

智慧金融量化分析

(Artificial Intelligence in Finance and Quantitative Analysis)

數據驅動財務金融 (Data-Driven Finance)

1101AIFQA06

MBA, IM, NTPU (M6132) (Fall 2021)

Tue 2, 3, 4 (9:10-12:00) (8F40)

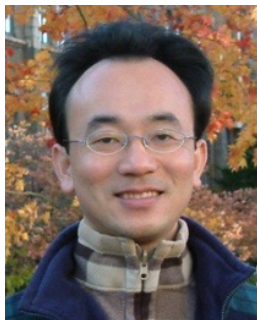
戴敏育 副教授

Min-Yuh Day, Ph.D, Associate Professor

國立臺北大學 資訊管理研究所

Institute of Information Management, National Taipei University

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



課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2021/09/28	智慧金融量化分析概論 (Introduction to Artificial Intelligence in Finance and Quantitative Analysis)
2	2021/10/05	AI 金融科技: 金融服務創新應用 (AI in FinTech: Financial Services Innovation and Application)
3	2021/10/12	投資心理學與行為財務學 (Investing Psychology and Behavioral Finance)
4	2021/10/19	財務金融事件研究法 (Event Studies in Finance)
5	2021/10/26	智慧金融量化分析個案研究 I (Case Study on AI in Finance and Quantitative Analysis I)
6	2021/11/02	財務金融理論 (Finance Theory)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
7	2021/11/09	數據驅動財務金融 (Data-Driven Finance)
8	2021/11/16	期中報告 (Midterm Project Report)
9	2021/11/23	金融計量經濟學 (Financial Econometrics)
10	2021/11/30	人工智慧優先金融 (AI-First Finance)
11	2021/12/07	智慧金融量化分析產業實務 (Industry Practices of AI in Finance and Quantitative Analysis)
12	2021/12/14	智慧金融量化分析個案研究 II (Case Study on AI in Finance and Quantitative Analysis II)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
13	2021/12/21	財務金融深度學習 (Deep Learning in Finance); 財務金融強化學習 (Reinforcement Learning in Finance)
14	2021/12/28	演算法交易 (Algorithmic Trading); 風險管理 (Risk Management); 交易機器人與基於事件的回測 (Trading Bot and Event-Based Backtesting)
15	2022/01/04	期末報告 I (Final Project Report I)
16	2022/01/11	期末報告 II (Final Project Report II)
17	2022/01/18	學生自主學習 (Self-learning)
18	2022/01/25	學生自主學習 (Self-learning)

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

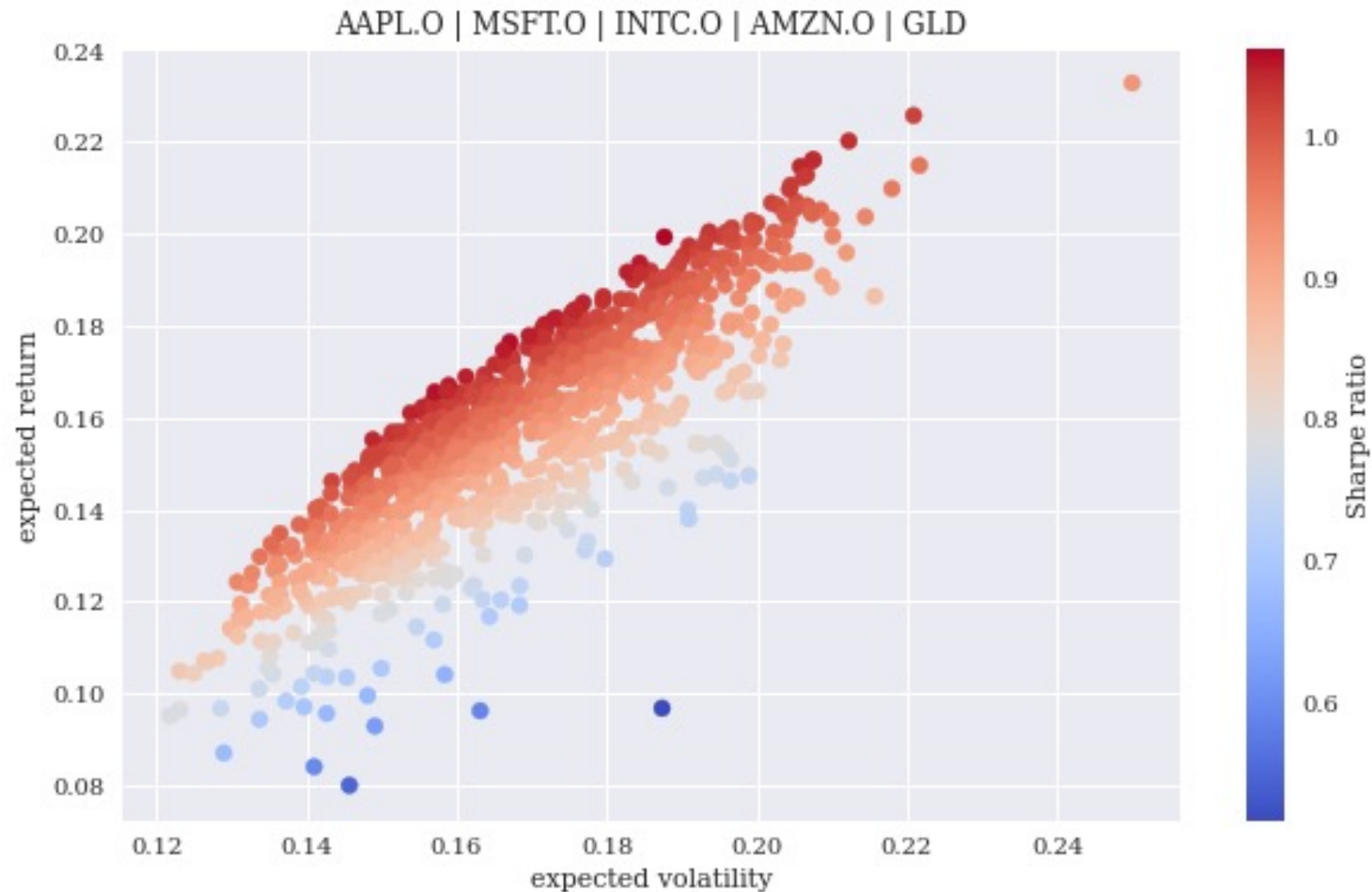
Normative Theories Revisited

- 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



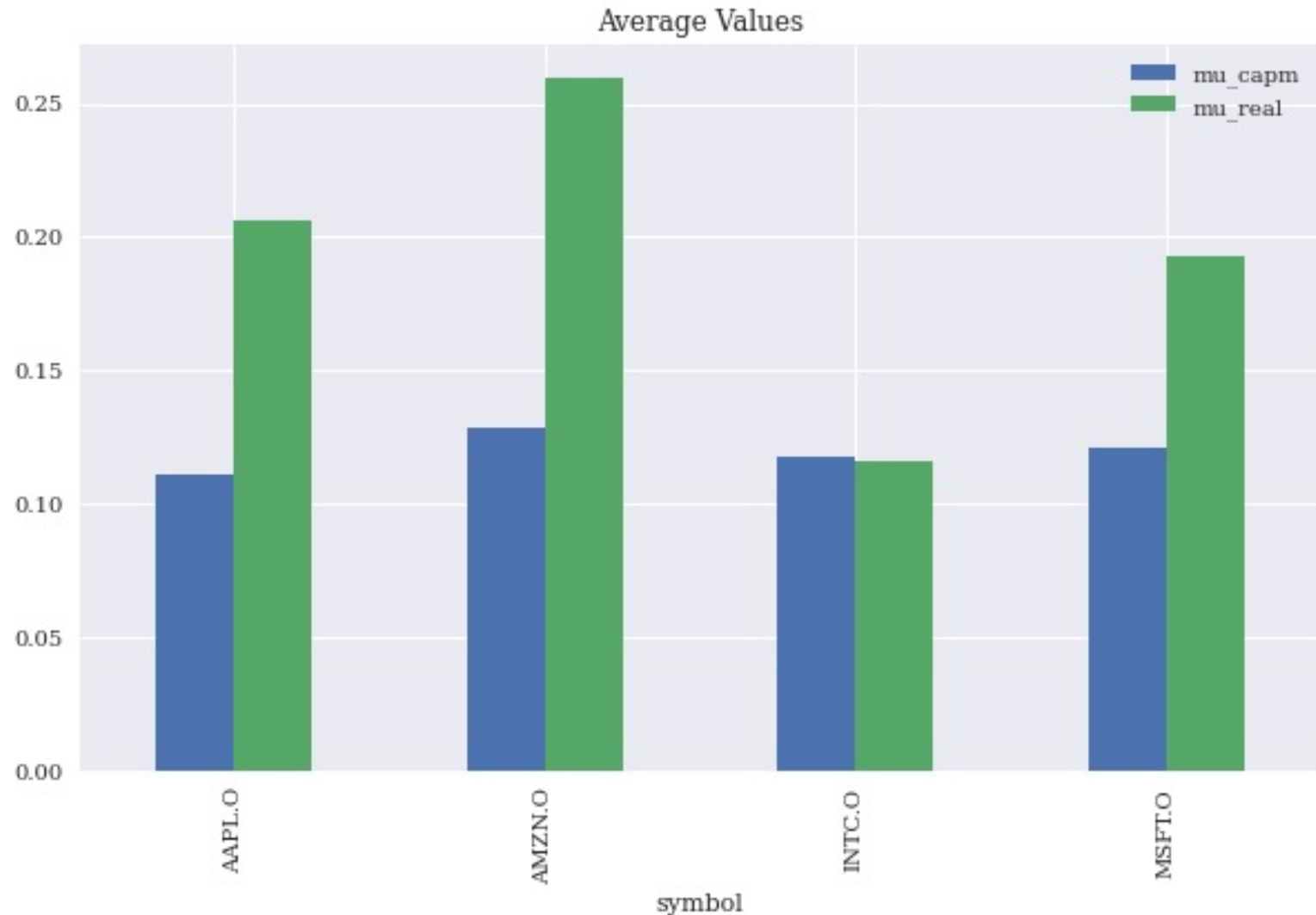
Expected versus realized portfolio Sharpe ratios



