智慧金融量化分析

(Artificial Intelligence in Finance and Quantitative Analysis)

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)

1101AIFQA10
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

2021-12-28

課程大綱 (Syllabus)

國立臺北大學 National Taipei University

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)

國立臺北大學 National Taipei University

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)

[演講主題：指數設計的方法論、數據分析與量化投資應用，演講者：李政剛，基金經理/元大投信]
[Invited Talk: Index Design – Methodology、Data Analysis and the Application of Quantitative Investing, Invited Speaker: Jervis J.G. Li, Fund Manager, Yuanta SITC]

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)

Algorithmic Trading
Risk Management
Trading Bot
Event-Based Backtesting

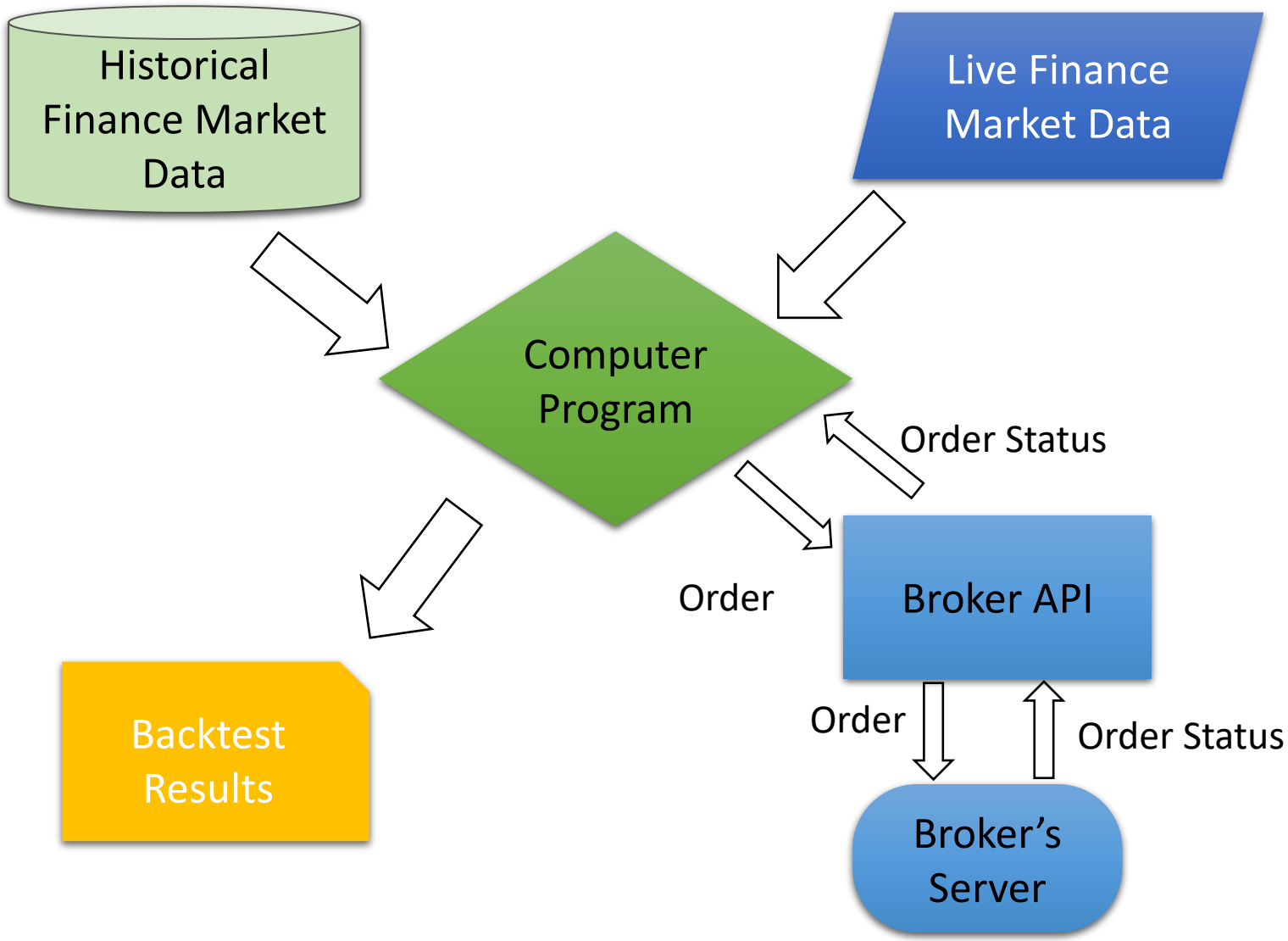
Outline

- Algorithmic Trading
- Risk Management
- Trading Bot
- Event-Based Backtesting

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

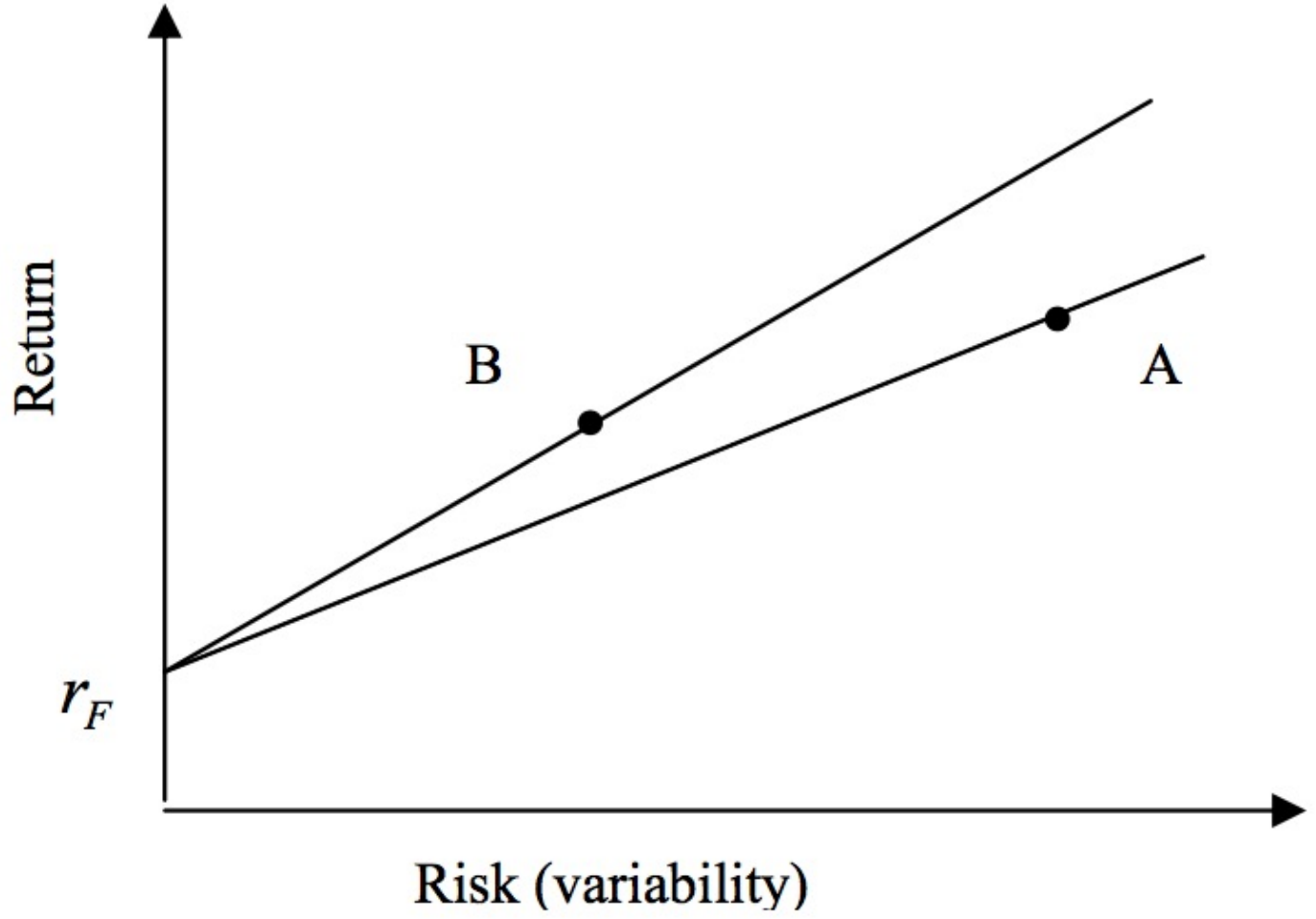
Algorithmic Trading

Algorithmic Trading



Source: Ernest P. Chan (2017), Machine Trading: Deploying Computer Algorithms to Conquer the Markets, Wiley

Risk and Return



Source: Bacon, Carl. "How sharp is the Sharpe-ratio?-Risk-adjusted Performance Measures." *Statpro White Paper* (2000).

Sharpe Ratio

$$\textbf{Sharpe Ratio} = \frac{Portofolio\ Return - Risk\ Free\ Return}{Portofolio\ Risk}$$

Source: Bacon, Carl. "How sharp is the Sharpe-ratio?-Risk-adjusted Performance Measures." *Statpro White Paper* (2000).

Sharpe Ratio

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

Where

r_P = portfolio return

r_F = risk free rate

σ_P = portfolio risk (variability, standard deviation of return)

Source: Bacon, Carl. "How sharp is the Sharpe-ratio?-Risk-adjusted Performance Measures." *Statpro White Paper* (2000).

Sortino Ratio

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

Where

r_P = portfolio return

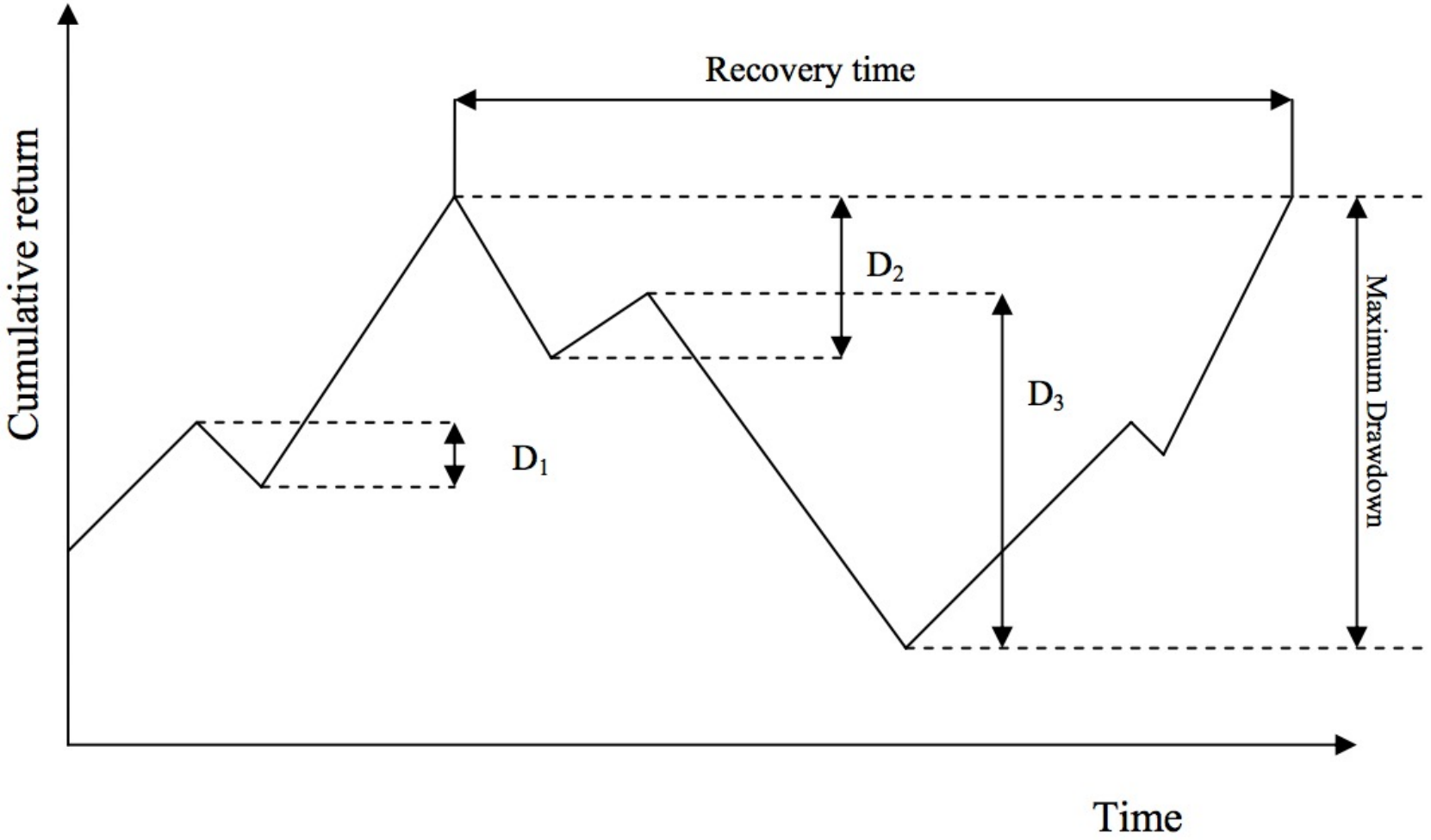
r_T = Minimum Target Return

σ_D = Downside Risk

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

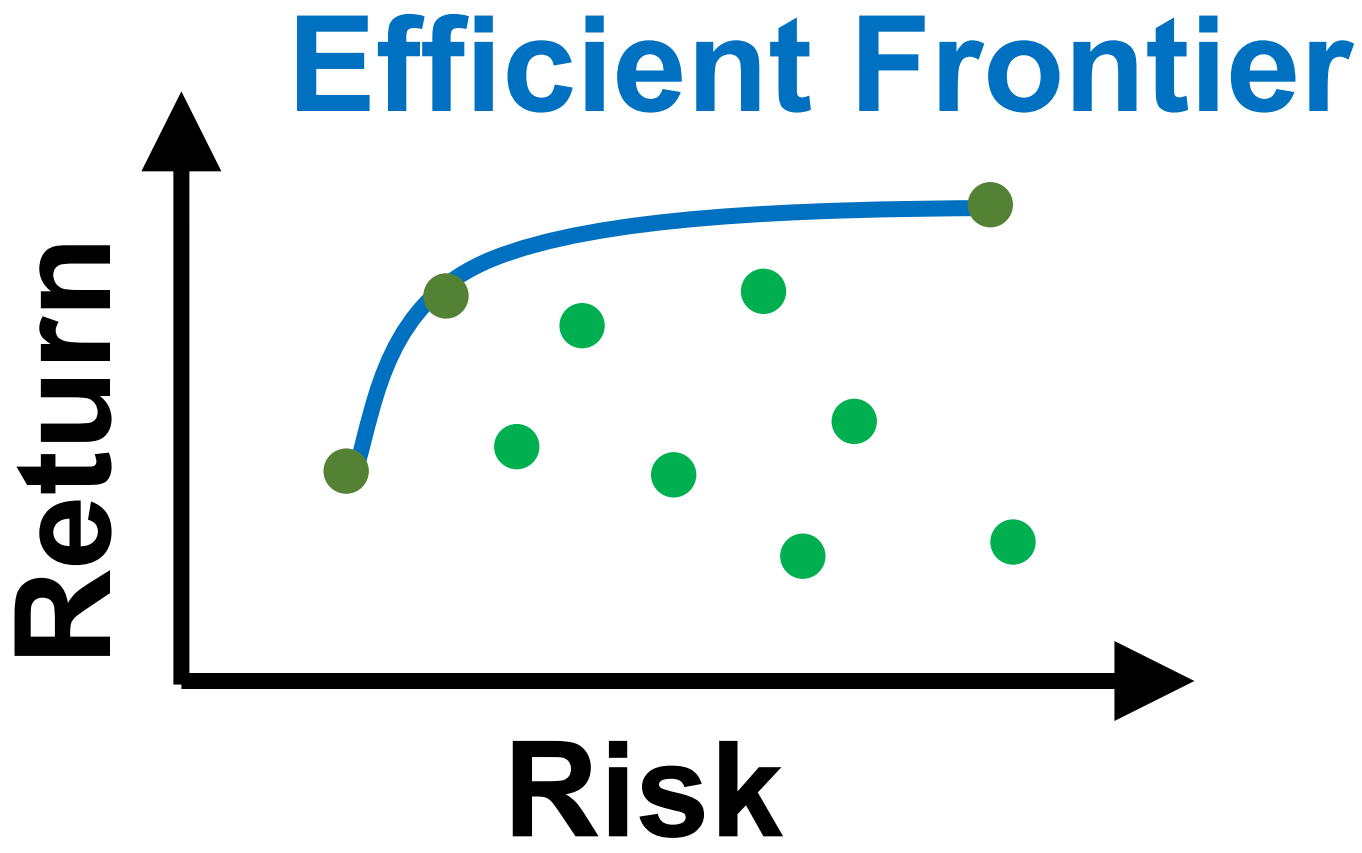
Source: Bacon, Carl. "How sharp is the Sharpe-ratio?-Risk-adjusted Performance Measures." *Statpro White Paper* (2000).

