智慧金融量化分析
(Artificial Intelligence in Finance and Quantitative Analysis)

金融計量經濟學
(Financial Econometrics)

1101AIFQA07
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

2021-11-23

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)

國立臺北大學 NTPU National Taipei University

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)

Financial Econometrics and Machine Learning

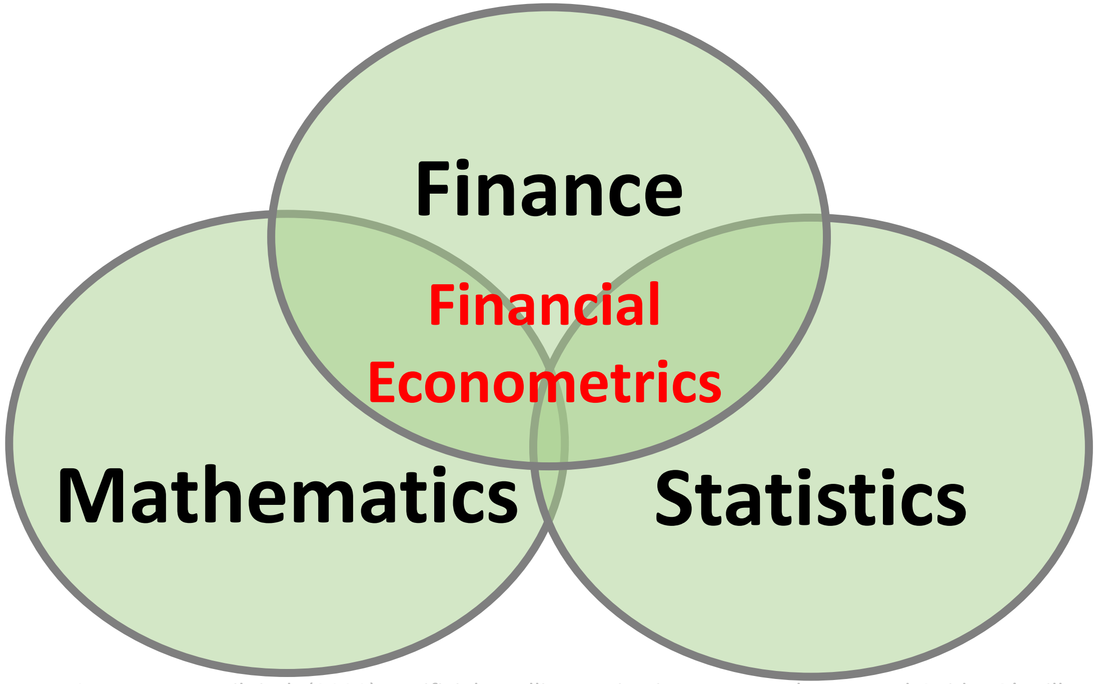
Financial Econometrics and Machine Learning

- Financial Econometrics
  - Financial Theories
  - OLS Regression
- Machine Learning
  - Learning
  - Evaluation
  - Bias and variance
  - Cross-validation

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

Financial Econometrics

- **The discipline at the intersection of mathematics, statistics, and finance that applies such methods to financial market data is typically called financial econometrics.**



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

Financial Econometrics

(Chris Brooks, 2019)

- Financial econometrics
  - the application of statistical techniques to problems in finance
- Financial econometrics can be useful for
  testing theories in finance,
  determining asset prices or returns,
  testing hypotheses concerning the relationships between variables,
  examining the effect on financial markets of changes in economic conditions,
  forecasting future values of financial variables and for financial decision-making.

Source: Chris Brooks (2019), Introductory Econometrics for Finance, 4th Edition, Cambridge University Press

Financial Econometrics

- [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.

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

Topics of Financial Econometrics

(Oliver Linton, 2019)

1. Econometric
2. Return Predictability and the Efficient Markets Hypothesis
3. Robust Tests and Tests of Nonlinear Predictability of Returns
4. Empirical Market Microstructure
5. Event Study Analysis
6. Portfolio Choice and Testing the Capital Asset Pricing Model
7. Multifactor Pricing Models

Source: Oliver Linton (2019), Financial Econometrics: Models and Methods, Cambridge University Press

Topics of Financial Econometrics

(Oliver Linton, 2019)

8. Present Value Relations
9. Intertemporal Equilibrium Pricing
10. Volatility
11. Continuous Time Processes
12. Yield Curve
13. Risk Management and Tail Estimation

Source: Oliver Linton (2019), Financial Econometrics: Models and Methods, Cambridge University Press

Applications of Financial Econometrics

(Chris Brooks, 2019)

1. **Testing whether financial markets are weak-form informationally efficient**
2. **Testing whether the capital asset pricing model (CAPM) or arbitrage pricing theory (APT) represent superior models for the determination of returns on risky assets**
3. **Measuring and forecasting the volatility of bond returns**
4. **Explaining the determinants of bond credit ratings used by the ratings agencies**
5. **Modelling long-term relationships between prices and exchange rates**

Source: Chris Brooks (2019), Introductory Econometrics for Finance, 4th Edition, Cambridge University Press

Applications of Financial Econometrics

(Chris Brooks, 2019)

6. Determining the optimal hedge ratio for a spot position in oil
7. Testing technical trading rules to determine which makes the most money
8. Testing the hypothesis that earnings or dividend announcements have no effect on stock prices
9. Testing whether spot or futures markets react more rapidly to news
10. Forecasting the correlation between the stock indices of two countries

Source: Chris Brooks (2019), Introductory Econometrics for Finance, 4th Edition, Cambridge University Press

Machine Learning and Financial Econometrics

- **ML and DL methods** are able to **discover statistical inefficiencies** and even **economic inefficiencies** that are not discoverable by **traditional econometric methods**, such as multivariate OLS regression.

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

Normative Financial Theories

- 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.

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

Normative Financial Theories

- 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.

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

Financial Econometrics and Regression

- One of the major tools in financial econometrics is regression, in both its univariate and multivariate forms
  - $\boldsymbol{y = \alpha + \beta\, x}$
  - $\boldsymbol{y = \alpha + \beta_1\, x_1 + \beta_2\, x_2}$
  - $\boldsymbol{y = \alpha + \beta_1\, x_1 + \beta_2\, x_2 + \beta_3\, x_3}$
- Regression is also a central tool in statistical learning in general

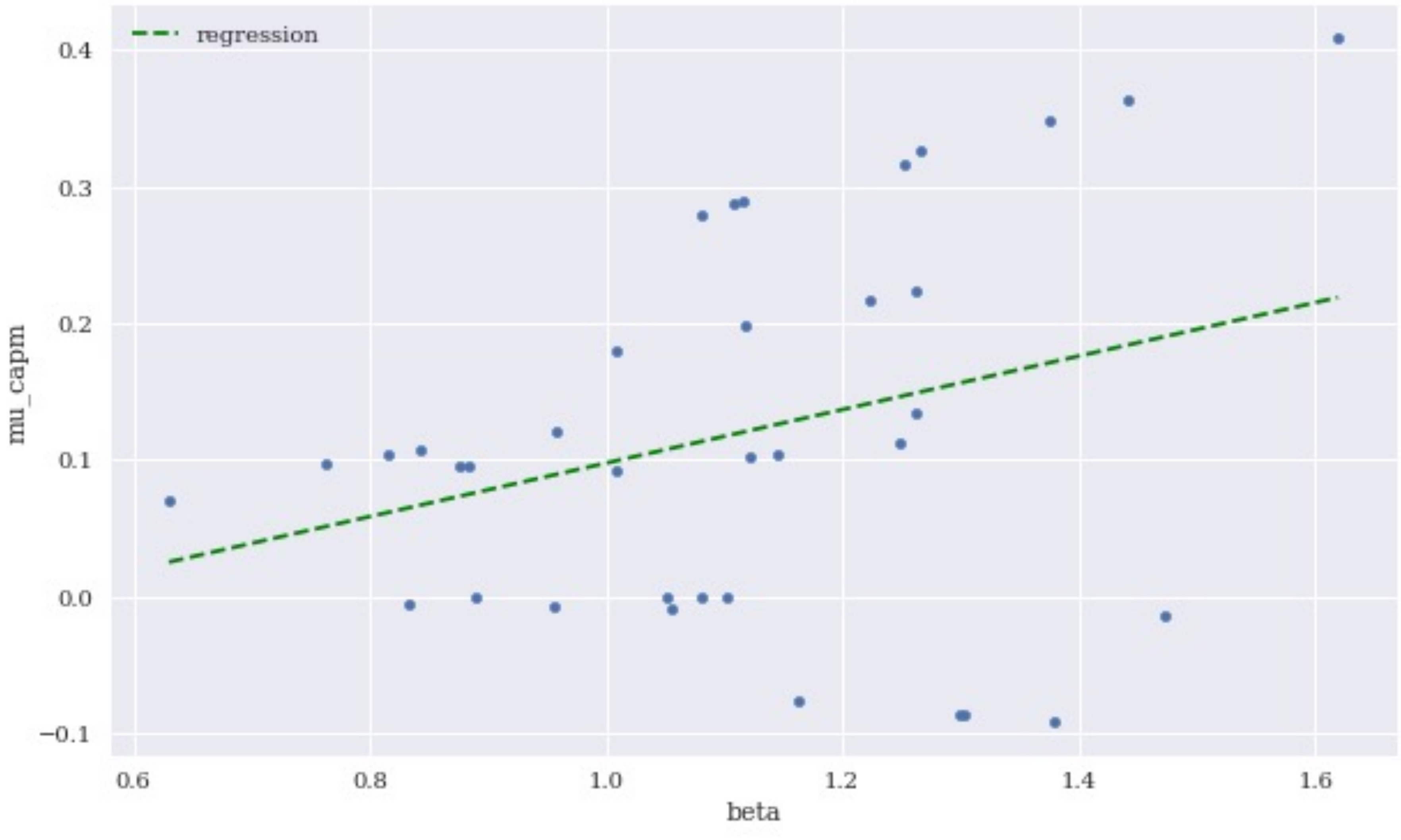
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

CAPM and APT
OLS regression

- Both the **CAPM** and the **APT** relate the **output variables** with the relevant **input factors** in **linear** fashion.
- From an econometric point of view, both models are implemented based on linear **ordinary least-squares (OLS) regression**.
- **CAPM**: univariate linear OLS regression
- **APT**: multivariate OLS regression

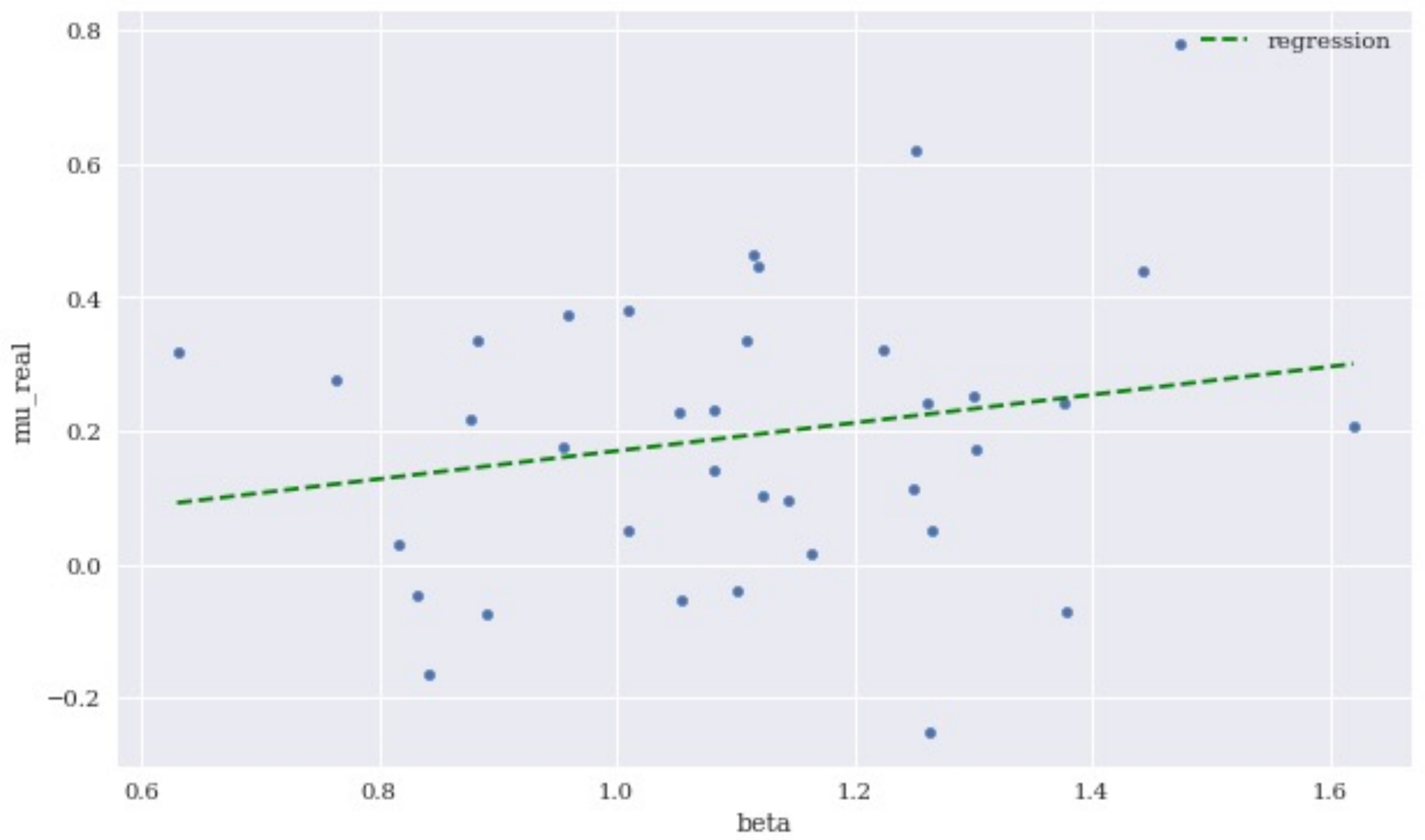
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Expected CAPM return versus beta (including linear regression)



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

Expected CAPM return versus beta (including linear regression)



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

Machine Learning

- Learning
- Data: Features, Labels
- Success (Loss Function): MSE
- Capacity (Model Fit)
- Evaluation
- Bias and variance
- Cross-validation

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

Learning (Mitchell, 1997)

- A computer program is said
  to learn from experience E
  with respect to some class of tasks T
  and performance measure P,
  if its performance at tasks in T,
  as measured by P,
  improves with experience E.

