability of financial data
 - Theoretical and empirical financial research was mainly driven by relatively small data sets and was mostly comprised of **end-of-day (EOD)** data
 - Types of financial and other data available in ever increasing **granularity, quantity, and velocity.**
- **Quality and quantity via programmatic APIs**
 - Finance professionals have relied on data terminals from **Refinitiv** or **Bloomberg**
 - Major breakthrough in data-driven finance via programmatic APIs

Relevant types of financial data

Time	Structured data	Unstructured data	Alternative data
Historical	Prices, fundamentals	News, texts	Web, social media, satellites
Streaming	Prices, volumes	News, filings	Web, social media, satellites, Internet of Things

Yahoo Finance World Indices

<https://finance.yahoo.com/world-indices/>



Search for news, symbols or companies



Sign in



Mail

YF Chartbook Calendars Trending Tickers Stocks: Most Actives Stocks: Gainers Stocks: Losers Top ETFs Futures World Indices Currencies ...

U.S. markets closed

S&P 500

4,217.04
-7.12 (-0.17%)



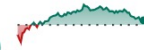
Dow 30

32,936.41
-190.87 (-0.58%)



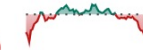
Nasdaq

13,018.33
+34.52 (+0.27%)



Russell 2000

1,665.88
-14.91 (-0.89%)



Crude Oil

86.04
+0.55 (+0.64%)



World Indices

Symbol	Name	Last Price	Change	% Change	Volume	Intraday High/Low	52 Week Range	Day Chart
^GSPC	S&P 500	4,217.04	-7.12	-0.17%	2.455B	4,189.22 - 4,255.84	3,698.15 - 4,607.07	
^DJI	Dow Jones Industrial Average	32,936.41	-190.87	-0.58%	345.435M	32,892.19 - 33,234.85	31,161.41 - 35,679.13	
^IXIC	NASDAQ Composite	13,018.33	+34.52	+0.27%	4.075B	12,848.83 - 13,143.23	10,207.47 - 14,446.55	
^NYA	NYSE COMPOSITE (DJ)	14,946.01	-87.30	-0.58%	0	14,916.63 - 15,082.62	14,085.42 - 16,458.89	
^XAX	NYSE AMEX COMPOSITE INDEX	4,518.31	-63.56	-1.39%	0	4,501.20 - 4,581.87	3,824.21 - 4,723.96	
^BUK100P	Cboe UK 100	736.03	-2.71	-0.37%	0	732.56 - 738.73	690.72 - 805.30	
^RUT	Russell 2000	1,665.88	-14.91	-0.89%	0	1,662.64 - 1,687.74	1,662.64 - 2,007.31	
^VIX	CBOE Volatility Index	20.37	-1.34	-6.17%	0	19.48 - 23.08	15.53 - 35.05	
^FTSE	FTSE 100	7,374.83	-27.31	-0.37%	0	7,338.59 - 7,402.20	6,914.70 - 8,047.10	
^GDAXI	DAX PERFORMANCE-INDEX	14,800.72	+2.25	+0.02%	0	14,630.21 - 14,838.01	12,747.38 - 16,528.97	

World Indices

```
import io
import requests
import pandas as pd
response = requests.get('https://finance.yahoo.com/world-indices/')
df = pd.read_html(io.StringIO(response.text))
worldidx = df[0]
worldidx.to_csv('world_indices.csv')
worldidx
```

Symbol	Name	Last Price	Change	% Change	Volume
^GSPC	S&P 500	3,797.34	+44.59	+1.19%	2.589B
^DJI	Dow Jones Industrial Average	31,499.62	+417.06	+1.34%	345.036M
^IXIC	NASDAQ Composite	10,952.61	+92.90	+0.86%	4.063B
^NYA	NYSE COMPOSITE (DJ)	14,226.11	+82.05	+0.58%	0
^XAX	NYSE AMEX COMPOSITE INDEX	4,295.57	-106.83	-2.43%	0
^BUK100P	Cboe UK 100	701.69	+5.39	+0.77%	0
^RUT	Russell 2000	1,748.40	+6.16	+0.35%	0
^VIX	Vix	29.85	+0.16	+0.54%	0

ffn: Financial Functions for Python

```
#^GSPC S&P 500
#^DJI Dow 30
#^IXIC Nasdaq
!pip install ffn
import ffn
%pylab inline
df = ffn.get('^gspc, ^dji, ^ixic', start='2010-01-01', end='2022-01-01')
print(df.head())
print(df.tail())
print(df.describe())
ax = df.plot(figsize=(12,9))
```

```
2021-12-28  4788.330059  36338.210000  15701.710727
2021-12-29  4793.060059  36488.628906  15766.219727
2021-12-30  4778.729980  36398.078125  15741.559570
2021-12-31  4766.180176  36338.300781  15644.969727
count      gspc          dji          ixic
mean    2260.488112  19756.317518  6004.283709
std       890.501675   6927.100147  3438.840186
min     1022.580017   9686.480469  2091.790039
25%     1461.400024  13557.000000  3131.489990
50%     2088.479980  17851.509766  4984.620117
75%     2798.360107  25332.179688  7669.169922
max     4793.060059  36488.628906  16057.440430
```

^GSPC: S&P 500, ^DJI: Dow 30, ^IXIC: Nasdaq

```
df = ffn.get('^gspc', '^dji', '^ixic', start='2010-01-01', end='2022-01-01')  
ax = df.plot(figsize=(12, 9))
```



ffn: Financial Functions for Python

```
!pip install ffn
import ffn
%pylab inline
df = ffn.get('^gspc, ^dji, ^ixic', start='2010-01-01', end='2022-01-01')
print(df.head())
print(df.tail())
print(df.describe())
ax = df.plot(figsize=(12, 9))

returns = df.to_returns().dropna()
ax = returns.hist(figsize=(14, 10))
returns.corr().as_format('.2f')
returns.plot_corr_heatmap()
ax = df.plot(figsize=(14, 10))

perf = df.calc_stats()
perf.plot(figsize=(14, 10))

print(perf.display())
```

CoinGeckoAPI(): Cryptocurrency Data API

```
from pycoingecko import CoinGeckoAPI
import pandas as pd
from datetime import datetime

start_date_obj = datetime.strptime('2020-01-01', '%Y-%m-%d')
end_date_obj = datetime.strptime('2022-12-31', '%Y-%m-%d')

cg = CoinGeckoAPI()

data = cg.get_coin_market_chart_range_by_id(id='bitcoin',
vs_currency='usd', from_timestamp=start_date_obj.timestamp(),
to_timestamp=end_date_obj.timestamp())

processed_data = [ {'date': datetime.utcfromtimestamp(price[0] /
1000).date(), 'price': price[1]} for price in data.get('prices', [])]
df = pd.DataFrame(processed_data)
df.to_csv('btcusd.csv')
```

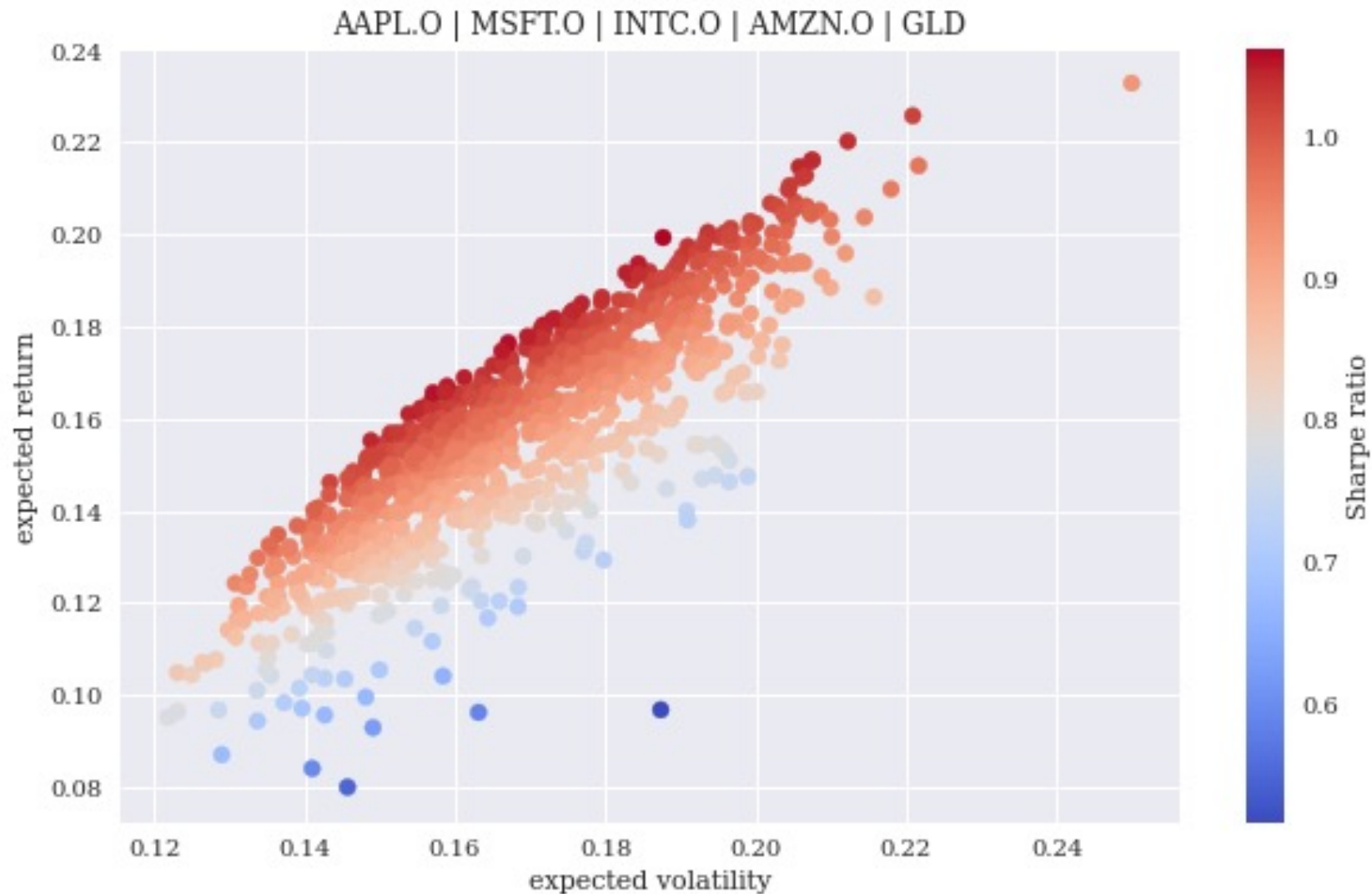
Normative Theories Revisited

- Revisits the normative theories and analyzes them based on real financial time series data
- **Expected Utility and Reality**
- **Mean-Variance Portfolio Theory (MVPT)**
- **Capital Asset Pricing Model (CAPM)**
- **Arbitrage Pricing Theory (APT)**

