Stock_Correlations

September 29, 2021

1 Stock Covariance & Correlations

Covariance measures the directional relationship between the returns on two assets. A positive covariance means that asset returns move together while a negative covariance means they move inversely. Covariance is calculated by analyzing at-return surprises (standard deviations from the expected return) or by multiplying the correlation between the two variables by the standard deviation of each variable. (https://www.investopedia.com/terms/c/covariance.asp)

Stock correlation explained the relationship that exists between two stocks and their respective price movements which has a value that must fall between -1.0 and +1.0.

A perfect positive correlation means that the correlation coefficient is exactly 1. This implies that as one security moves, either up or down, the other security moves in lockstep, in the same direction. A perfect negative correlation means that two assets move in opposite directions, while a zero correlation implies no relationship at all. (https://www.investopedia.com/terms/c/correlation.asp)

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math

import warnings
warnings.filterwarnings("ignore")

# fix_yahoo_finance is used to fetch data
import fix_yahoo_finance as yf
yf.pdr_override()
```

1.1 Two Securities Correlation

[2]:
```
# input
symbols = ['AMD','INTC']
start = '2012-01-01'
end = '2019-01-01'

# Read data
```

```
dataset = yf.download(symbols,start,end)['Adj Close']

# View Columns
dataset.head()
```

```
[*********************100%***********************]  2 of 2 downloaded
```

[2]:

```
             AMD       INTC
Date
2012-01-03  5.48  19.332613
2012-01-04  5.46  19.781666
2012-01-05  5.46  20.010126
2012-01-06  5.43  19.891949
2012-01-09  5.59  20.065273
```

[3]:

```
stocks_returns = np.log(dataset / dataset.shift(1))
```

[4]:

```
AMD = stocks_returns['AMD'].var()
AMD
```

[4]: 0.0013675618444054449

[5]:

```
INTC = stocks_returns['INTC'].var()
INTC
```

[5]: 0.00022174925461484408

[6]:

```
AMD = stocks_returns['AMD'].var() * 250
AMD
```

[6]: 0.34189046110136123

[7]:

```
INTC = stocks_returns['INTC'].var() * 250
INTC
```

[7]: 0.05543731365371102

[8]:

```
cov_matrix = stocks_returns.cov()
cov_matrix
```

[8]:

	AMD	INTC
AMD	0.001368	0.000172
INTC	0.000172	0.000222

[9]:

```
print('Covariance Matrix')
cov_matrix = stocks_returns.cov()*250
cov_matrix
```

Covariance Matrix

[9]:

	AMD	INTC
AMD	0.341890	0.043085
INTC	0.043085	0.055437

[10]:
```
print('Correlation Matrix')
corr_matrix = stocks_returns.corr()*250
corr_matrix
```

Correlation Matrix

[10]:

	AMD	INTC
AMD	250.000000	78.239458
INTC	78.239458	250.000000

1.2 Four Securities Correlation

[11]:
```
# input
symbols = ['AAPL','MSFT','AMD','NVDA']
start = '2012-01-01'
end = '2019-01-01'

# Read data
dataset = yf.download(symbols,start,end)['Adj Close']

# View Columns
dataset.head()
```

```
[*********************100%***********************]  4 of 4 downloaded
```

[11]:

	AAPL	AMD	MSFT	NVDA
Date				
2012-01-03	51.269413	5.48	22.156071	12.939396
2012-01-04	51.544937	5.46	22.677486	13.086854
2012-01-05	52.117188	5.46	22.909233	13.556875
2012-01-06	52.662014	5.43	23.265116	13.400198
2012-01-09	52.578468	5.59	22.958887	13.400198

[12]:
```
stocks_returns = np.log(dataset / dataset.shift(1))
```

[13]:
```
AAPL = stocks_returns['AAPL'].var()
AAPL
```

[13]: 0.0002580753578116192

[14]:
```
MSFT = stocks_returns['MSFT'].var()
MSFT
```

[14]: 0.0002109675276041315

[15]:
```
AMD = stocks_returns['AMD'].var()
AMD
```

[15]: 0.0013675618444054449

[16]:
```
NVDA = stocks_returns['NVDA'].var()
NVDA
```

[16]: 0.0005325810343486495

[17]:
```
AAPL = stocks_returns['AAPL'].var() * 250
AAPL
```

[17]: 0.0645188394529048

[18]:
```
MSFT = stocks_returns['MSFT'].var() * 250
MSFT
```

[18]: 0.05274188190103288

[19]:
```
AMD = stocks_returns['AMD'].var() * 250
AMD
```

[19]: 0.34189046110136123

[20]:
```
NVDA = stocks_returns['NVDA'].var() * 250
NVDA
```

[20]: 0.1331452585871624

[21]:
```
cov_matrix = stocks_returns.cov()
cov_matrix
```

[21]:

	AAPL	AMD	MSFT	NVDA
AAPL	0.000258	0.000133	0.000094	0.000121
AMD	0.000133	0.001368	0.000136	0.000348
MSFT	0.000094	0.000136	0.000211	0.000143
NVDA	0.000121	0.000348	0.000143	0.000533

[22]:
```
print('Covariance Matrix')
cov_matrix = stocks_returns.cov()*250
cov_matrix
```

```
Covariance Matrix
```

[22]:

	AAPL	AMD	MSFT	NVDA
AAPL	0.064519	0.033303	0.023565	0.030209
AMD	0.033303	0.341890	0.034009	0.086893
MSFT	0.023565	0.034009	0.052742	0.035644
NVDA	0.030209	0.086893	0.035644	0.133145

[23]:

```
print('Correlation Matrix')
corr_matrix = stocks_returns.corr()*250
corr_matrix
```

```
Correlation Matrix
```

[23]:

	AAPL	AMD	MSFT	NVDA
AAPL	250.000000	56.057056	100.993983	81.482575
AMD	56.057056	250.000000	63.316364	101.816637
MSFT	100.993983	63.316364	250.000000	106.338055
NVDA	81.482575	101.816637	106.338055	250.000000