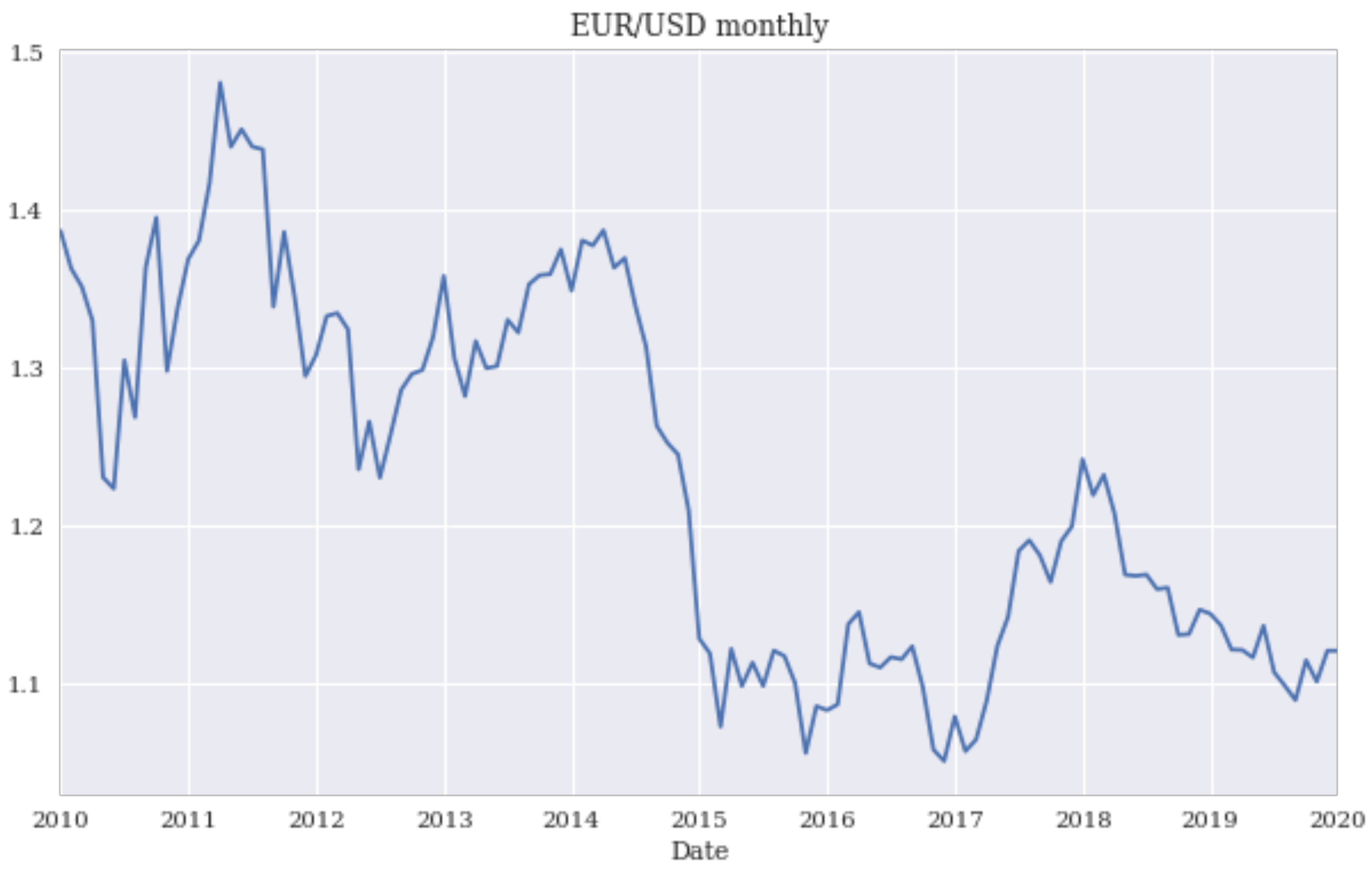
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Performance Measure

- The measure of success for estimation problems
  - mean-squared error (MSE)
- Classification problems
  - accuracy ratio

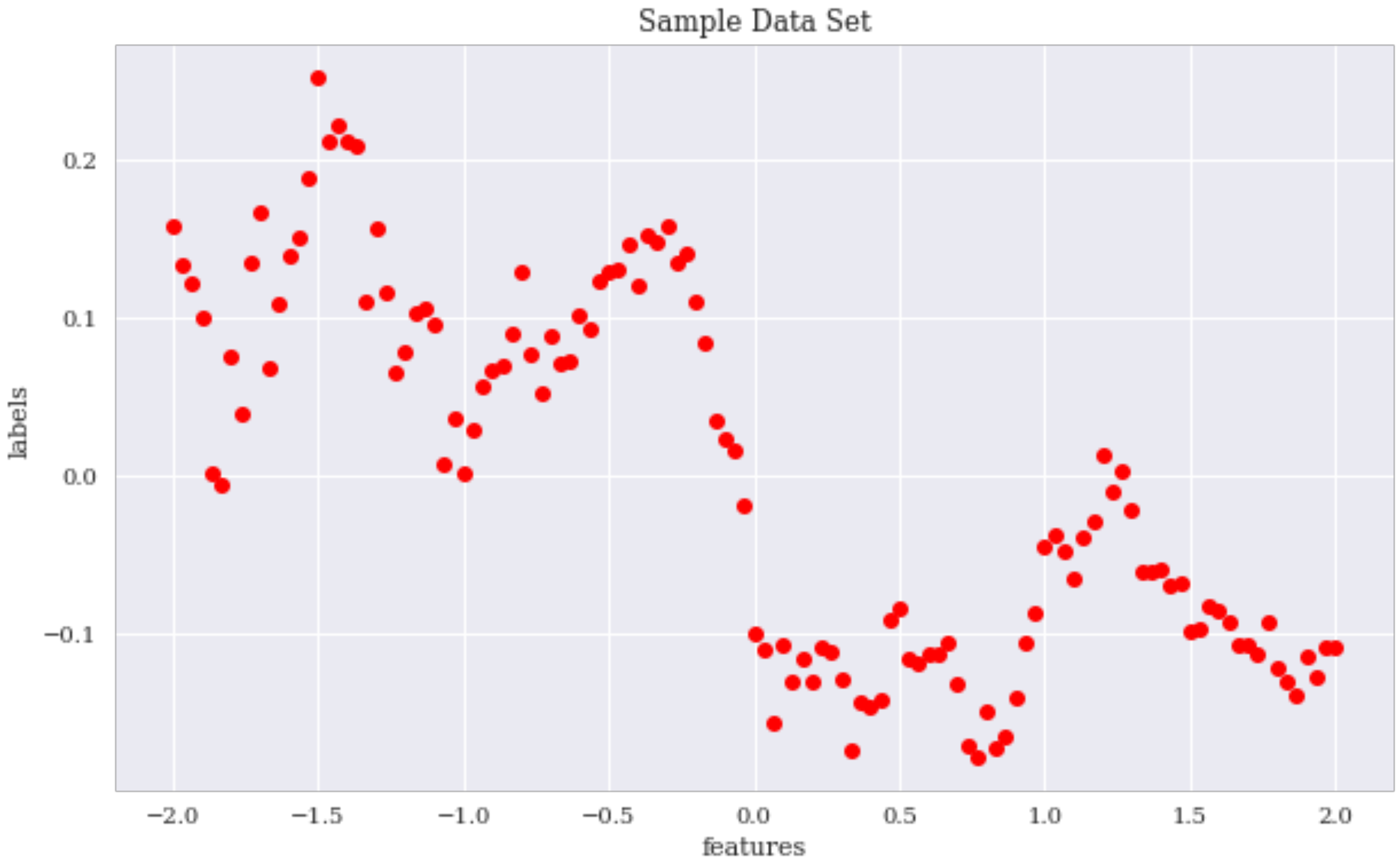
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

EUR/USD exchange rate as time series (monthly)



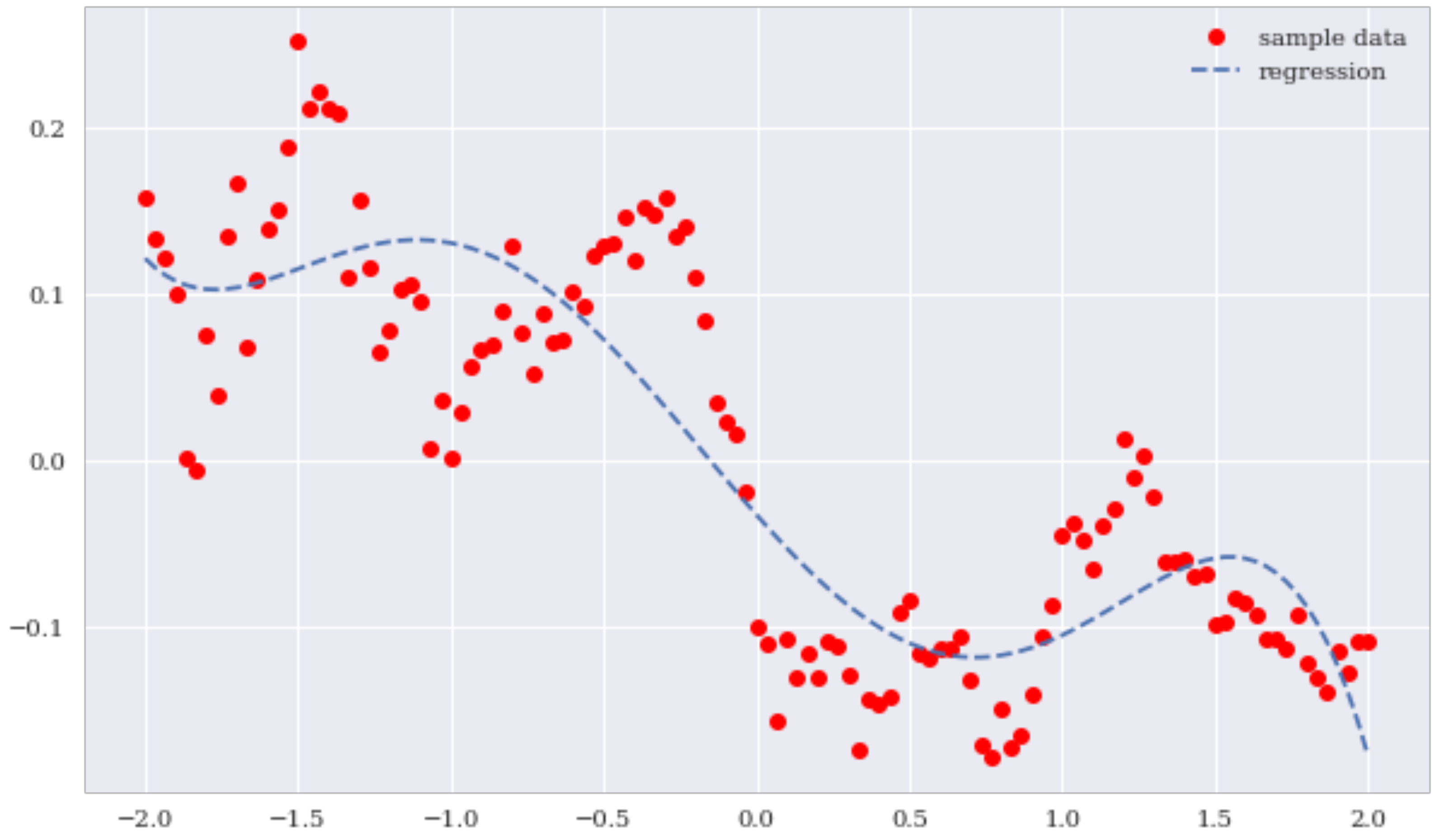
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Sample data set