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

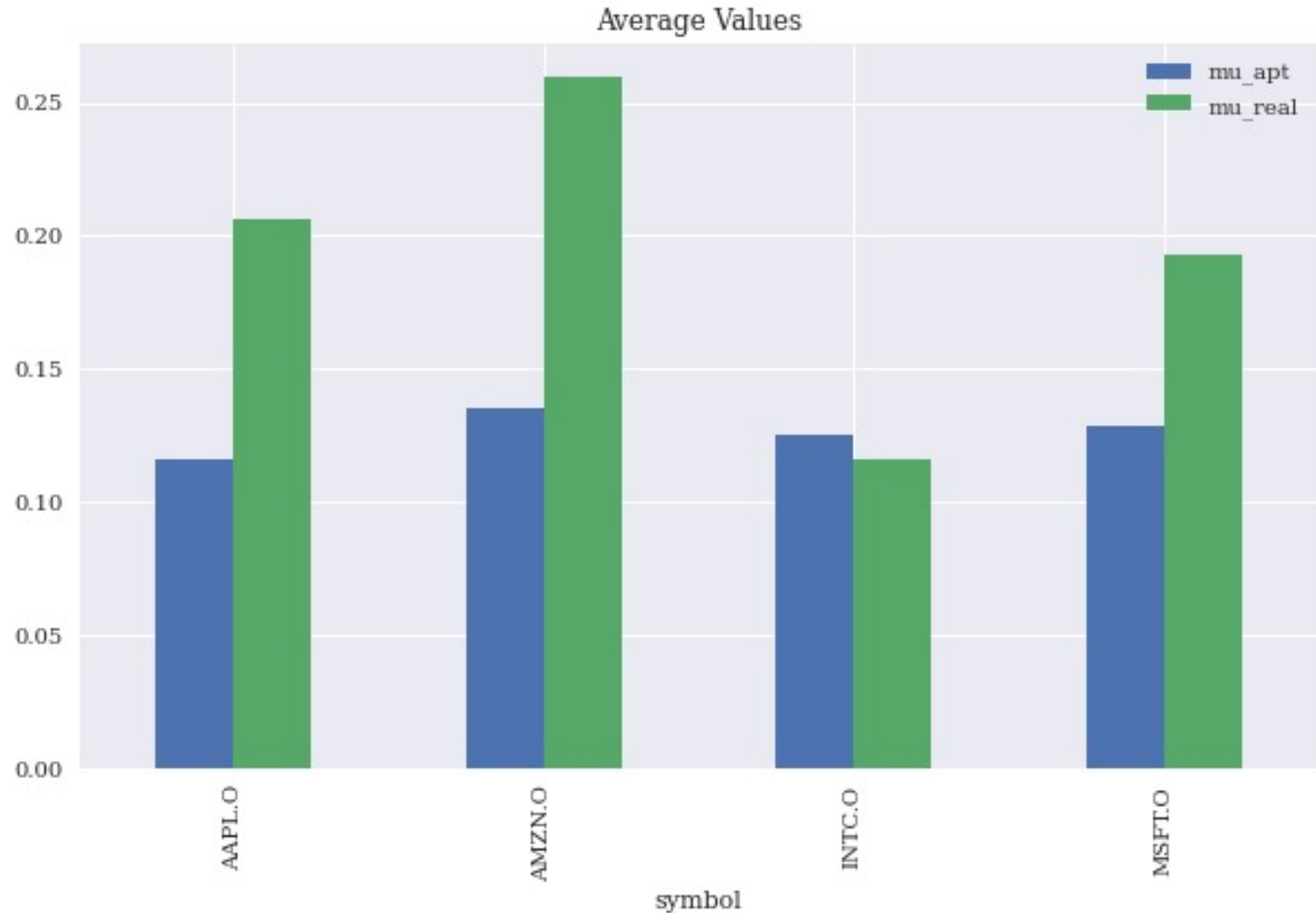
Factor	Description
Market	MSCI World Gross Return Daily USD (PUS = Price Return)
Size	MSCI World Equal Weight Price Net Index EOD
Volatility	MSCI World Minimum Volatility Net Return
Value	MSCI World Value Weighted Gross (NUS for Net)
Risk	MSCI World Risk Weighted Gross USD EOD
Growth	MSCI World Quality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



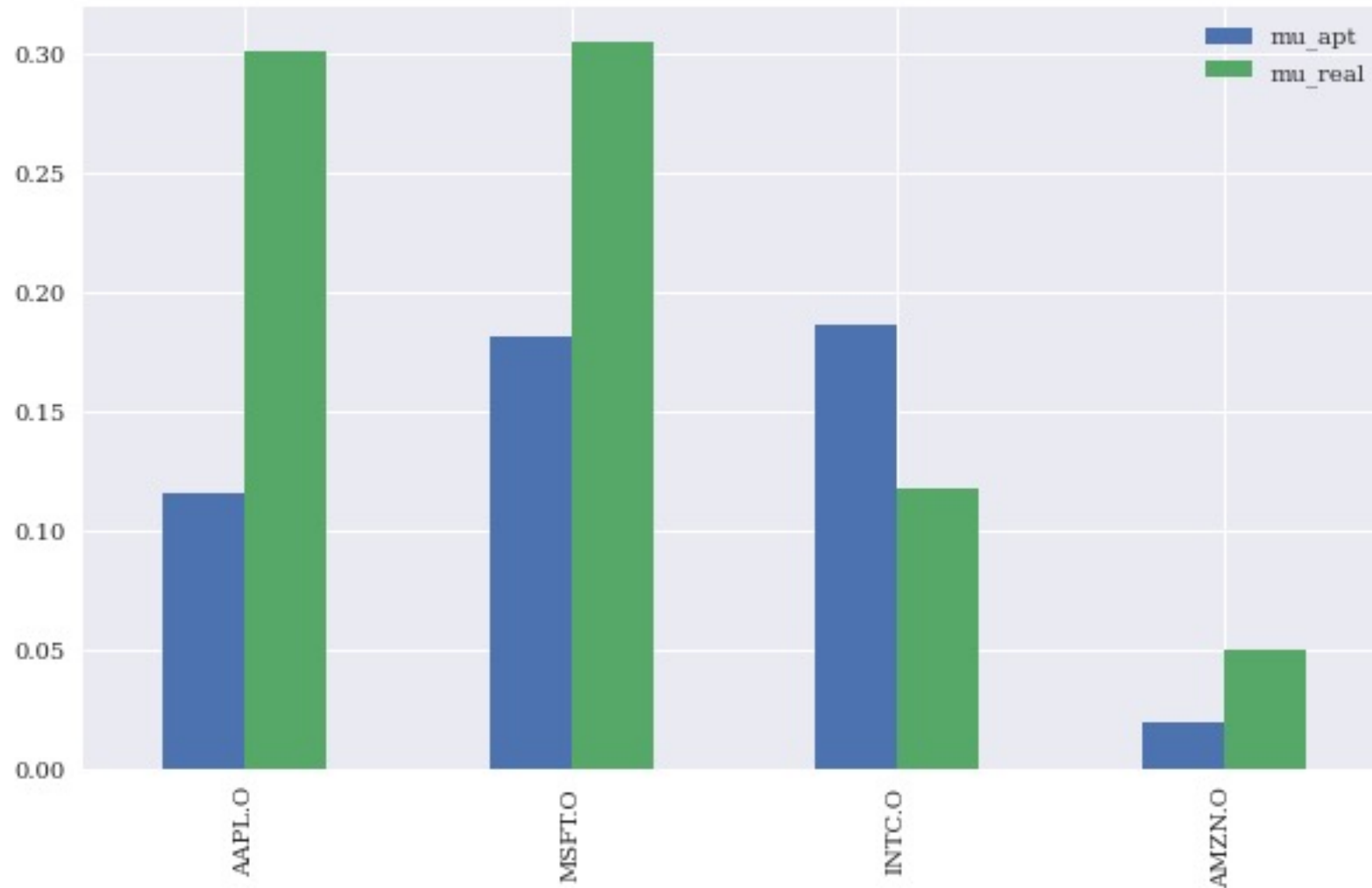
Average APT-predicted versus average realized stock returns for multiple stocks