Normalized financial time series data



Simulated portfolio volatilities, returns, and Sharpe ratios



Expected versus realized portfolio volatilities



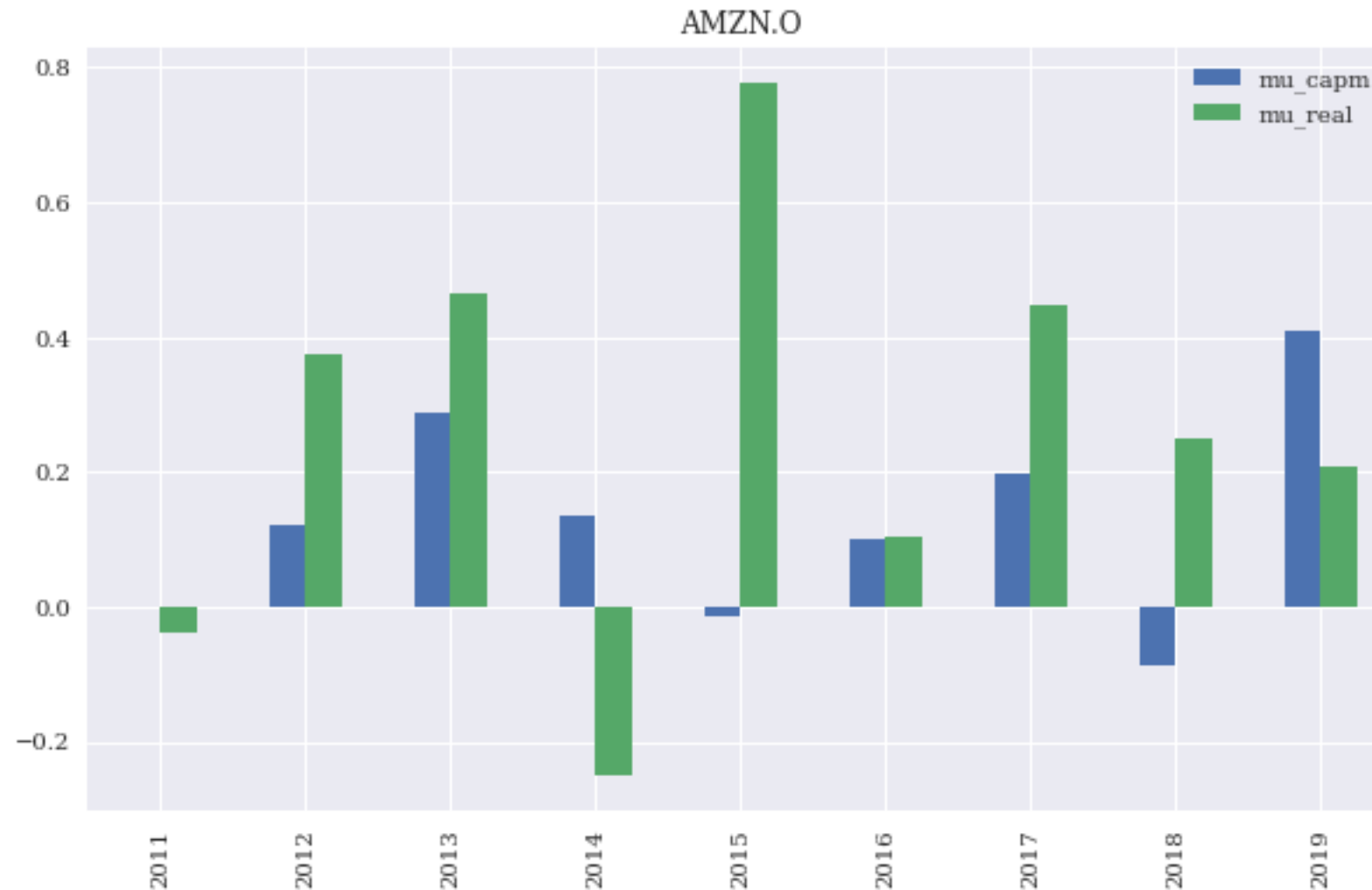
Expected versus realized portfolio returns



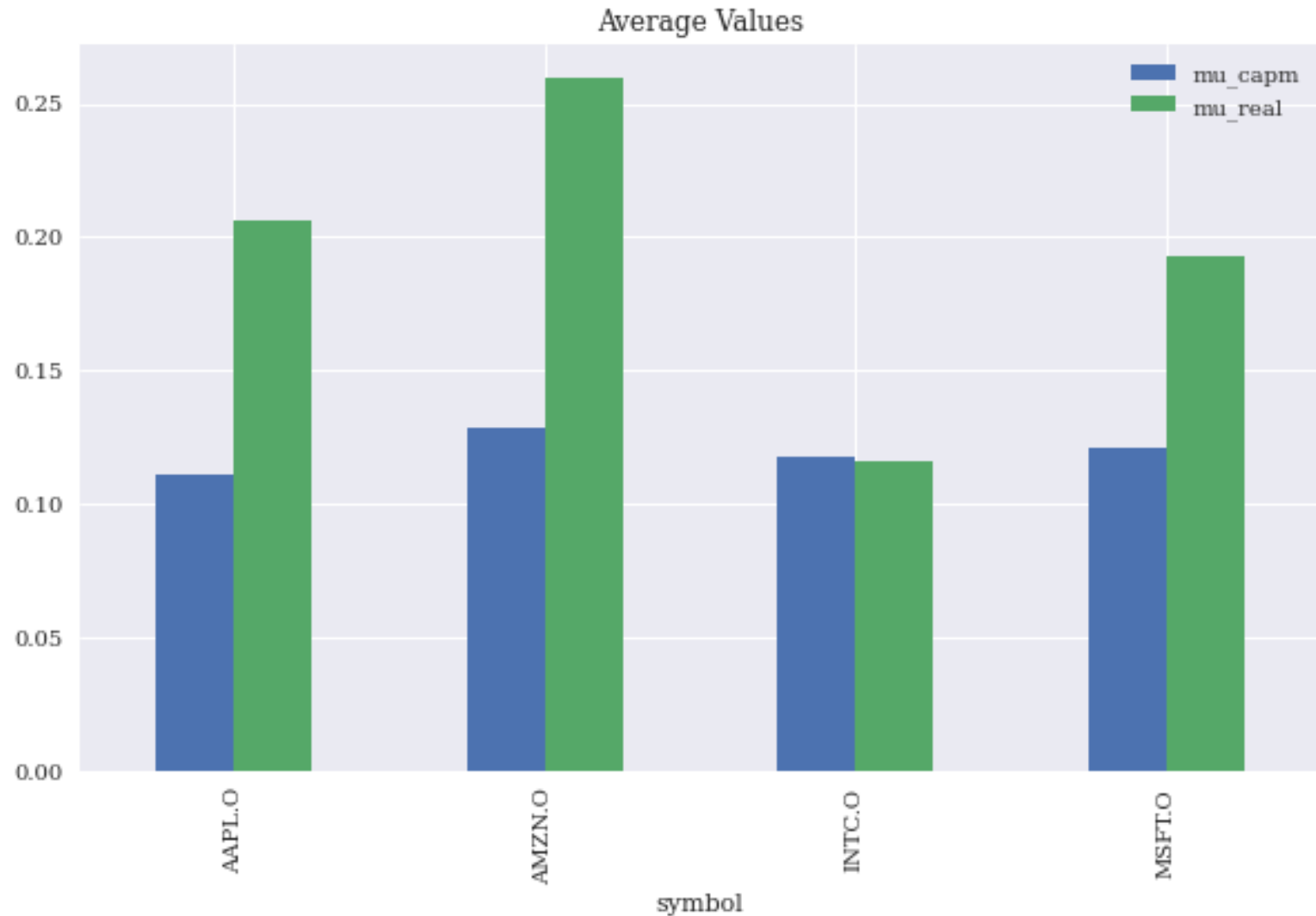
Expected versus realized portfolio Sharpe ratios



CAPM-predicted versus realized stock returns for a single stock



Average CAPM-predicted versus average realized stock returns for multiple stocks



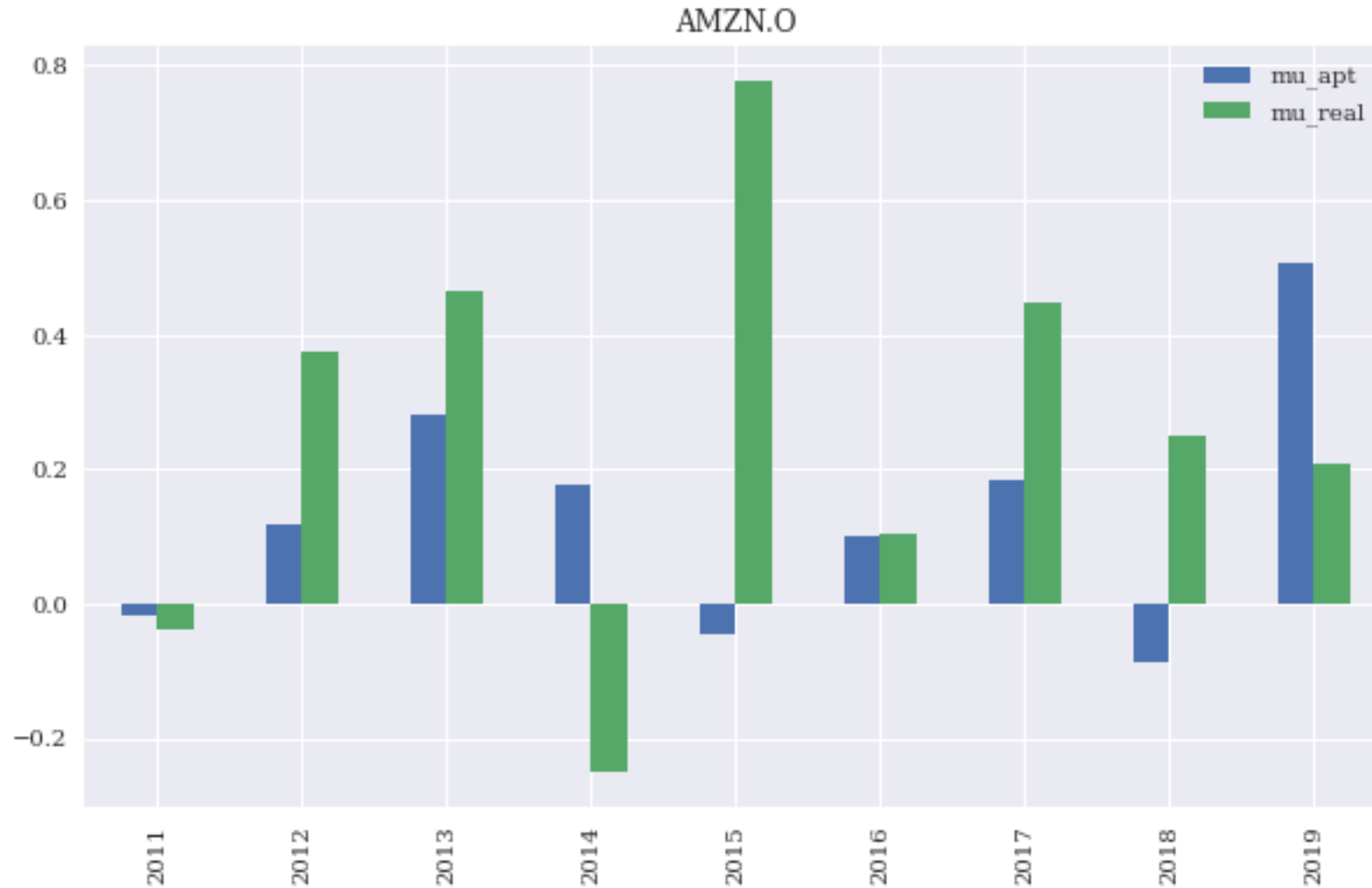
Arbitrage Pricing Theory (APT)

Relevant types of financial data

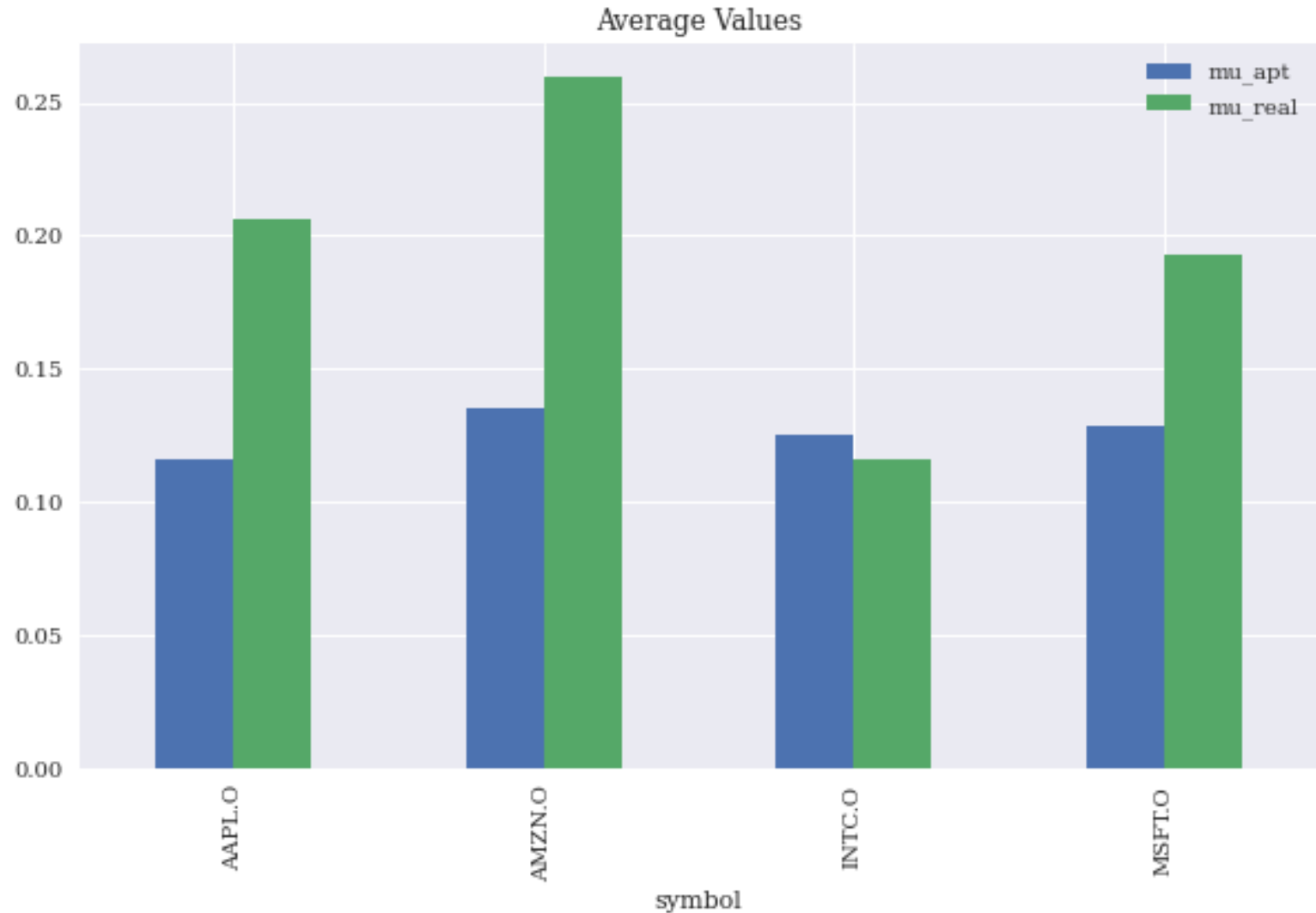
Factor	Description
Market	MSCI World Gross Return Daily USD (PUS = Price Return)
Size	MSCI World Equal Weight Price Net Index EOD
Volatility	MSCI World Minimum Volatility Net Return
Value	MSCI World Value Weighted Gross (NUS for Net)
Risk	MSCI World Risk Weighted Gross USD EOD
Growth	MSCI World Quality Net Return USD
Momentum	MSCI World Momentum Gross Index USD EOD

```
factors = pd.read_csv('http://hilpisch.com/aiif_eikon_eod_factors.csv',  
                    index_col=0, parse_dates=True)
```