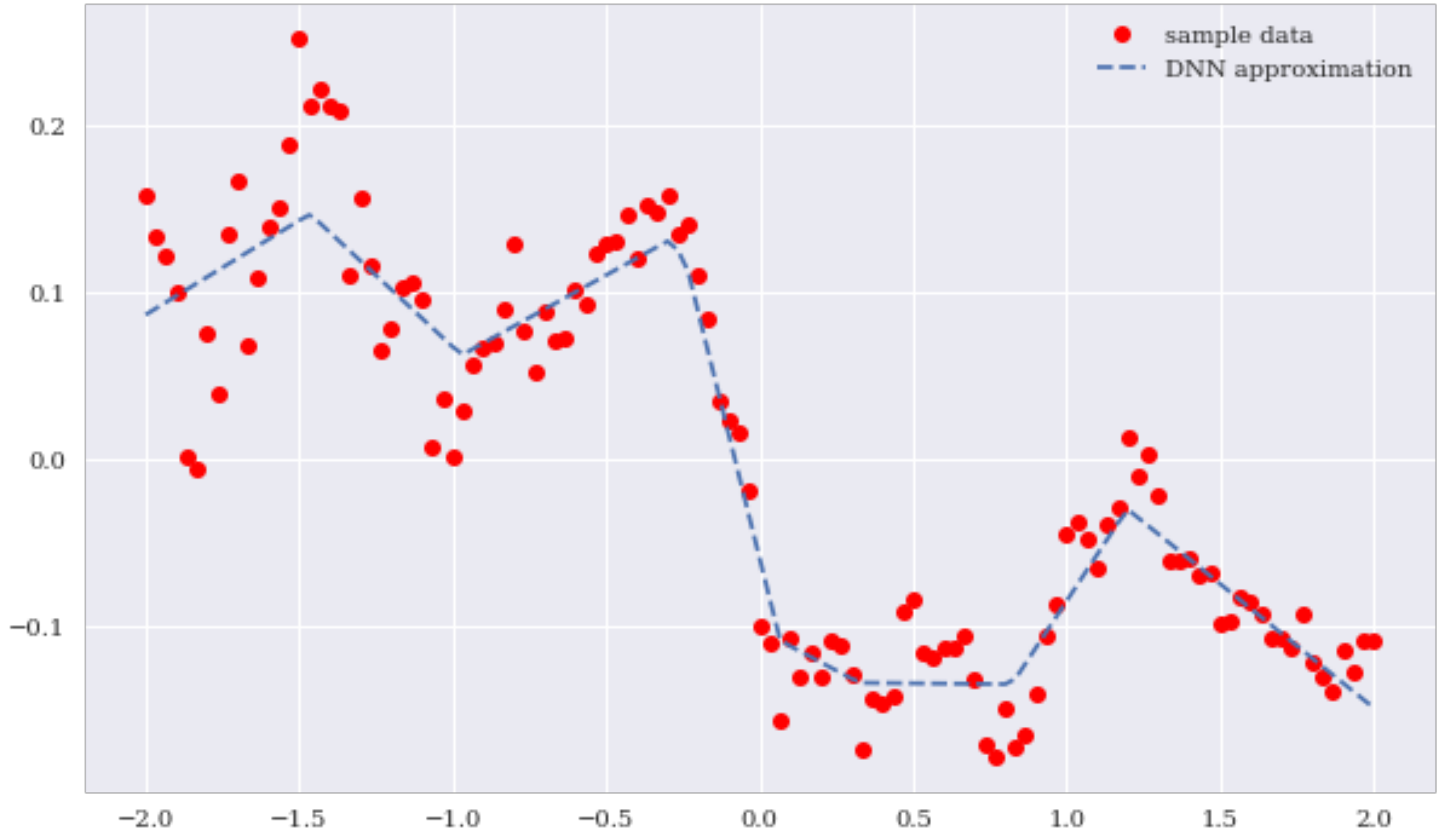
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Sample data and cubic regression line



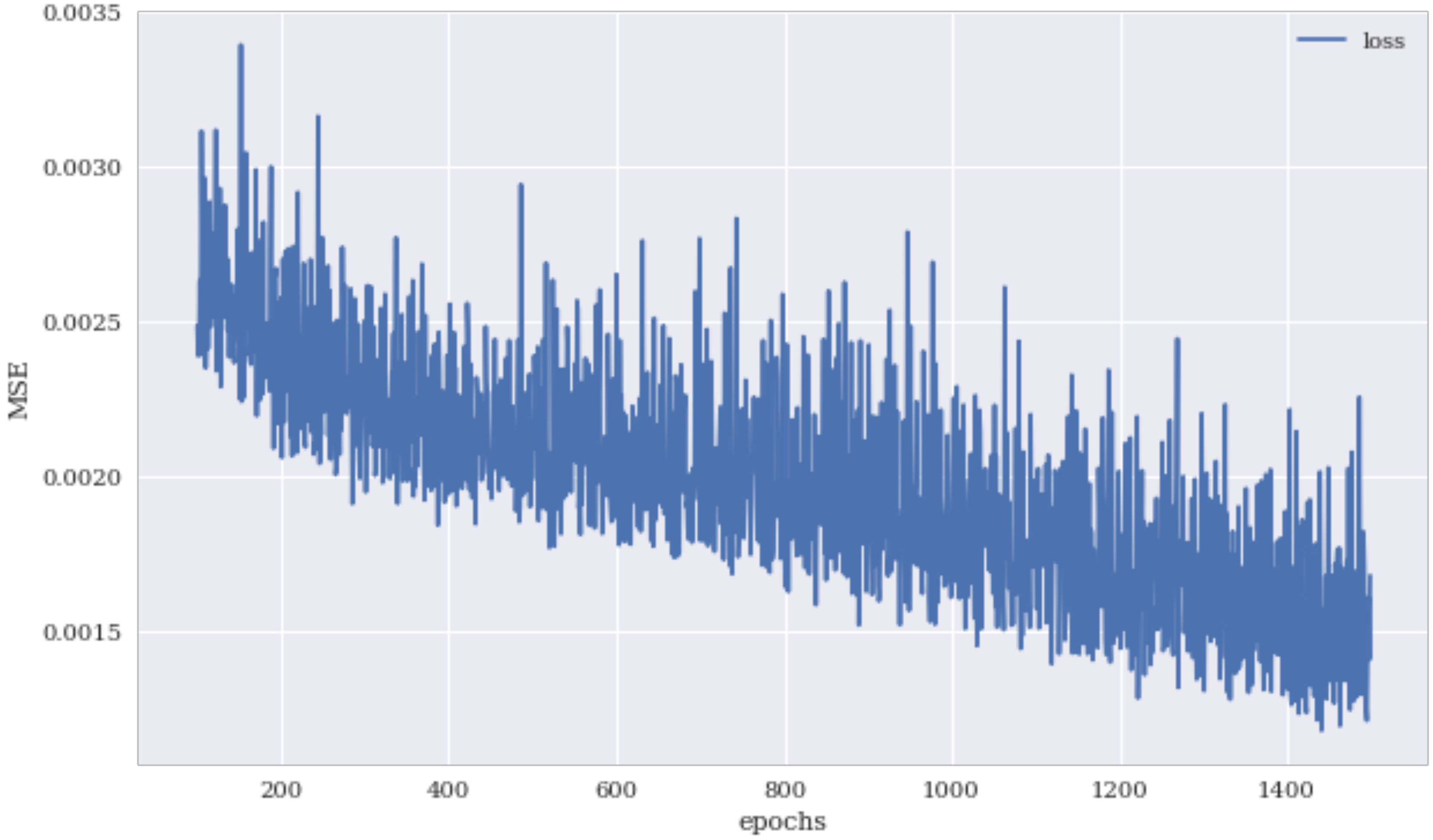
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Sample data and neural network approximation



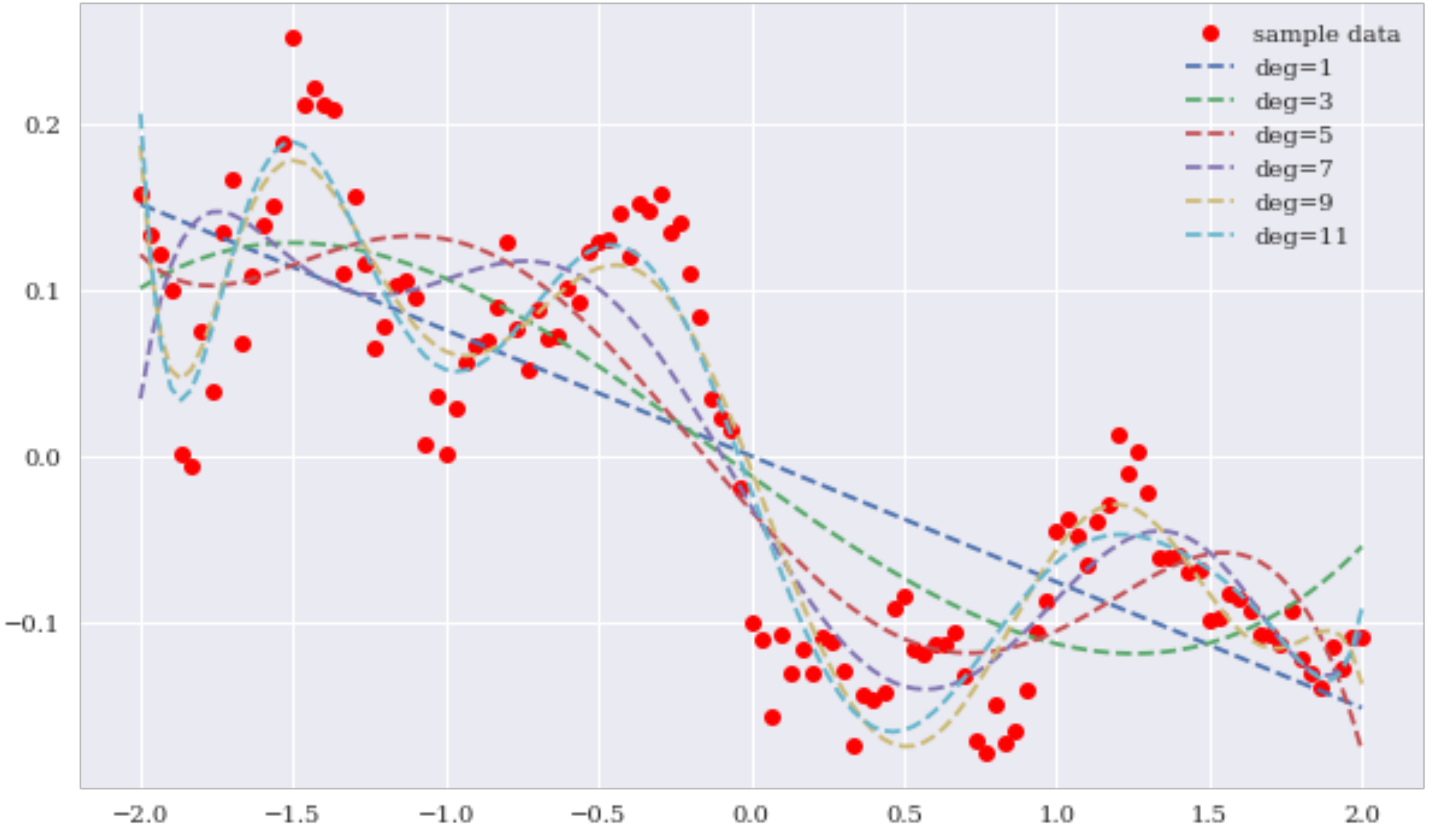
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