Normalized factors time series data



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



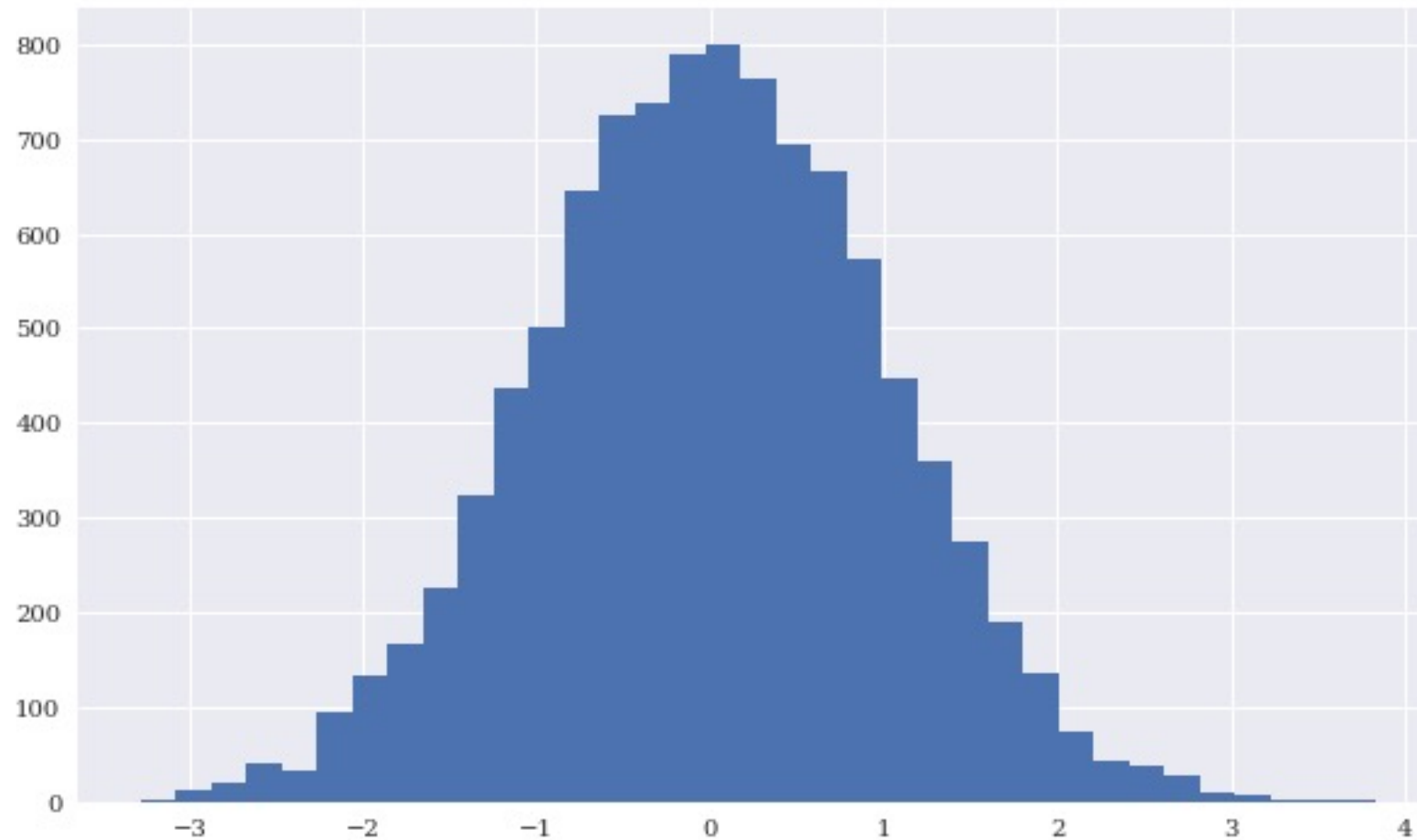
APT-predicted performance and real performance over time (gross)



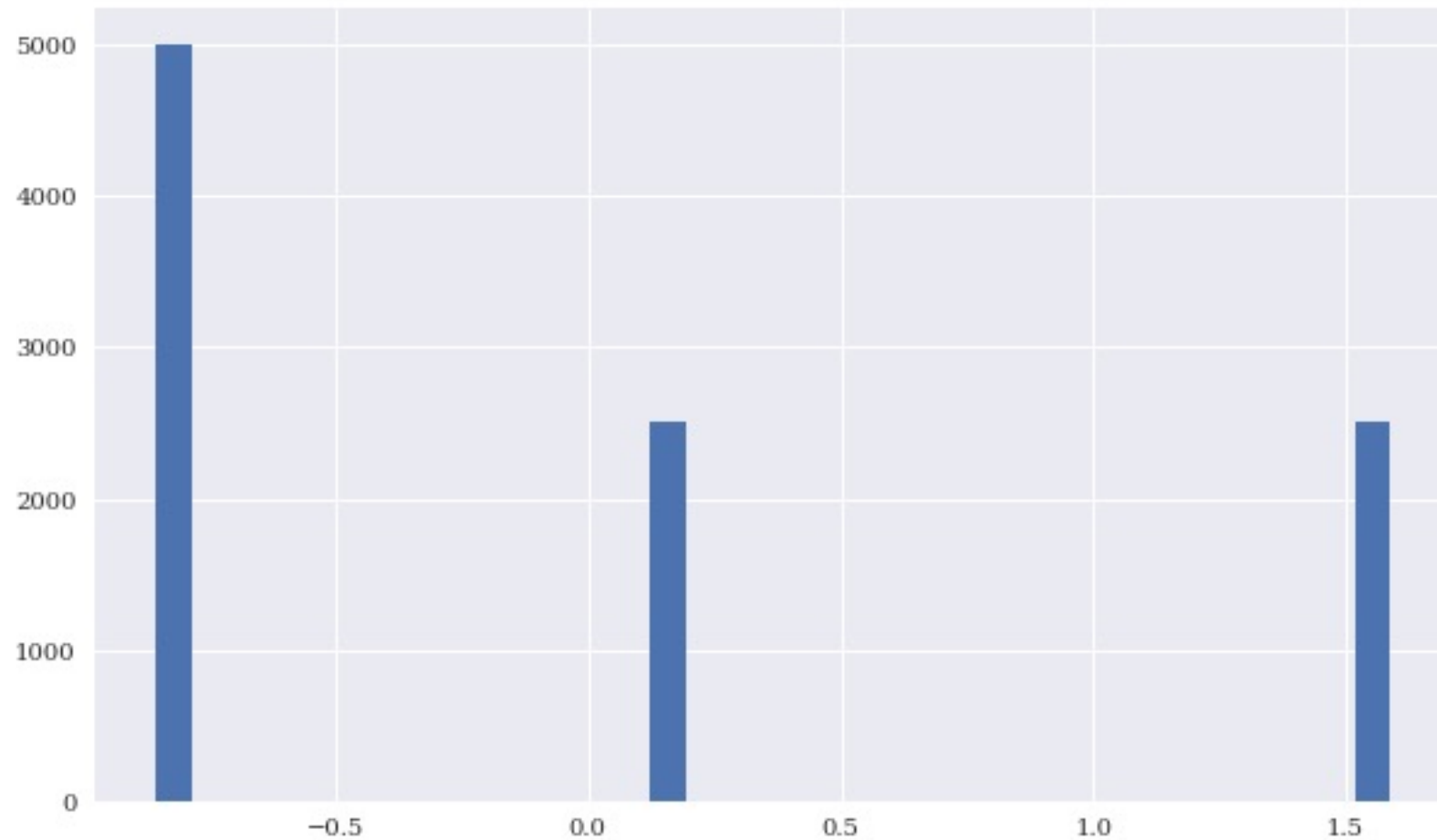
Debunking Central Assumptions in Finance

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

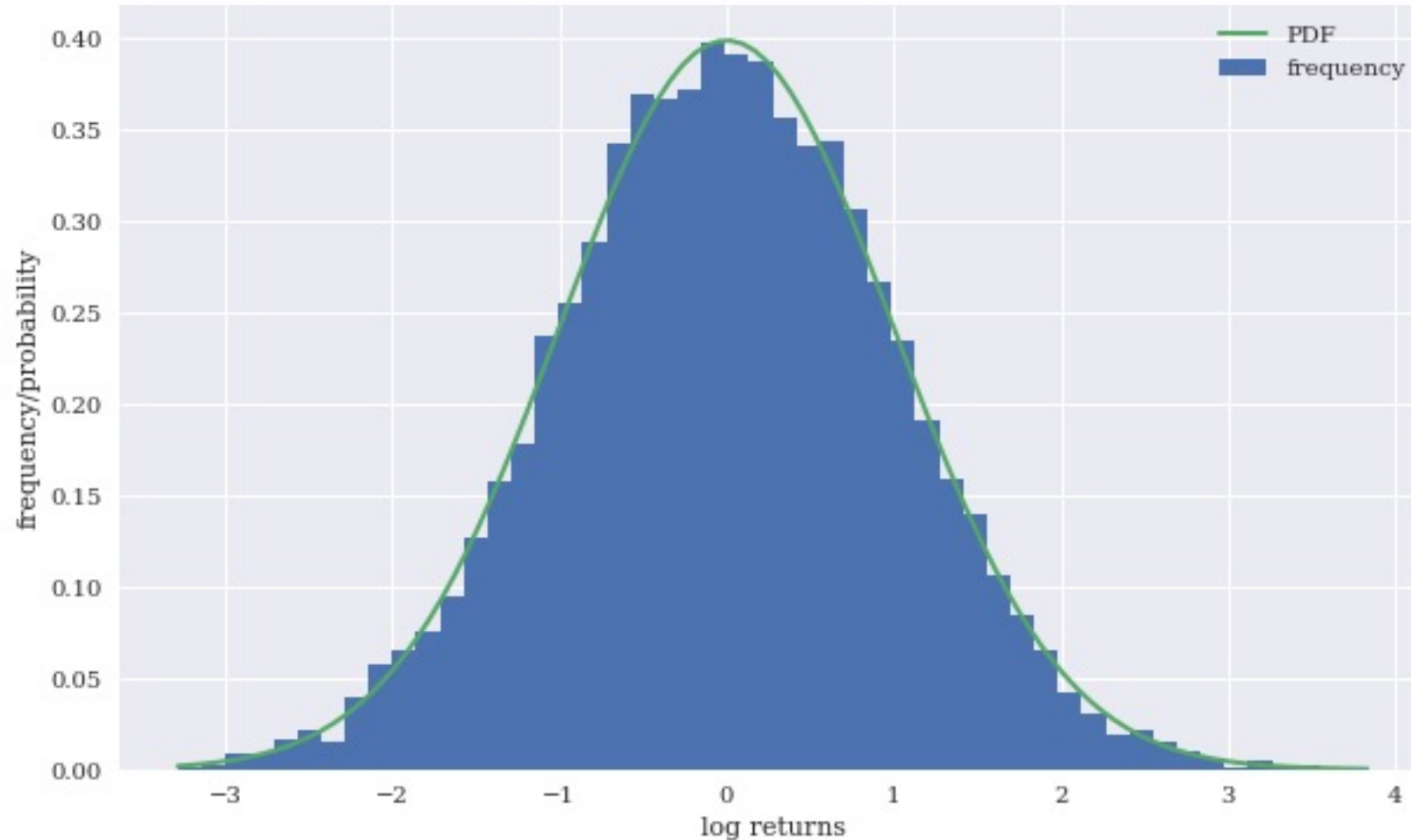
Standard normally distributed random numbers