Max Drawdown



Source: Bacon, Carl. "How sharp is the Sharpe-ratio?-Risk-adjusted Performance Measures." *Statpro White Paper* (2000).

Portfolio Optimization
Efficient Frontier



Source: Tucker Balch (2012), Investment Science: Portfolio Optimization, https://www.youtube.com/watch?v=5qbMhXXq0vI

Time series data for EUR/USD and SMAs

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

Time series data for EUR/USD, SMAs, and resulting positions

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

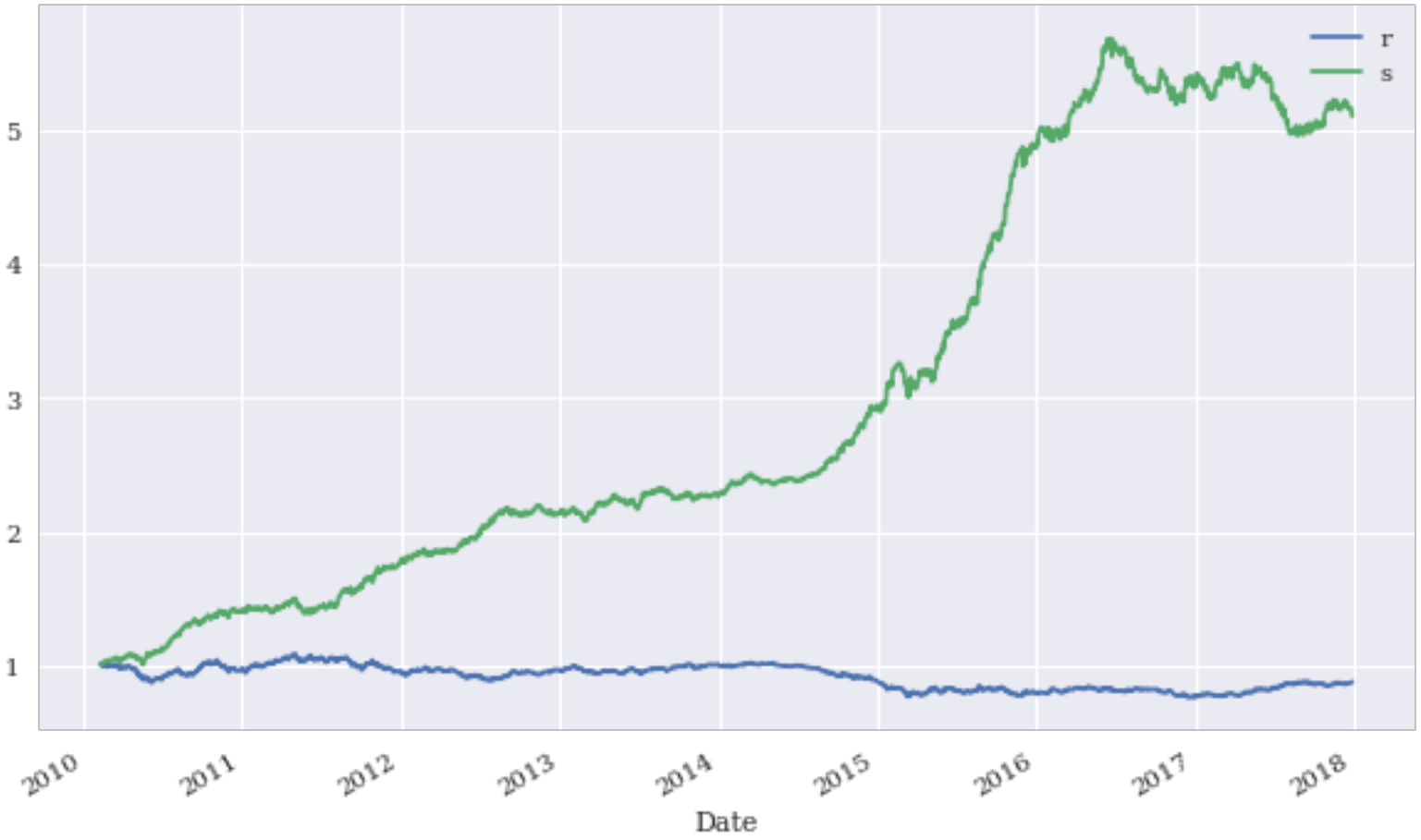
Gross performance of passive benchmark investment and SMA strategy

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

Gross performance of the SMA strategy before and after transaction costs

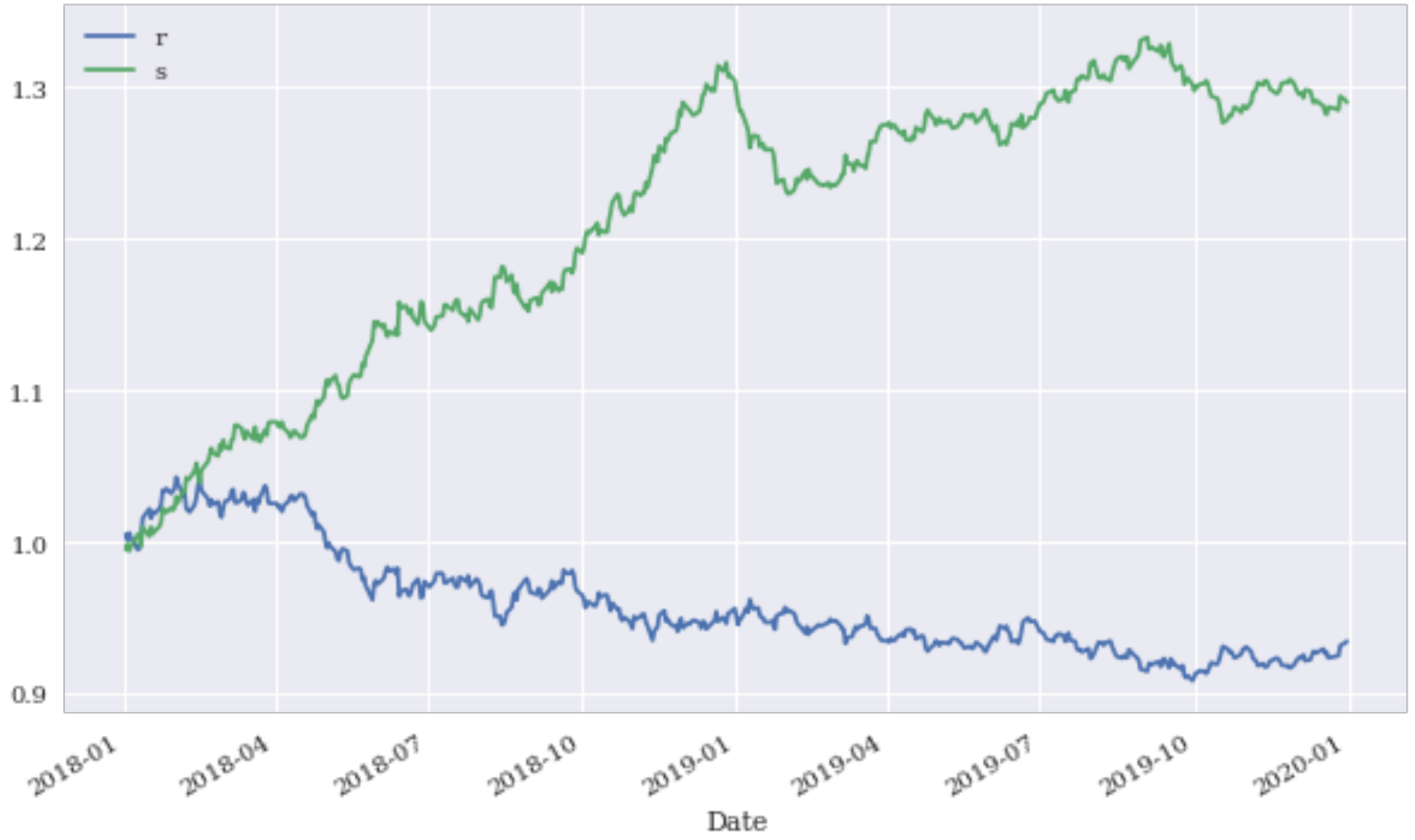
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Gross performance of the passive benchmark investment and the daily DNN strategy (in-sample)



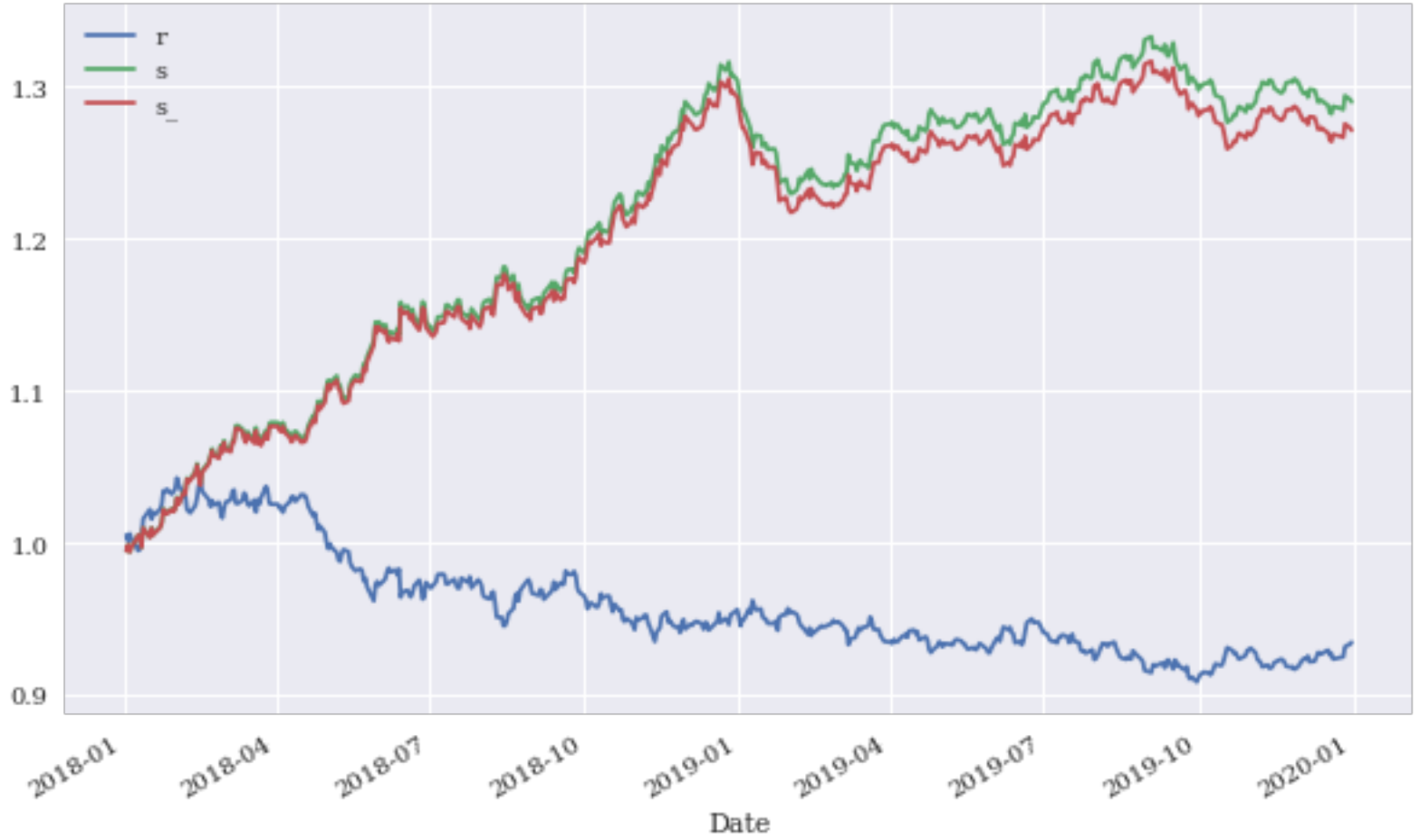
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Gross performance of the passive benchmark investment and the daily DNN strategy (out-of-sample)



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

Gross performance of the daily DNN strategy before and after transaction costs (out-of-sample)

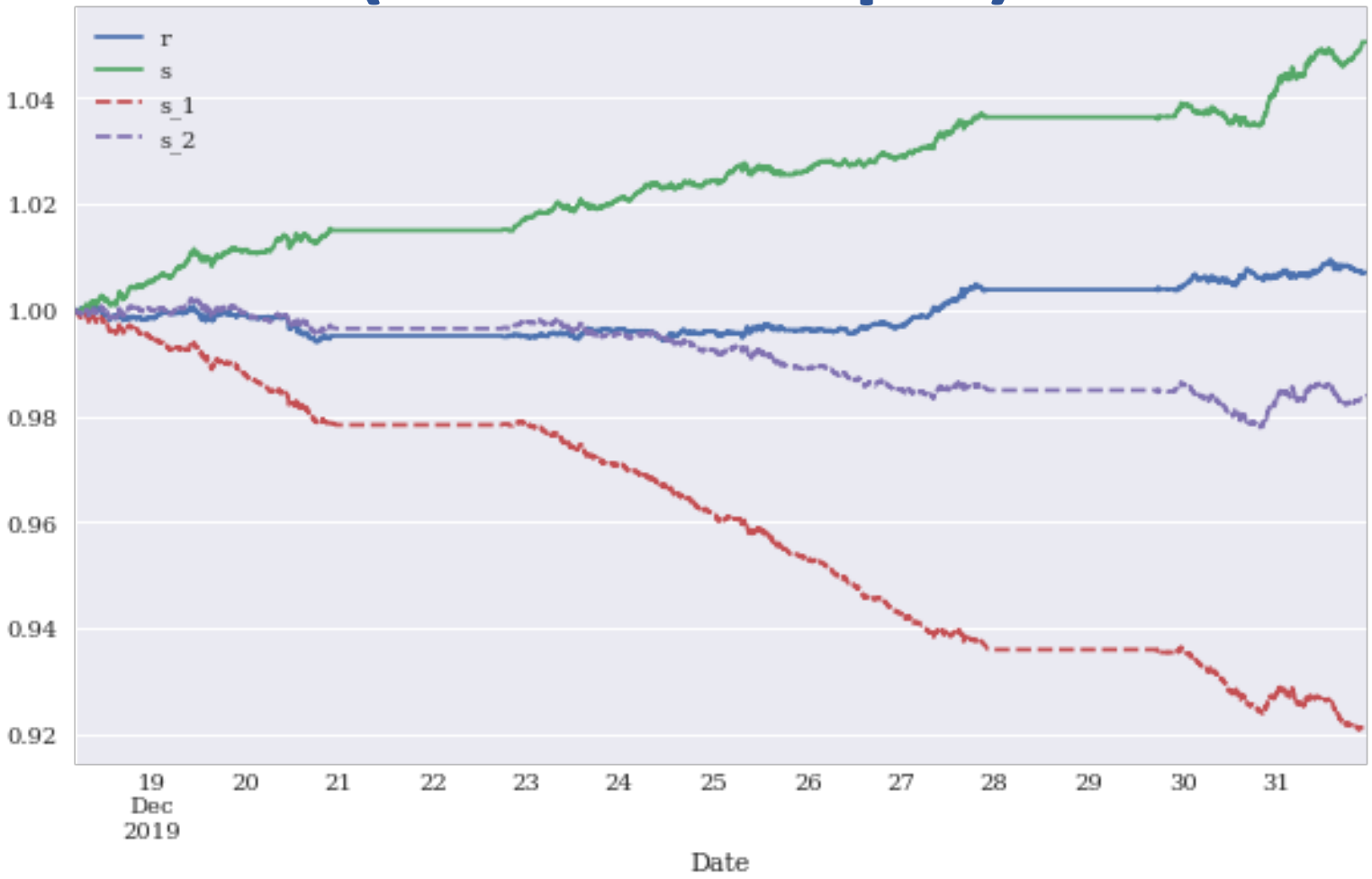


Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Gross performance of the passive benchmark investment and the DNN intraday strategy (out-of-sample)