APT-predicted versus realized stock returns for a stock



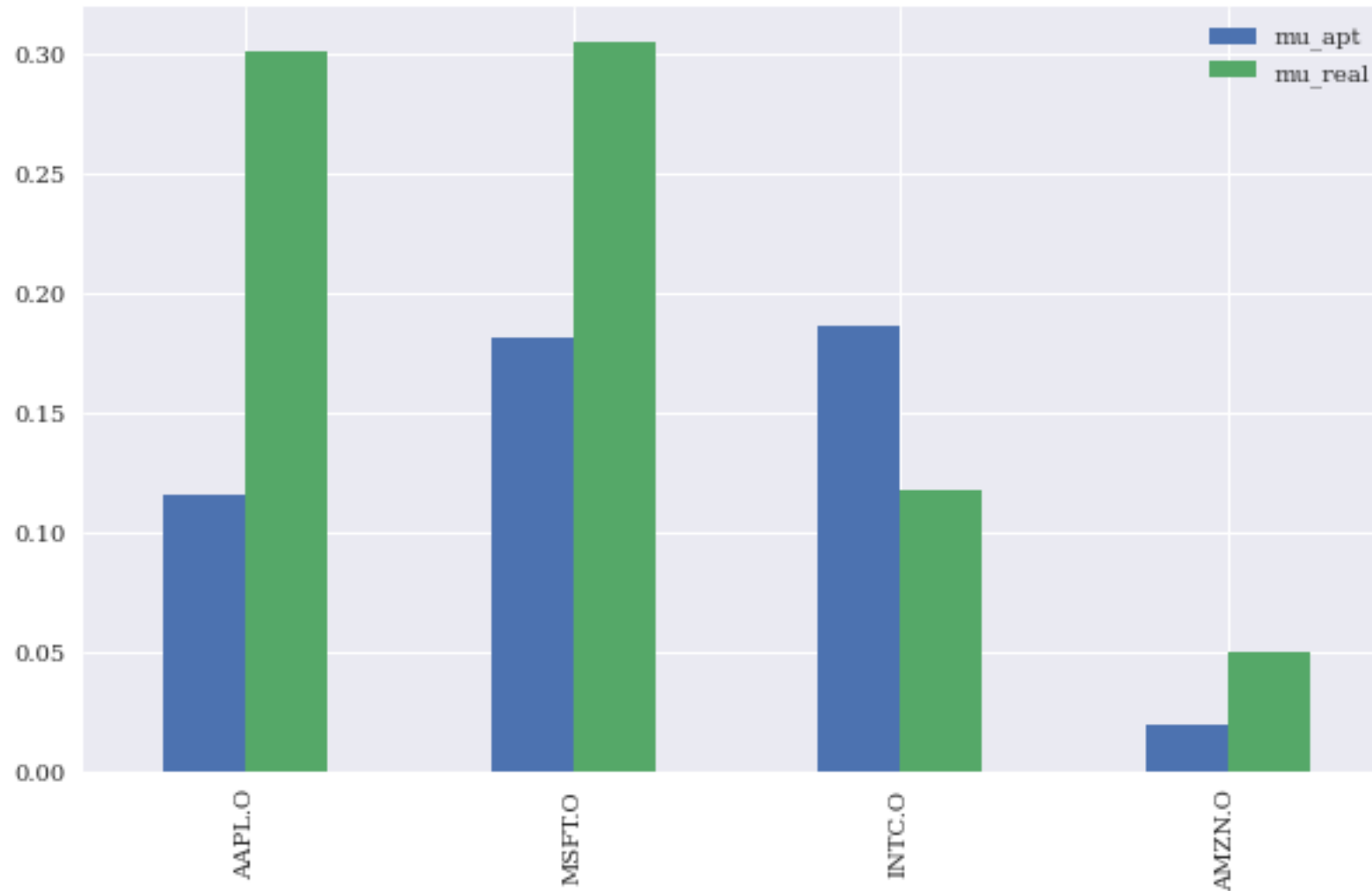
Average APT-predicted versus average realized stock returns for multiple stocks



Normalized factors time series data



APT-predicted returns based on typical factors compared to realized returns



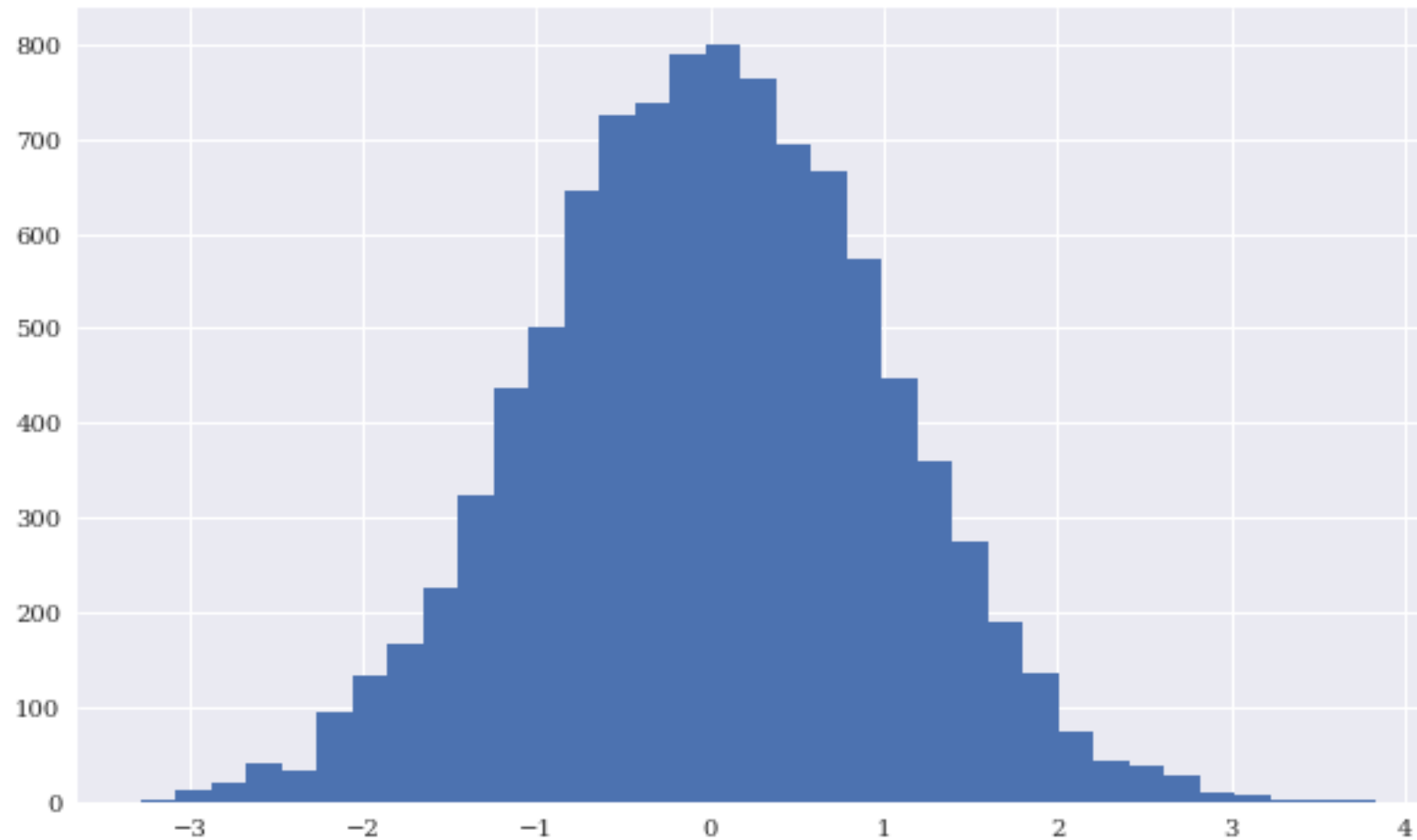
APT-predicted performance and real performance over time (gross)



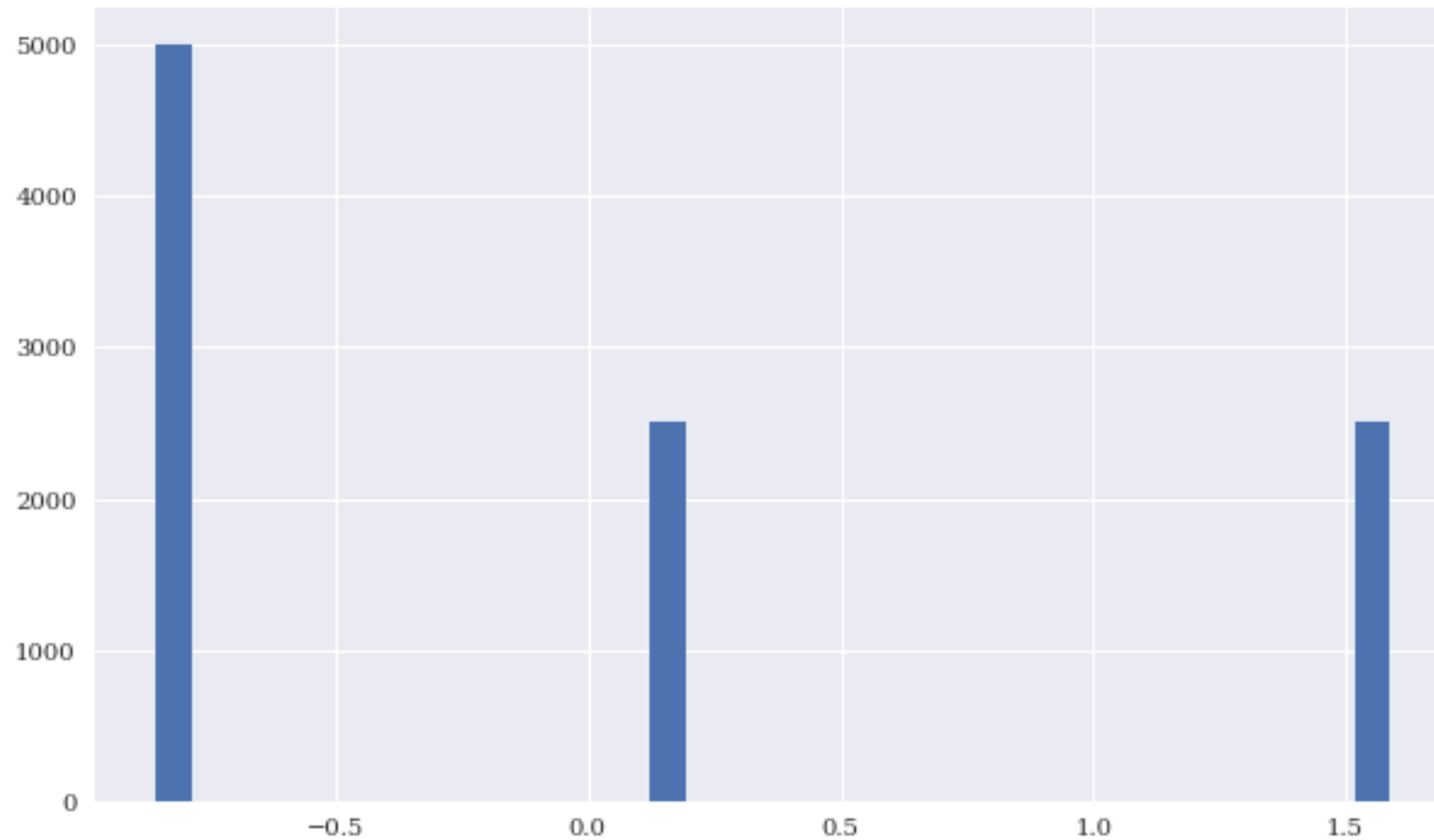
Debunking Central Assumptions in Finance