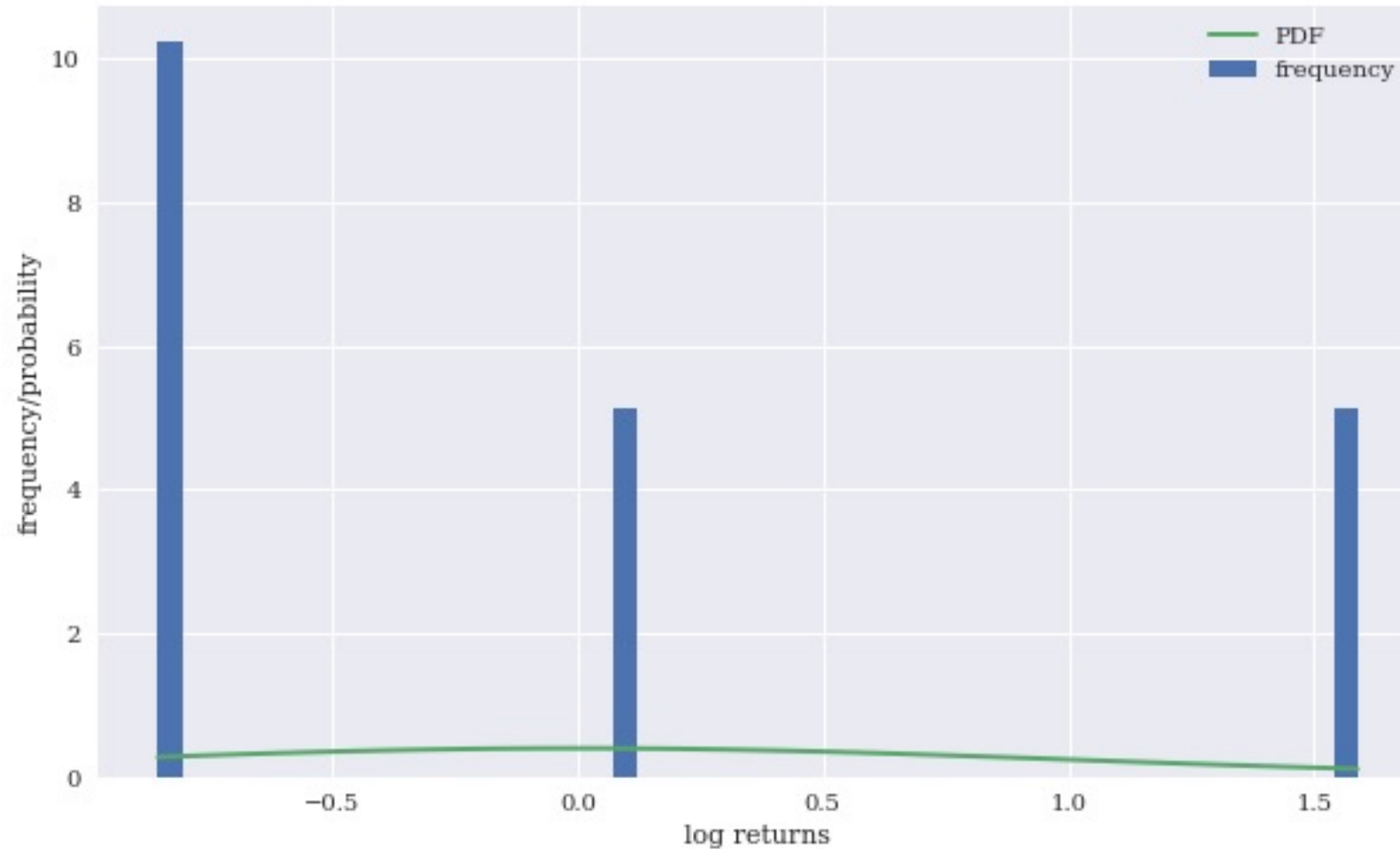
Distribution with first and second moment of 0.0 and 1.0, respectively



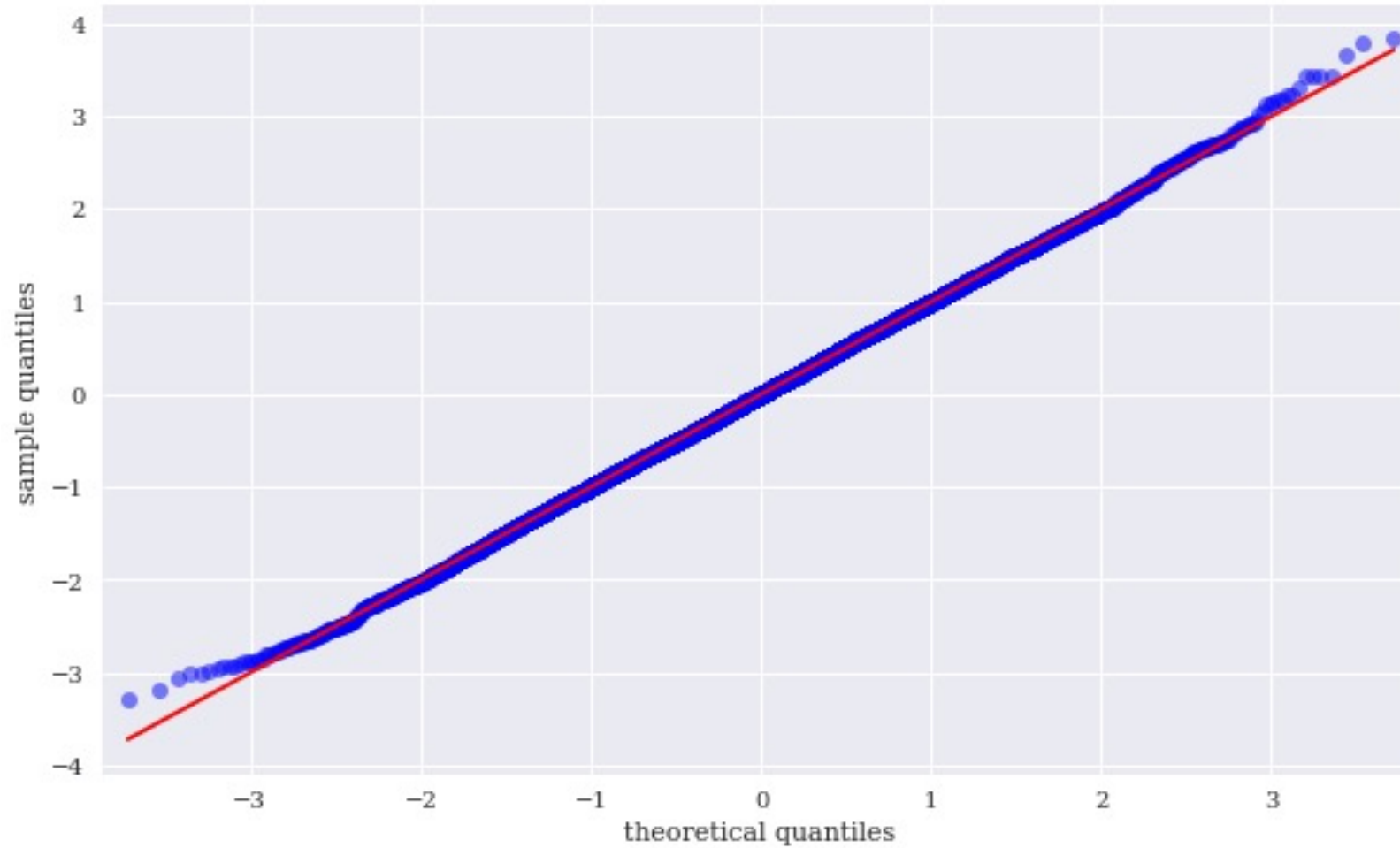
Histogram and PDF for standard normally distributed numbers



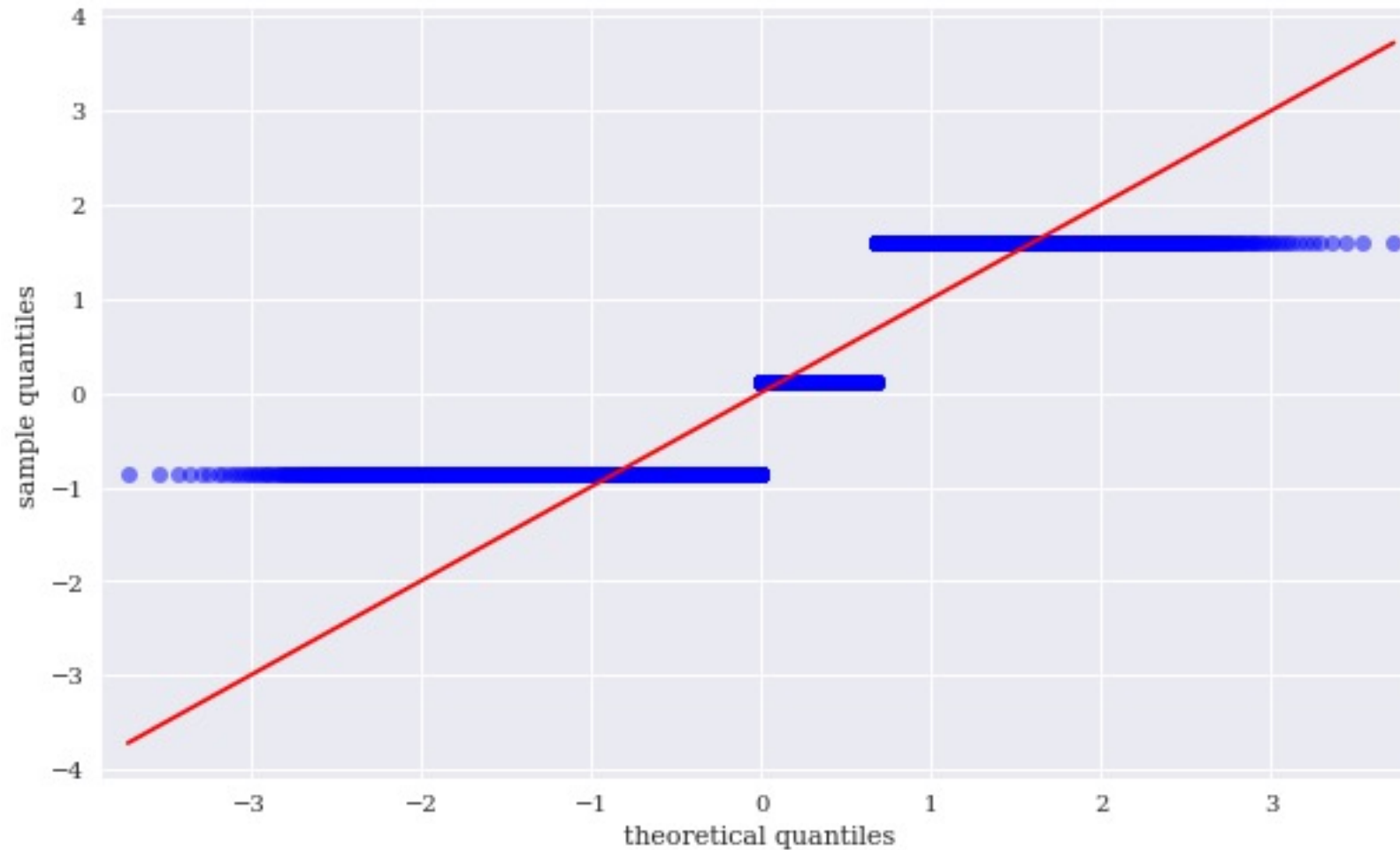
Histogram and normal PDF for discrete numbers