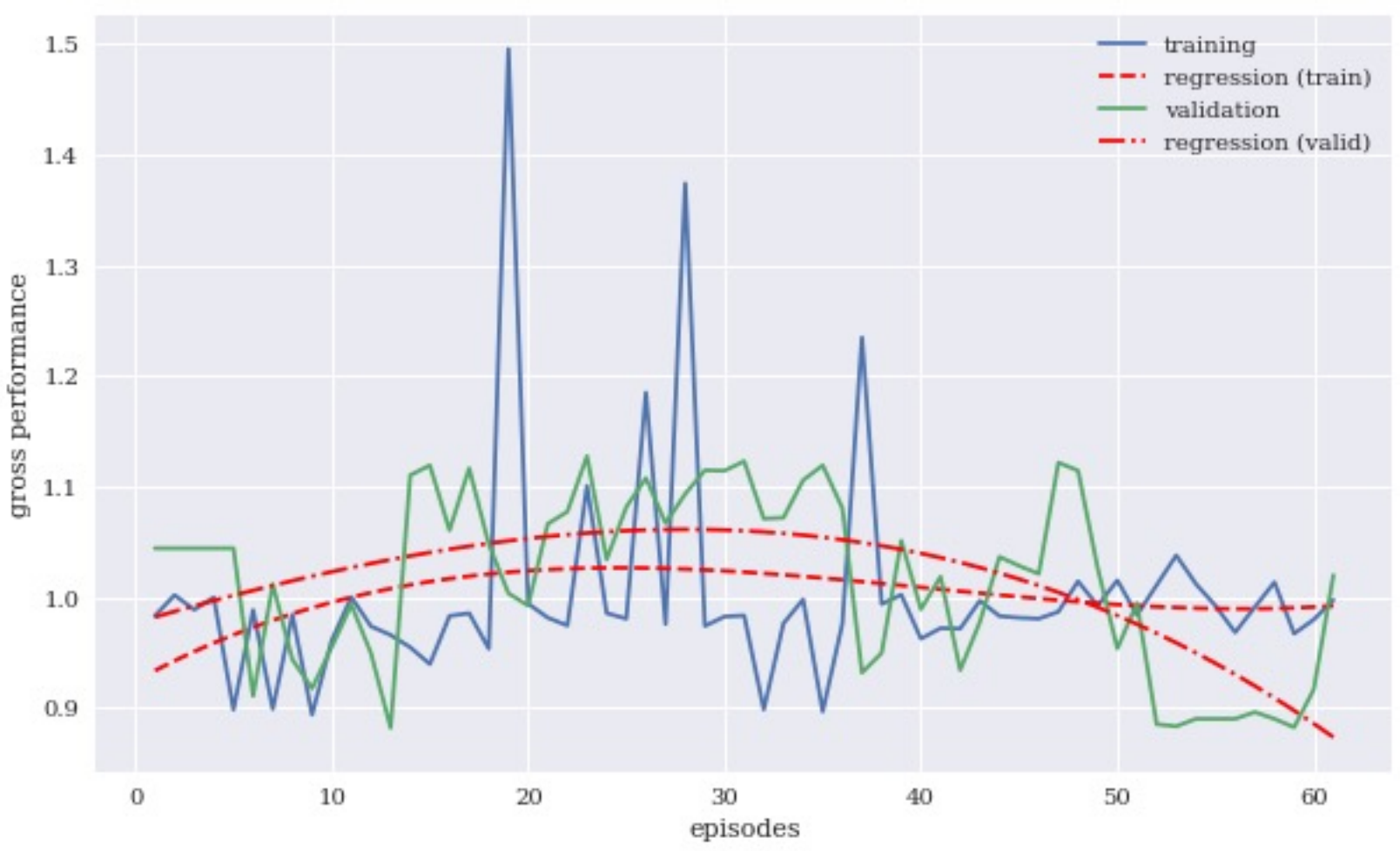
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Gross performance of the DNN intraday strategy before and after higher/ lower transaction costs (out-of-sample)



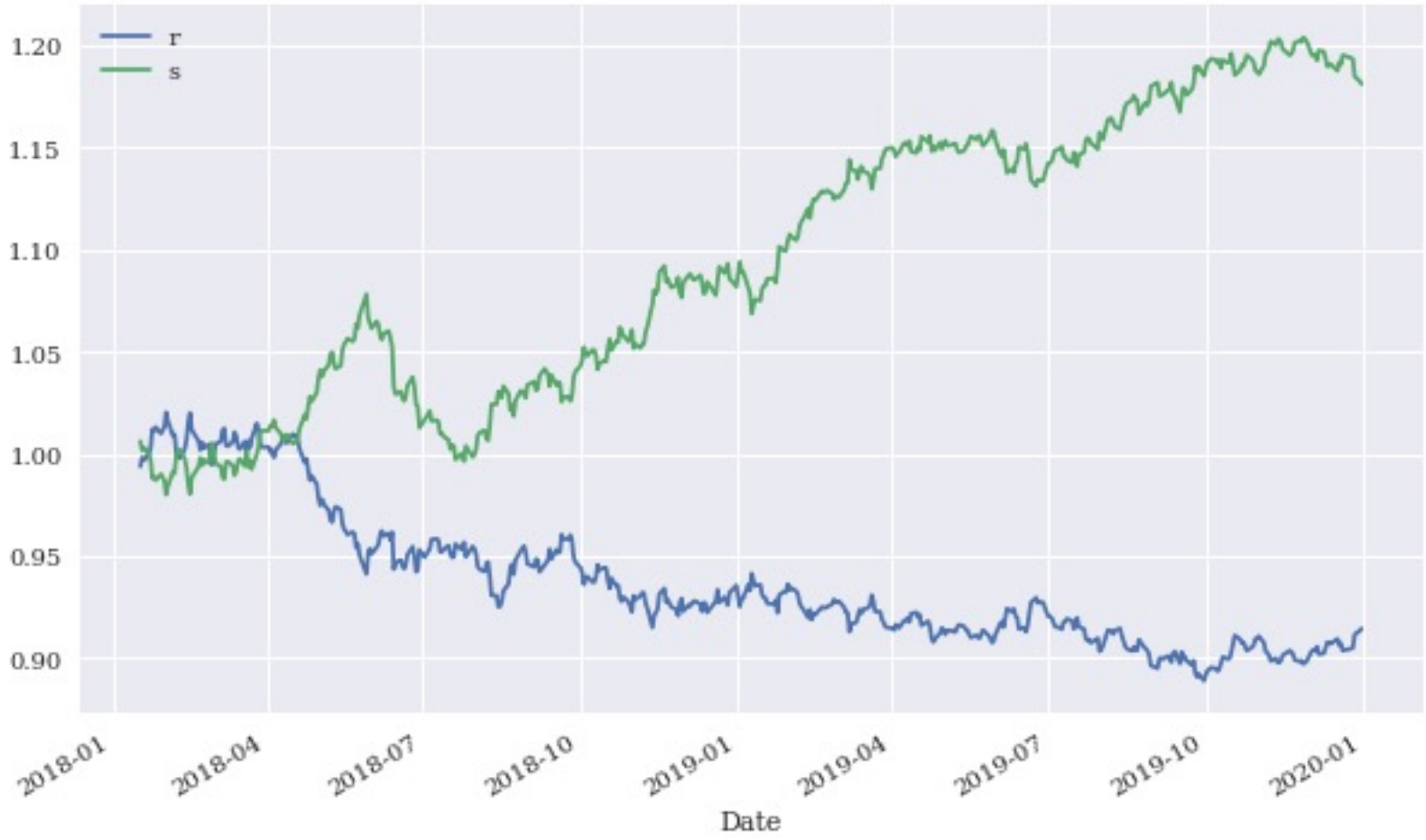
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Gross performance on training and validation data set



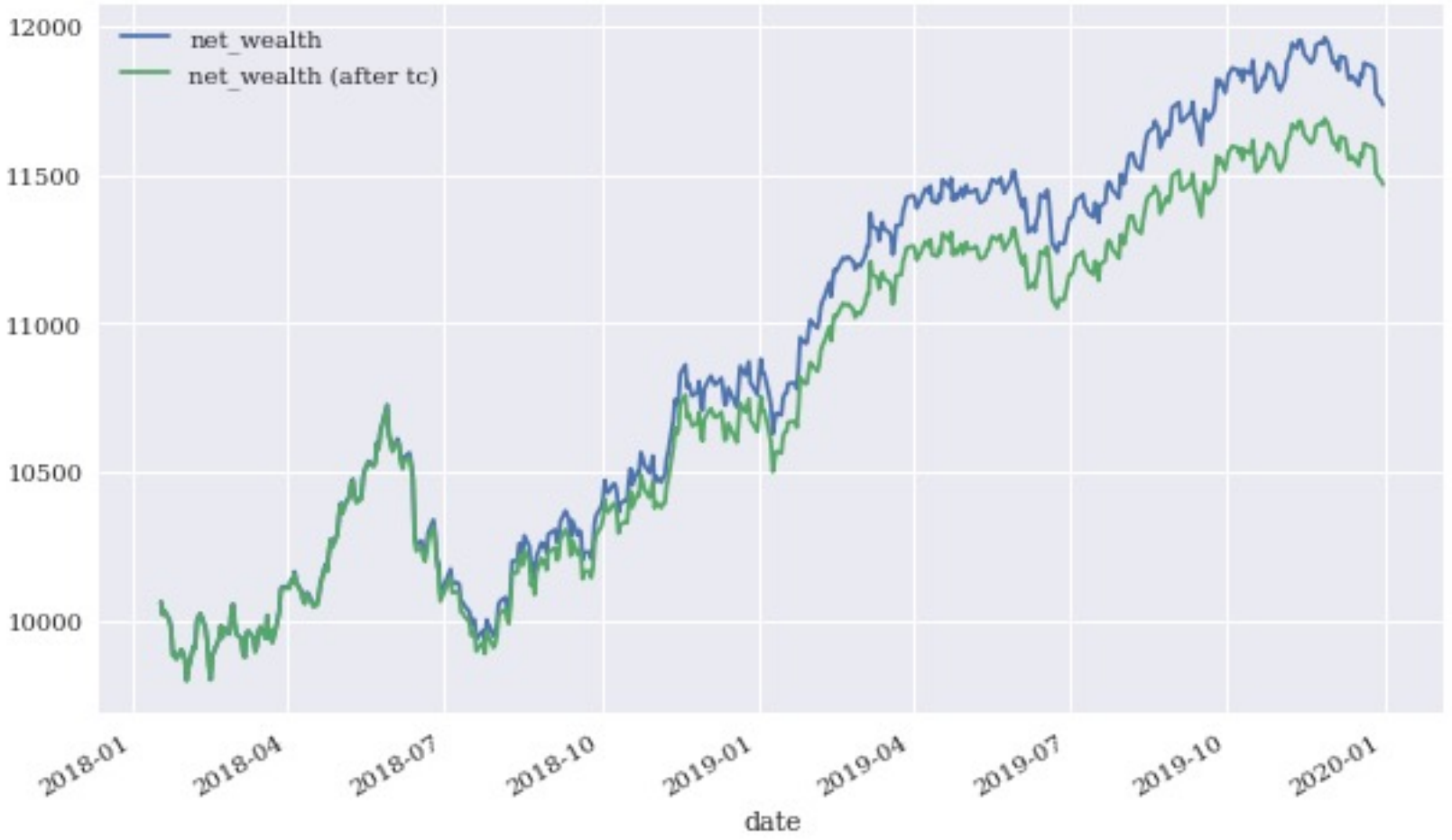
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Gross performance of the passive benchmark investment and the trading bot (out-of-sample)



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

Gross performance of the trading bot before and after transaction costs (in-sample)

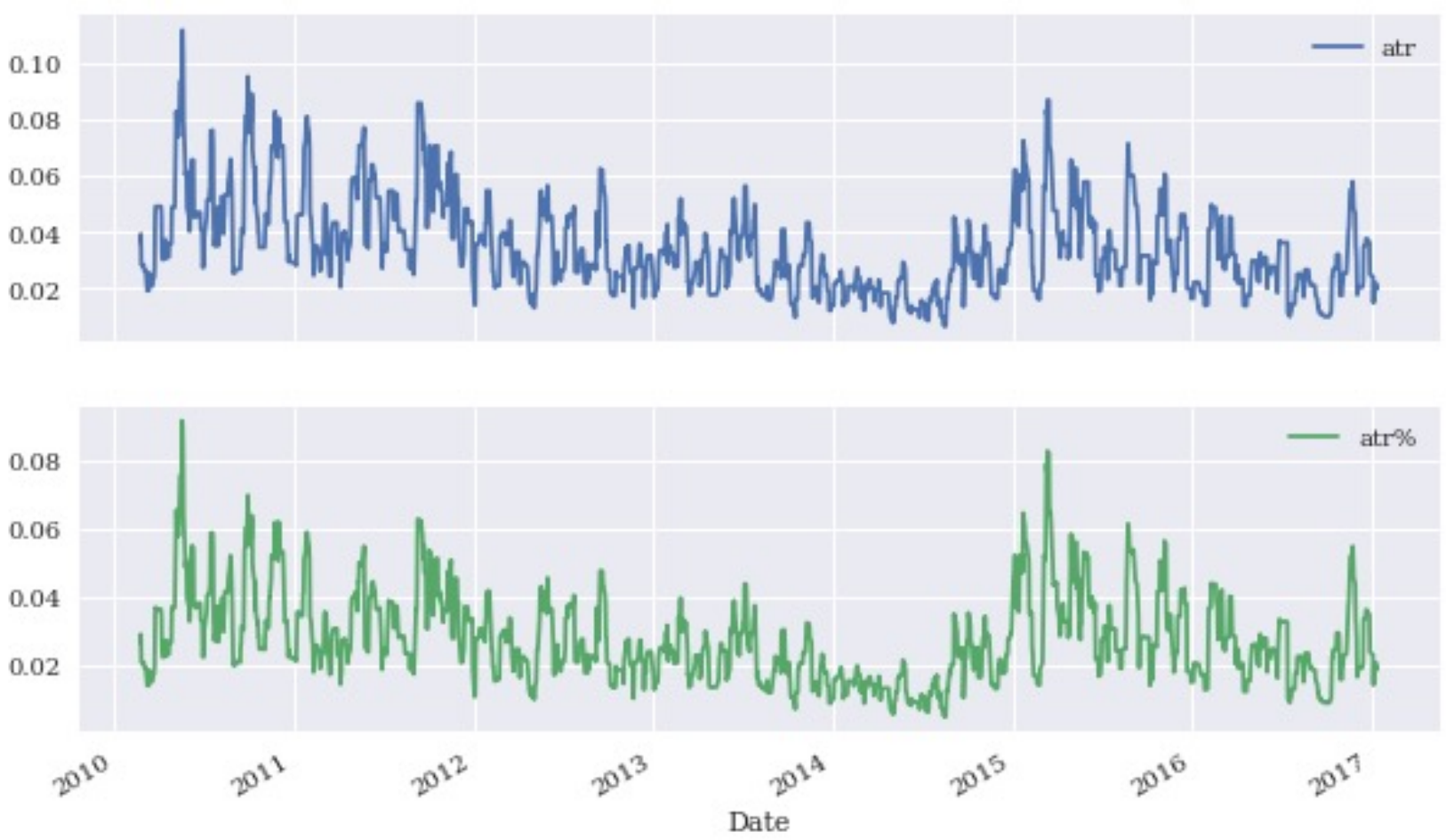


Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Gross performance of the passive benchmark investment and the trading bot (vectorized and event-based backtesting)

