

(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)



<u>戴敏育</u>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





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)





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)

[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]

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)





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

# Algorithmic Trading

### **Algorithmic Trading**



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



#### **Sharpe Ratio**

#### **Sharpe Ratio** Portofolio Return – Risk Free Return

#### Portofolio Risk

#### **Sharpe Ratio**

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

Where  $r_P = \text{portfolio return}$   $r_F = \text{risk free rate}$  $\sigma_P = \text{portfolio risk}$  (variability, standard deviation of return)

#### **Sortino Ratio**

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

Where

 $r_P$  = portfolio return

 $r_T$  = Minimum Target Return

 $\sigma_D$  = Downside Risk

**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



Portfolio Optimization Efficient Frontier



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

### 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



## Gross performance of passive benchmark investment and SMA strategy



#### Gross performance of the SMA strategy before and after transaction costs



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



#### 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)



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



## Gross performance on training and validation data set



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



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



### 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



#### **BTC-USD**



#### **BTC-USD Returns**

btcusd returns



#### **BTC-USD Returns Box**

btcusd returns box



#### **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,



Upisch / aiif Public	https://github.co	om/yhilpisch/aiif	Notifications 🛱 Star 93	8 % Fork 77		
<> Code <ul> <li>Issues I</li> </ul>	Pull requests ( Actions ( Projects	🖽 Wiki 🔃 Security 🗠 Insights				
<mark>਼ਿੰ main →</mark> <b>ਿੈ 1</b> branch	<b>⊙ 0</b> tags	Go to file Code -	About			
<b>yves</b> Code updates for TF	2.3.	e334251 on Dec 8, 2020 🕚 4 commits	Jupyter Notebooks and code for the book Artificial Intelligence in Finance (O'Reilly) by			
code	Code updates for TF 2.3.	11 months ago	2 home the in/hooks/aii	4		
🗅 .gitignore	Code updates for TF 2.3.	11 months ago				
LICENSE.txt	Code updates.	11 months ago		O'REILLY'		
🖺 README.md	Code updates.	11 months ago		Artificial		
i≘ README.md			Releases	Intelligen	ce	
Artificial Inte	lligence in Finance		No releases published	A Python-Based Guide	;	
			Packages		1	
About this Repos	sitory		No packages published		201	
This repository provides <b>Finance</b> book published	Python code and Jupyter Notebooks accomp by O'Reilly.	panying the Artificial Intelligence in	Languages	HAR COM	Yves Hilpisc	
<b>O'REILLY</b> °			<ul> <li>Jupyter Notebook 97.4%</li> </ul>	• Python 2.6%		

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



Source: <a href="https://github.com/yhilpisch/aiif/tree/main/code">https://github.com/yhilpisch/aiif/tree/main/code</a>

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

co python101.ipynb - Colaboratory × +	
← → C https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT?authuser=2#scrollTo=wsh36fLxDKC3	☆ ◙ 0 :
CO A python101.ipynb 📩 File Edit View Insert Runtime Tools Help	r 🚓 share 🗛
CODE TEXT A CELL CELL	editing
<pre></pre>	:
[→ 194.87	
<pre>[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))</pre>	
[→ 194.87	
<pre>[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))</pre>	
[→ 194.87	
<pre>[13] 1 # Python if else 2 score = 80 3 if score &gt;=60 : 4      print("Pass") 6      else: 6           print("Fail").</pre>	
[→ Pass	

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



C	<b>O A python101.ipynb</b> $\overleftrightarrow$ File Edit View Insert Runtime To	ols	Help	All changes saved		omment		Share	\$	A
≣	Table of contents X		+ Coo	de + Text	V RA	M sk	•	🎤 Ed	liting	^
Q	Data Driven Finance		- Da	ta Driven Finance						
<>	Financial Econometrics and Regression									
	Data Availability		▼ Fir	ancial Econometrics and Regression						
$\{x\}$	Normative Theories Revisited									
	Mean-Variance Portfolio Theory Capital Asset Pricing Model Arbitrage-Pricing Theory Debunking Central Assumptions Normality Sample Data Sets		✔ [18]	<pre>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</pre>						
	Real Financial Returns			array([-4, -3, -2, -1, 0, 1, 2, 3, 4])						
	Linear Relationships		S O	1 y = f(x)						
	Deep Learning for Financial Time Series			2 y						
	Portfolio Optimization and Algorithmic Trading	;	C→	array([ 0.00, 0.50, 1.00, 1.50, 2.00, 2.50, 3.00, 3.50, 4.00	])	↑ ↓	<b>G</b>	4	<u>。</u>	ī:
	Investment Portfolio Optimisation with Python		os D	1 print('x', x) 2 3 print('y', y)						
	Efficient Frontier Portfolio Optimisation in Python			4 5 beta = np.cov(x, y, ddof=0)[0, 1] / x.var()						
=	Investment Portfolio Optimization			6 print('beta', beta)						