MSE values against number of training epochs



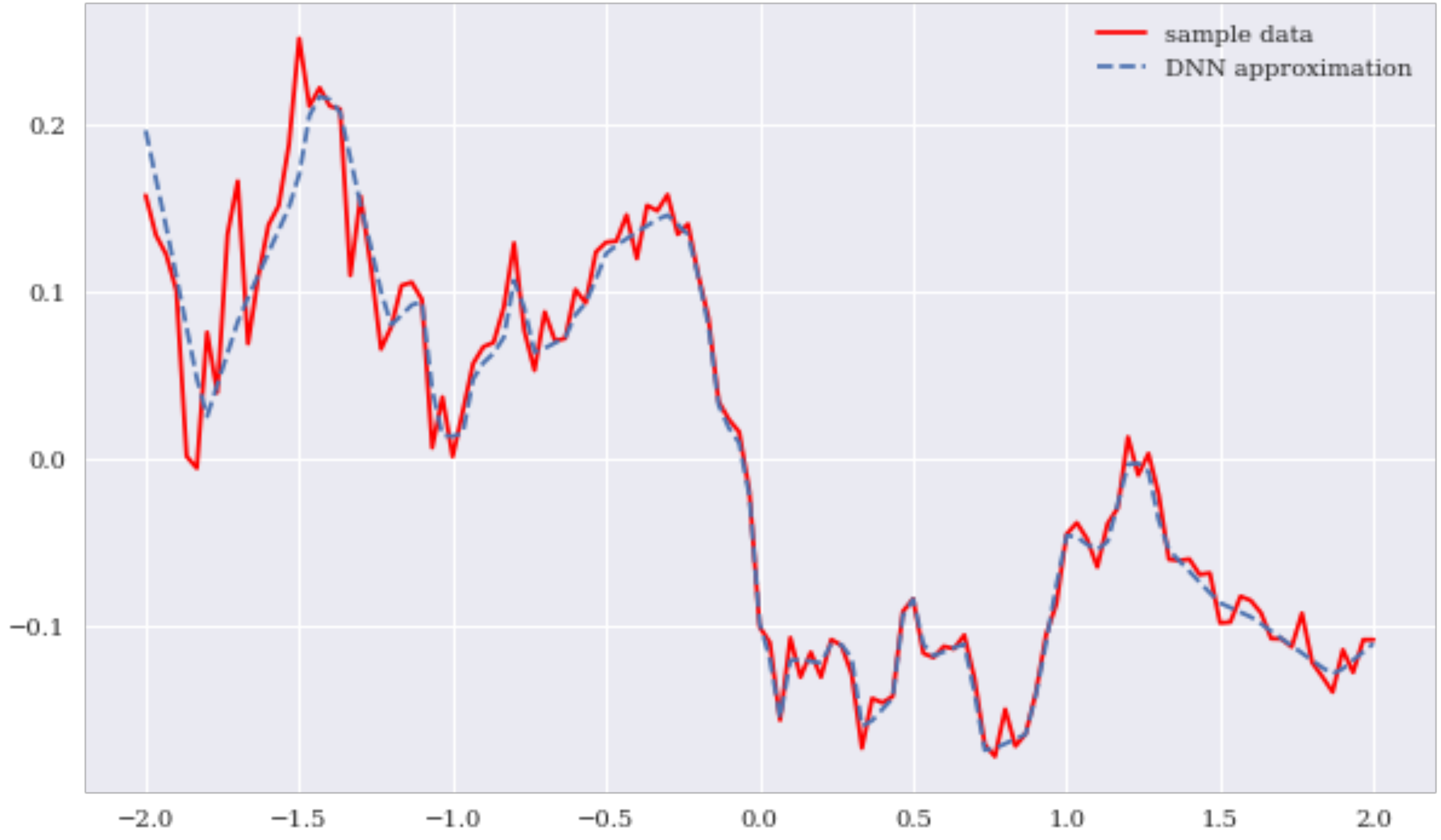
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Regression lines for different highest degrees



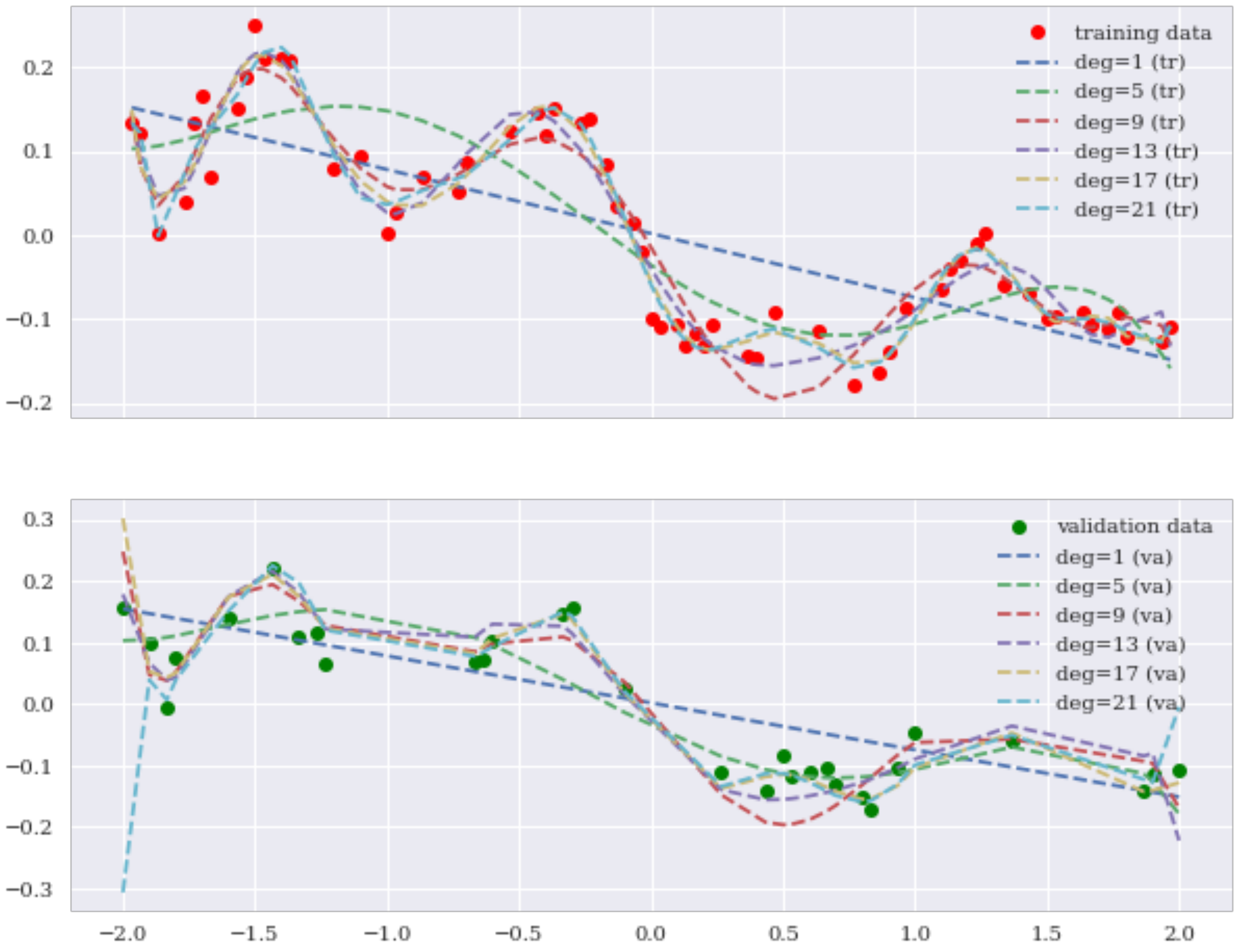
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Sample data and DNN approximation (higher capacity)



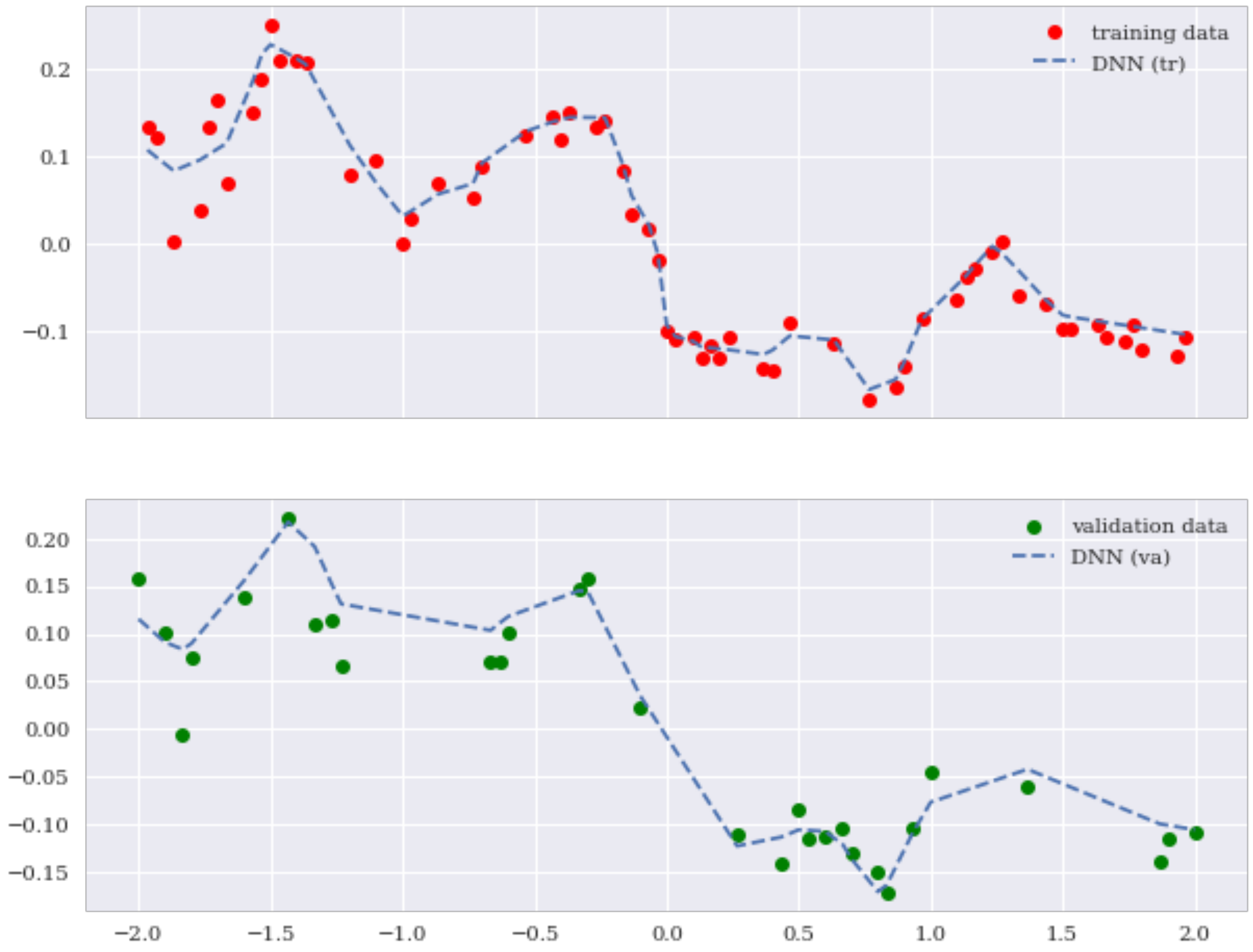
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Training and validation data including regression fits