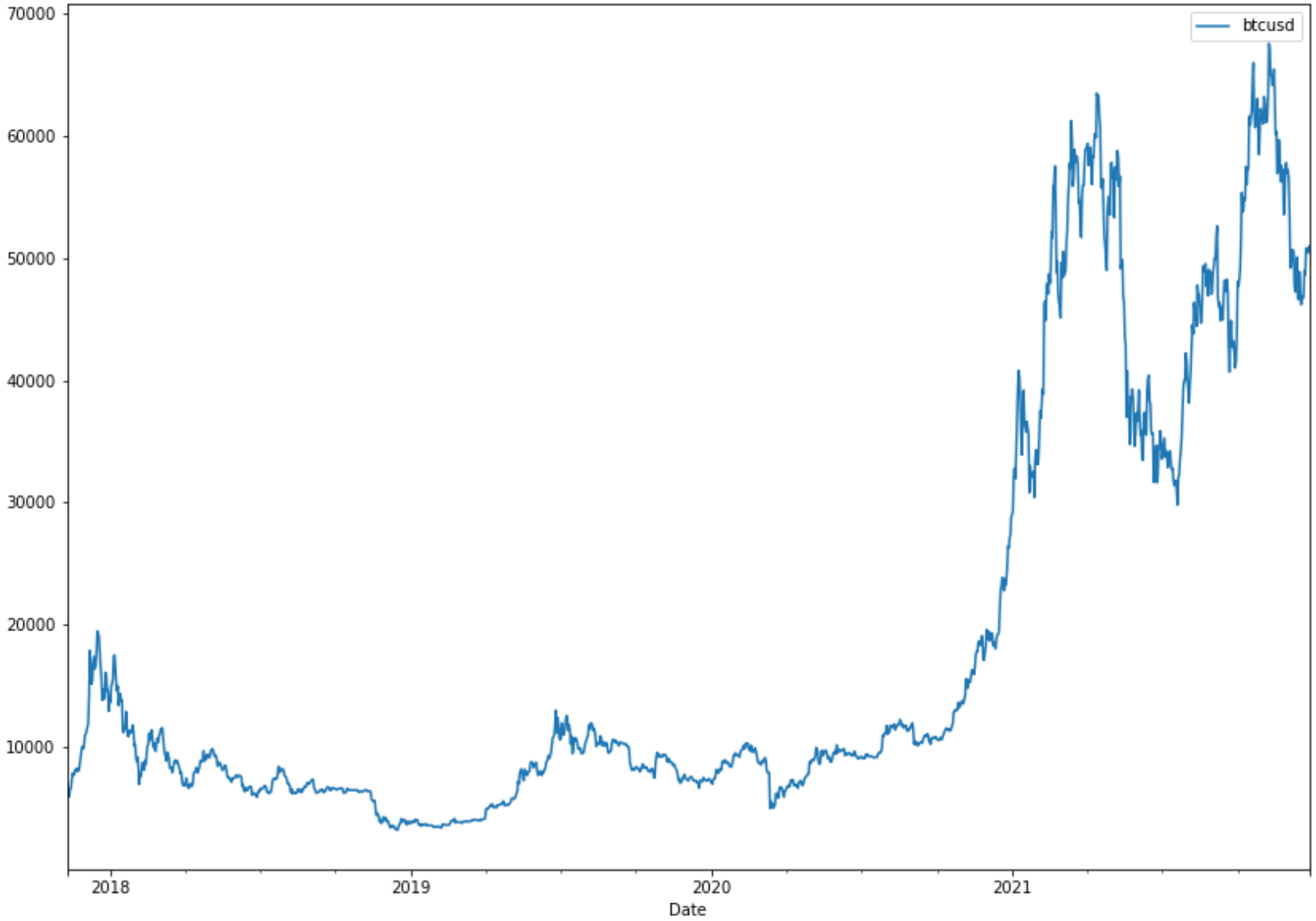
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Average true range (ATR) in absolute (price) and relative (%) terms



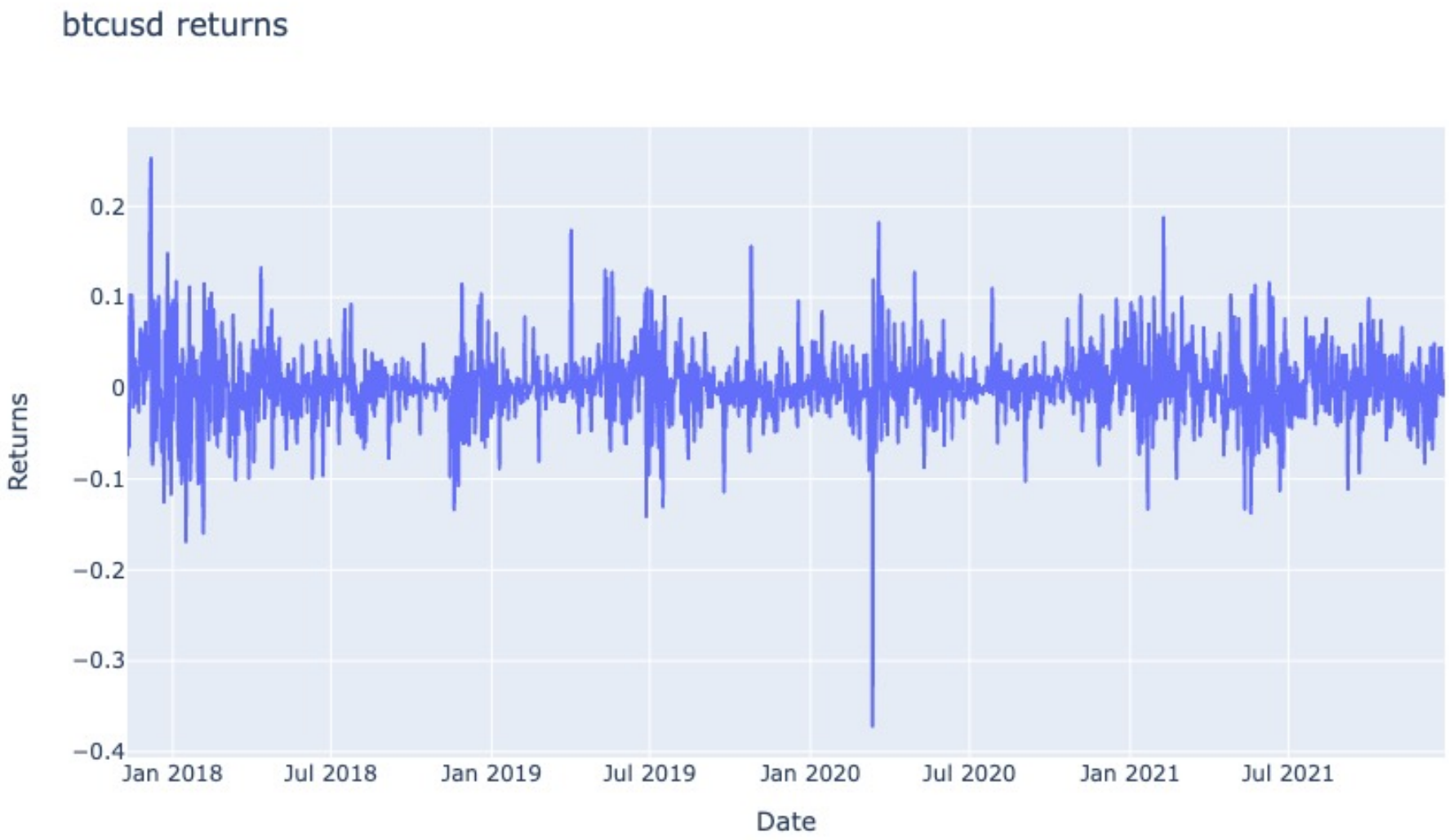
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

BTC-USD



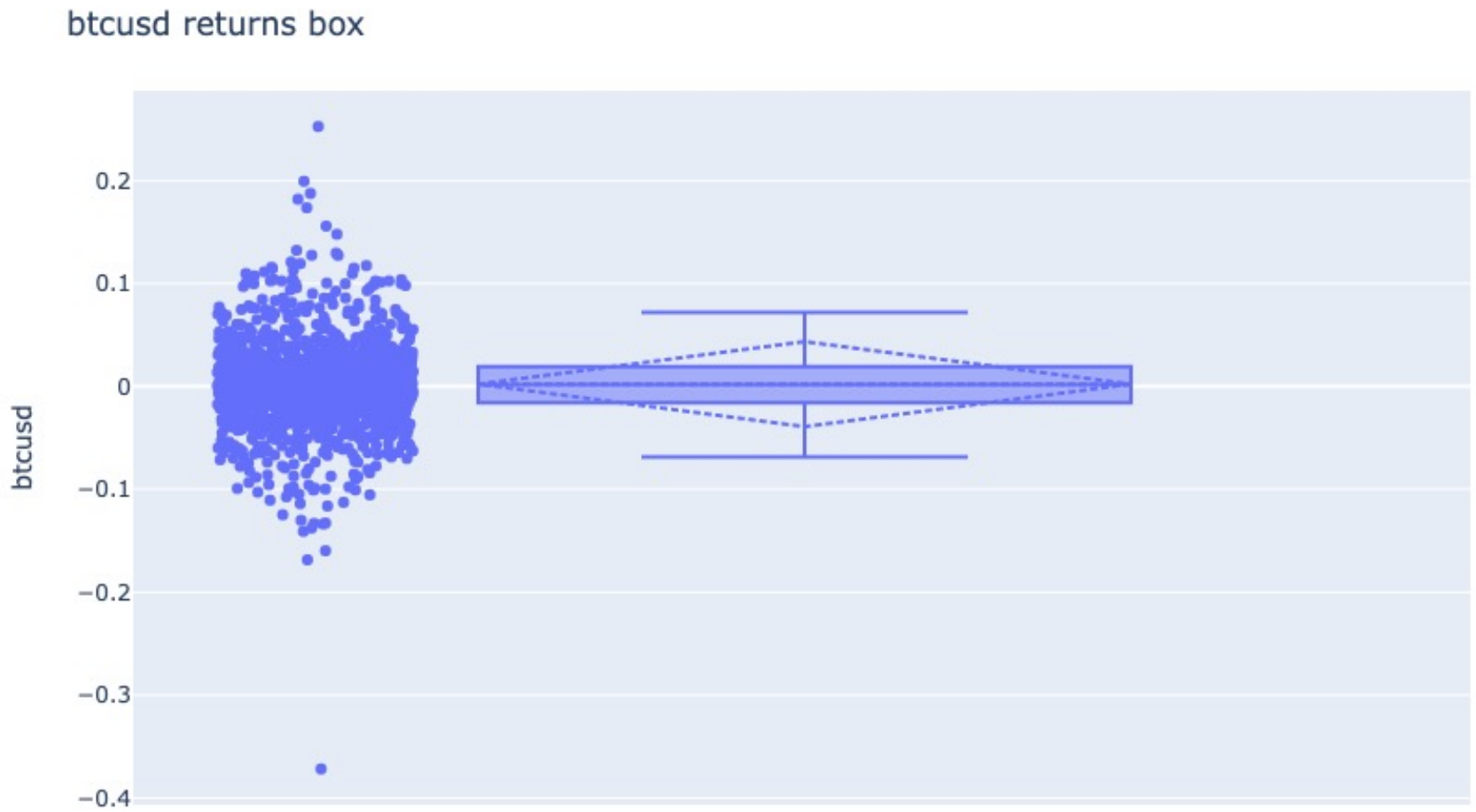
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

BTC-USD Returns



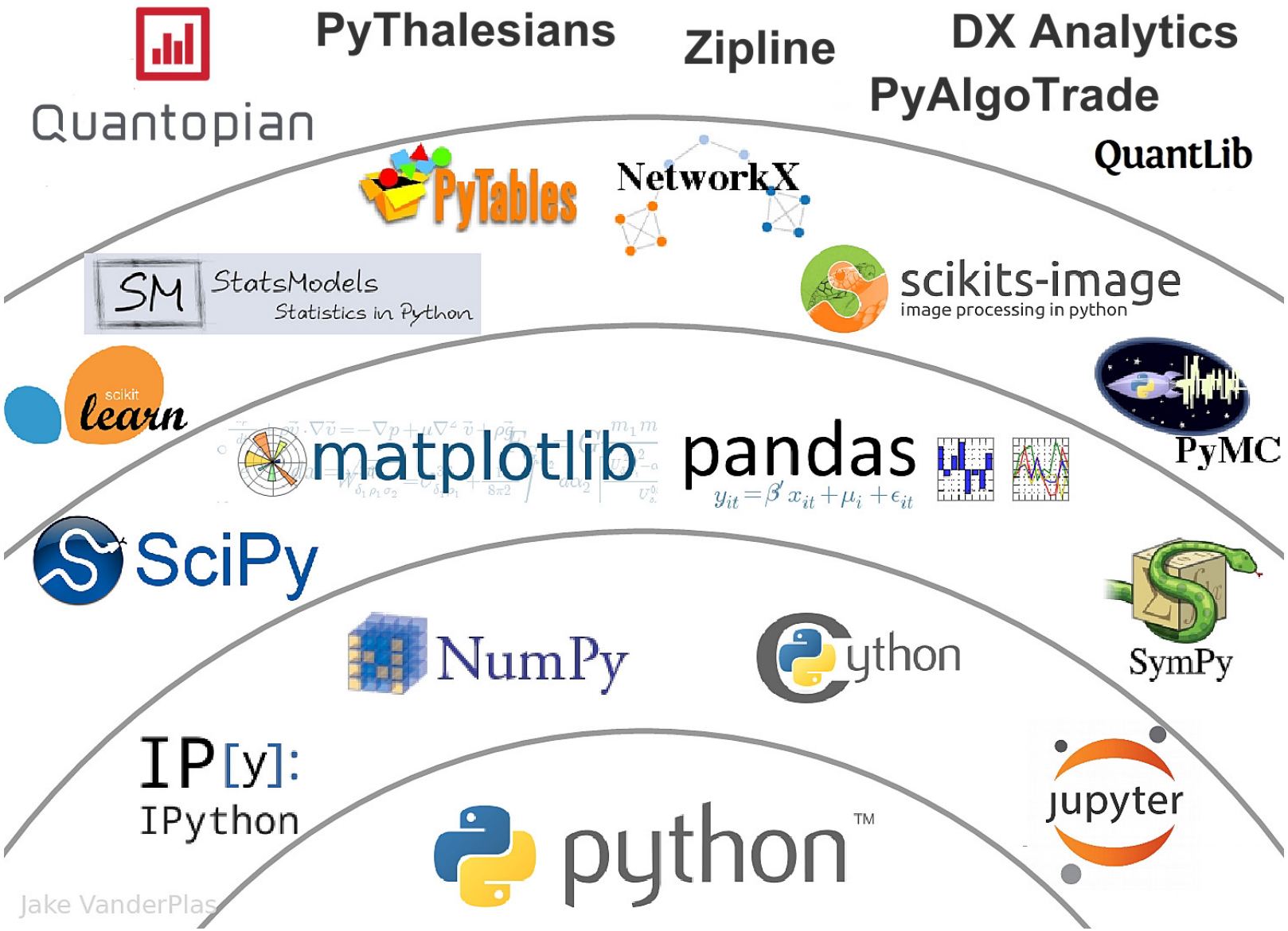
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