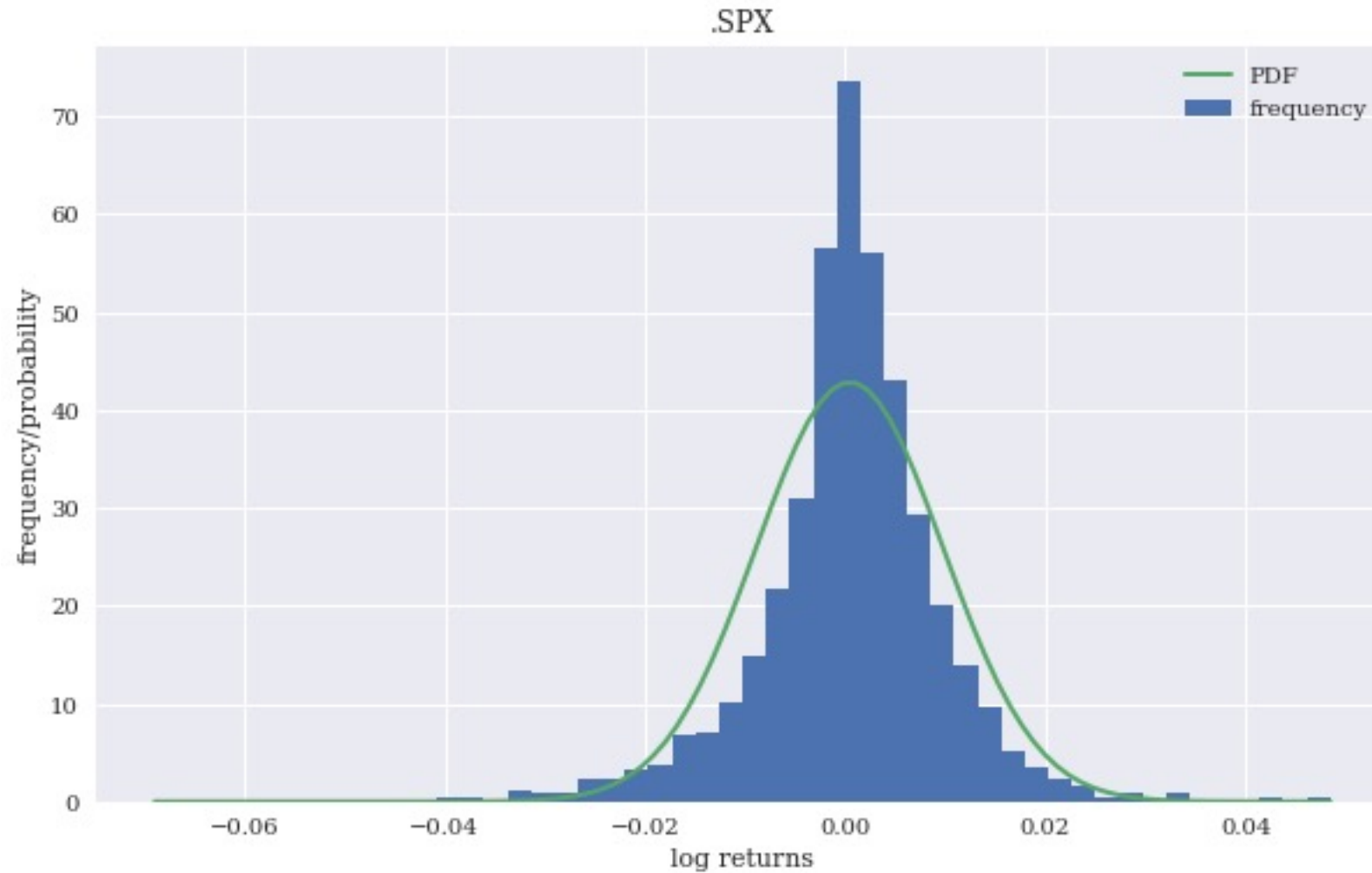
Q-Q plot for standard normally distributed numbers