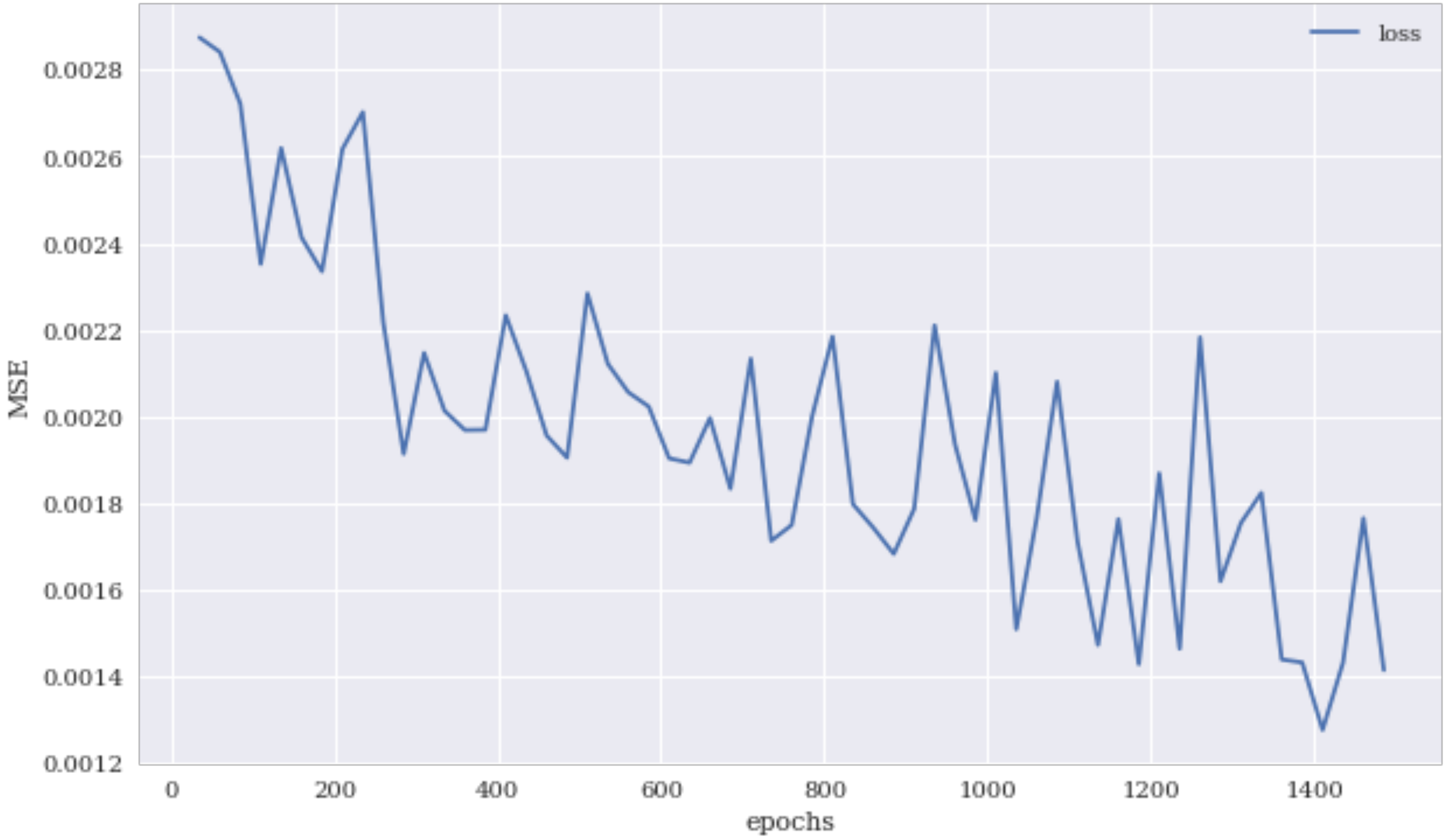
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Training and validation data including DNN predictions



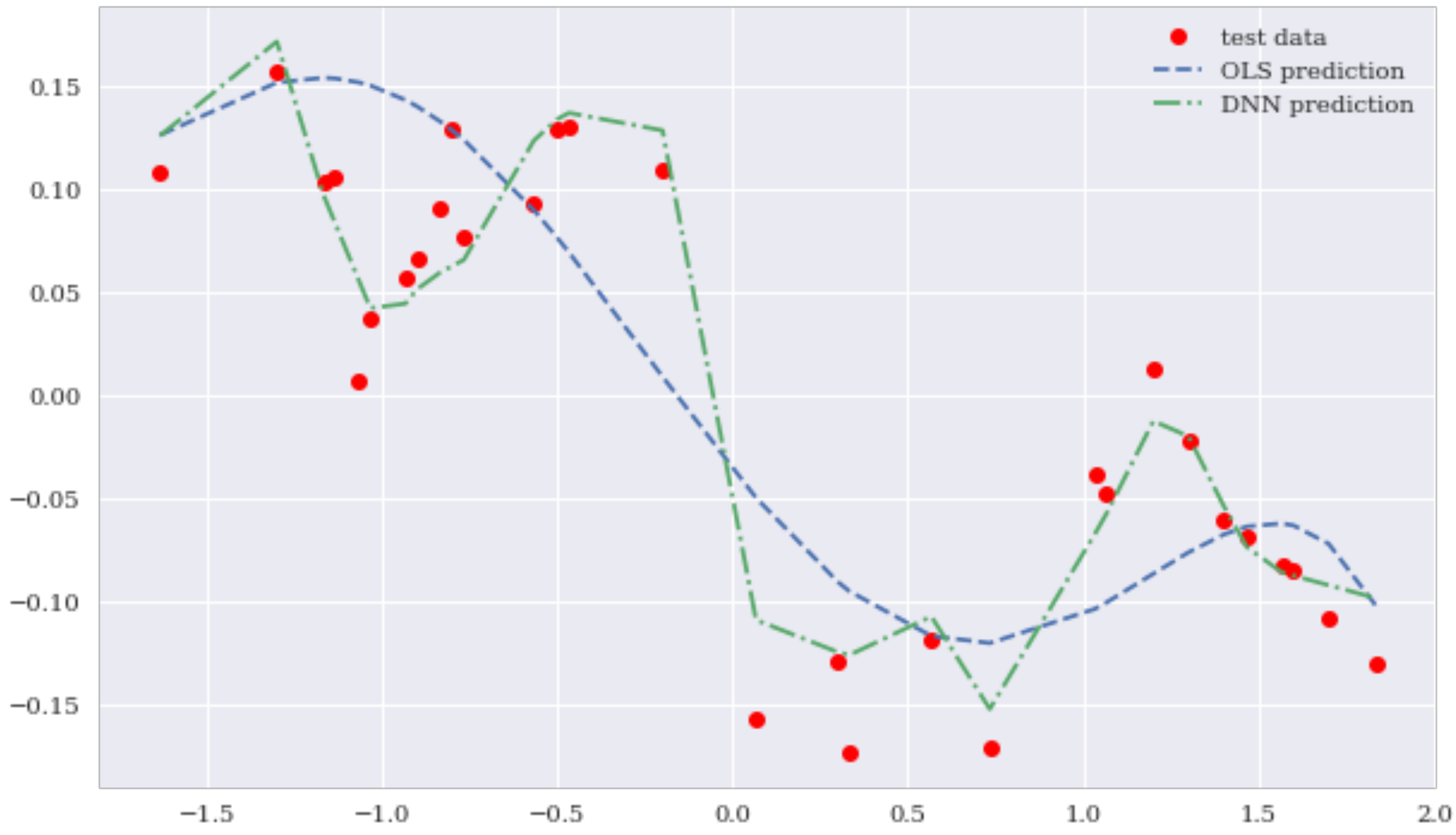
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

MSE values for DNN model on the training and validation data sets



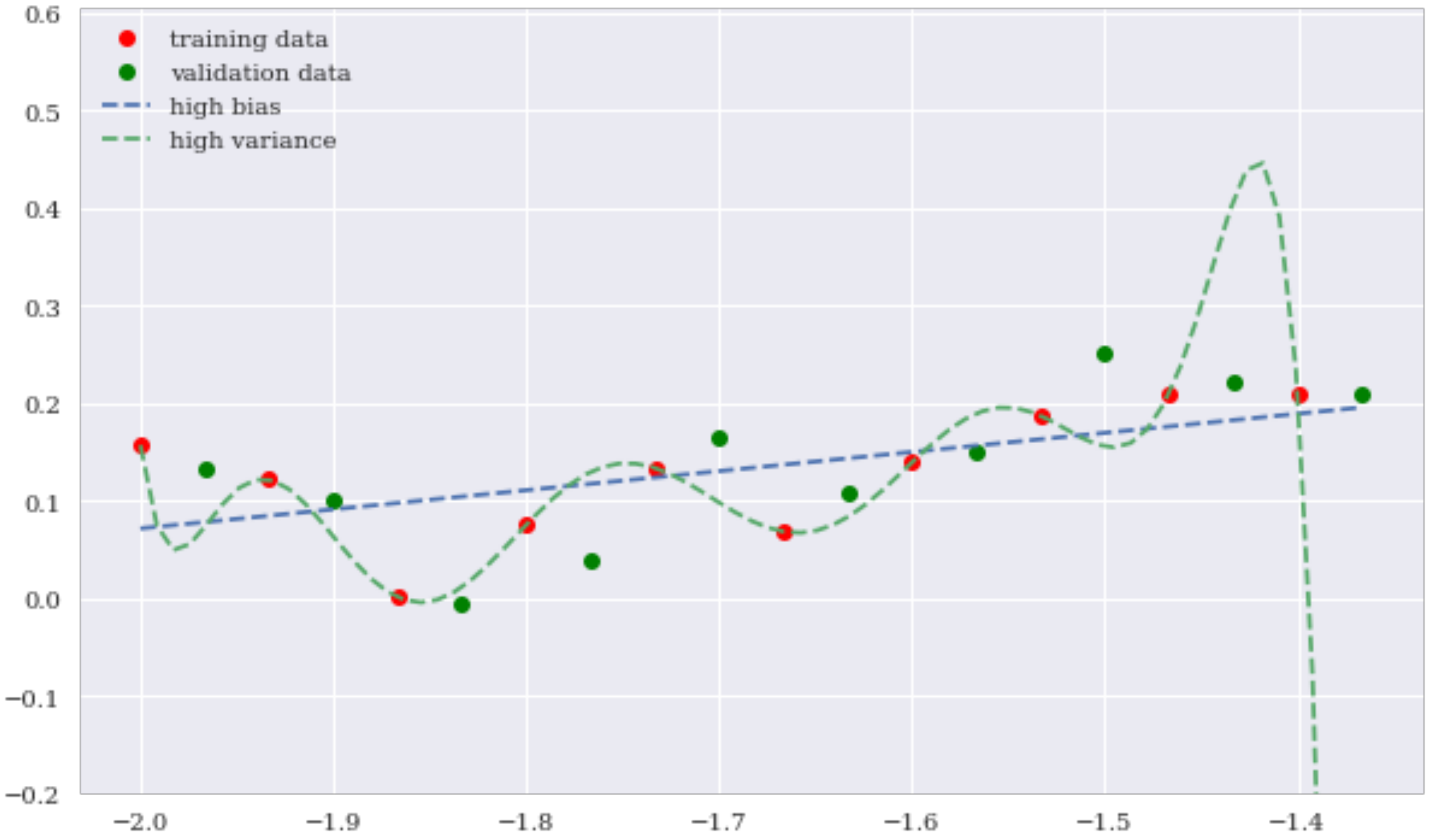
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Test data and predictions from OLS regression and the DNN model



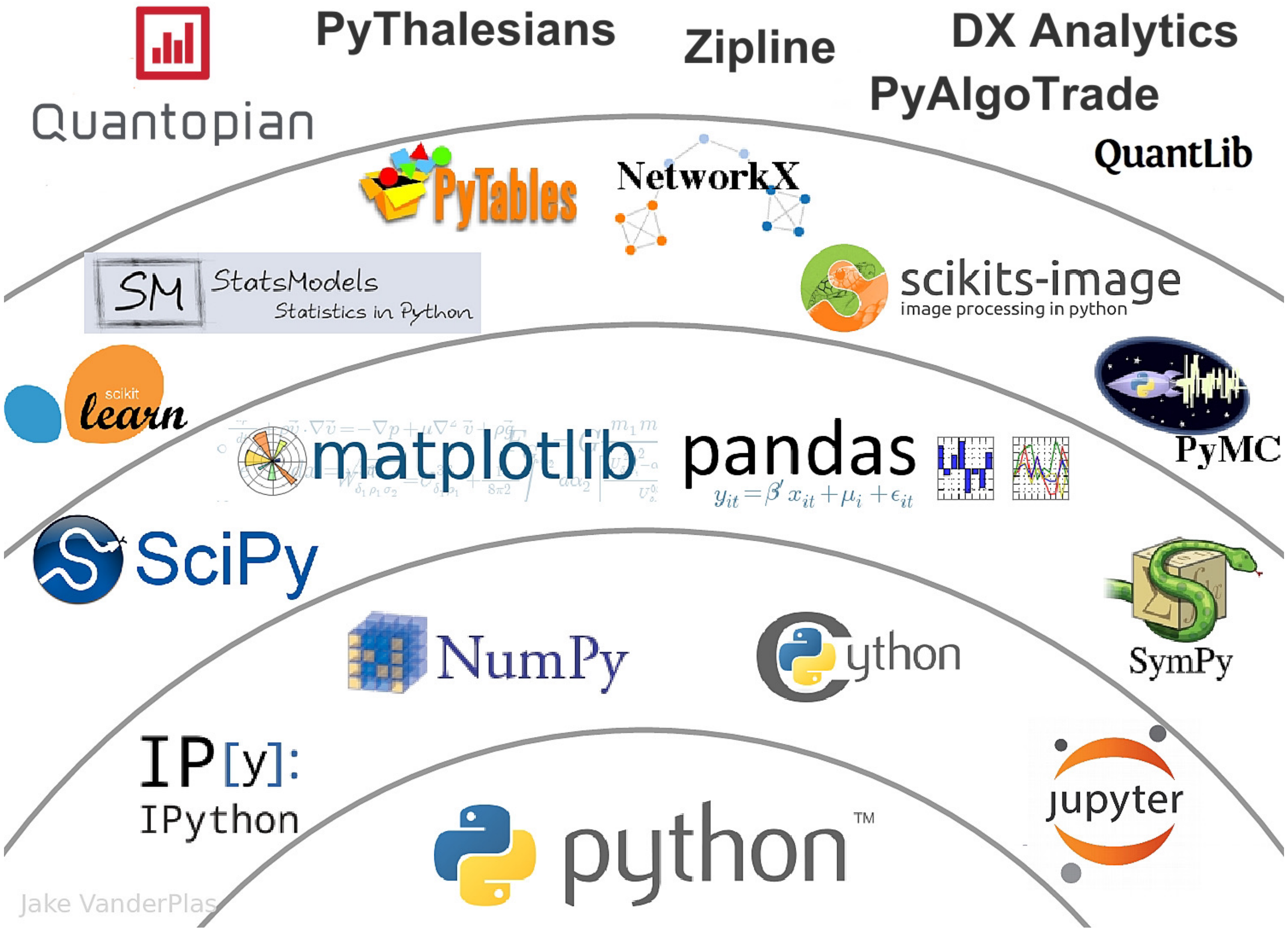
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