BTC-USD Returns Box



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

The Quant Finance PyData Stack



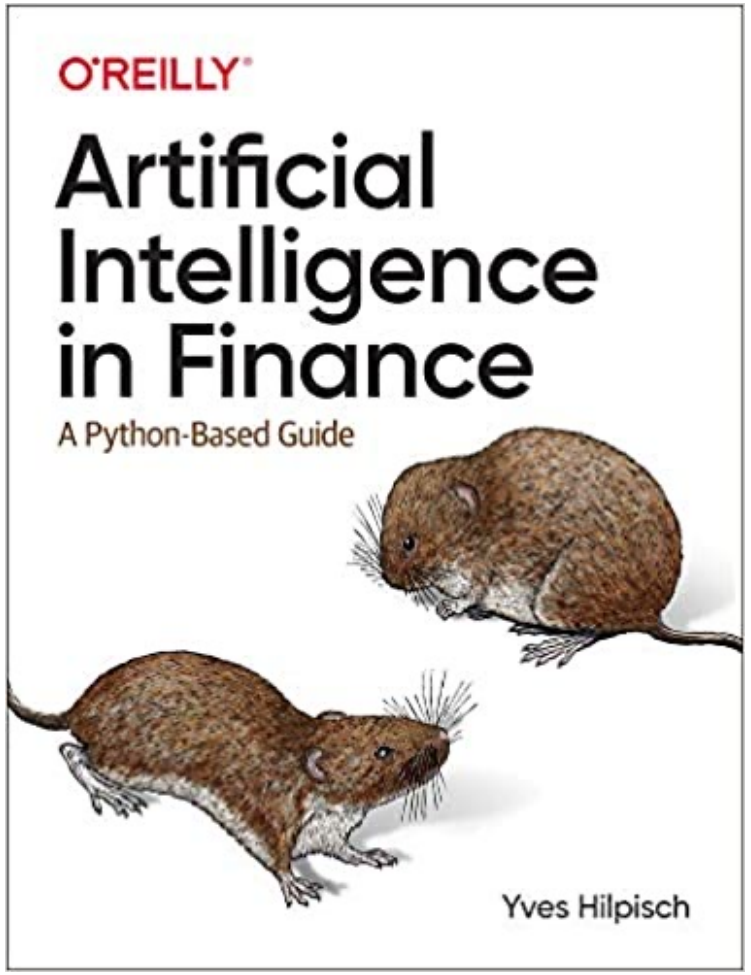
Source: http://nbviewer.jupyter.org/format/slides/github/quantopian/pyfolio/blob/master/pyfolio/examples/overview_slides.ipynb#/5

Yves Hilpisch (2020),

Artificial Intelligence in Finance:

A Python-Based Guide,

O'Reilly



Source: https://www.amazon.com/Artificial-Intelligence-Finance-Python-Based-Guide/dp/1492055433

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.

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

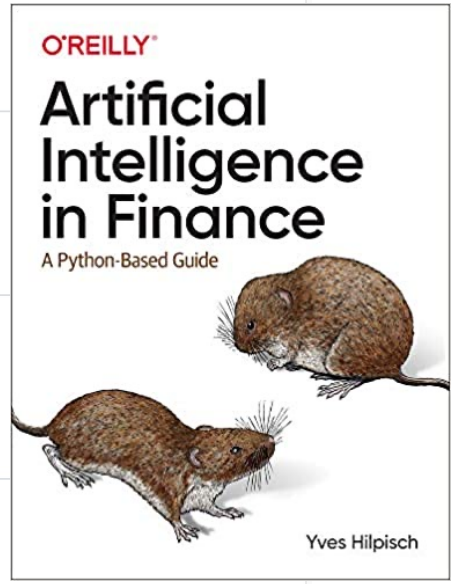
No releases published

Packages

No packages published

Languages

Jupyter Notebook 97.4% Python 2.6%



Source: https://github.com/yhilpisch/aiif

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

yhilpisch / aiif Public

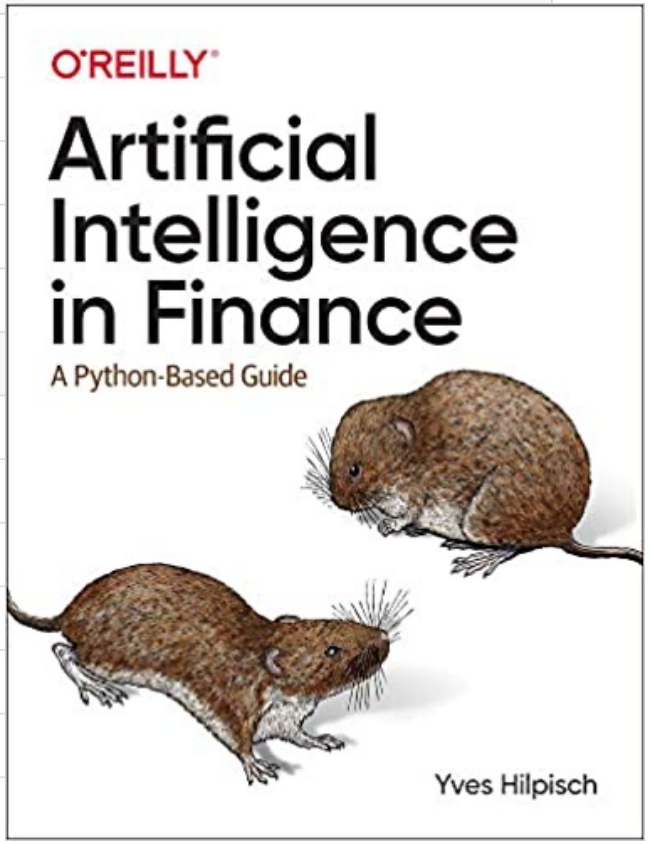
Notifications | Star 98 | Fork 77

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

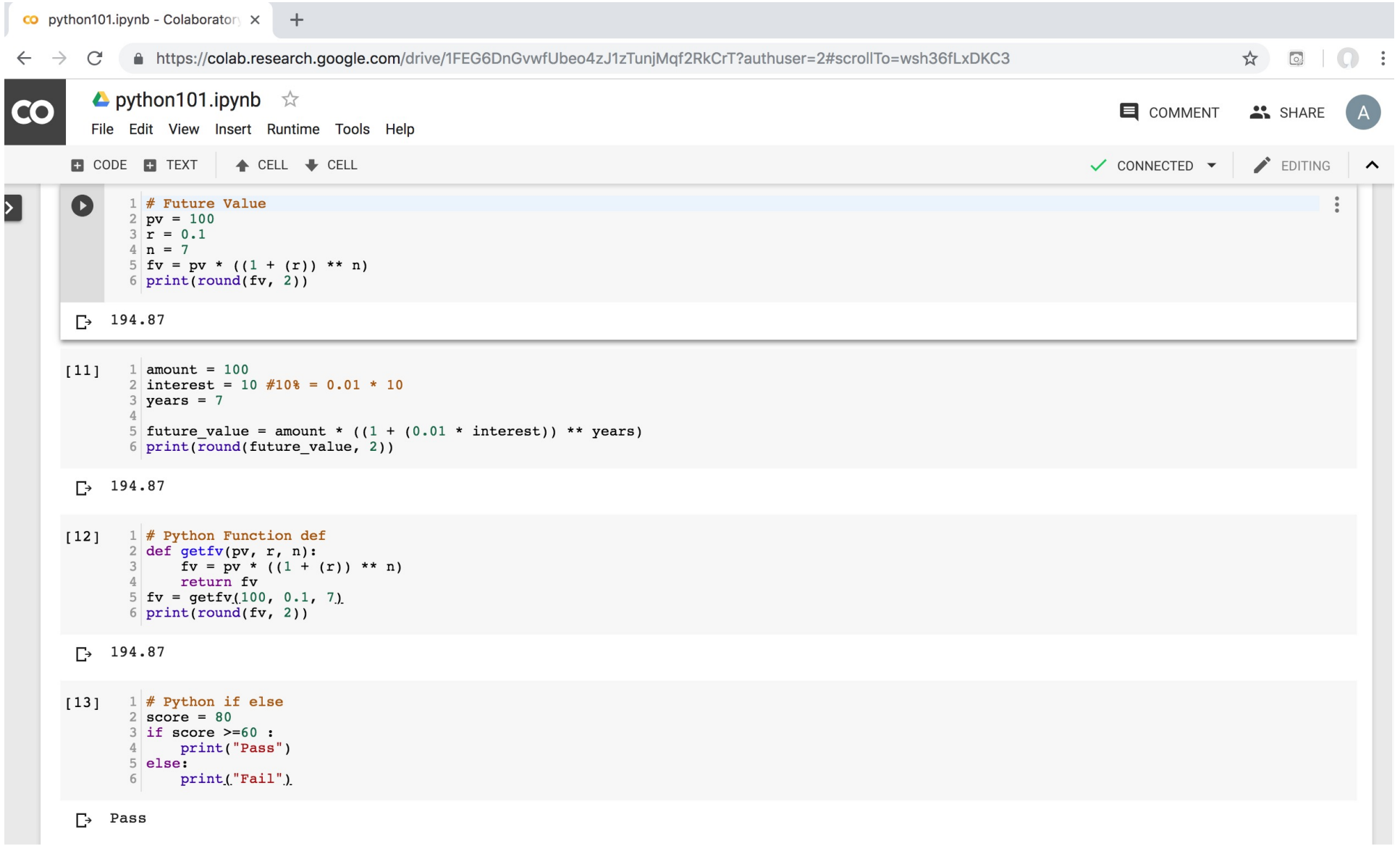
..	
oanda	Code updates for TF 2.3.
01_artificial_intelligence.ipynb	Code updates for TF 2.3.
02_superintelligence.ipynb	Code updates for TF 2.3.
03_normative_finance.ipynb	Code updates for TF 2.3.
04_data_driven_finance_a.ipynb	Initial commit.
04_data_driven_finance_b.ipynb	Initial commit.
05_machine_learning.ipynb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

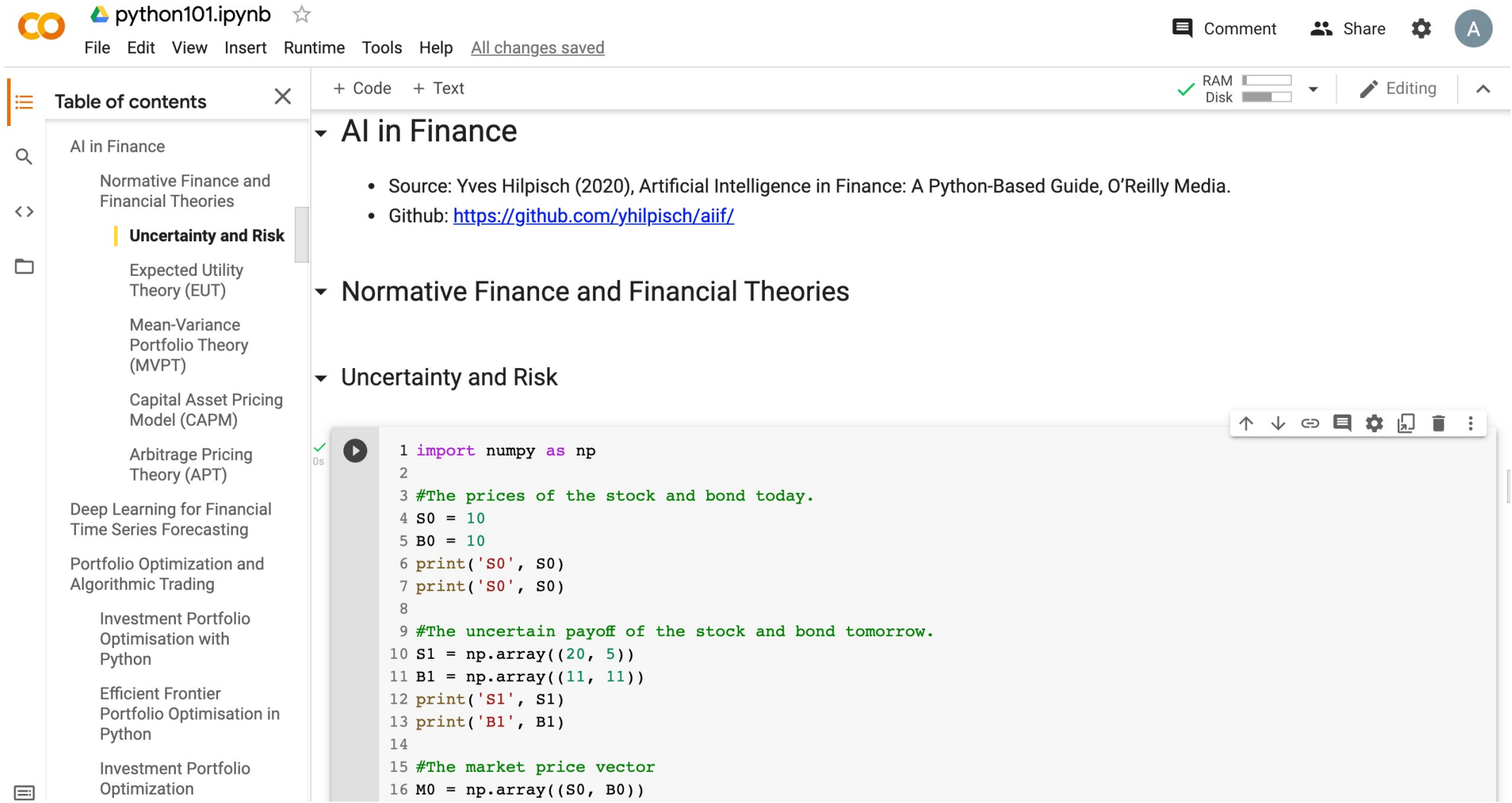
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