CO Copython File Edit	1 <b>101.ipynb</b> 🛧 /iew Insert Runtime Tools	s Help <u>All changes saved</u>	🗏 Comment 👫 Share 🏟 🗛
⊟ Table of conte	nts ×	+ Code + Text	✓ RAM Disk ✓ Editing ∧
Financial RQRegressionData Avail	conometrics and n ability	Machine Learning	
<> Normative {x} Mean- Capita	Theories Revisited Variance Portfolio Theory I Asset Pricing Model	- Data	
C Arbitra Debunking Normality Samp Real F Linear Rel Financial Ecor Learning	Arbitrage-Pricing Theory Debunking Central Assumptions Normality Sample Data Sets Real Financial Returns Linear Relationships Financial Econometrics and Machine Learning	<pre>     1 import numpy as np     2 import pandas as pd     3 from pylab import plt, mpl     4 np.random.seed(100)     5 plt.style.use('seaborn')     6 mpl.rcParams['savefig.dpi'] = 300     7 mpl.rcParams['font.family'] = 'serif'     8     9 url = 'http://hilpisch.com/aiif_eikon_eod_data.csv'     10     11 raw = pd.read_csv(url, index_col=0, parse_dates=True)['EUR=']     12 raw.head() </pre>	
Image: Machine L         Data         Success         Capacity         Evaluation         Bias & Var         E:         Cross-Vali	earning ance dation	<pre> C→ Date 2010-01-01 1.4323 2010-01-04 1.4411 2010-01-05 1.4368 2010-01-06 1.4412 2010-01-07 1.4318 Name: EUR=, dtype: float64  </pre> [2] 1 raw.tail()	

C	O Contraction Cont	Tools Help	All changes saved	E Comment	Share	<b>\$</b>
	Table of contents	× + c	ode + Text	✓ RAM Disk	🗸 🖌 🎤 Ed	iting ^
Q <> {x}	Mean-Variance Portfolio Theory Capital Asset Pricing Model Arbitrage-Pricing Theory Debunking Central Assumptions Normality Sample Data Sets Real Financial Returns Linear Relationships	· E	<pre>ficient 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</pre>	↑ ↓	c) 🗐 🇘	
	Financial Econometrics and Machine Learning Machine Learning		<pre>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')</pre>			
	Data Data Success Capacity Evaluation Bias & Variance Cross-Validation AI-First Finance Efficient Markets Market Prediction Based on Returns Data Market Prediction With More Features	3	<pre><matplotlib.axessubplots.axessubplot 0x7f29f972f210="" at=""></matplotlib.axessubplots.axessubplot></pre>			
≕	Market Prediction Intraday		2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2027	)		

C	O Apython101.ipynb 🕁 File Edit View Insert Runtim	ne -	Fools Help <u>All changes saved</u>	Ę	Comment		Share	\$	A
≣	Table of contents $\qquad  imes$		+ Code + Text		Connect	•		Editing	^
Q <>> {x}	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	•	Deep Learning (DL) in Finance <ul> <li>Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, Of</li> <li>Github: <a href="https://github.com/yhilpisch/aiif/">https://github.com/yhilpisch/aiif/</a></li> </ul> Dense Neural Networks (DNN) I import os <ul> <li>import numpy as np</li> <li>import pandas as pd</li> <li>from pylab import plt, mpl</li> <li>plt.style.use('seaborn')</li> <li>mpl.rcParams['font.family'] = 'serif'</li> <li>pd.set_option('precision', 4)</li> <li>np.set_printoptions(suppress=True, precision=4)</li> <li>os.environ['PYTHONHASHSEED'] = '0'</li> </ul>	'Reilly	^ ↓ Media.				
	Financial Features Deep RNNs Convolutional Neural Networks (CNN)		<pre>[ ] 1 url = '<u>http://hilpisch.com/aiif_eikon_id_eur_usd.csv</u>' 2 symbol = 'EUR_USD' 3 raw = pd.read_csv(url, index_col=0, parse_dates=True) 4 raw.head()</pre>						
	Reinforcement Learning (RL) in Finance		HIGH LOW OPEN CLOSE						