High bias and high variance OLS regression fits



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

The Quant Finance PyData Stack

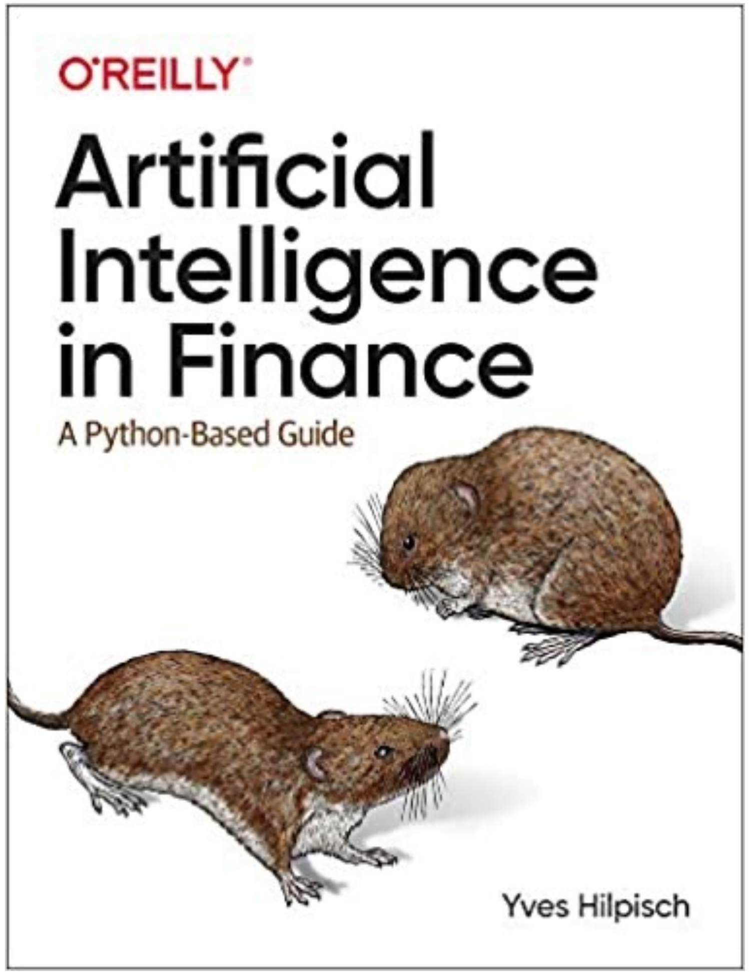


Source: http://nbviewer.jupyter.org/format/slides/github/quantopian/pyfolio/blob/master/pyfolio/examples/overview_slides.ipynb#/5

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

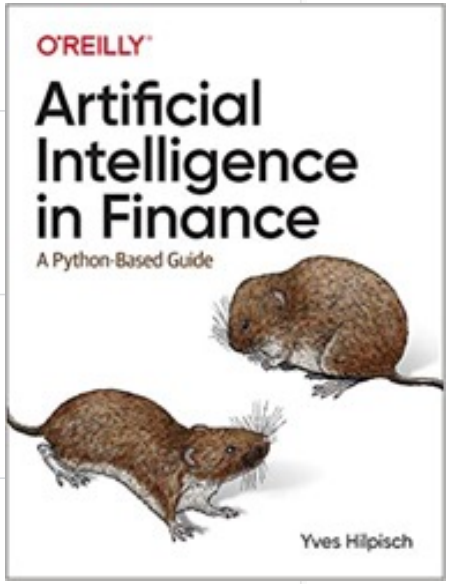
No releases published

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

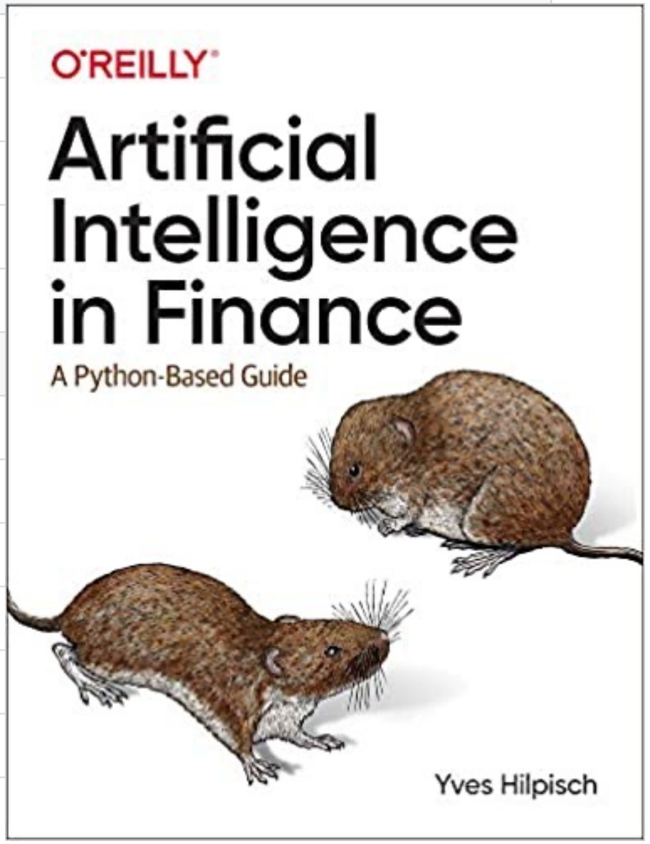
main | aiif / code /

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

python101.ipynb - Colaboratory

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT?authuser=2#scrollTo=wsh36fLxDKC3

python101.ipynb

File Edit View Insert Runtime Tools Help

COMMENT SHARE

CODE TEXT CELL CELL CONNECTED EDITING

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

```
194.87
```

[11]

```
amount = 100
interest = 10 #10% = 0.01 * 10
years = 7

future_value = amount * ((1 + (0.01 * interest)) ** years)
print(round(future_value, 2))
```

```
194.87
```

[12]

```
# Python Function def
def getfv(pv, r, n):
    fv = pv * ((1 + (r)) ** n)
    return fv
fv = getfv(100, 0.1, 7)
print(round(fv, 2))
```

```
194.87
```

[13]

```
# Python if else
score = 80
if score >=60 :
    print("Pass")
else:
    print("Fail")
```

```
Pass
```

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

+ Code + Text RAM Disk Editing

Table of contents

AI in Finance

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

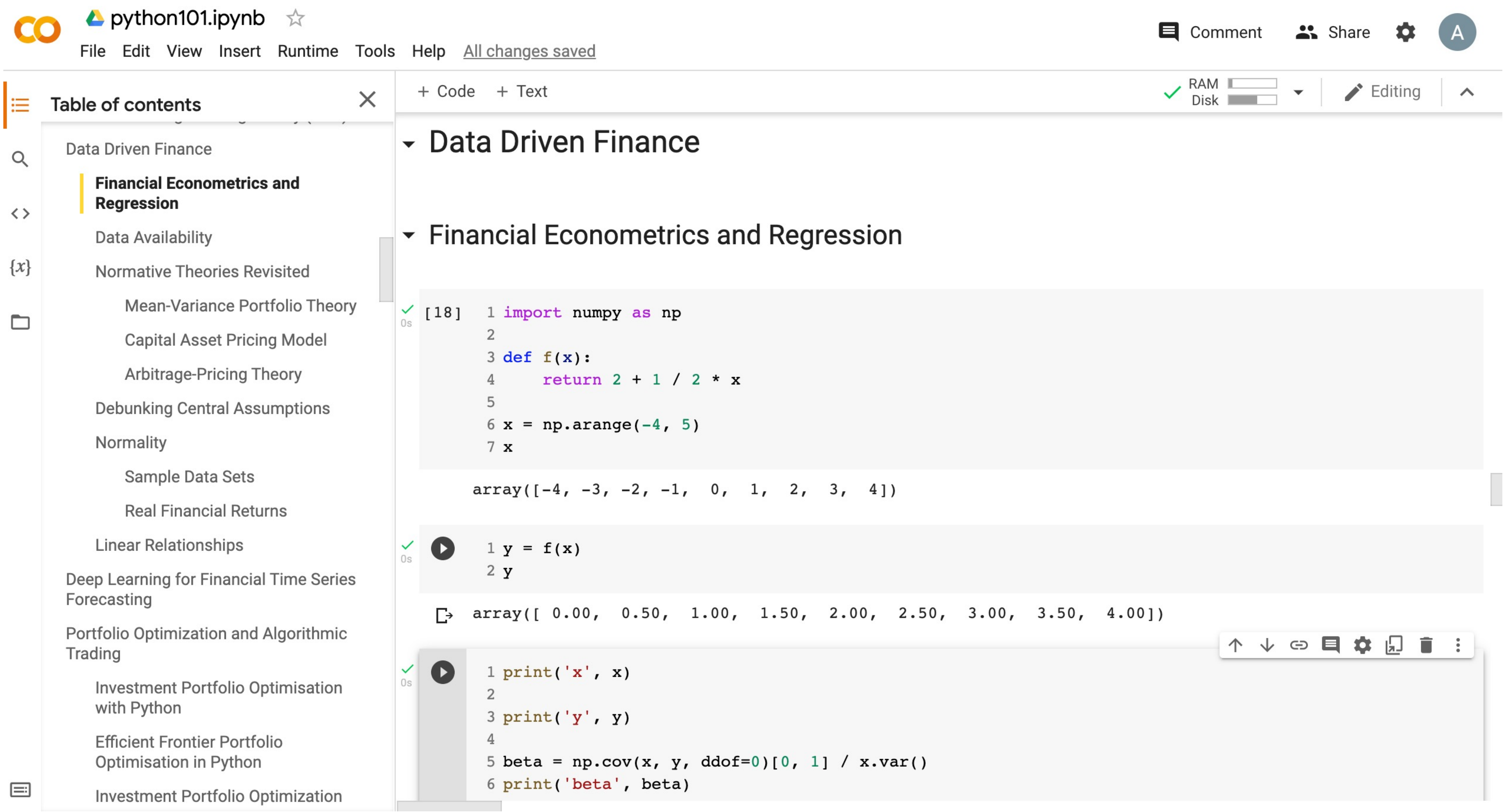
Normative Finance and Financial Theories

Uncertainty and Risk

```python
import numpy as np

#The prices of the stock and bond today.
S0 = 10
B0 = 10
print('S0', S0)
print('S0', S0)

#The uncertain payoff of the stock and bond tomorrow.
S1 = np.array((20, 5))
B1 = np.array((11, 11))
print('S1', S1)
print('B1', B1)

#The market price vector
M0 = np.array((S0, B0))
```