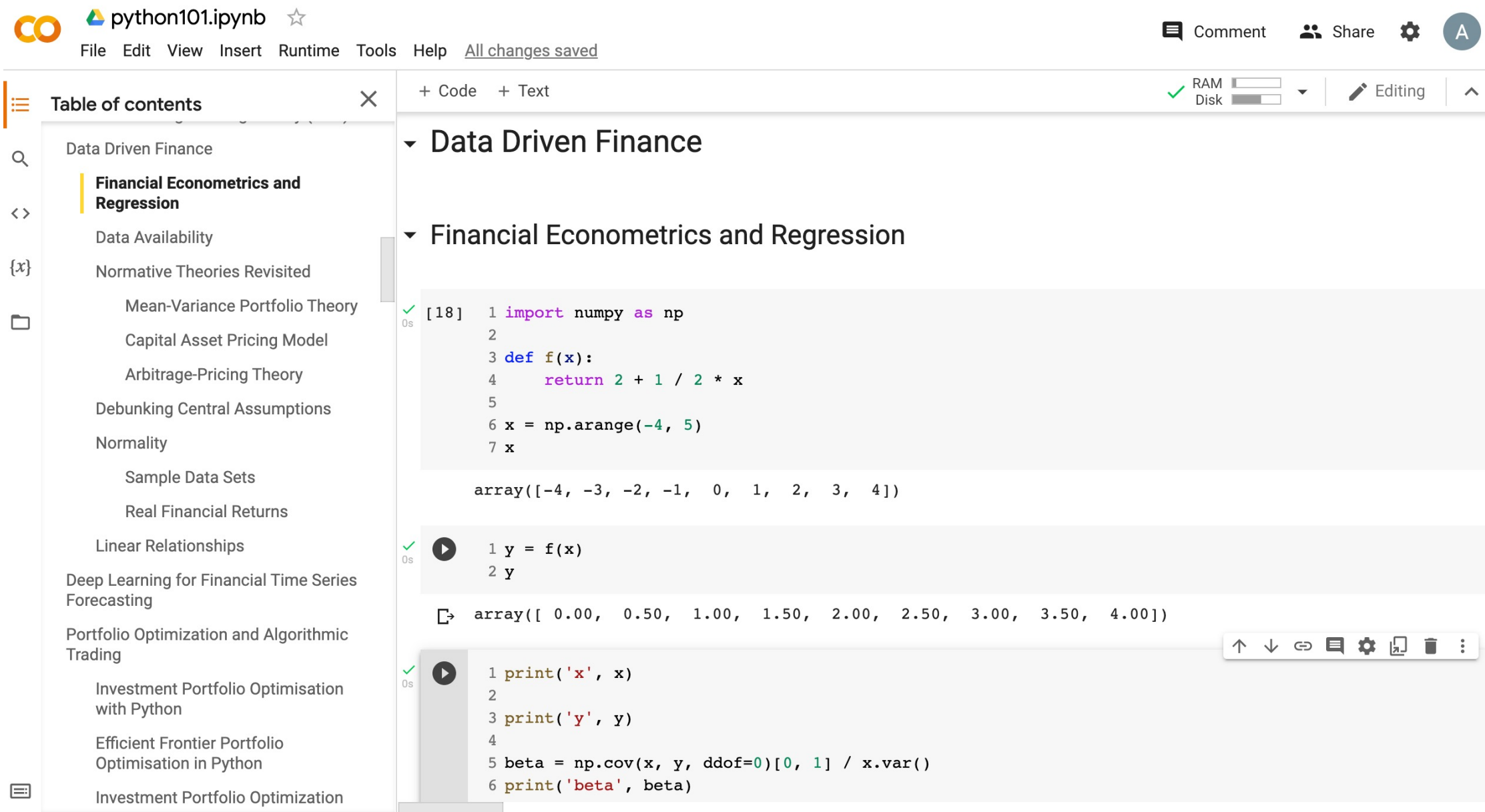
Python in Google Colab (Python101)

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



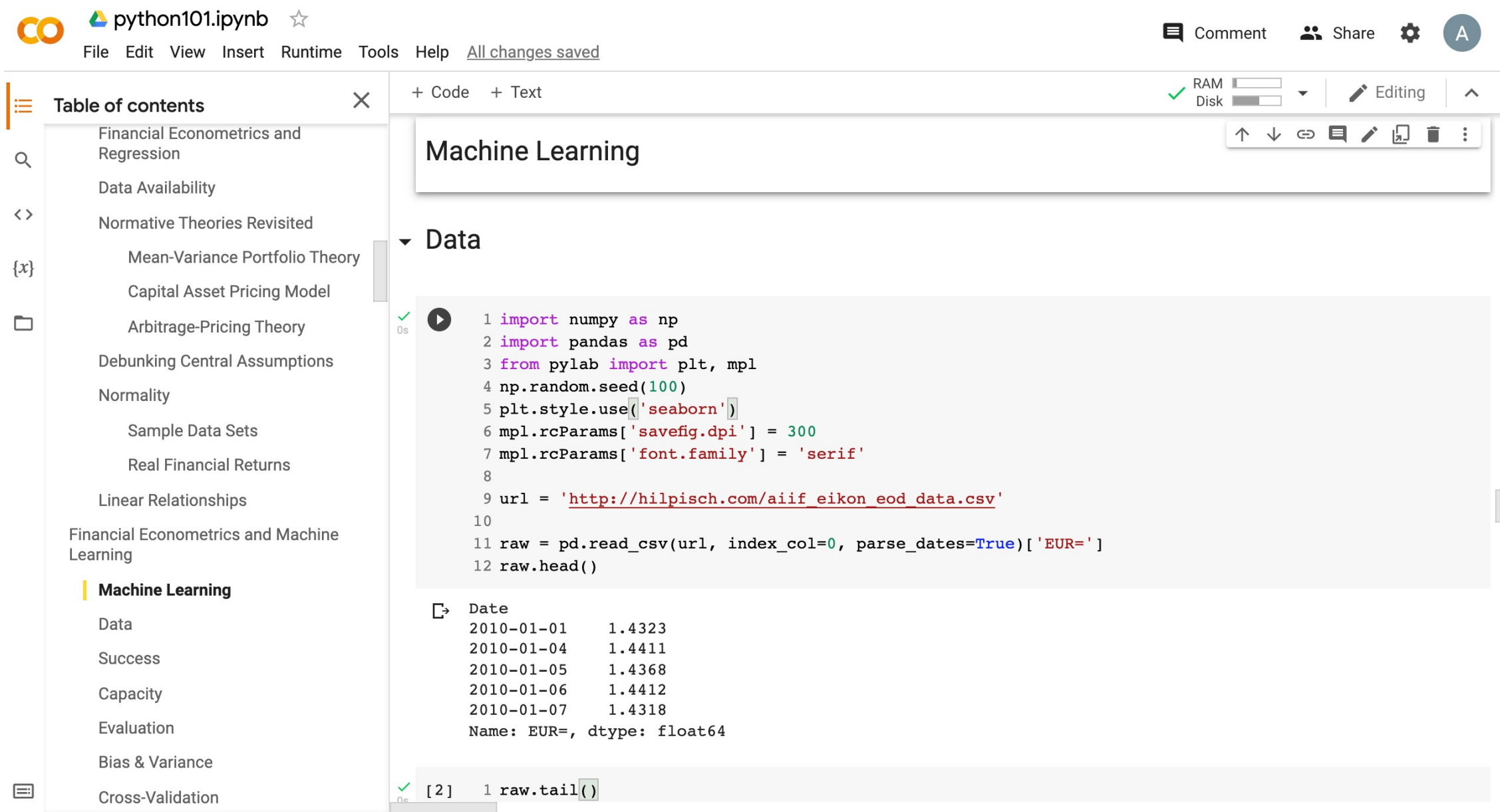
https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



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

- 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
- Cross-Validation
- AI-First Finance
- **Efficient Markets**
- Market Prediction Based on Returns Data
- Market Prediction With More Features
- Market Prediction Intraday

+ Code + Text RAM Disk Editing

Efficient Markets

```
1 import numpy as np
2 import pandas as pd
3 from pylab import plt, mpl
4 plt.style.use('seaborn')
5 mpl.rcParams['savefig.dpi'] = 300
6 mpl.rcParams['font.family'] = 'serif'
7 pd.set_option('precision', 4)
8 np.set_printoptions(suppress=True, precision=4)
9
10 url = 'http://hilpisch.com/aiif_eikon_eod_data.csv'
11 data = pd.read_csv(url, index_col=0, parse_dates=True).dropna()
12 (data / data.iloc[0]).plot(figsize=(10, 6), cmap='coolwarm')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f29f972f210>
```

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

- Deep Learning (DL) in Finance
  - Dense Neural Networks (DNN)
    - Baseline Prediction
    - Normalization
    - Dropout
    - Regularization
    - Bagging
    - Optimizers
  - Recurrent Neural Networks (RNN)
    - First Example
    - Second Example
    - Financial Price Series
    - Financial Return Series
    - Financial Features
    - Deep RNNs
  - Convolutional Neural Networks (CNN)
- Reinforcement Learning (RL) in Finance

+ Code + Text Connect Editing

Deep Learning (DL) in Finance

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

Dense Neural Networks (DNN)

```
1 import os
2 import numpy as np
3 import pandas as pd
4 from pylab import plt, mpl
5 plt.style.use('seaborn')
6 mpl.rcParams['savefig.dpi'] = 300
7 mpl.rcParams['font.family'] = 'serif'
8 pd.set_option('precision', 4)
9 np.set_printoptions(suppress=True, precision=4)
10 os.environ['PYTHONHASHSEED'] = '0'
```

```
[ ] 1 url = 'http://hilpisch.com/aiif_eikon_id_eur_usd.csv'
    2 symbol = 'EUR_USD'
    3 raw = pd.read_csv(url, index_col=0, parse_dates=True)
    4 raw.head()
```

	HIGH	LOW	OPEN	CLOSE

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

- Financial Features
- Deep RNNs
- Convolutional Neural Networks (CNN)
- **Reinforcement Learning (RL) in Finance**
- Reinforcement Learning (RL)
- CartPole Environment
- Dimensionality Reduction
- Action Rule
- Total Reward per Episode
- Simple Learning
- Testing the Results
- DNN Learning
- Q Learning
- Finance Environment
- Improved Finance Environment
- Improved Financial QL Agent

+ Code + Text Connect Editing

Reinforcement Learning (RL) in Finance

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

Reinforcement Learning (RL)

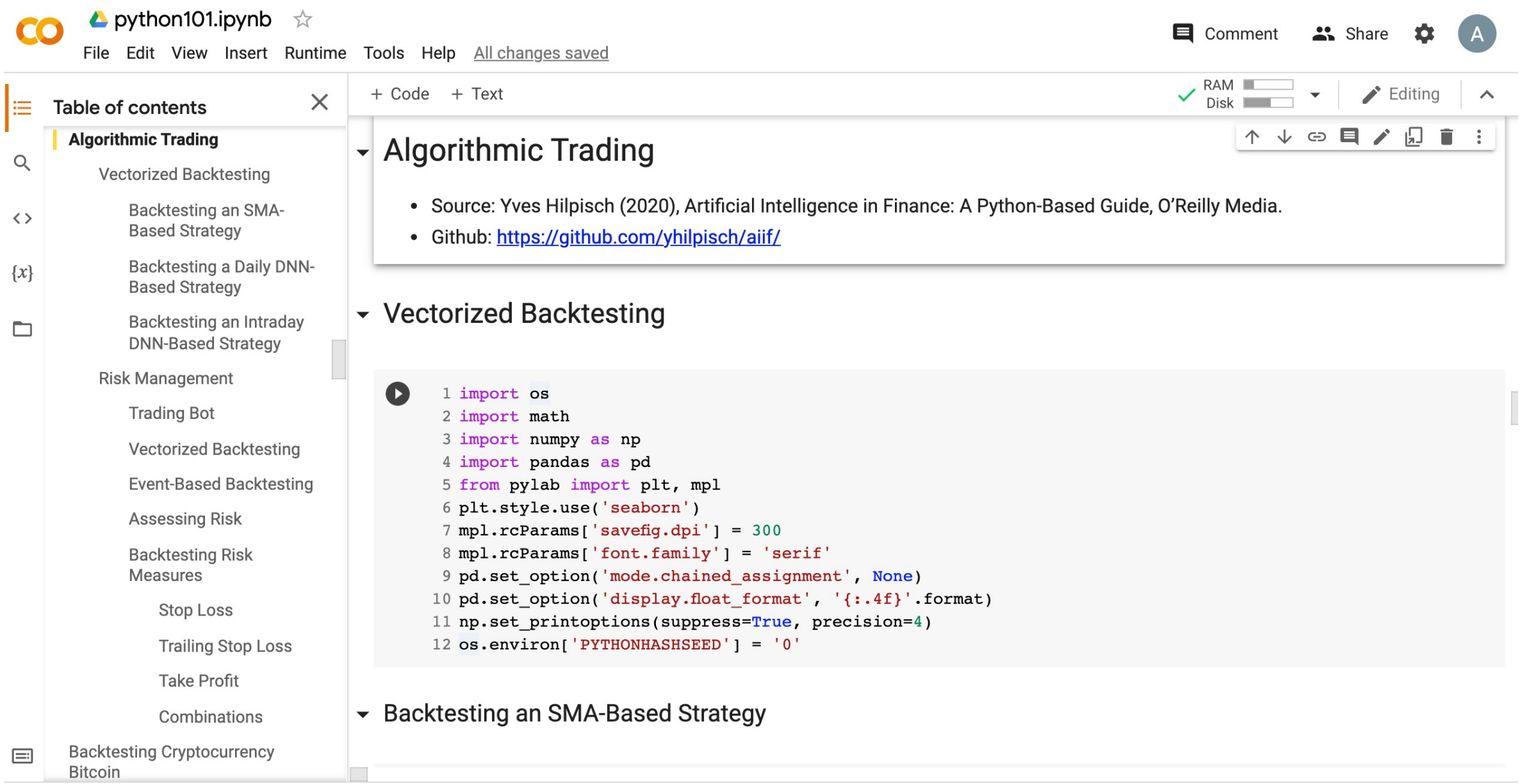
```
1 import os
2 import math
3 import random
4 import numpy as np
5 import pandas as pd
6 from pylab import plt, mpl
7 plt.style.use('seaborn')
8 mpl.rcParams['savefig.dpi'] = 300
9 mpl.rcParams['font.family'] = 'serif'
10 np.set_printoptions(precision=4, suppress=True)
11 os.environ['PYTHONHASHSEED'] = '0'
```