C	🔾 🝐 python101.ipynb 🕁 File Edit View Insert Runtime	Tools Help All changes saved	Comment	👪 Sh	are 🕻	
≣	Table of contents $\times$	+ Code + Text	Connect	•	Editi	ng 🔨 🔨
Q	Deep RNNs	- Reinforcement Learning (RL) in Finance	$\wedge \downarrow$	⇔ 🗖	/	<b>i</b> :
<>	Convolutional Neural Networks (CNN)	<ul> <li>Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'</li> <li>Github: https://github.com/yhilpisch/aiif/</li> </ul>	Reilly Media.			
$\{x\}$	Reinforcement Learning (RL) in Finance	Sittab. <u>Intps.//gittab.com/yinipisci/ain/</u>				
	Reinforcement Learning (RL) CartPole Environment	<ul> <li>Reinforcement Learning (RL)</li> </ul>				
	Dimensionality Reduction Action Rule Total Reward per Episode Simple Learning Testing the Results DNN Learning Q Learning Finance Environment	<pre> 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' </pre>				
	Improved Finance Environment					
	Agent					

42



C	D 🍐 python101.ipynb 🕁 File Edit View Insert Runtime	Tools Help All changes saved	Comment	🖺 Share	۵	A
⊫	Table of contents X	+ Code + Text	V RAM Disk	- /	Editing	^
Q	Algorithmic Trading Vectorized Backtesting	<ul> <li>Vectorized Backtesting</li> </ul>	$\uparrow \downarrow$	c 🔲 🖊	آ <u>ل</u>	
<>	Backtesting an SMA- Based Strategy					
{ <i>x</i> }	Backtesting a Daily DNN- Based Strategy	1 import os 2 import math				
	Backtesting an Intraday DNN-Based Strategy	3 import numpy as np 4 import pandas as pd 5 from pylab import plt, mpl				
	Risk Management	6 plt.style.use('seaborn') 7 mpl.rcParams['savefig.dpi'] = 300				
	Vectorized Backtesting	<pre>9 pd.set_option('mode.chained_assignment', None) 10 pd.set_option('display.float_format', '{:.4f}'.format)</pre>				
	Assessing Risk	<pre>11 np.set_printoptions(suppress=True, precision=4) 12 os.environ['PYTHONHASHSEED'] = '0'</pre>				
	Backtesting Risk Measures Stop Loss	<ul> <li>Backtesting an SMA-Based Strategy</li> </ul>				
	Trailing Stop Loss Take Profit Combinations	<pre>[ ] 1 url = '<u>http://hilpisch.com/aiif_eikon_eod_data.csv</u>' 2 symbol = 'EUR=' 3 data = pd.DataFrame(pd.read_csv(url, index_col=0, 4</pre>				
≕	Backtesting Cryptocurrency Bitcoin	5 data.info()				





C	<b>O A</b> python101.ipynb 🕁 File Edit View Insert Runtime	Tools Help All changes saved	📮 Comment 🛛 👫 Share 🌼 🧖
≣	Table of contents $\qquad  imes$	+ Code + Text	✓ RAM ► Editing
Q <>> {x}	Algorithmic Trading Vectorized Backtesting Backtesting an SMA- Based Strategy Backtesting a Daily DNN- Based Strategy Backtesting an Intraday DNN-Based Strategy <b>Risk Management</b> Trading Bot Vectorized Backtesting Event-Based Backtesting	<pre> • Risk Management [ ] 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('mode.chained_assignment', None) 9 pd.set_option('display.float_format', '{:.4f}'.format) 10 np.set_printoptions(suppress=True, precision=4) 11 os.environ['PYTHONHASHSEED'] = '0'</pre>	
	Assessing Risk Backtesting Risk Measures		
II	Stop Loss Trailing Stop Loss Take Profit Combinations Backtesting Cryptocurrency Bitcoin	<pre>[ ] 1 # import finance 2 # finance.py 3 # Finance Environment 4 # 5 # (c) Dr. Yves J. Hilpisch 6 # Artificial Intelligence in Finance 7 #</pre>	