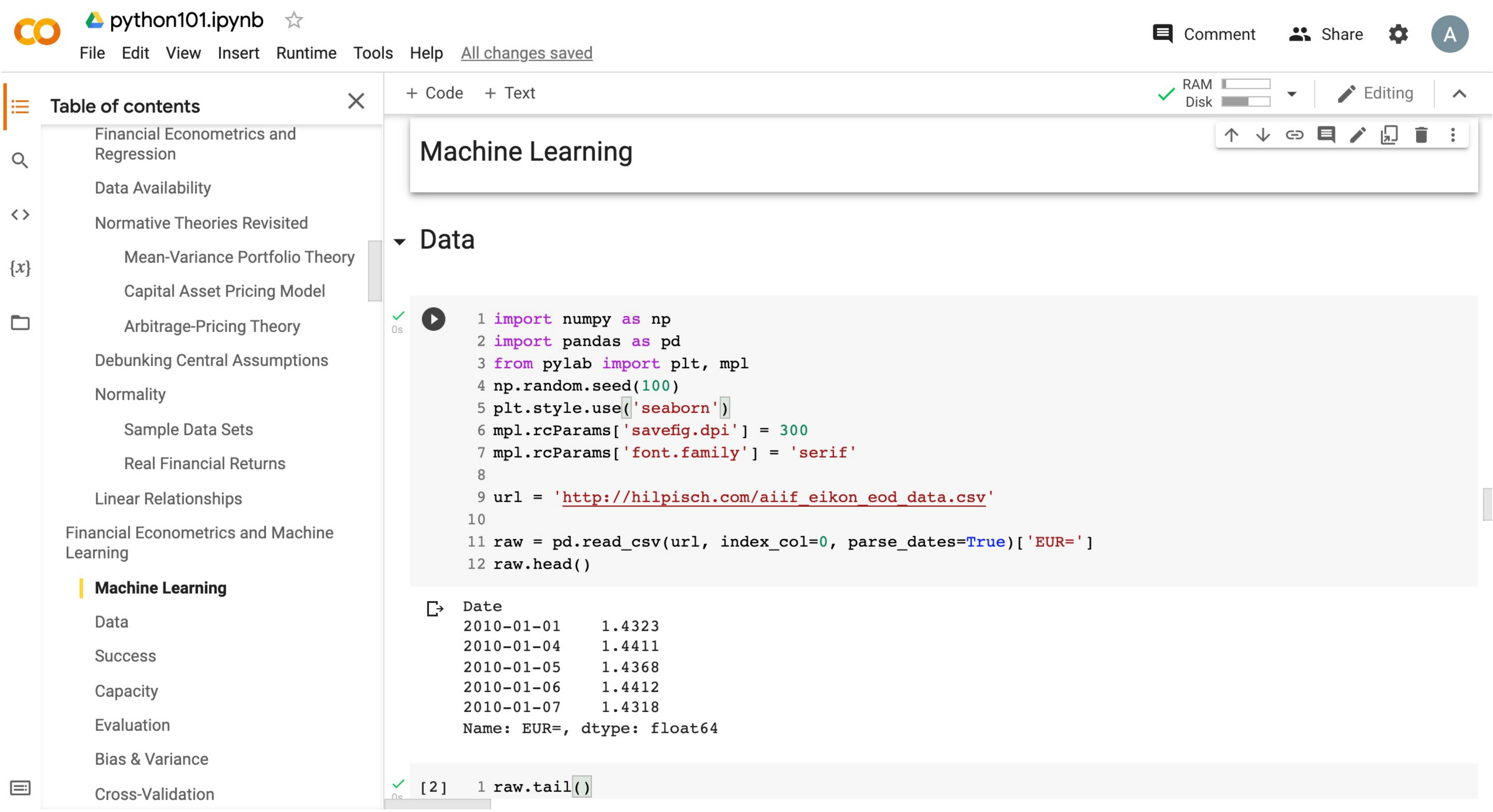
https://tinyurl.com/aintpupython101

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 Saving...

Comment Share

Table of contents

+ Code + Text

RAM Disk Editing

Success

```python
def MSE(l, p):
    return np.mean((l - p) ** 2)
```

[9]
```python
reg = np.polyfit(f, l, deg=5)
reg
```

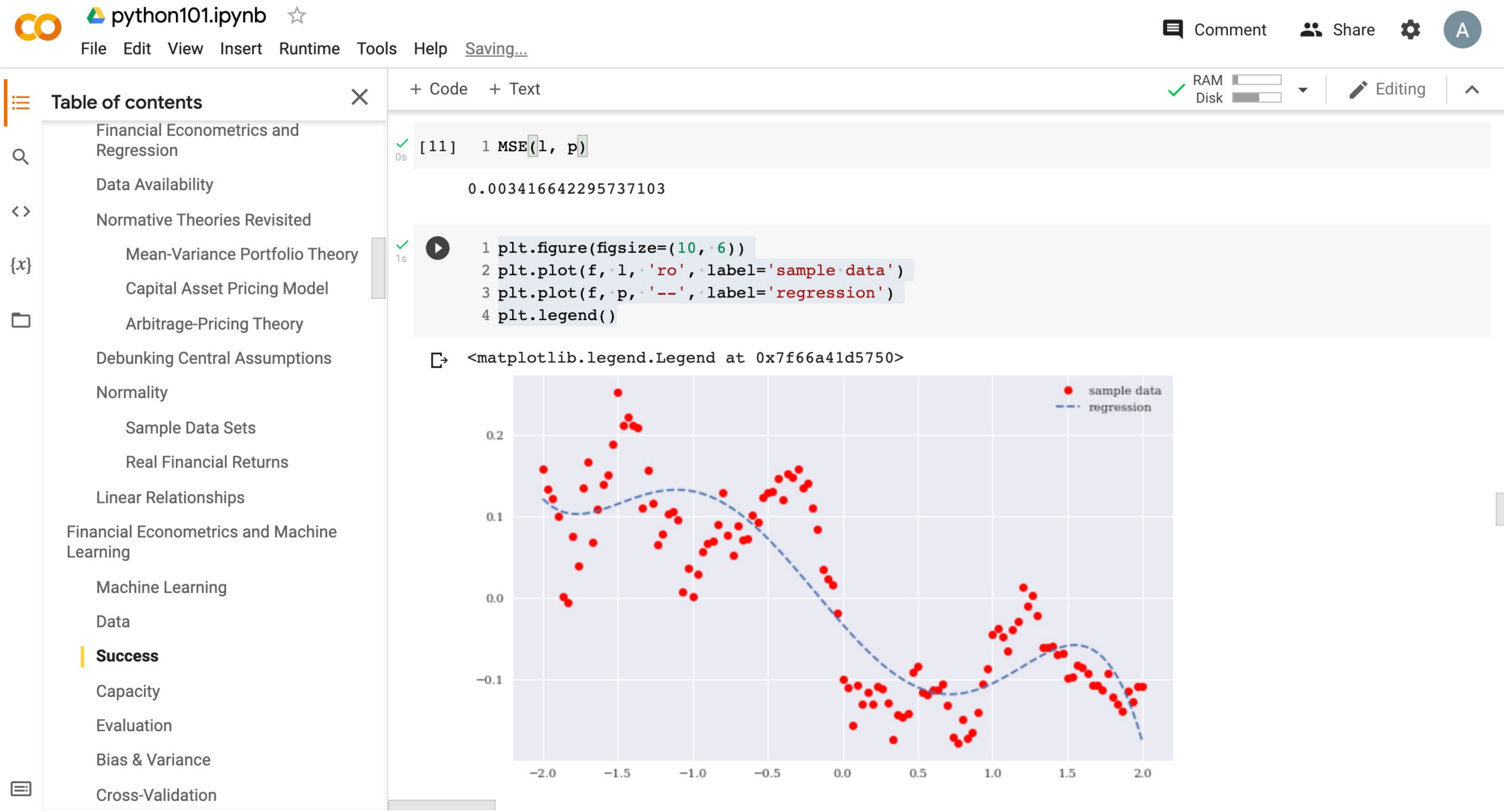
```
array([-0.01910626, -0.0147182 ,  0.10990388,  0.06007211, -0.20833598,
       -0.03275423])
```

[10]
```python
p = np.polyval(reg, f)
p
```

```
array([ 0.12088427,  0.11526131,  0.11080193,  0.10739461,  0.10493286,
        0.10331514,  0.10244475,  0.10222973,  0.10258281,  0.10342126,
        0.10466683,  0.10624564,  0.1080881 ,  0.1101288 ,  0.11230643,
        0.11456366,  0.11684709,  0.11910711,  0.12129784,  0.123377  ,
        0.12530587,  0.12704913,  0.12857481,  0.1298542 ,  0.1308617 ,
        0.1315748 ,  0.13197395,  0.13204243,  0.13176634,  0.13113443,
        0.13013803,  0.12877097,  0.12702948,  0.12491207,  0.12241947,
        0.11955452,  0.11632208,  0.11272891,  0.10878364,  0.1044966 ,
        0.09987977,  0.09494668,  0.0897123 ,  0.08419296,  0.07840627,
        0.07237098,  0.06610693,  0.05963494,  0.05297671,  0.04615473,
        0.03919218,  0.03211286,  0.02494106,  0.01770149,  0.01041918,
        0.00311939, -0.00417251, -0.0114311 , -0.01863101, -0.02574704,
       -0.03275423, -0.03962796, -0.04634406, -0.05287887, -0.05920936,
       -0.06531322, -0.07116897, -0.07675602, -0.08205478, -0.08704677,
```

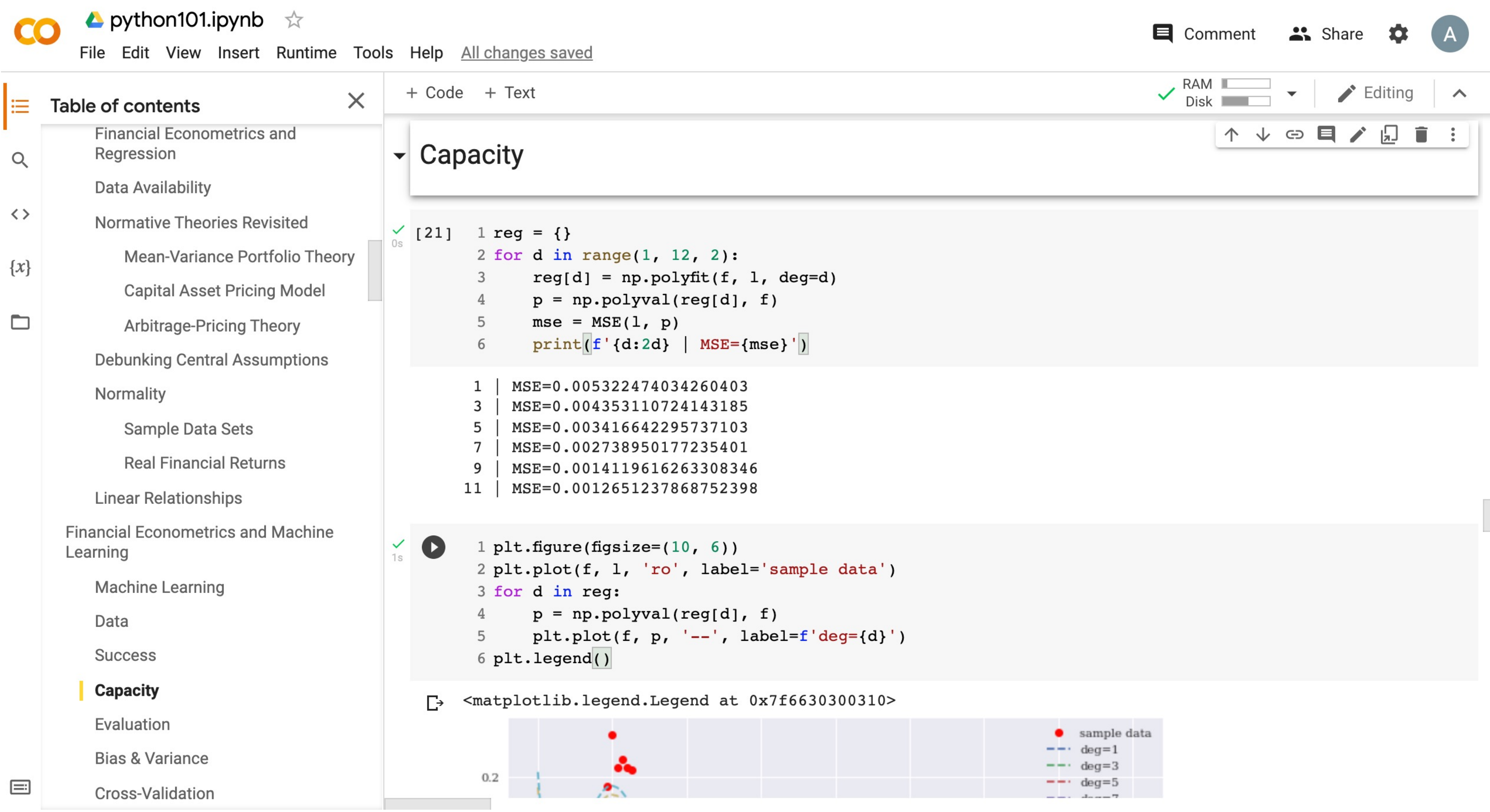
https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