- Debunks two of the most commonly found **assumptions in financial models and theories**
 - **Normality of returns**
 - **Linear relationships**

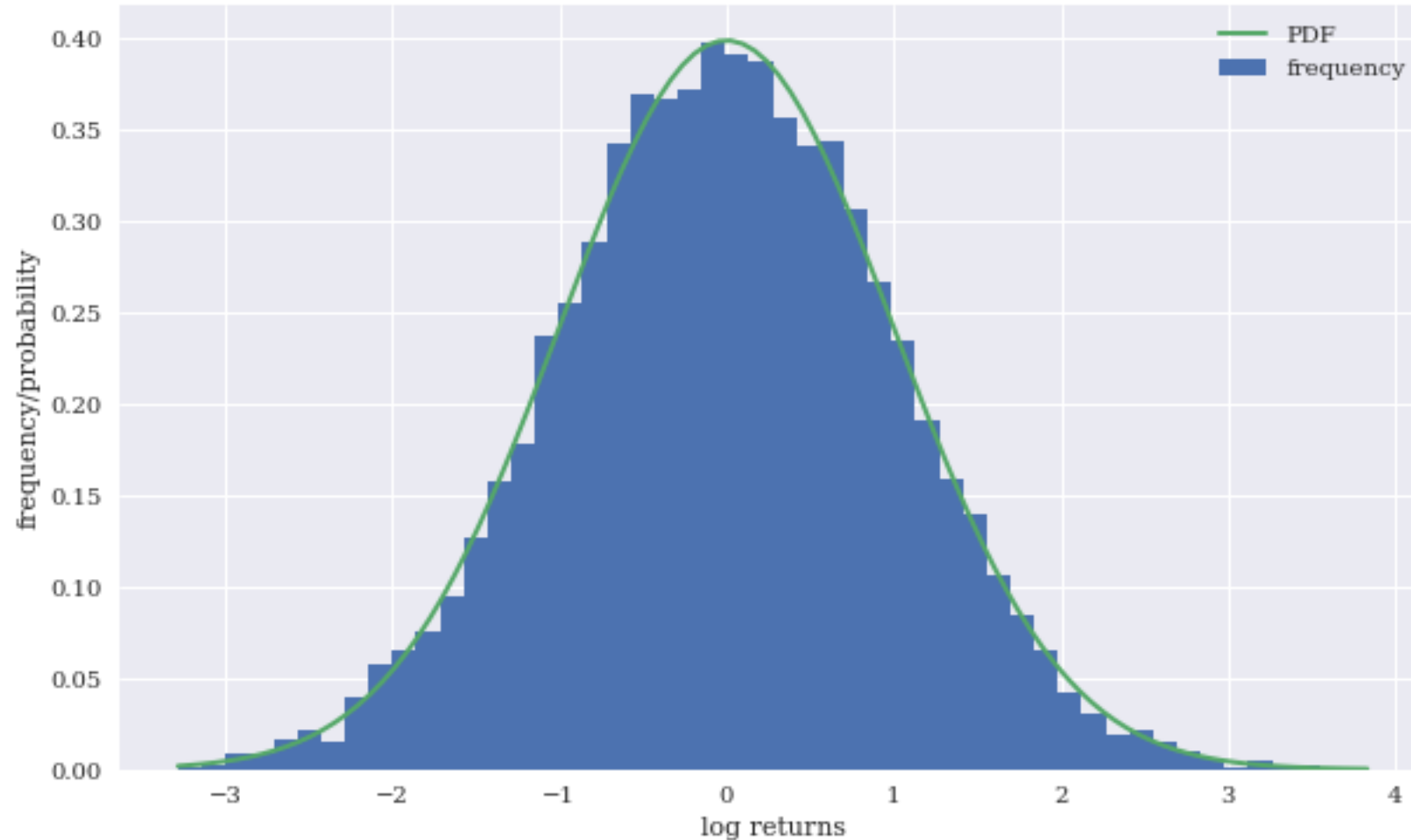
Standard normally distributed random numbers



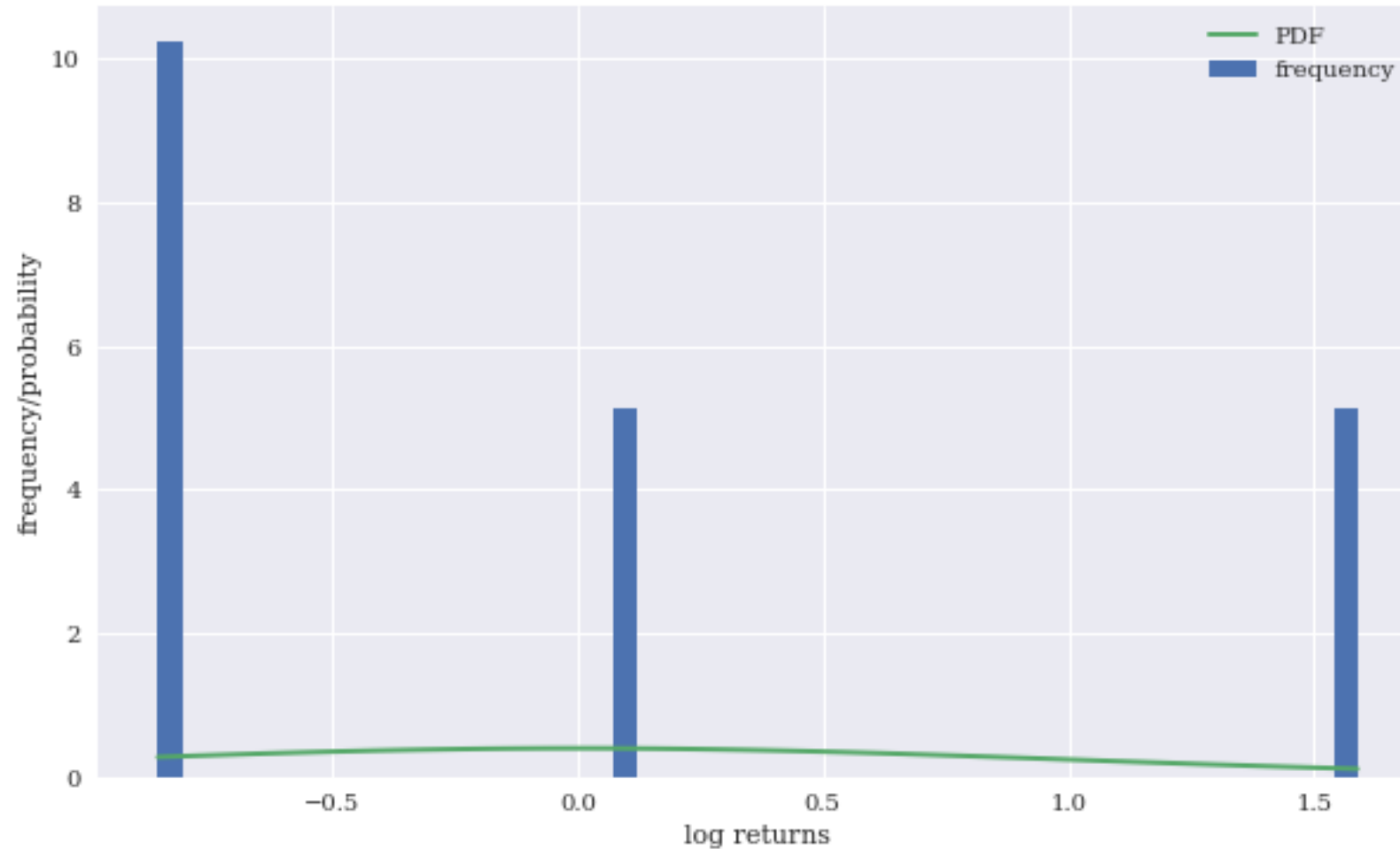
Distribution with first and second moment of 0.0 and 1.0, respectively



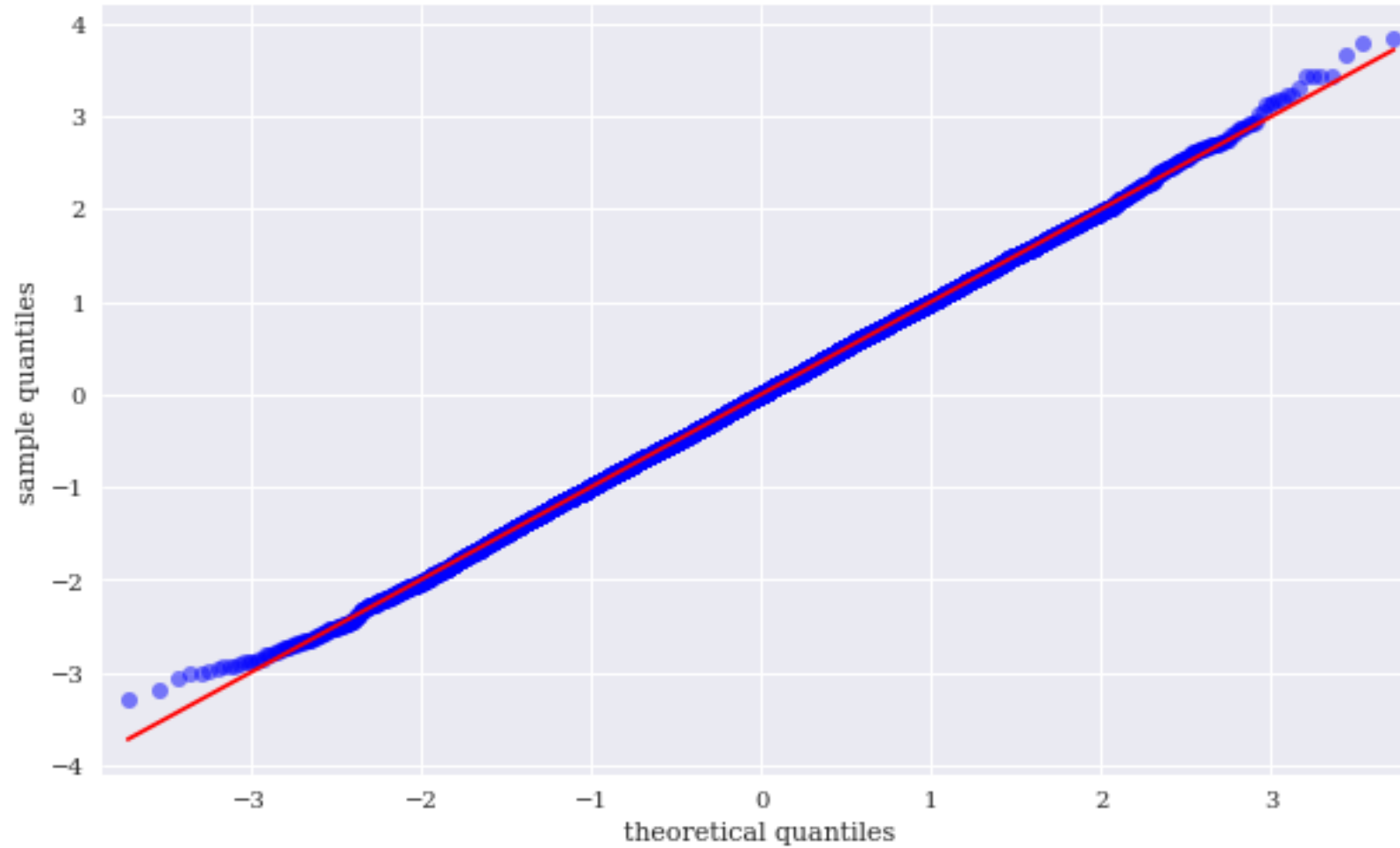
Histogram and PDF for standard normally distributed numbers