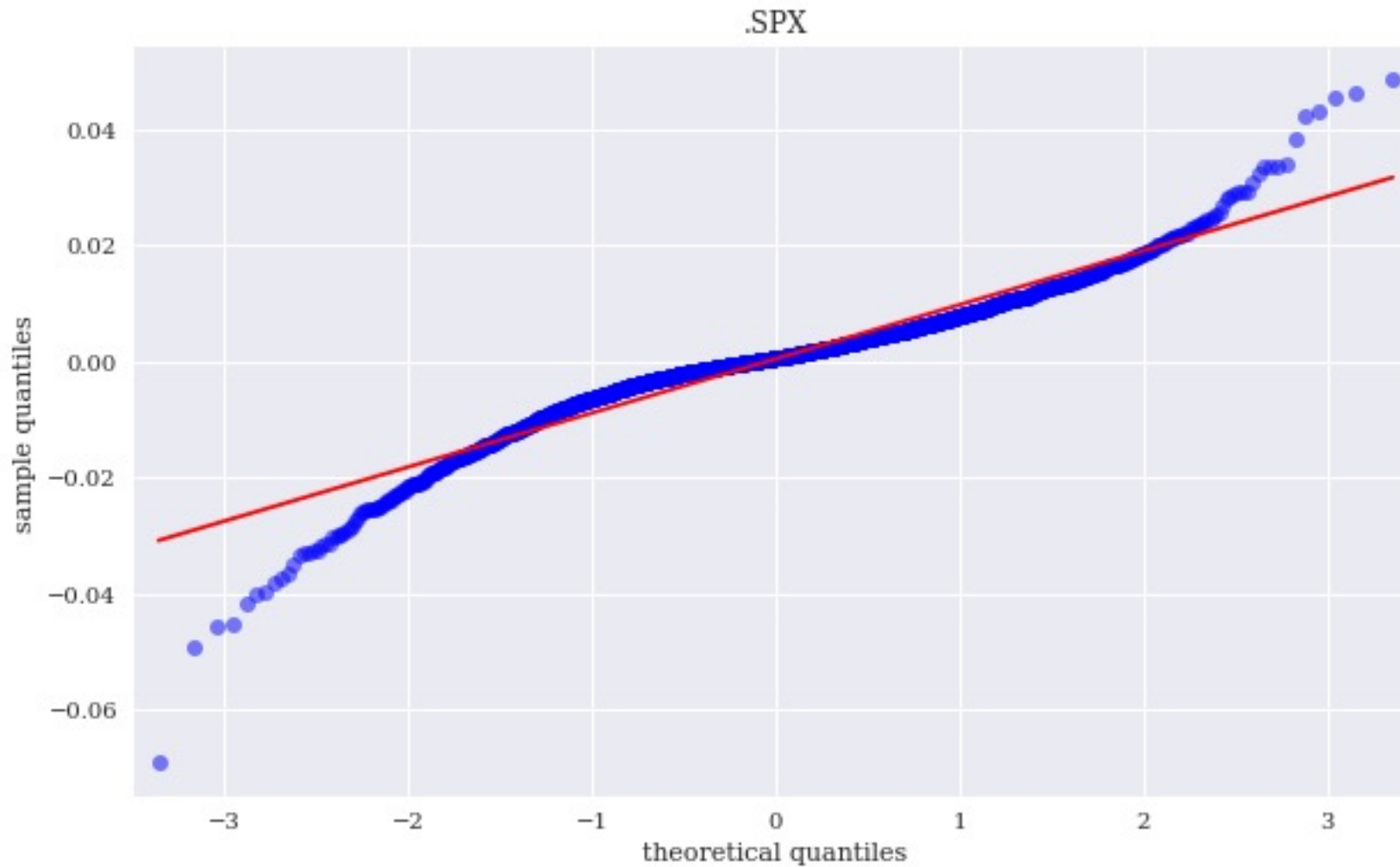
Q-Q plot for discrete numbers



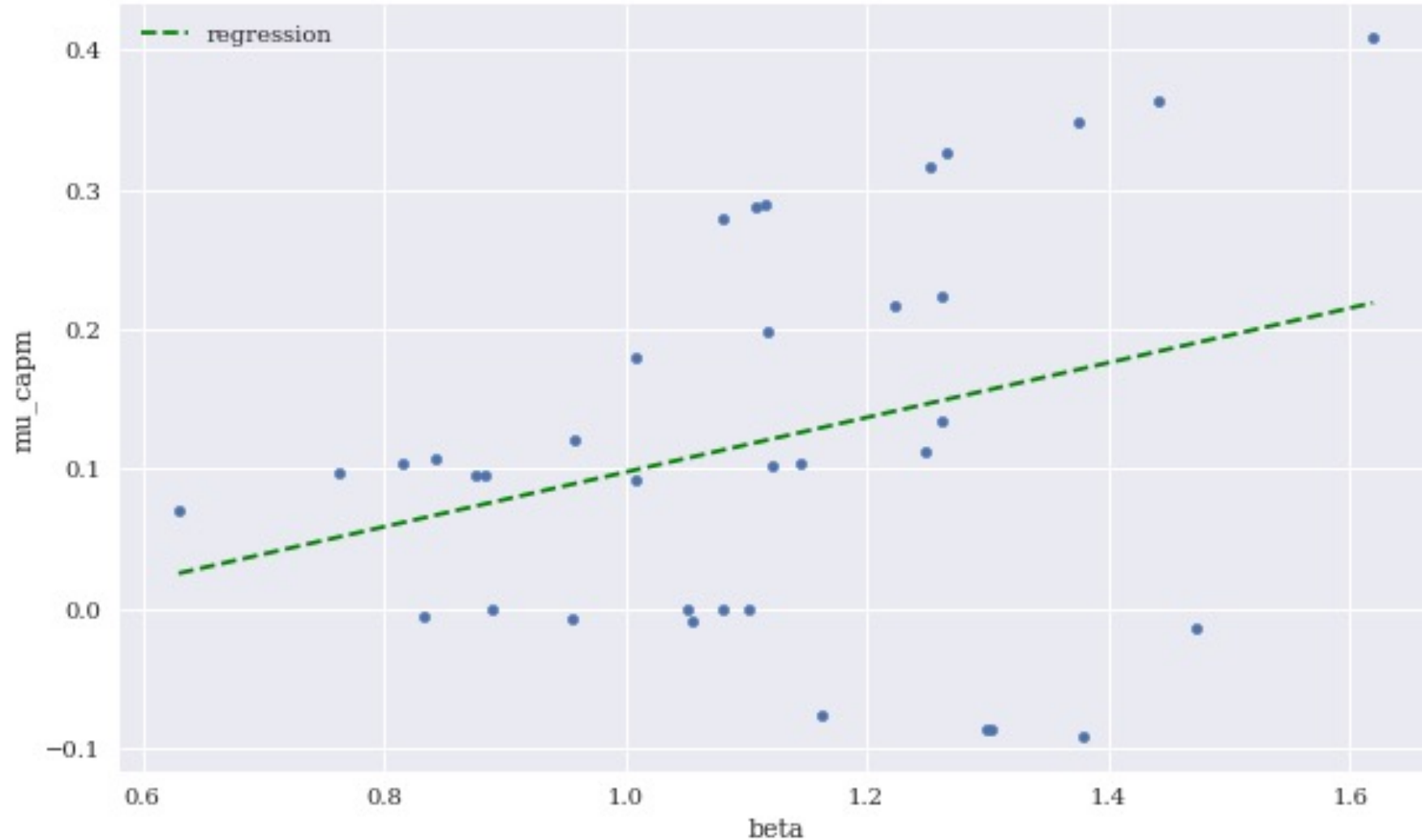
Frequency distribution and normal PDF for S&P 500 log returns



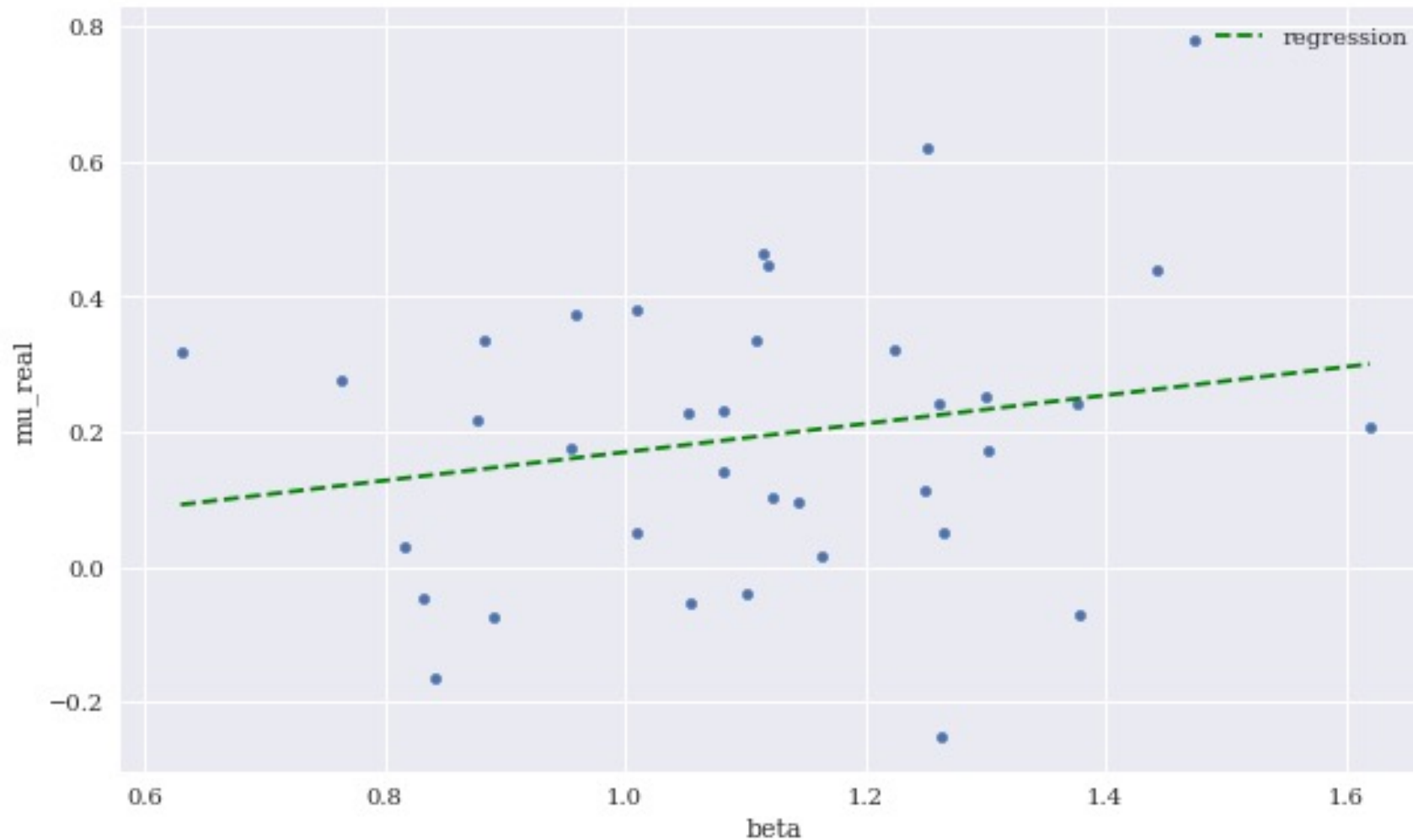
Q-Q for S&P 500 log returns



Expected CAPM return versus beta (including linear regression)



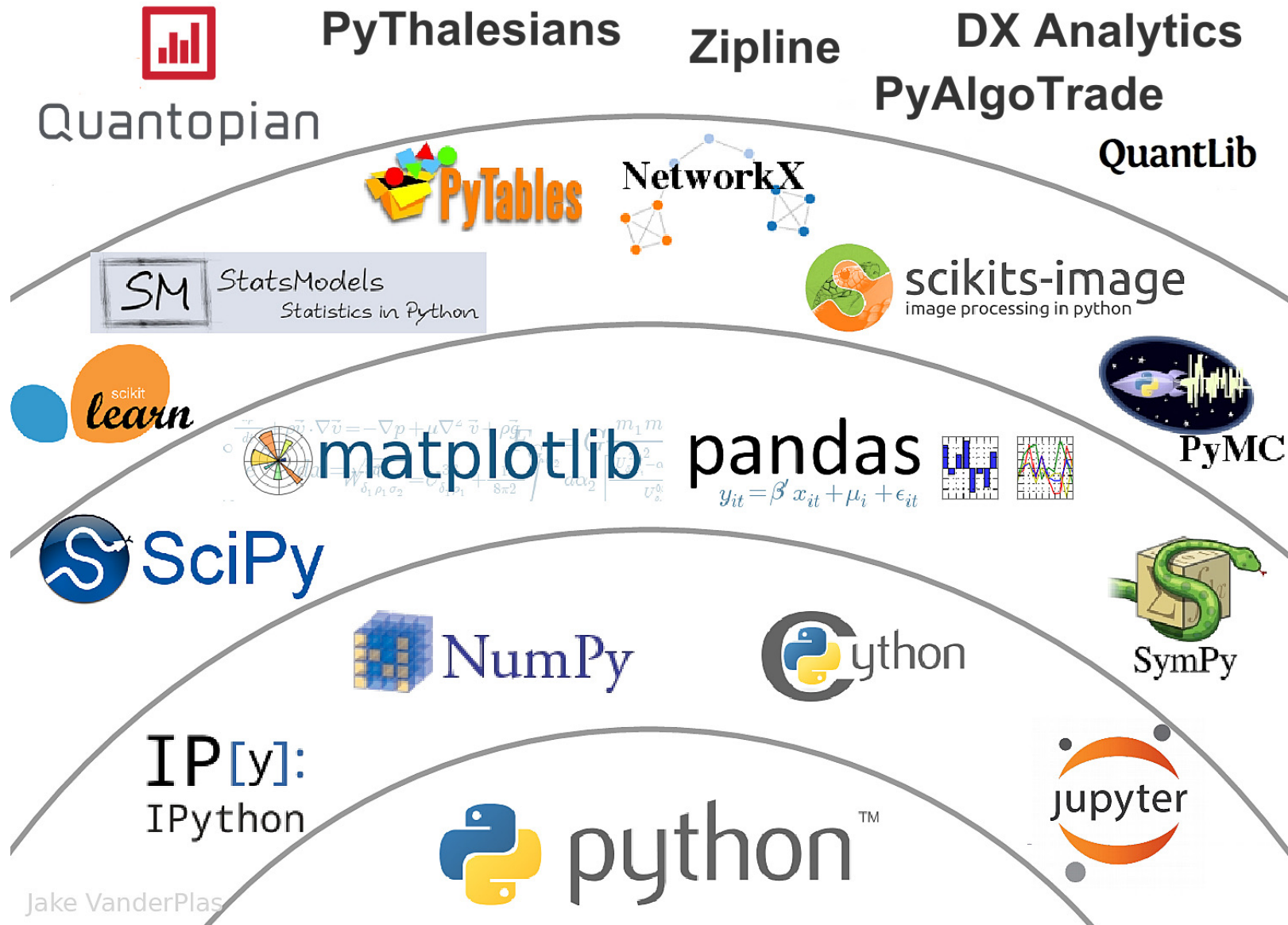
Expected CAPM return versus beta (including linear regression)



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.