https://tinyurl.com/aintpupython101

C	O A python101.ipynb 🕁 File Edit View Insert Runtime	Tools Help All changes saved	Ę	Comment	*	Share	۵	A
≣	Table of contents $\qquad  imes$	+ Code + Text	~	Disk	•		Editing	^
Q	Algorithmic Trading Vectorized Backtesting	- Event-Based Backtesting		$\uparrow \downarrow$	Θ	□ /	i ري	
<>	Backtesting an SMA- Based Strategy	1 #import backtesting as bt 2						
{ <i>x</i> }	Backtesting a Daily DNN- Based Strategy	3 # backtesting.py 4 # Event-Based Backtesting						
	Backtesting an Intraday DNN-Based Strategy	5 #Base Class (1) 6 # 7 # (c) Dr. Yves J. Hilpisch						
	Risk Management	8 # Artificial Intelligence in Finance						
	Trading Bot	9 <b>#</b> 10						
	Vectorized Backtesting	11 class BacktestingBase:						I
	Event-Based Backtesting	<pre>12 definit(self, env, model, amount, ptc, ftc, verbose=False): 13 self.env = env</pre>						
	Assessing Risk	14 self.model = model						
	Backtesting Risk Measures	<pre>15 self.initial_amount = amount 16 self.current_balance = amount 17 self.ptc = ptc</pre>						
	Stop Loss	18 self.ftc = ftc						
	Trailing Stop Loss	19self.verbose = verbose20self.units = 0						
	Take Profit	21 self.trades = 0						
	Combinations	<pre>22 23 def get_date_price(self, bar):</pre>						
≡	Backtesting Cryptocurrency Bitcoin	24 ''' Returns date and price for a given bar.						

https://tinyurl.com/aintpupython101

C	<b>O A</b> python101.ipynb 🕁 File Edit View Insert Runtime	Tools Help <u>All changes saved</u>	🗐 Comment 🛛 👫 Share 🏟 🔥
≔	Table of contents $\times$	+ Code + Text	✓ RAM → Fediting ∧
Q	Algorithmic Trading	Combinations	
<>	Vectorized Backtesting Backtesting an SMA- Based Strategy	<pre> 1 tb.backtest_strategy(sl=0.015, tsl=None, 2</pre>	
{ <i>x</i> }	Backtesting a Daily DNN- Based Strategy	C→ ====================================	
	Backtesting an Intraday DNN-Based Strategy	2018-01-17   current balance = 10000.00	
	Risk Management	*** STOP LOSS (SHORT   -0.0203) ***	
	Trading Bot	*** STOP LOSS (SHORT   -0.0152) ***	
	Vectorized Backtesting	*** TAKE PROFIT (SHORT   0.0189) ***	
	Event-Based Backtesting	*** TAKE PROFIT (SHORT   0.0219) ***	
	Assessing Risk	*** TAKE PROFIT (SHORT   0.0192) ***	
	Backtesting Risk Measures	*** STOP LOSS (LONG   -0.0154) ***	
	Stop Loss	*** TAKE PROFIT (SHORT   0.0214) ***	
	Trailing Stop Loss Take Profit	*** STOP LOSS (SHORT   -0.0158) ***	
	Combinations	*** TAKE PROFIT (SHORT   0.0223) ***	
=	Backtesting Cryptocurrency Bitcoin	*** STOP LOSS (SHORT   -0.0162) ***	

C	Python101.ipynb File Edit View Insert Runtime	Tools Help All changes saved	🗐 Comment 🛛 😫 Share 🏟 🛕
≣	Table of contents $\qquad  imes$	+ Code + Text	✓ RAM ► ✓ Editing ∧
Q	Algorithmic Trading Vectorized Backtesting	<ul> <li>Backtesting Cryptocurrency Bitcoin</li> </ul>	
<>	Backtesting an SMA- Based Strategy	<ul> <li>Financial Functions (ffn): <u>https://pmorissette.github.io/ffn/</u></li> <li>backtesting.py: <u>https://kernc.github.io/backtesting.py/</u></li> </ul>	
{ <i>x</i> }	Based Strategy	1 Inin install fr	
	Backtesting an Intraday DNN-Based Strategy	2 import ffn 3 import plotly.express as px	
	Risk Management	4 <b>%pylab</b> inline	
	Trading Bot	5 #BTC-USD Bitcoin USD 6 df = ffn.get('btc-usd', start='2016-01-01', end='2021-12-31')	
	Vectorized Backtesting	<pre>7 print('df') 8 print(df.head())</pre>	
	Assessing Risk	<pre>9 print(df.tail()) 10 print(df.describe())</pre>	
	Backtesting Risk Measures	<pre>11 df.plot(figsize=(14,10)) 12 13 returns = df.to_returns().dropna()</pre>	
	Stop Loss	14 print('returns') 15 print(returns_bead())	
	Trailing Stop Loss	16 print(returns.tail())	
	Take Profit	<pre>17 print(returns.describe()) 18 #ax = df.plot(figsize=(12,9))</pre>	
	Combinations	19	
=	Backtesting Cryptocurrency Bitcoin	<pre>20 perf = df.calc_stats() 21 perf.plot(figsize=(14, 10))</pre>	





https://tinyurl.com/aintpupython101



https://tinyurl.com/aintpupython101

### Summary

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

### References

- Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media, <u>https://github.com/yhilpisch/aiif</u>.
- 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, <a href="https://tinyurl.com/aintpupython101">https://tinyurl.com/aintpupython101</a>