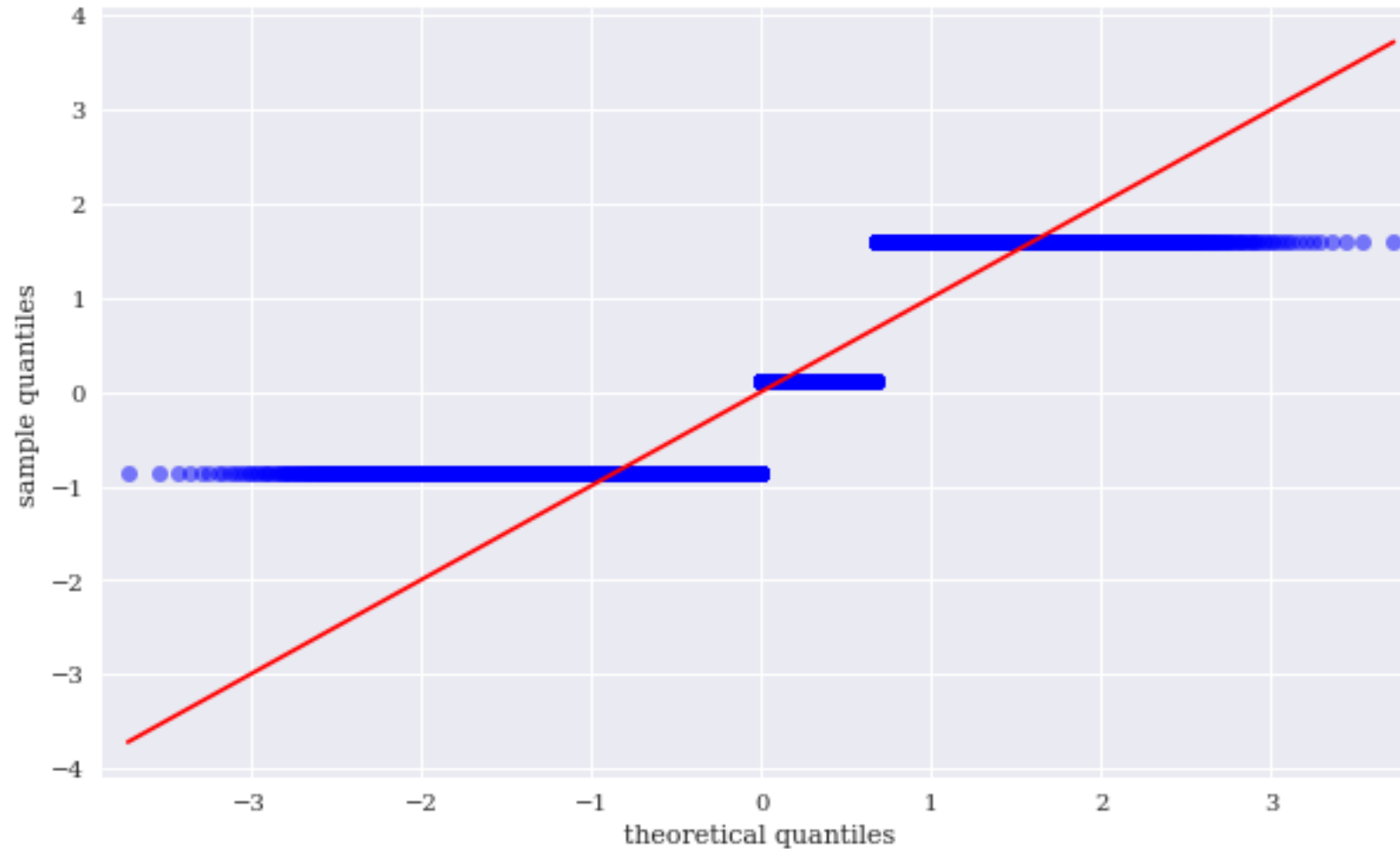
Histogram and normal PDF for discrete numbers



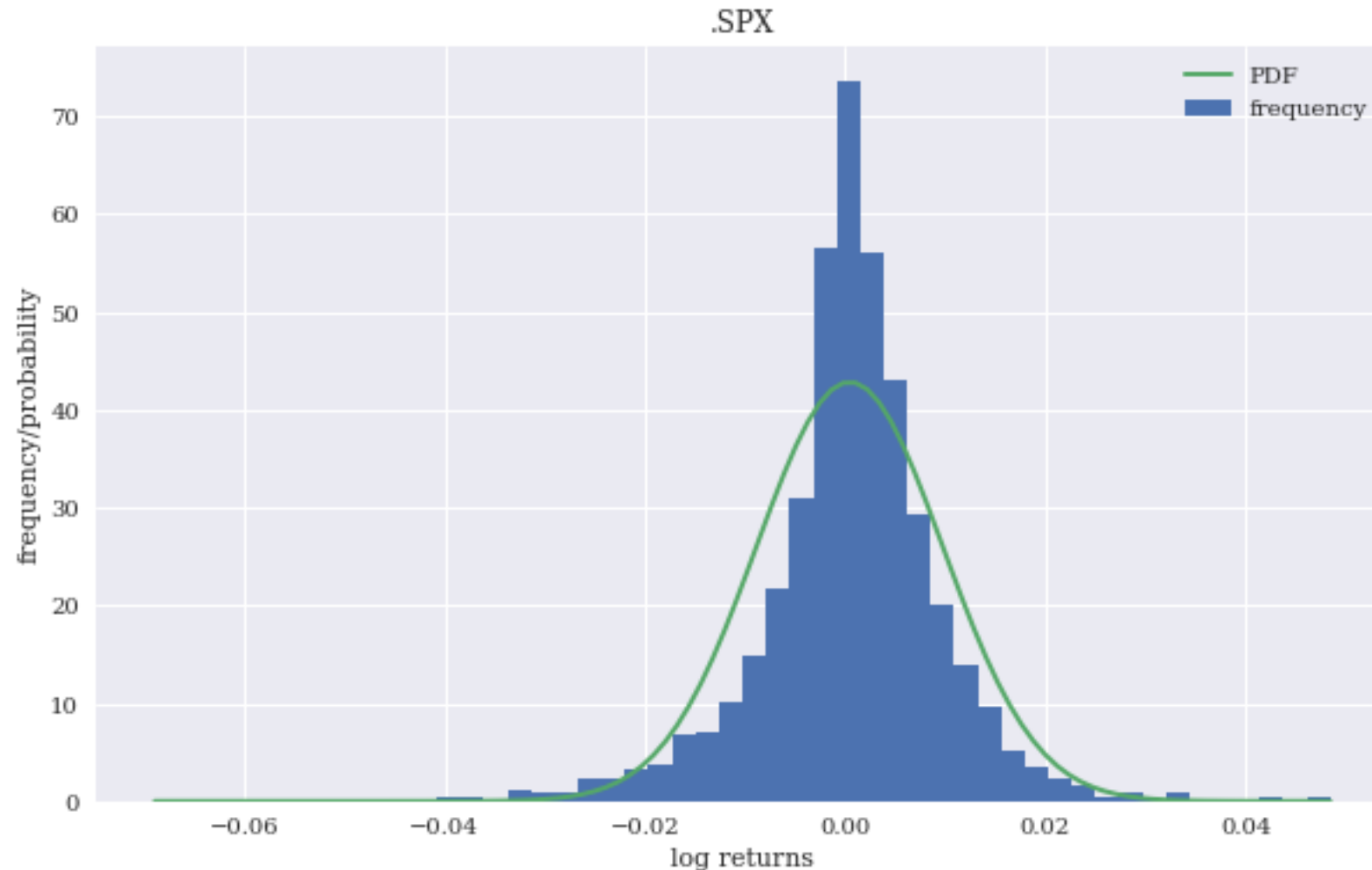
Q-Q plot for standard normally distributed numbers