CartPole Environment

```
[ ]  1 import gym
```

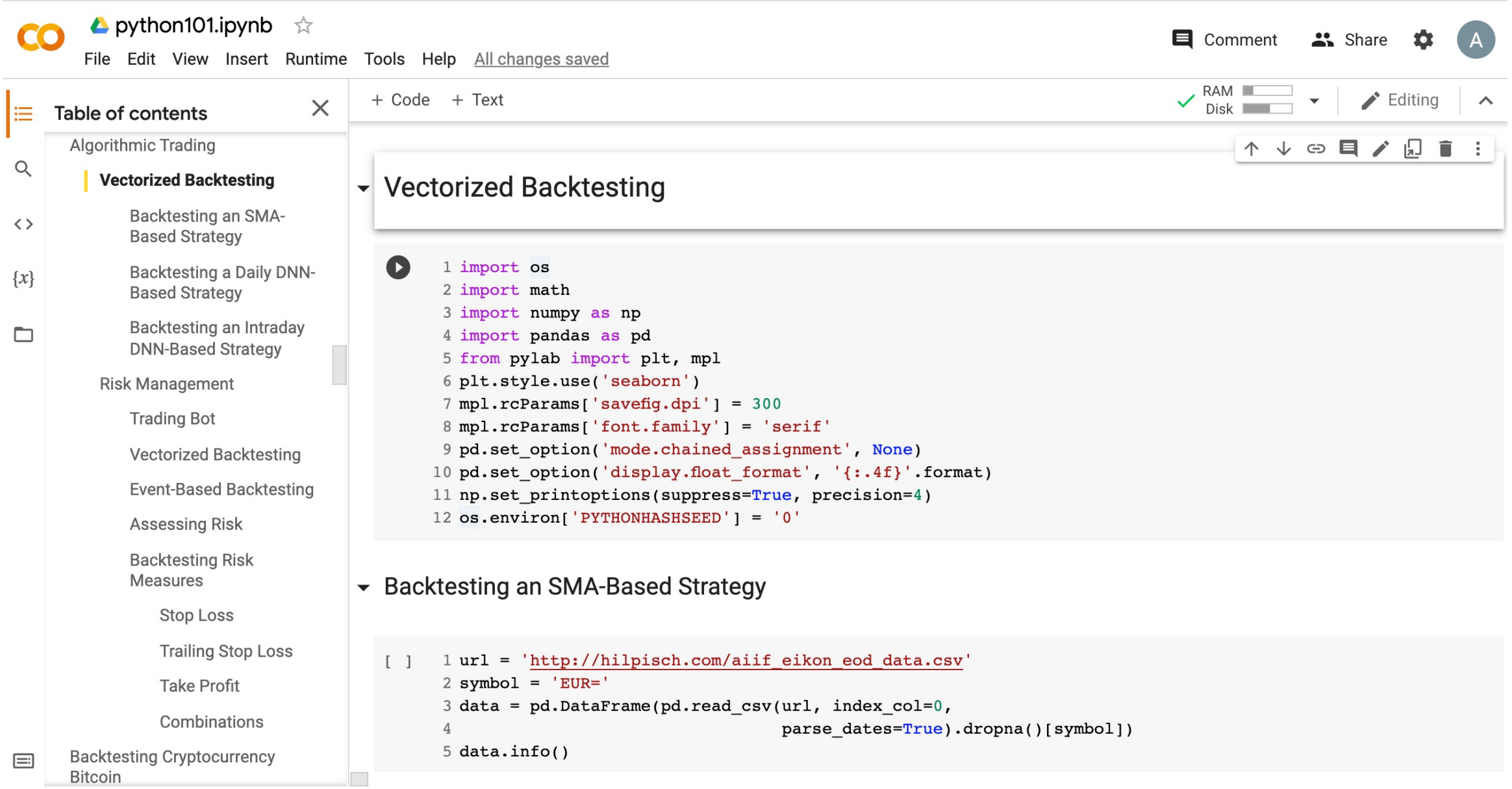
https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



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

- Algorithmic Trading
  - **Vectorized Backtesting**
    - Backtesting an SMA-Based Strategy
    - Backtesting a Daily DNN-Based Strategy
    - Backtesting an Intraday DNN-Based Strategy
  - Risk Management
    - Trading Bot
    - Vectorized Backtesting
    - Event-Based Backtesting
    - Assessing Risk
    - Backtesting Risk Measures
      - Stop Loss
      - Trailing Stop Loss
      - Take Profit
      - Combinations
- Backtesting Cryptocurrency Bitcoin

+ Code + Text

RAM Disk Editing

```
[ ]  1 data['r'] = np.log(data[symbol] / data[symbol].shift(1))
     2 data.dropna(inplace=True)
     3 data['s'] = data['p'] * data['r']
     4 data[['r', 's']].sum().apply(np.exp)  # gross performance
     5 data[['r', 's']].sum().apply(np.exp) - 1  # net performance
     6 data[['r', 's']].cumsum().apply(np.exp).plot(figsize=(10, 6))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd9f404fed0>
```

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

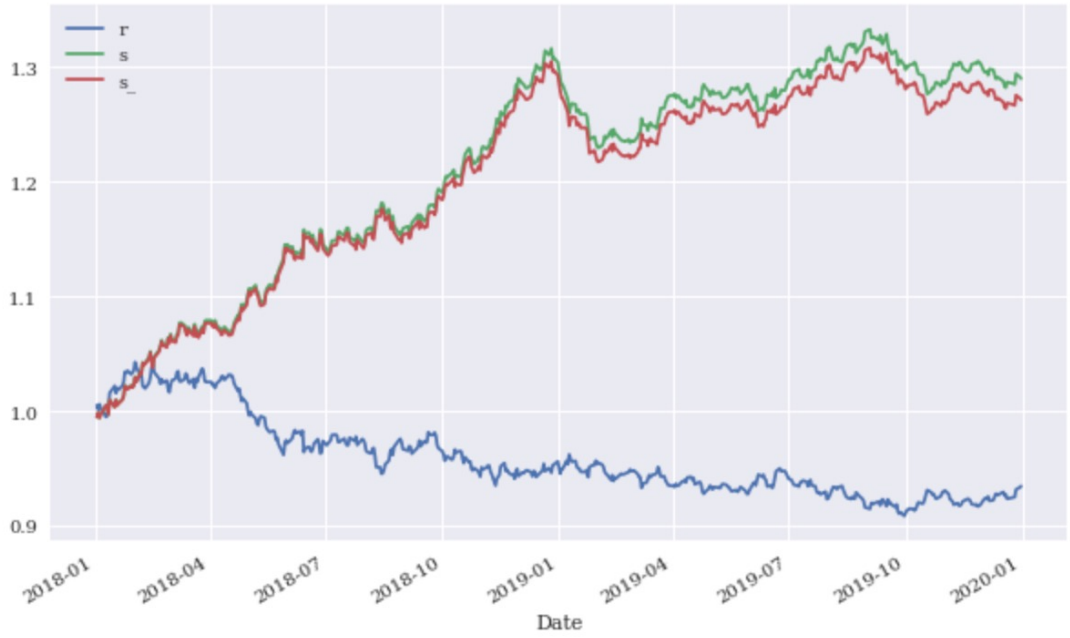
- Algorithmic Trading
  - Vectorized Backtesting
    - Backtesting an SMA-Based Strategy
    - **Backtesting a Daily DNN-Based Strategy**
    - Backtesting an Intraday DNN-Based Strategy
  - Risk Management
    - Trading Bot
    - Vectorized Backtesting
    - Event-Based Backtesting
    - Assessing Risk
    - Backtesting Risk Measures
      - Stop Loss
      - Trailing Stop Loss
      - Take Profit
      - Combinations
- Backtesting Cryptocurrency Bitcoin

+ Code + Text

RAM Disk Editing

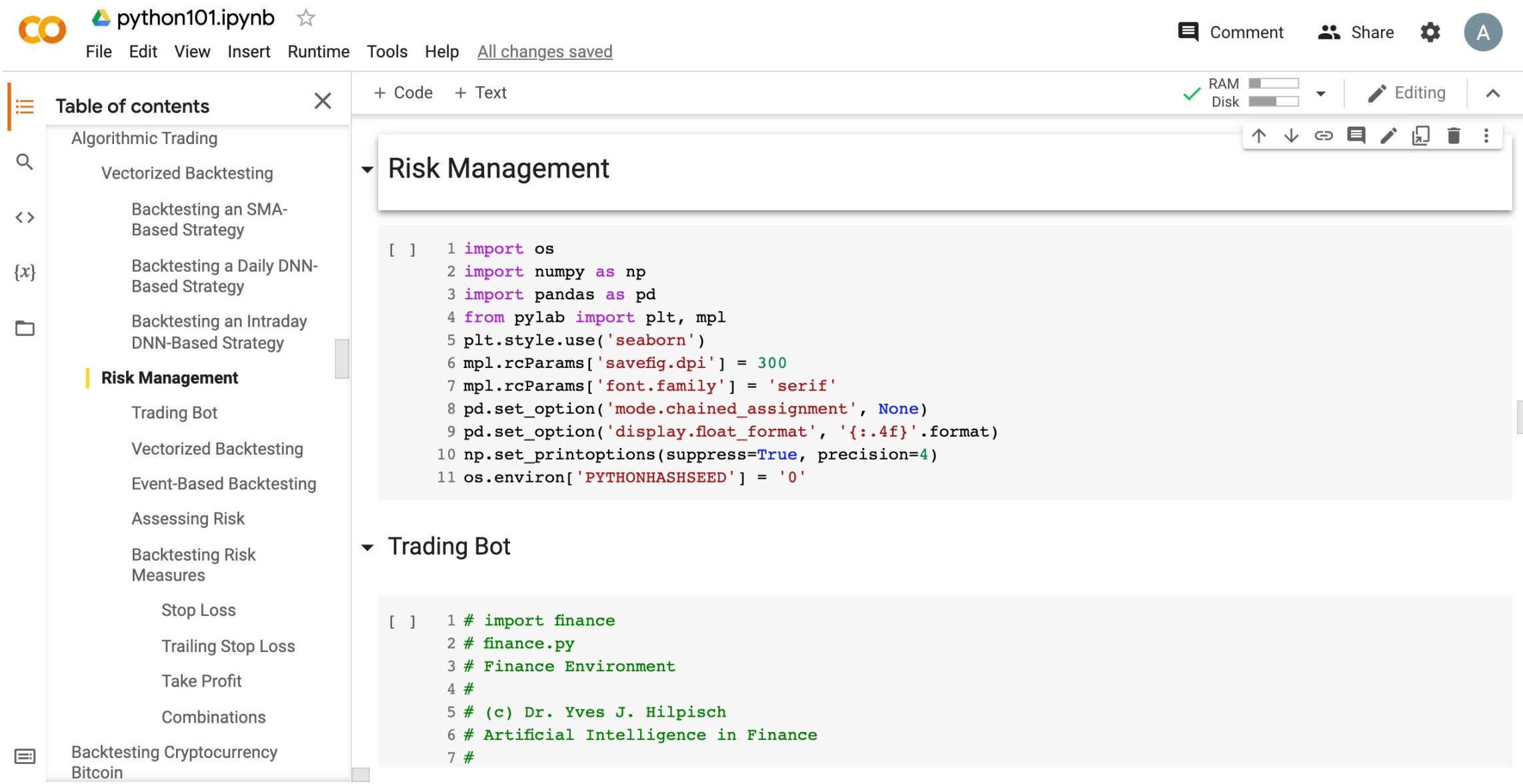
```
test['s_'] = np.where(test['p'].diff() != 0,
                      test['s'] - pc, test['s'])
# test['s_'].iloc[0] -= pc
test['s_'].iloc[-1] -= pc
test[['r', 's', 's_']].sum().apply(np.exp)
test[['r', 's', 's_']].sum().apply(np.exp) - 1
test[['r', 's', 's_']].cumsum().apply(np.exp).plot(figsize=(10, 6))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd901d89910>
```



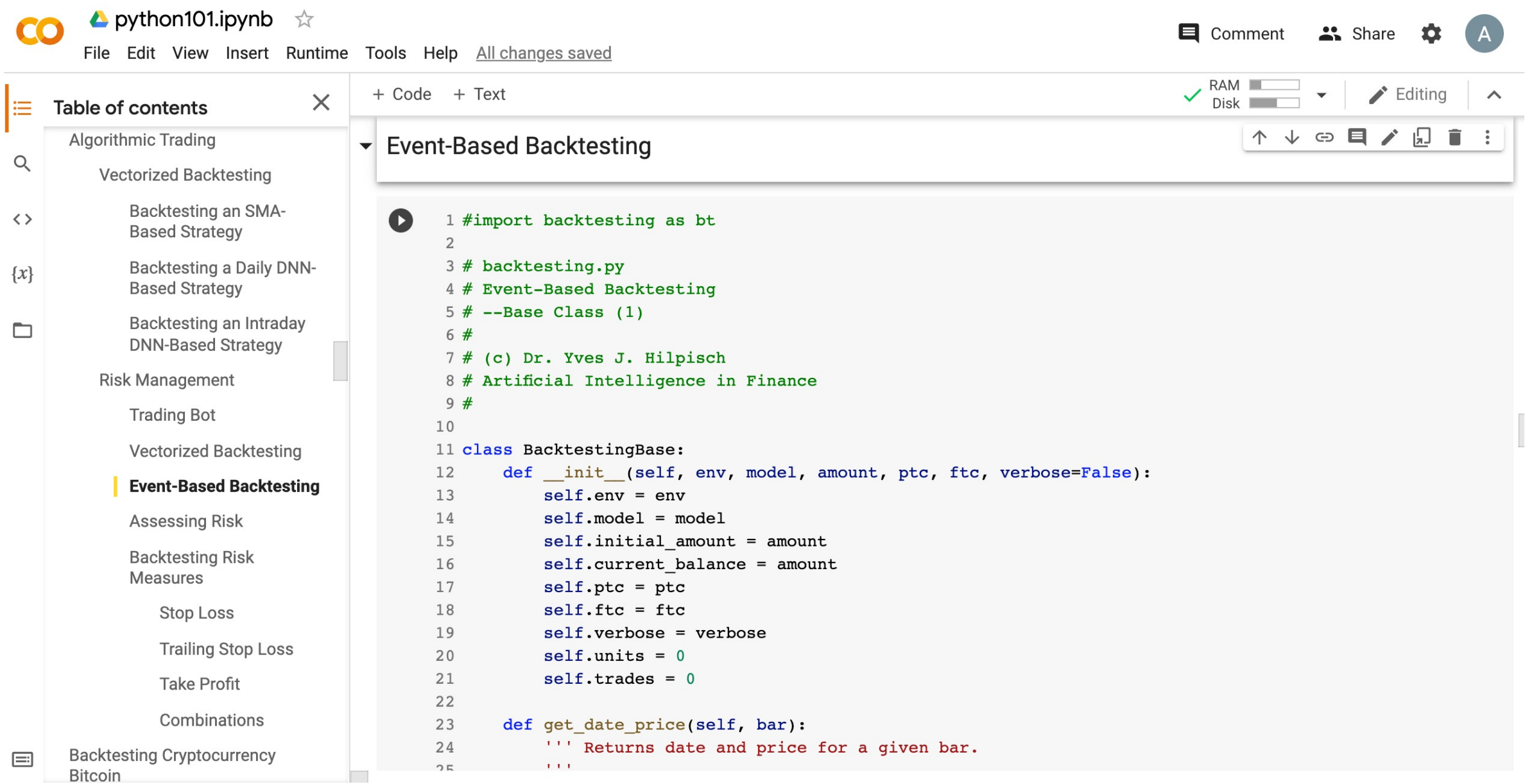
https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