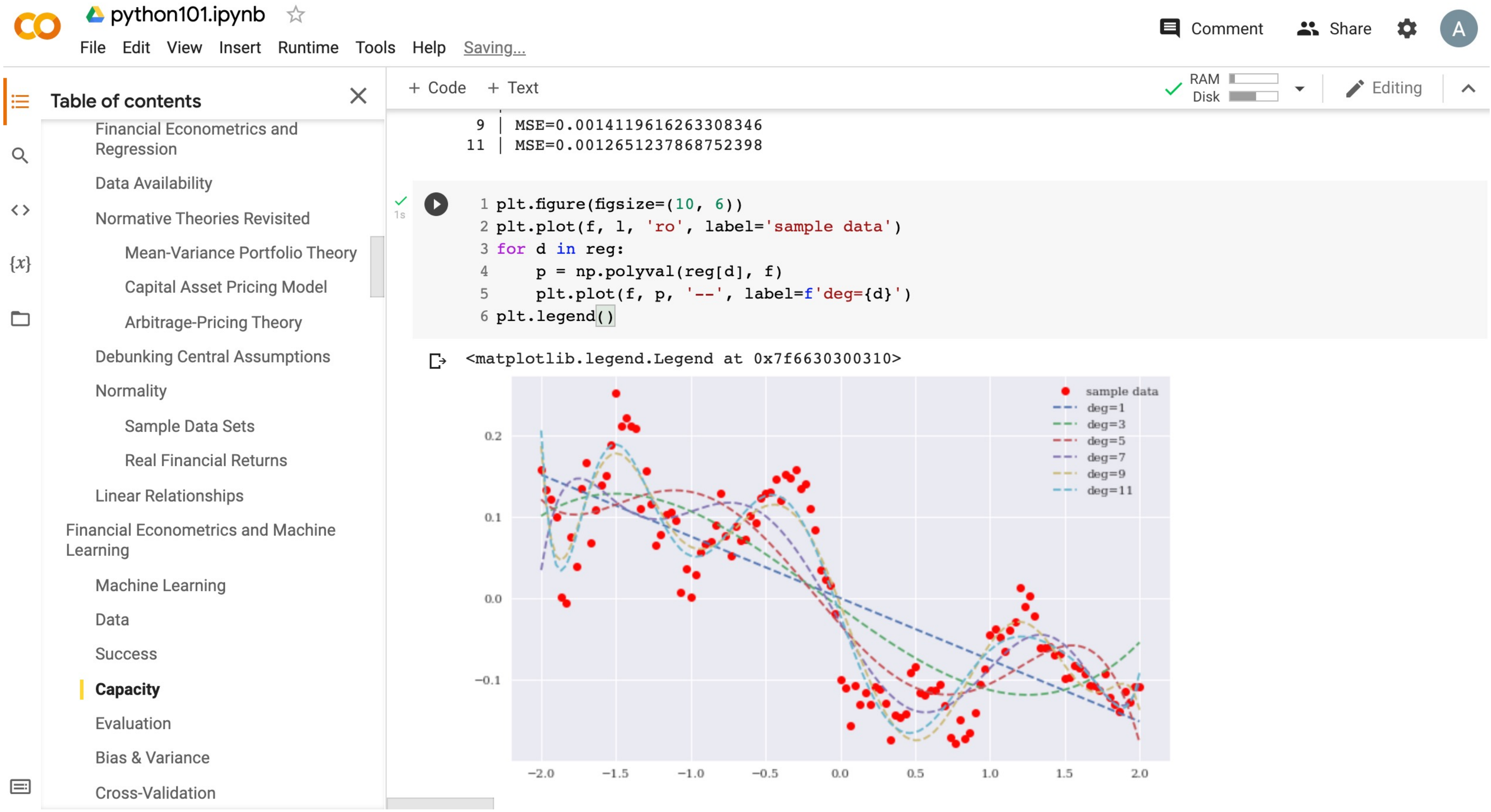
https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

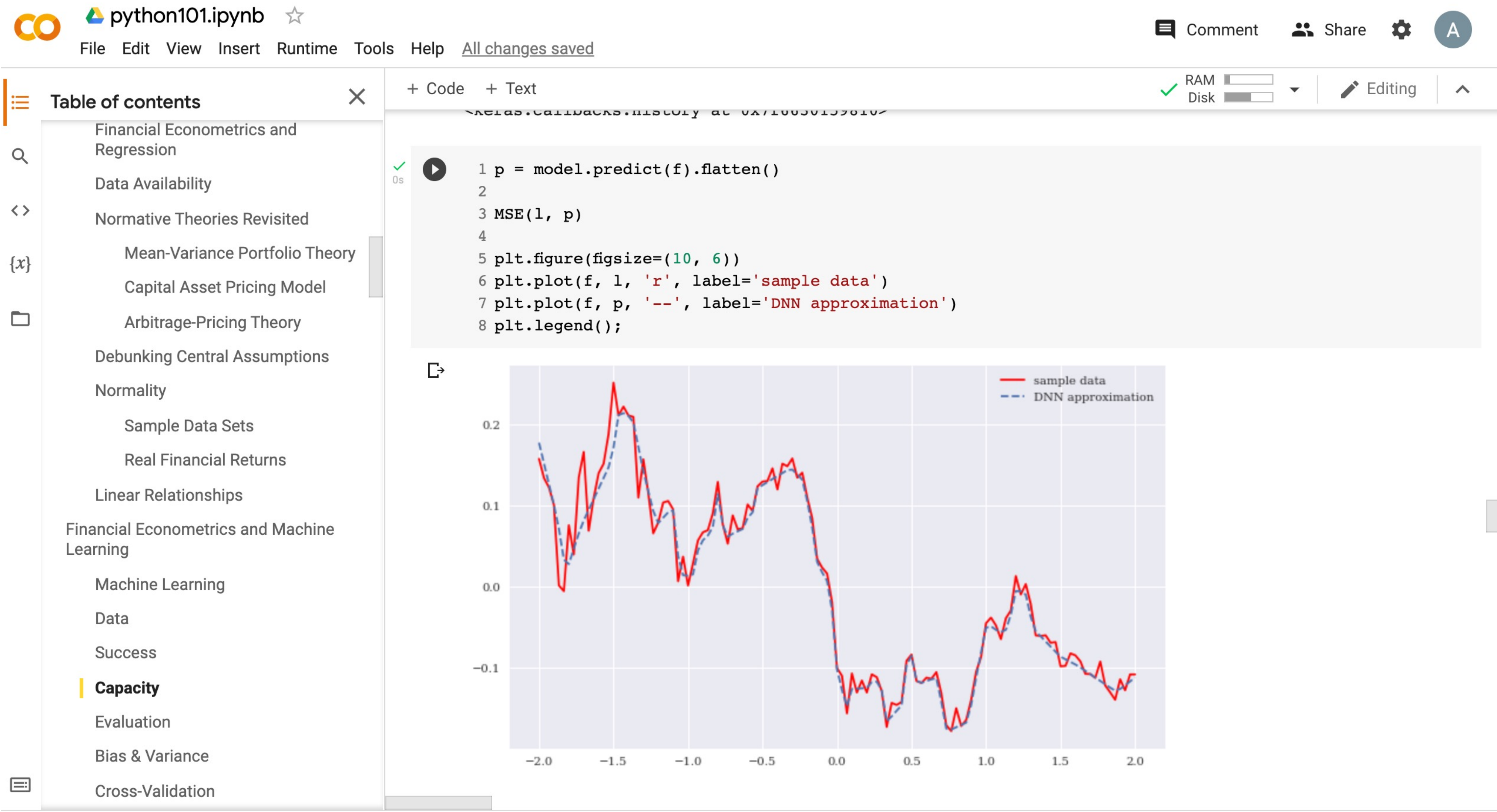
Python in Google Colab (Python101)

```
def create_dnn_model(hl=1, hu=256):
    ''' Function to create Keras DNN model.
    Parameters
    ==========
    hl: int
    number of hidden layers
    hu: int
    number of hidden units (per layer)
    '''
    model = Sequential()
    for _ in range(hl):
        model.add(Dense(hu, activation='relu', input_dim=1))
    model.add(Dense(1, activation='linear'))
    model.compile(loss='mse', optimizer='rmsprop')
    return model

model = create_dnn_model(3)

model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_2 (Dense)             (None, 256)               512

 dense_3 (Dense)             (None, 256)               65792

 dense_4 (Dense)             (None, 256)               65792

 dense_5 (Dense)             (None, 1)                 257

=================================================================
Total params: 132,353
Trainable params: 132,353
Non-trainable params: 0
_________________________________________________________________
```

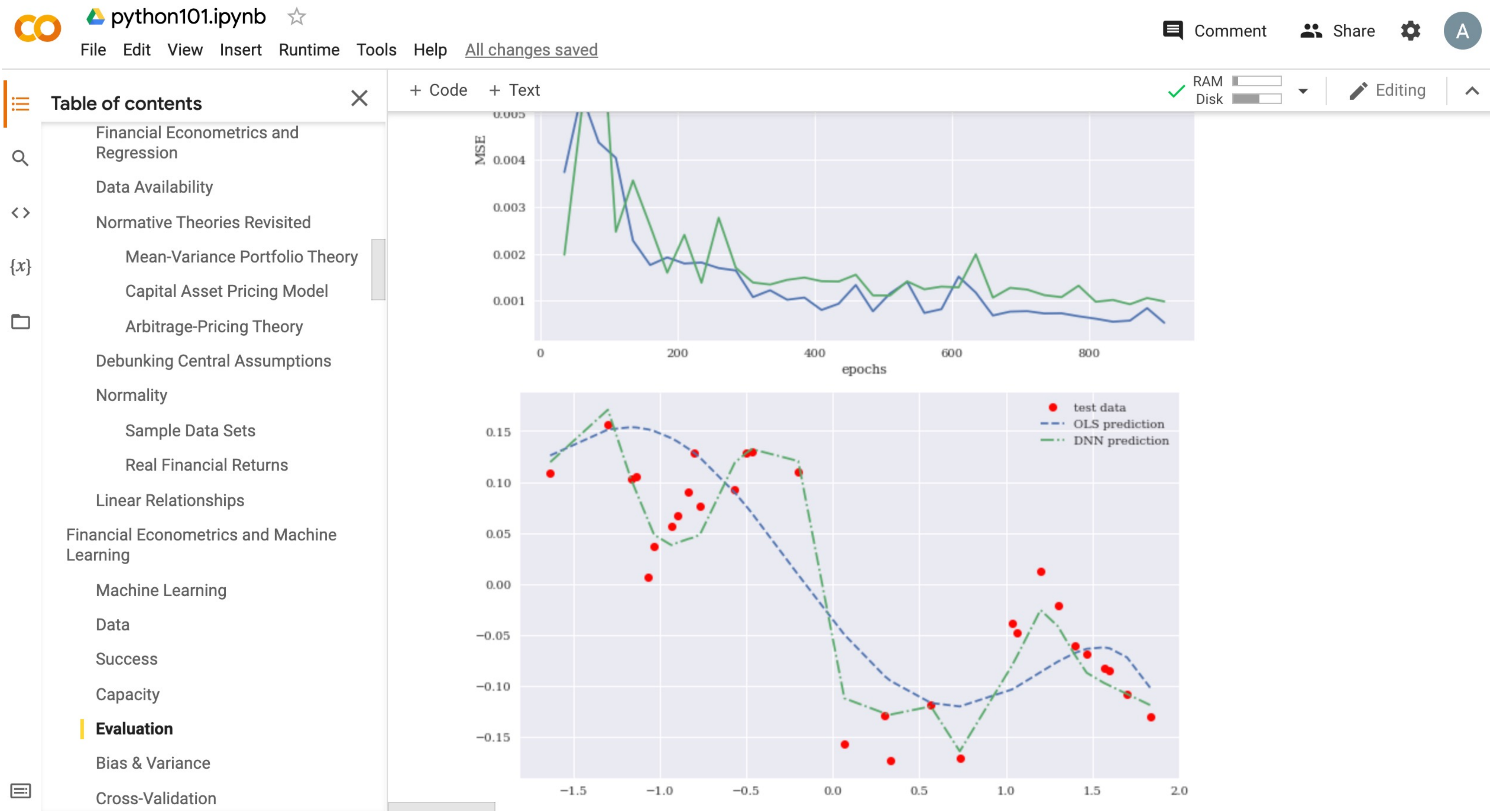
https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

Summary

- Financial Econometrics
  - Financial Theories
  - OLS Regression
- Machine Learning
  - Learning
  - Evaluation
  - Bias and variance
  - Cross-validation

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 .
- Chris Brooks (2019), Introductory Econometrics for Finance, 4th Edition, Cambridge University Press
- Oliver Linton (2019), Financial Econometrics: Models and Methods, Cambridge University Press
- Tom Mitchell (1997), Machine Learning, McGraw-Hill.
- Min-Yuh Day (2021), Python 101, https://tinyurl.com/aintpupython101