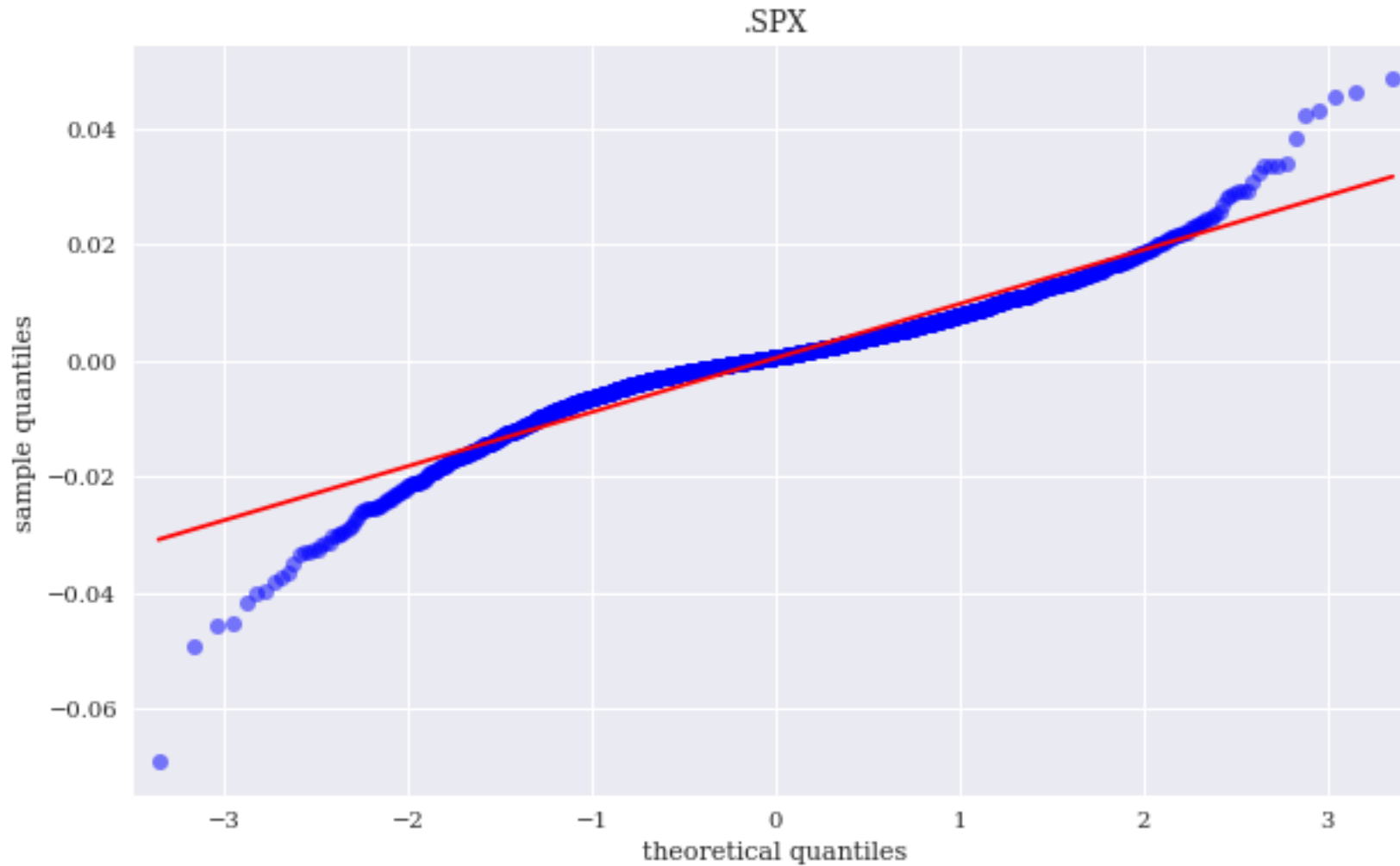
Q-Q plot for discrete numbers



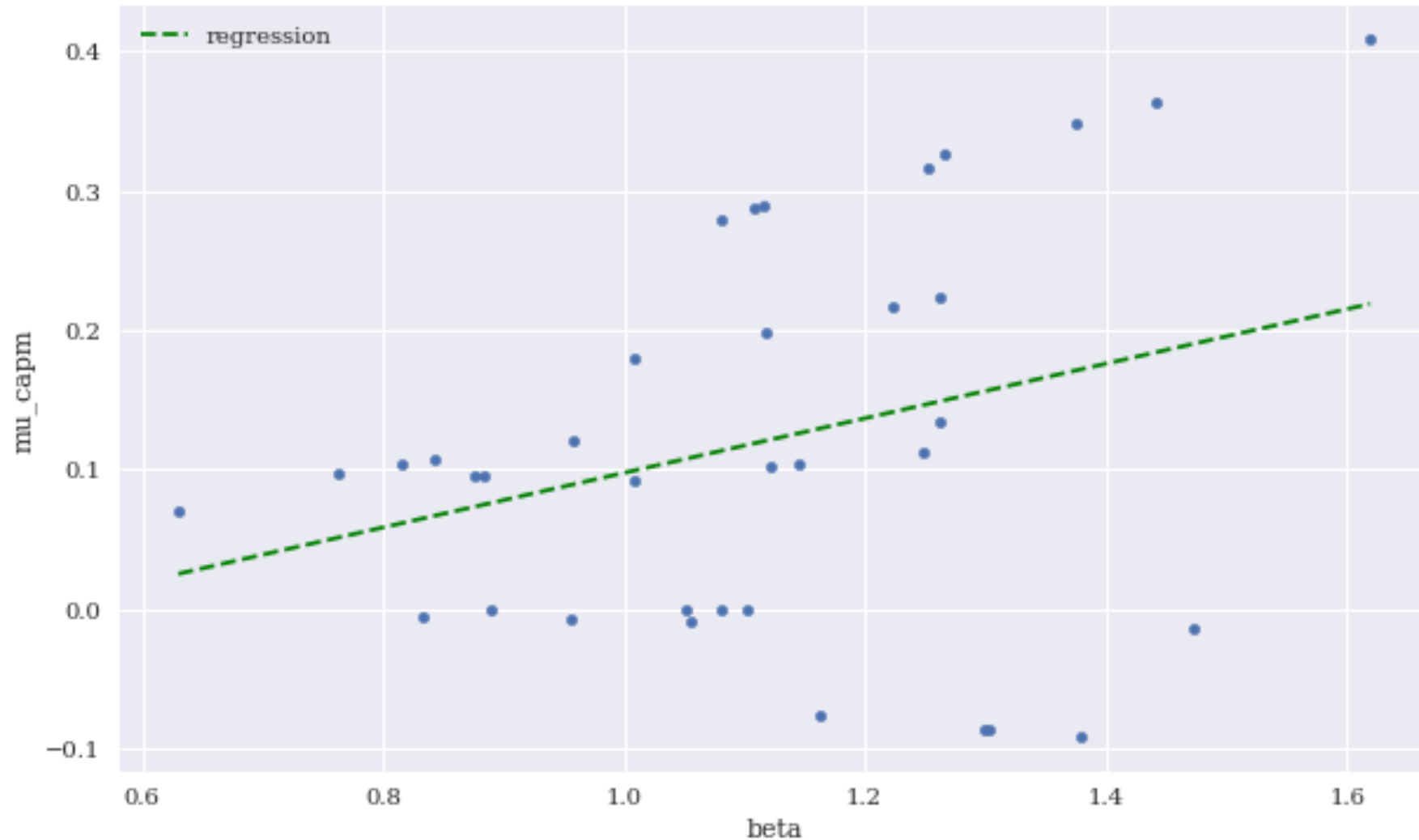
Frequency distribution and normal PDF for S&P 500 log returns



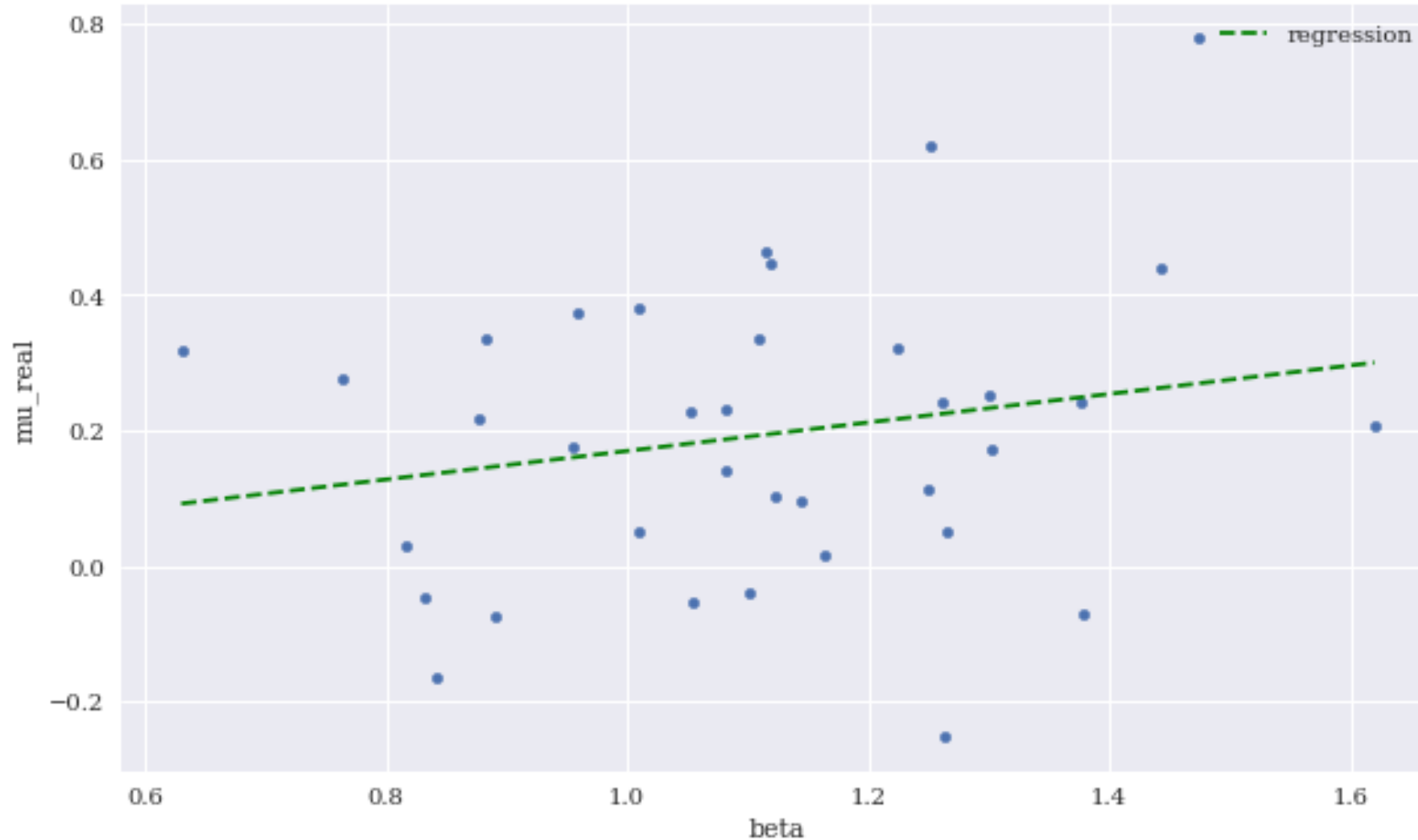
Q-Q for S&P 500 log returns



Expected CAPM return versus beta (including linear regression)



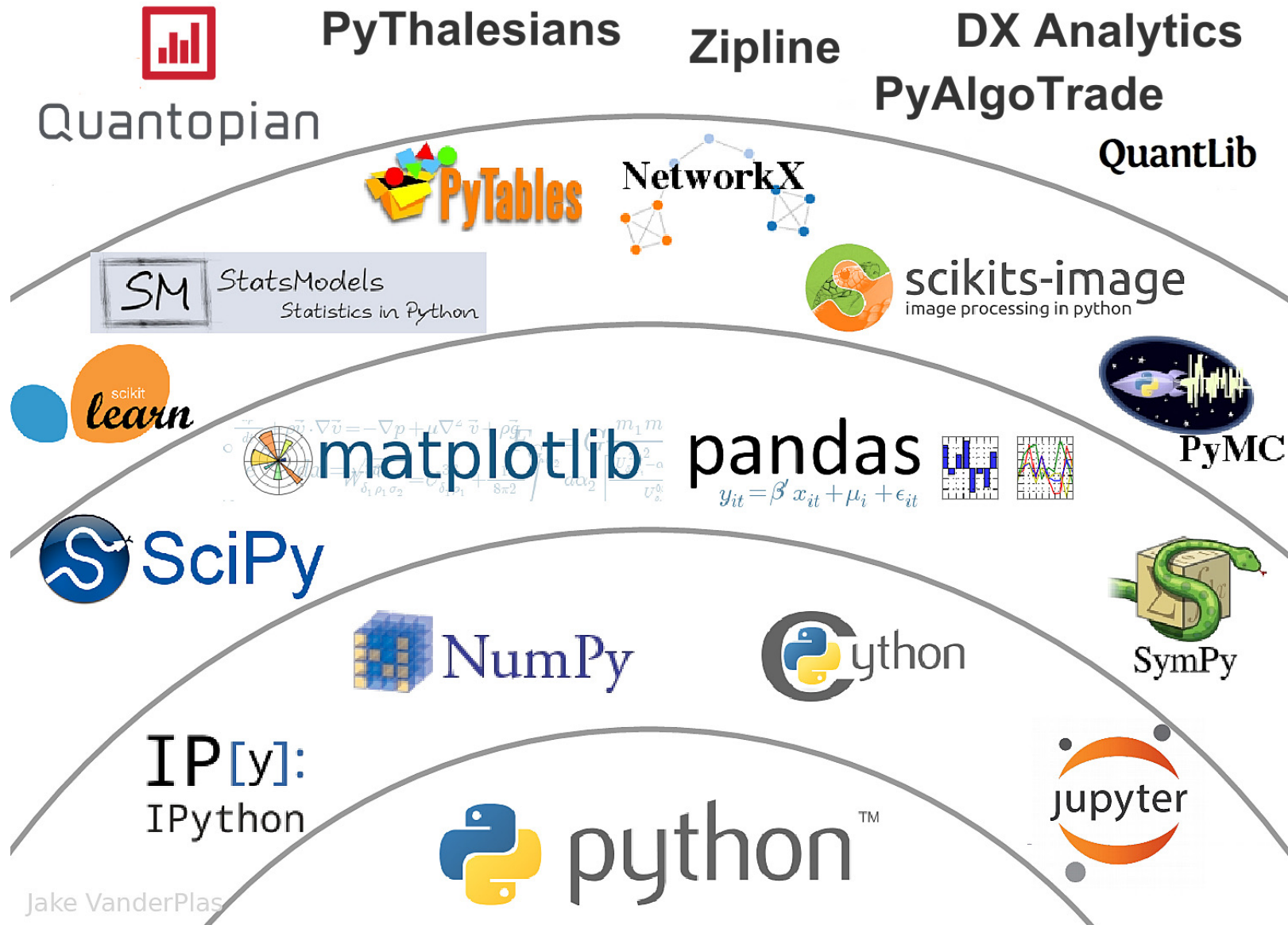
Expected CAPM return versus beta (including linear regression)



Theory-First to Data-Driven Finance

- Finance used to be characterized by **normative theories** based on **simplified mathematical models** of the financial markets, relying on **assumptions** such as **normality of returns** and **linear relationships**.
- The almost universal and comprehensive availability of (financial) data has led to a shift in focus from a **theory-first approach** to **data-driven finance**.
- Several examples based on real financial data illustrate that many popular financial models and theories cannot survive a confrontation with financial market realities.
- Although elegant, they might be too simplistic to capture the complexities, changing nature, and nonlinearities of financial markets.