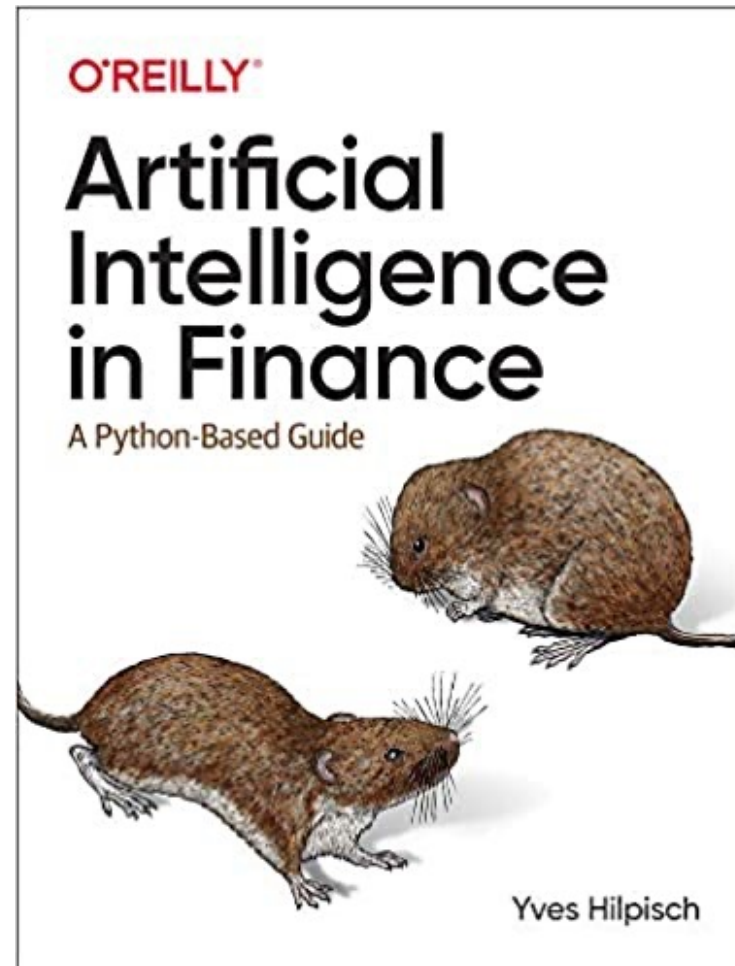
The Quant Finance PyData Stack



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](#).



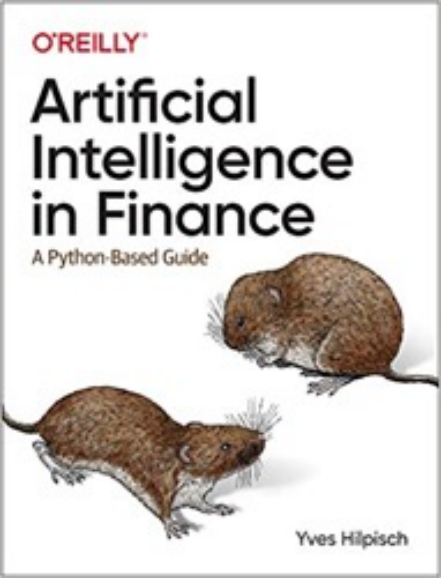
About
Jupyter Notebooks and code for the book **Artificial Intelligence in Finance** (O'Reilly) by Yves Hilpisch.
home.tpq.io/books/aiif
Readme
View license

Releases
No releases published

Packages
No packages published

Languages

Jupyter Notebook	97.4%
Python	2.6%



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

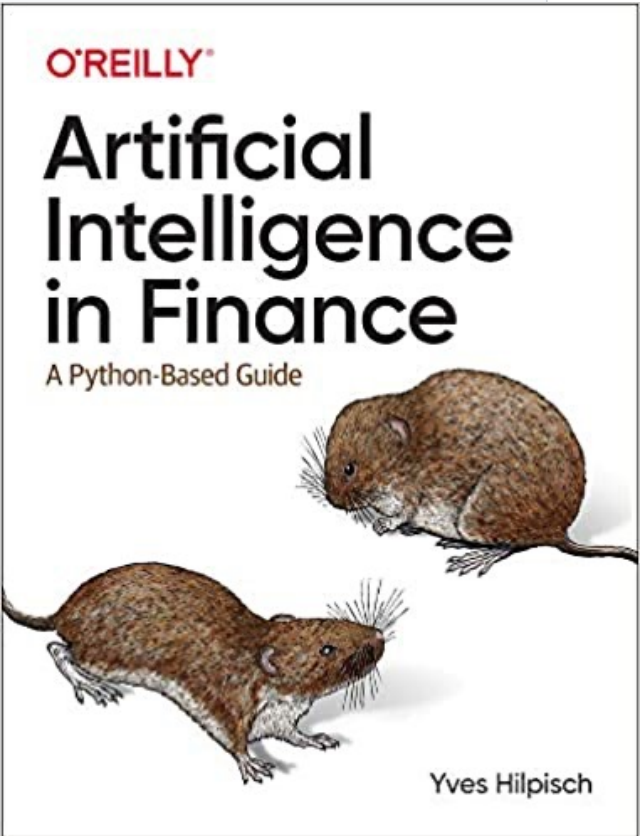
yhilpisch / aiif Public Notifications Star 98 Fork 77

<> 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

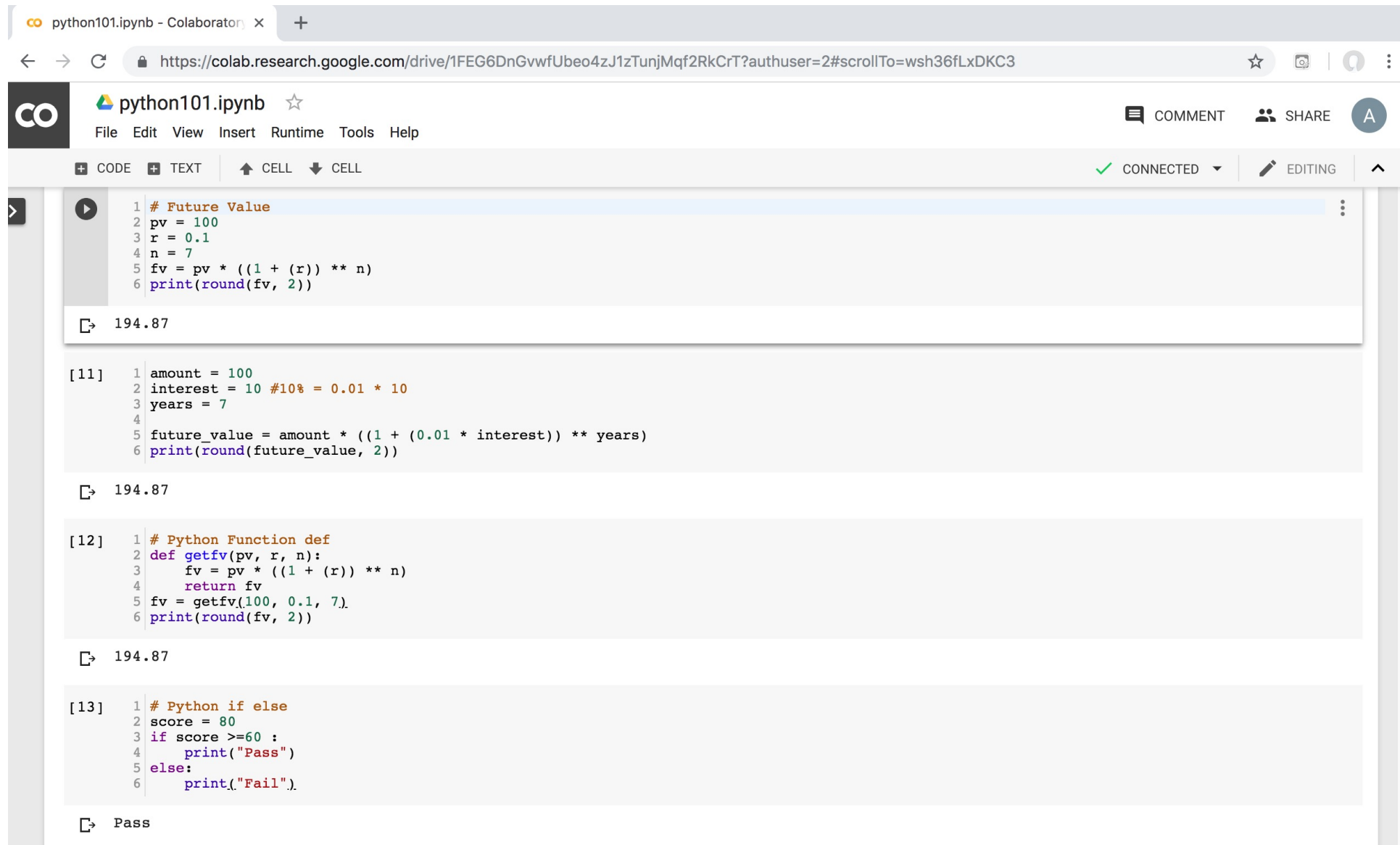
..	
oanda	Code updates for TF 2.3.
01_artificial_intelligence.ipynb	Code updates for TF 2.3.
02_superintelligence.ipynb	Code updates for TF 2.3.
03_normative_finance.ipynb	Code updates for TF 2.3.
04_data_driven_finance_a.ipynb	Initial commit.
04_data_driven_finance_b.ipynb	Initial commit.
05_machine_learning.ipynb	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>