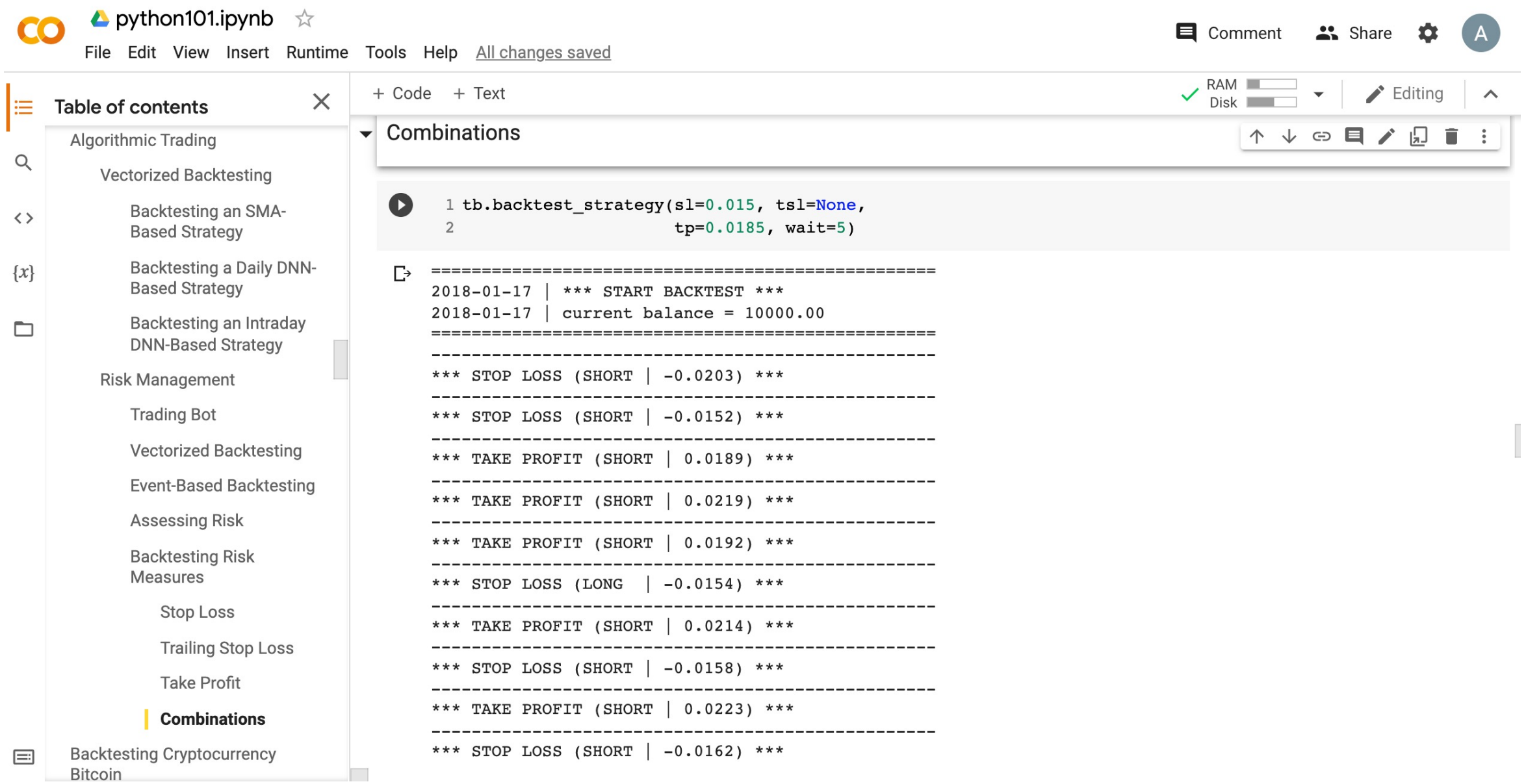
https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



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

- Algorithmic Trading
  - Vectorized Backtesting
    - Backtesting an SMA-Based Strategy
    - Backtesting a Daily DNN-Based Strategy
    - Backtesting an Intraday DNN-Based Strategy
  - Risk Management
    - Trading Bot
    - Vectorized Backtesting
    - Event-Based Backtesting
    - Assessing Risk
    - Backtesting Risk Measures
      - Stop Loss
      - Trailing Stop Loss
      - Take Profit
      - Combinations
- **Backtesting Cryptocurrency Bitcoin**

+ Code + Text RAM Disk Editing

Backtesting Cryptocurrency Bitcoin

- Financial Functions (ffn): https://pmorissette.github.io/ffn/
- backtesting.py: https://kernc.github.io/backtesting.py/

15s

```
!pip install ffn
import ffn
import plotly.express as px
%pylab inline
#BTC-USD Bitcoin USD
df = ffn.get('btc-usd', start='2016-01-01', end='2021-12-31')
print('df')
print(df.head())
print(df.tail())
print(df.describe())
df.plot(figsize=(14,10))

returns = df.to_returns().dropna()
print('returns')
print(returns.head())
print(returns.tail())
print(returns.describe())
#ax = df.plot(figsize=(12,9))

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

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

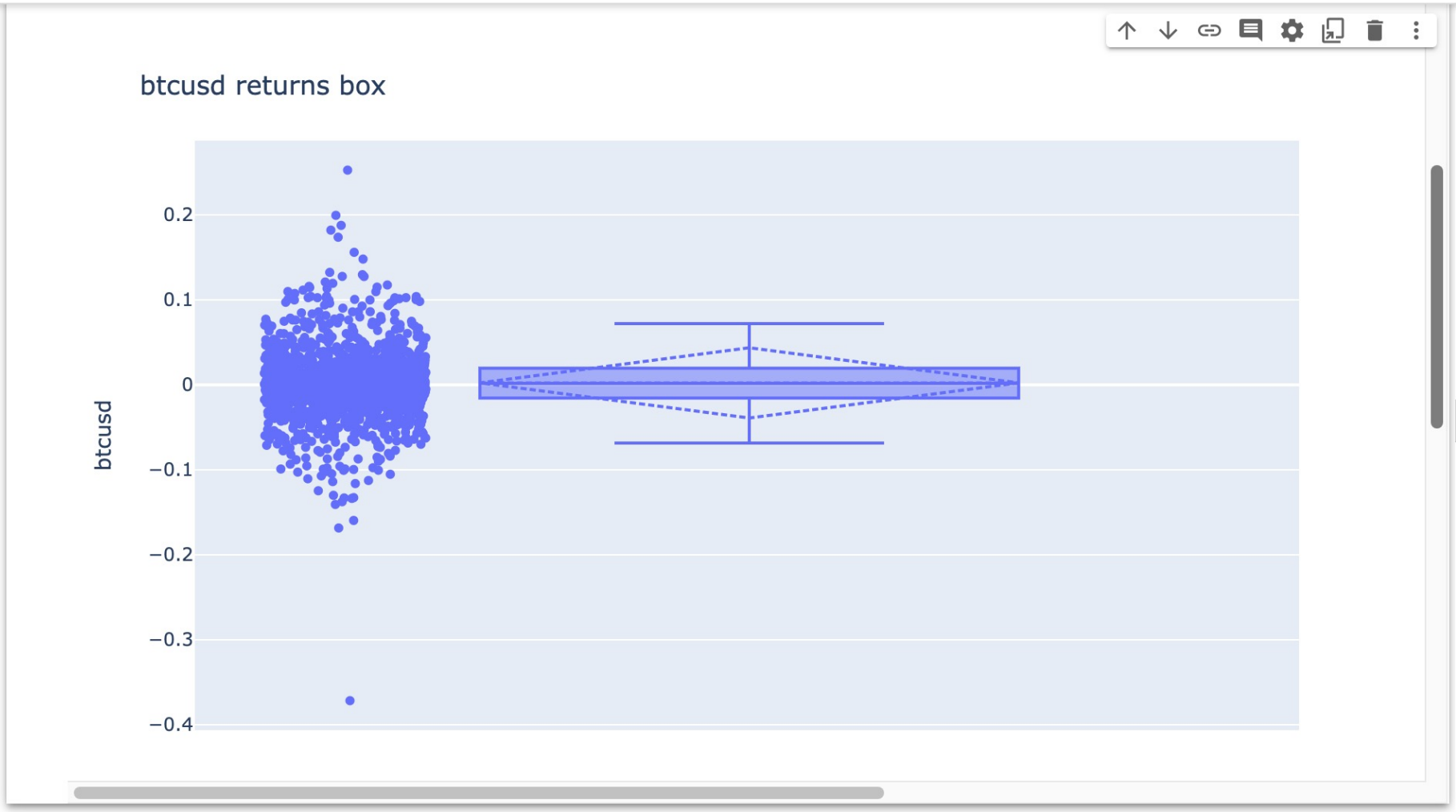
- Algorithmic Trading
  - Vectorized Backtesting
    - Backtesting an SMA-Based Strategy
    - Backtesting a Daily DNN-Based Strategy
    - Backtesting an Intraday DNN-Based Strategy
  - Risk Management
    - Trading Bot
    - Vectorized Backtesting
    - Event-Based Backtesting
    - Assessing Risk
    - Backtesting Risk Measures
      - Stop Loss
      - Trailing Stop Loss
      - Take Profit
      - Combinations

Backtesting Cryptocurrency Bitcoin

+ Code + Text

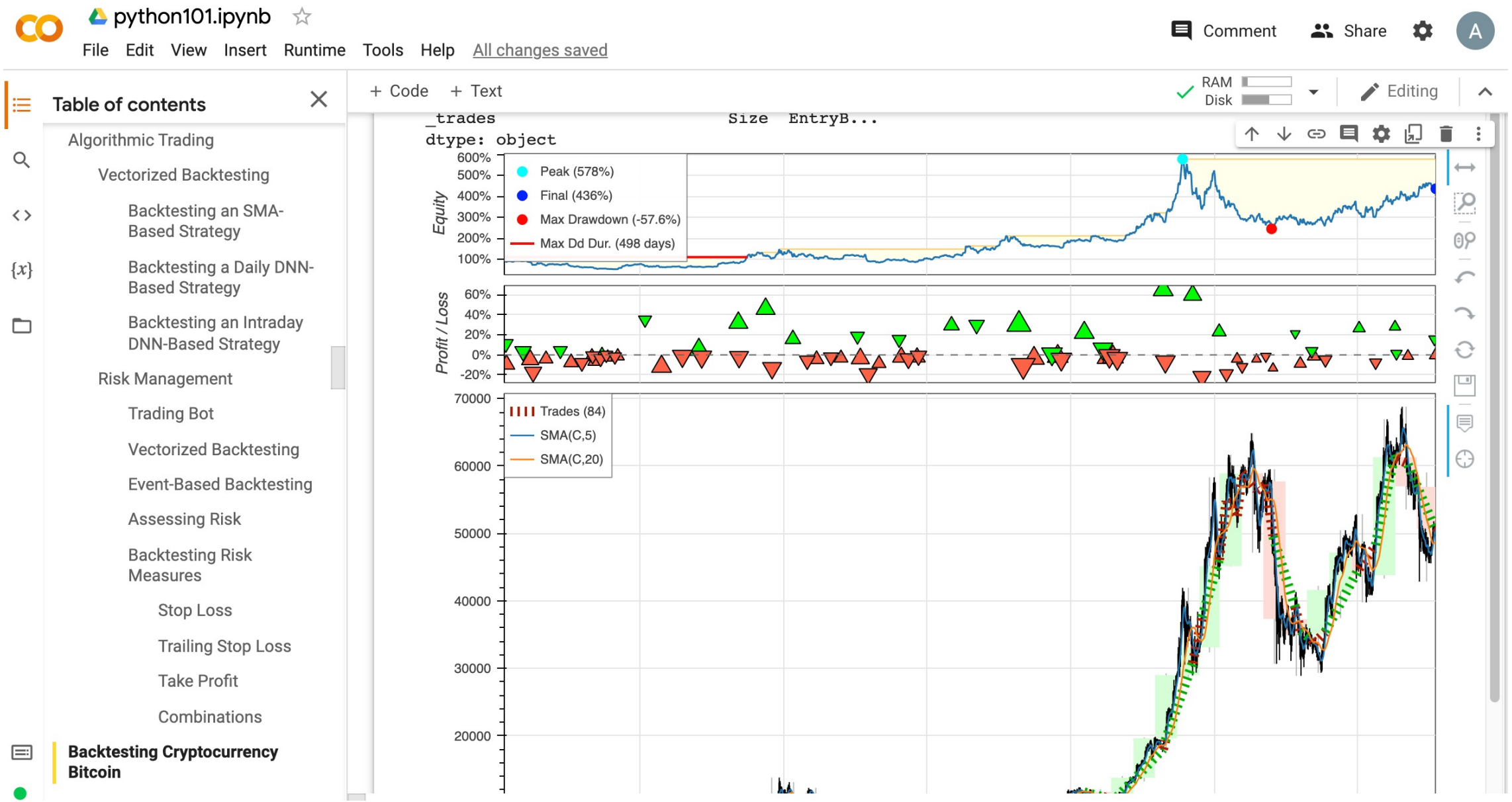
RAM Disk

Editing

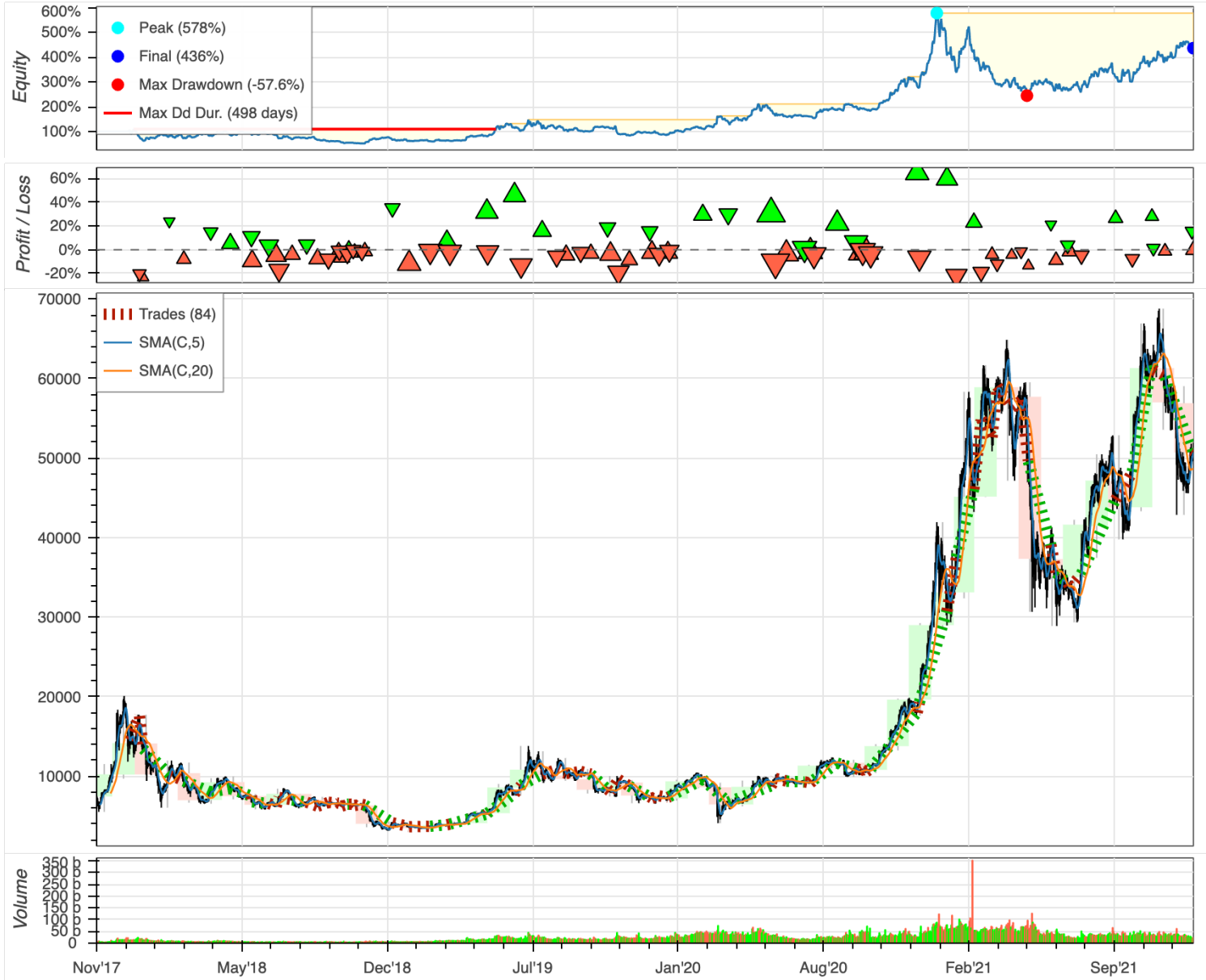


https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101



https://tinyurl.com/aintpupython101

Summary

- **Algorithmic Trading**
- **Risk Management**
- **Trading Bot**
- **Event-Based Backtesting**

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

References

- Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media, https://github.com/yhilpisch/aiif .
- Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition, O'Reilly Media.
- Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.
- Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu (2020), "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019." Applied Soft Computing 90 (2020): 106181.
- Min-Yuh Day (2021), Python 101, https://tinyurl.com/aintpupython101