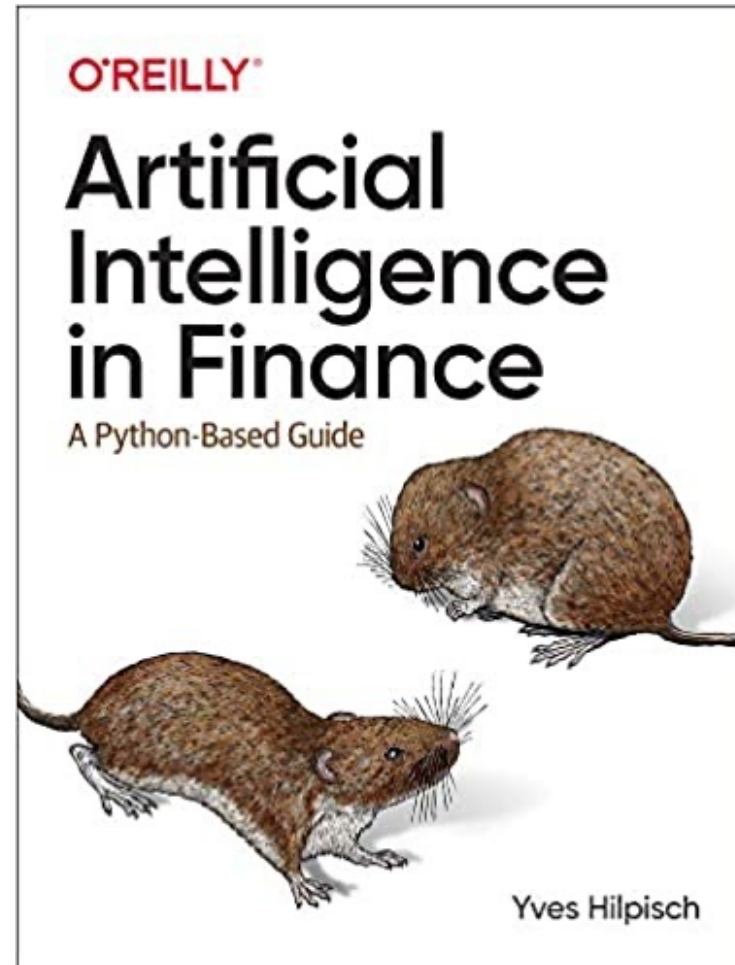
The Quant Finance PyData Stack



Jake VanderPlas

Source: http://nbviewer.jupyter.org/format/slides/github/quantopian/pyfolio/blob/master/pyfolio/examples/overview_slides.ipynb#/5

Yves Hilpisch (2020),
Artificial Intelligence in Finance:
A Python-Based Guide,
O'Reilly



Yves Hilpisch (2020), **Artificial Intelligence in Finance: A Python-Based Guide**, O'Reilly

yhilpisch / aiif Public <https://github.com/yhilpisch/aiif> Notifications Star 98 Fork 77

Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags Go to file Code

yves Code updates for TF 2.3. e334251 on Dec 8, 2020 4 commits


code	Code updates for TF 2.3.	11 months ago
.gitignore	Code updates for TF 2.3.	11 months ago
LICENSE.txt	Code updates.	11 months ago
README.md	Code updates.	11 months ago

README.md

Artificial Intelligence in Finance

About this Repository

This repository provides Python code and Jupyter Notebooks accompanying the **Artificial Intelligence in Finance** book published by [O'Reilly](#).



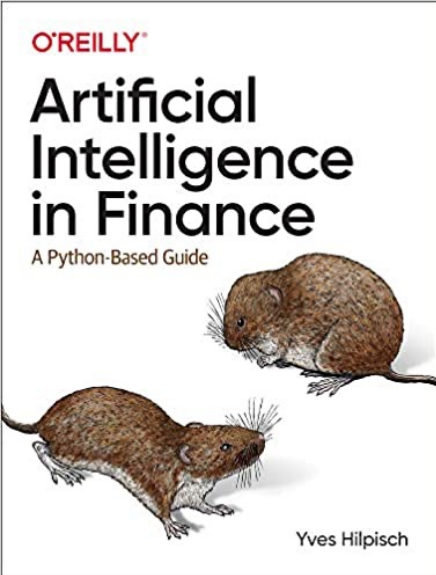
Releases
No releases published

Packages
No packages published

Languages

- Jupyter Notebook 97.4%
- Python 2.6%

About
Jupyter Notebooks and code for the book **Artificial Intelligence in Finance** (O'Reilly) by Yves Hilpisch.
home.tpq.io/books/aiif
Readme
View license



Yves Hilpisch

Yves Hilpisch (2020), **Artificial Intelligence in Finance: A Python-Based Guide**, O'Reilly

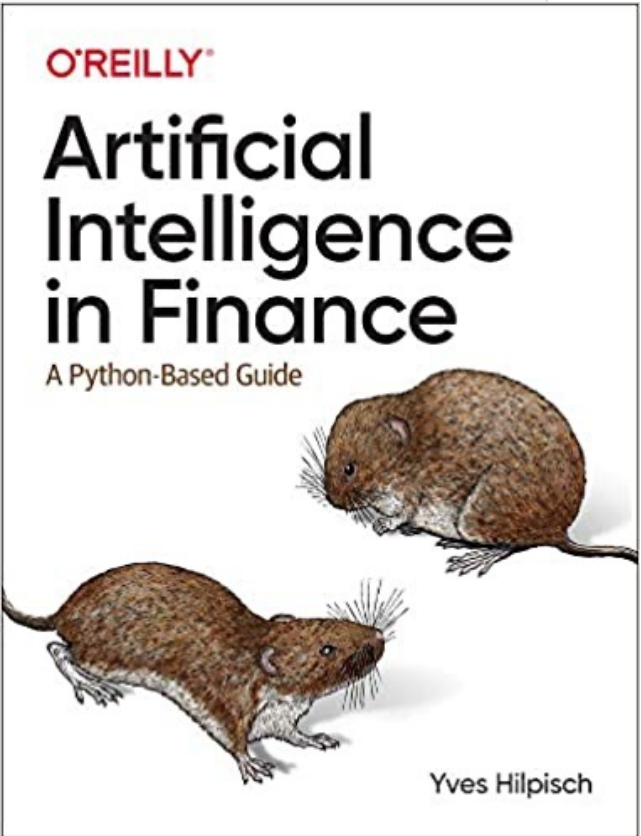
yhilpisch / aiif Public Notifications Star 98 Fork 77

<> Code Issues Pull requests Actions Projects Wiki Security Insights

main aiif / code / <https://github.com/yhilpisch/aiif/tree/main/code> Go to file

yves Code updates for TF 2.3. e334251 on Dec 8, 2020 History

..	
oanda	Code updates for TF 2.3.
01_artificial_intelligence.ipynb	Code updates for TF 2.3.
02_superintelligence.ipynb	Code updates for TF 2.3.
03_normative_finance.ipynb	Code updates for TF 2.3.
04_data_driven_finance_a.ipynb	Initial commit.
04_data_driven_finance_b.ipynb	Initial commit.
05_machine_learning.ipynb	Code updates for TF 2.3.
06_ai_first_finance.ipynb	Code updates for TF 2.3.
07_dense_networks.ipynb	Code updates for TF 2.3.
08_recurrent_networks.ipynb	Code updates for TF 2.3.
09_reinforcement_learning_a.ipynb	Code updates.
09_reinforcement_learning_b.ipynb	Code updates for TF 2.3.



Source: <https://github.com/yhilpisch/aiif/tree/main/code>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share ⚙️ A

RAM Disk Editing ↕

Table of contents

- AI in Finance
 - Normative Finance and Financial Theories
 - Uncertainty and Risk**
 - Expected Utility Theory (EUT)
 - Mean-Variance Portfolio Theory (MVPT)
 - Capital Asset Pricing Model (CAPM)
 - Arbitrage Pricing Theory (APT)
 - Deep Learning for Financial Time Series Forecasting
 - Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python
 - Investment Portfolio Optimization

▼ AI in Finance

- Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.
- Github: <https://github.com/yhilpisch/aiif/>

▼ Normative Finance and Financial Theories

▼ Uncertainty and Risk

```
1 import numpy as np
2
3 #The prices of the stock and bond today.
4 S0 = 10
5 B0 = 10
6 print('S0', S0)
7 print('B0', B0)
8
9 #The uncertain payoff of the stock and bond tomorrow.
10 S1 = np.array((20, 5))
11 B1 = np.array((11, 11))
12 print('S1', S1)
13 print('B1', B1)
14
15 #The market price vector
16 M0 = np.array((S0, B0))
```

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)



python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings Profile

RAM Disk Editing

Table of contents

- Data Driven Finance
 - Financial Econometrics and Regression**
 - Data Availability
 - Normative Theories Revisited
 - Mean-Variance Portfolio Theory
 - Capital Asset Pricing Model
 - Arbitrage-Pricing Theory
 - Debunking Central Assumptions
 - Normality
 - Sample Data Sets
 - Real Financial Returns
 - Linear Relationships
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python
 - Investment Portfolio Optimization

▼ Data Driven Finance

▼ Financial Econometrics and Regression

```
[18] 1 import numpy as np
      2
      3 def f(x):
      4     return 2 + 1 / 2 * x
      5
      6 x = np.arange(-4, 5)
      7 x
```

```
array([-4, -3, -2, -1,  0,  1,  2,  3,  4])
```

```
1 y = f(x)
2 y
```

```
array([ 0.00,  0.50,  1.00,  1.50,  2.00,  2.50,  3.00,  3.50,  4.00])
```

```
1 print('x', x)
2
3 print('y', y)
4
5 beta = np.cov(x, y, ddof=0)[0, 1] / x.var()
6 print('beta', beta)
```

Python in Google Colab (Python101)



python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment

Share



A

RAM
Disk

Editing



Table of contents



- Data Driven Finance
 - Financial Econometrics and Regression
 - Data Availability
 - Normative Theories Revisited
 - Mean-Variance Portfolio Theory**
 - Capital Asset Pricing Model
 - Arbitrage-Pricing Theory
 - Debunking Central Assumptions
 - Normality
 - Sample Data Sets
 - Real Financial Returns
 - Linear Relationships
- Financial Econometrics and Machine Learning
 - Machine Learning
 - Data
 - Success
 - Capacity
 - Evaluation
 - Bias & Variance

+ Code + Text

Normative Theories Revisited

Mean-Variance Portfolio Theory

```
1 import numpy as np
2 import pandas as pd
3 from pylab import plt, mpl
4 from scipy.optimize import minimize
5 plt.style.use('seaborn')
6 mpl.rcParams['savefig.dpi'] = 300
7 mpl.rcParams['font.family'] = 'serif'
8 np.set_printoptions(precision=5, suppress=True,
9                     formatter={'float': lambda x: f'{x:6.3f}'})
10
11 url = 'http://hilpisch.com/aiif_eikon_eod_data.csv'
12
13 raw = pd.read_csv(url, index_col=0, parse_dates=True).dropna()
14 raw.info()
15
16 symbols = ['AAPL.O', 'MSFT.O', 'INTC.O', 'AMZN.O', 'GLD']
17
18 rets = np.log(raw[symbols] / raw[symbols].shift(1)).dropna()
19
20 (raw[symbols[:]] / raw[symbols[:]].iloc[0]).plot(figsize=(10, 6));
21
22 weights = len(rets.columns) * [1 / len(rets.columns)]
23 weights
```

<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)



python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment

Share



A

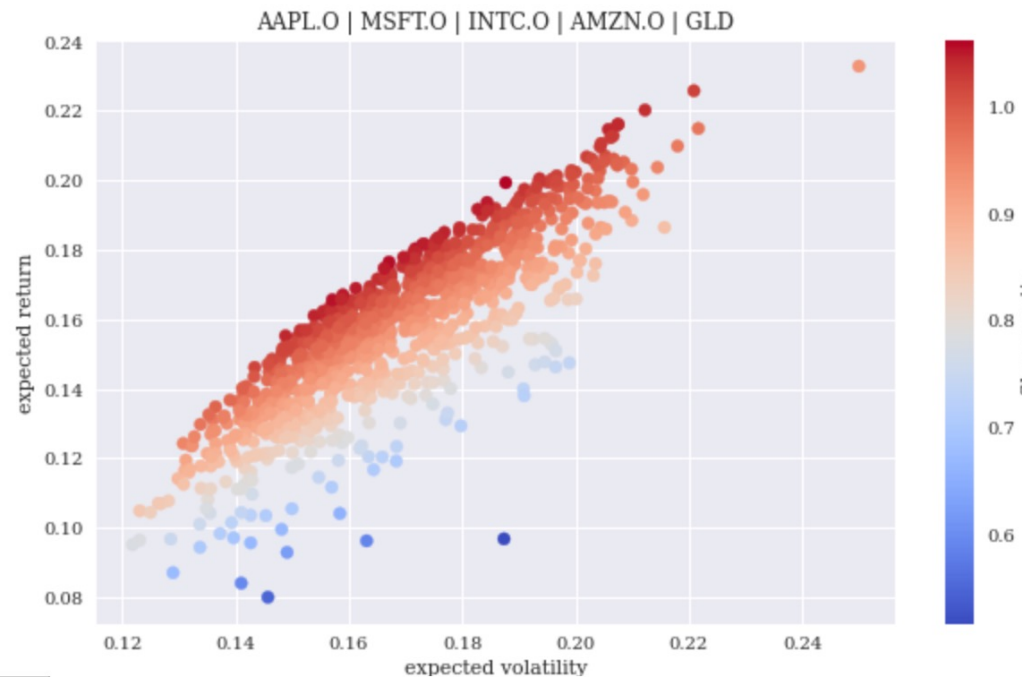
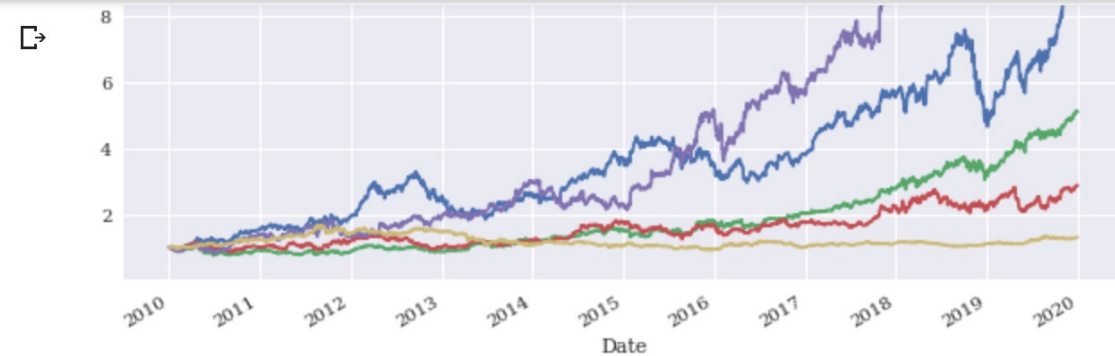
Table of contents