The screenshot shows a Google Colab notebook interface. At the top, there's a browser address bar with 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 like 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))
```

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))
```

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))
```

194.87

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

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 under "AI in Finance", with "Uncertainty and Risk" currently selected. The main content area displays a list of topics: "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", and "Investment Portfolio Optimization". Below the list, a code cell is visible, containing Python code that imports numpy and defines variables for stock and bond prices today and tomorrow, along with a market price vector.

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)



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

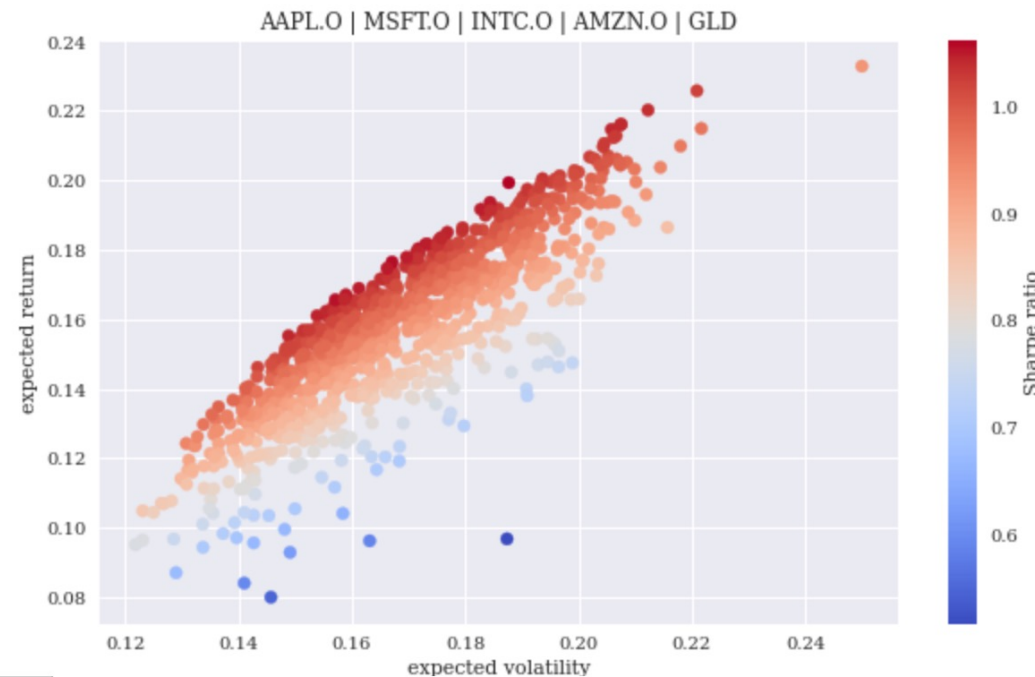
Table of contents

- 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

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



Python in Google Colab (Python101)



python101.ipynb ☆

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

Comment Share Settings 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
- 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

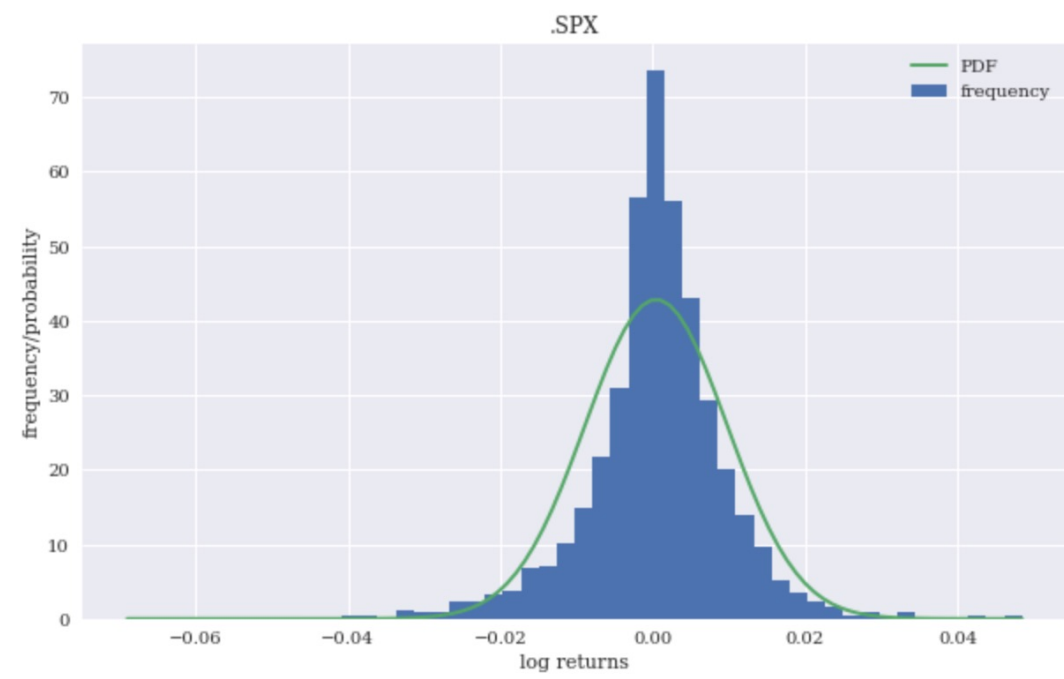
```
+ Code + Text
92 rets_sym.cumsum().apply(np.exp).plot(figsize=(10, 6));
93
94 rets_sym['same'] = (np.sign(rets_sym[sym + '_apt']) ==
95                    np.sign(rets_sym[sym + '_real']))
96
97 rets_sym['same'].value_counts()
98
99 rets_sym['same'].value_counts()[True] / len(rets_sym)
```



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)

python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings 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

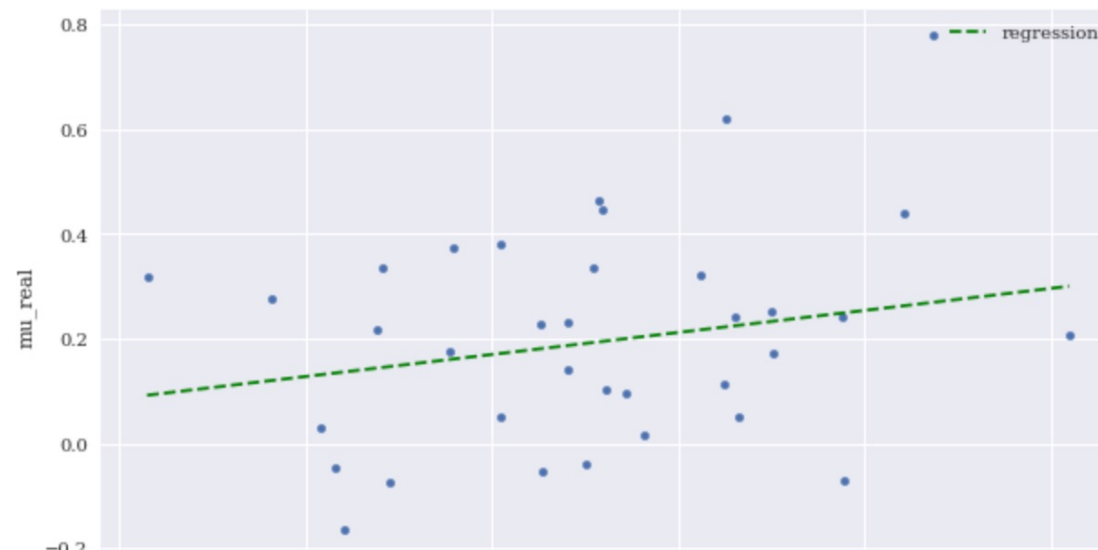
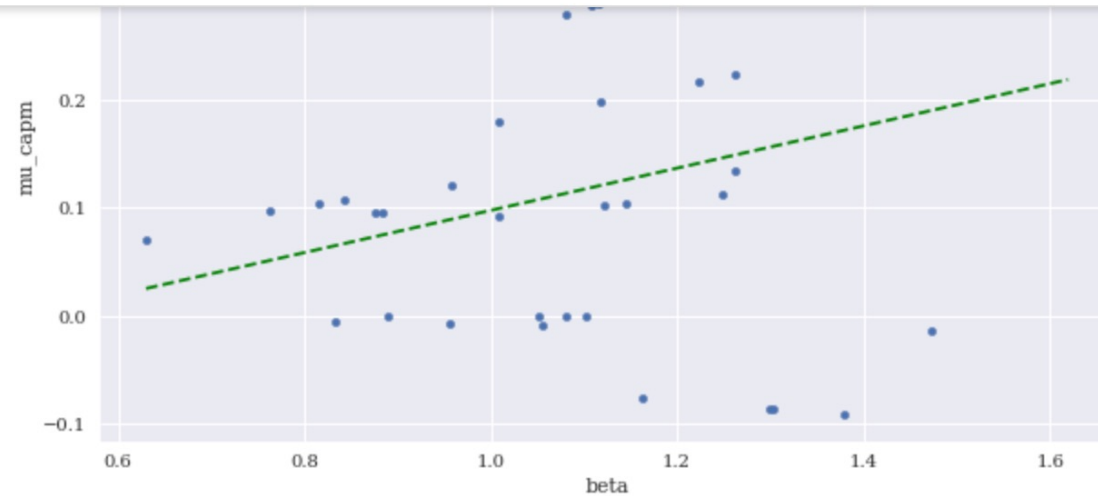
Investment Portfolio Optimisation with Python

Efficient Frontier Portfolio Optimisation in Python

Investment Portfolio Optimization

+ Code + Text

RAM Disk Editing



<https://tinyurl.com/aintpupython101>

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.
- Min-Yuh Day (2021), Python 101, <https://tinyurl.com/aintpupython101>