- Data Driven Finance
 - Financial Econometrics and Regression
 - Data Availability
 - Normative Theories Revisited
 - Mean-Variance Portfolio Theory**
 - Capital Asset Pricing Model
 - Arbitrage-Pricing Theory
 - Debunking Central Assumptions
 - Normality
 - Sample Data Sets
 - Real Financial Returns
 - Linear Relationships
 - Deep Learning for Financial Time Series Forecasting
 - Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python
 - Investment Portfolio Optimization

+ Code + Text

RAM Disk

Editing



<https://tinyurl.com/aintpupython101>

Python in Google Colab (Python101)



python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment

Share



A

Table of contents

Data Driven Finance

Financial Econometrics and Regression

Data Availability

Normative Theories Revisited

Mean-Variance Portfolio Theory

Capital Asset Pricing Model

Arbitrage-Pricing Theory

Debunking Central Assumptions

Normality

Sample Data Sets

Real Financial Returns

Linear Relationships

Deep Learning for Financial Time Series Forecasting

Portfolio Optimization and Algorithmic Trading

Investment Portfolio Optimisation with Python

Efficient Frontier Portfolio Optimisation in Python

Investment Portfolio Optimization

+ Code + Text

RAM Disk

Editing

AMZN.O

Year	beta	mu_capm	mu_real
2011	1.102	-0.001	-0.039
2012	0.958	0.122	0.374
2013	1.116	0.289	0.464
2014	1.262	0.135	-0.251
2015	1.473	-0.013	0.778
2016	1.122	0.102	0.104
2017	1.118	0.199	0.446
2018	1.300	-0.086	0.251
2019	1.619	0.408	0.207



Python in Google Colab (Python101)



python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings Profile

RAM Disk Editing



- Table of contents
- Data Driven Finance
 - Financial Econometrics and Regression
 - Data Availability
 - Normative Theories Revisited
 - Mean-Variance Portfolio Theory
 - Capital Asset Pricing Model
 - Arbitrage-Pricing Theory**
 - Debunking Central Assumptions
 - Normality
 - Sample Data Sets
 - Real Financial Returns
 - Linear Relationships
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python
 - Investment Portfolio Optimization

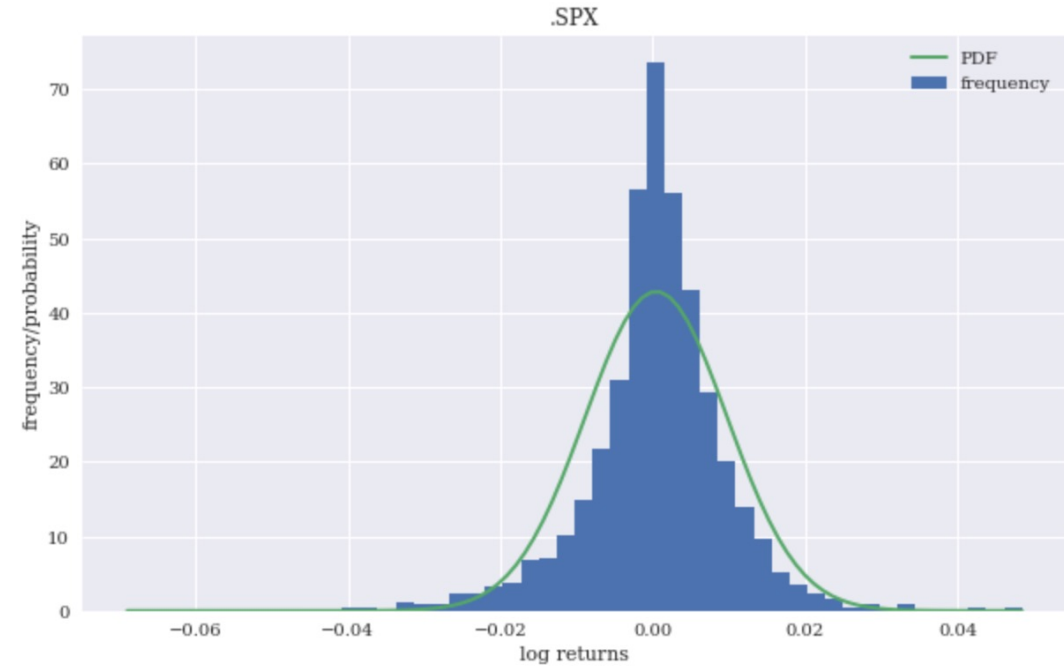
```
92 rets_sym.cumsum().apply(np.exp).plot(figsize=(10, 6));
93
94 rets_sym['same'] = (np.sign(rets_sym[sym + '_apt']) ==
95                    np.sign(rets_sym[sym + '_real']))
96
97 rets_sym['same'].value_counts()
98
99 rets_sym['same'].value_counts()[True] / len(rets_sym)
```



Python in Google Colab (Python101)

- Table of contents
- Data Driven Finance
 - Financial Econometrics and Regression
 - Data Availability
 - Normative Theories Revisited
 - Mean-Variance Portfolio Theory
 - Capital Asset Pricing Model
 - Arbitrage-Pricing Theory
 - Debunking Central Assumptions
 - Normality
 - Sample Data Sets
 - Real Financial Returns**
 - Linear Relationships
- Deep Learning for Financial Time Series Forecasting
- Portfolio Optimization and Algorithmic Trading
 - Investment Portfolio Optimisation with Python
 - Efficient Frontier Portfolio Optimisation in Python
 - Investment Portfolio Optimization

```
GLD
=====
RETURN SAMPLE STATISTICS
-----
Skew of Sample Log Returns -0.581025
Skew Normal Test p-value 0.000000
-----
Kurt of Sample Log Returns 5.899701
Kurt Normal Test p-value 0.000000
-----
Normal Test p-value 0.000000
-----
```



Python in Google Colab (Python101)

Table of contents

Data Driven Finance

Financial Econometrics and Regression

Data Availability

Normative Theories Revisited

Mean-Variance Portfolio Theory

Capital Asset Pricing Model

Arbitrage-Pricing Theory

Debunking Central Assumptions

Normality

Sample Data Sets

Real Financial Returns

Linear Relationships

Deep Learning for Financial Time Series Forecasting

Portfolio Optimization and Algorithmic Trading

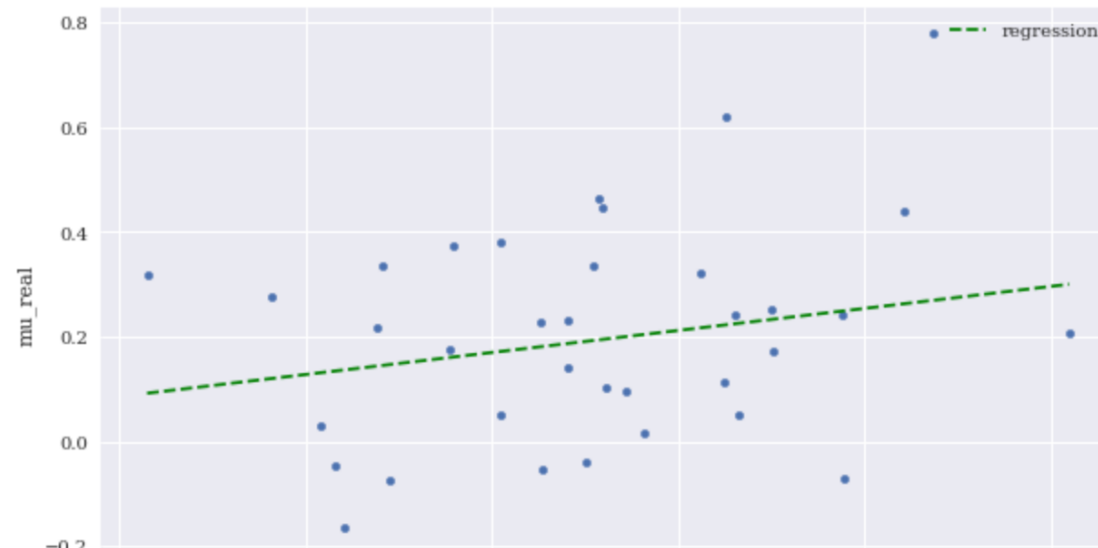
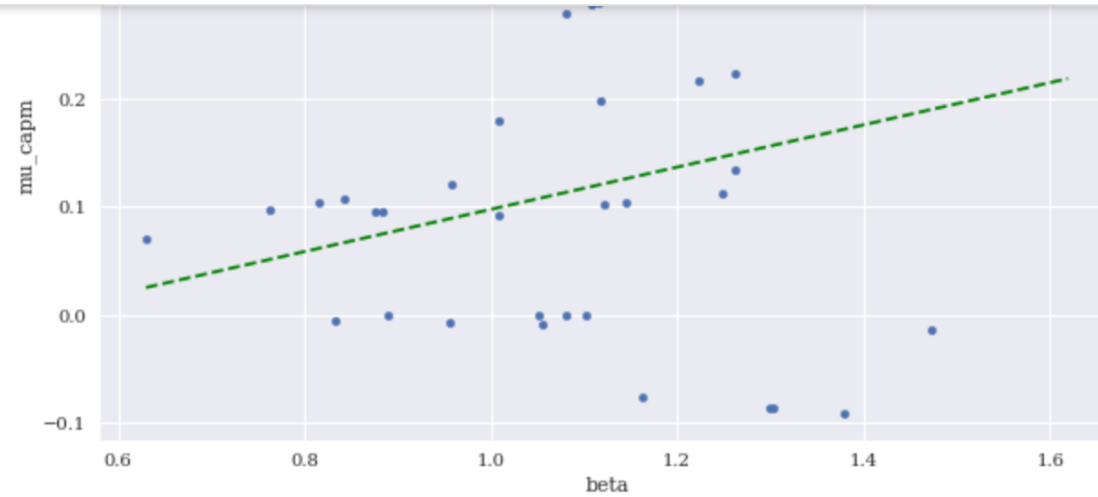
Investment Portfolio Optimisation with Python

Efficient Frontier Portfolio Optimisation in Python

Investment Portfolio Optimization

+ Code + Text

RAM Disk Editing



Summary

- **Data-Driven Finance**
- **Scientific Method**
- **Financial Econometrics and Regression**
- **Data Availability**
- **Normative Theories Revisited**
- **Debunking Central Assumptions in Finance**

References

- Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media, <https://github.com/yhilpisch/aiif> .
- Fishburn, P. C. (1968). Utility theory. Management science, 14(5), 335-378.
- Fama, E. F., & French, K. R. (2004). The capital asset pricing model: Theory and evidence. Journal of economic perspectives, 18(3), 25-46.
- Markowitz, H. (1952). PORTFOLIO SELECTION. The Journal of Finance, 7(1), 77-91.
- Pratt, J. W. (1964). Risk aversion in the small and in the large, econometrics 32, jan.
- Ross, S. A. (1976). The arbitrage theory of capital asset pricing. Journal of Economic Theory, 13(3), 341-60.
- Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. The journal of finance, 19(3), 425-442.
- Hui Gong (2026) "AI Agents in Financial Markets: Architecture, Applications, and Systemic Implications." arXiv preprint arXiv:2603.13942.
- Min-Yuh Day (2026), Python 101, <https://tinyurl.com/aintpupython101>