INTRO TO PORTFOLIO RISK MANAGEMENT IN PYTHON

Financial Returns

Dakota Wixom
Quantitative Analyst | QuantCourse.com

Course Overview

Learn how to analyze investment return distributions, build portfolios and reduce risk, and identify key factors which are driving portfolio returns.

- Univariate Investment Risk
- Portfolio Investing
- Factor Investing
- Forecasting and Reducing Risk

Investment Risk

What is Risk?

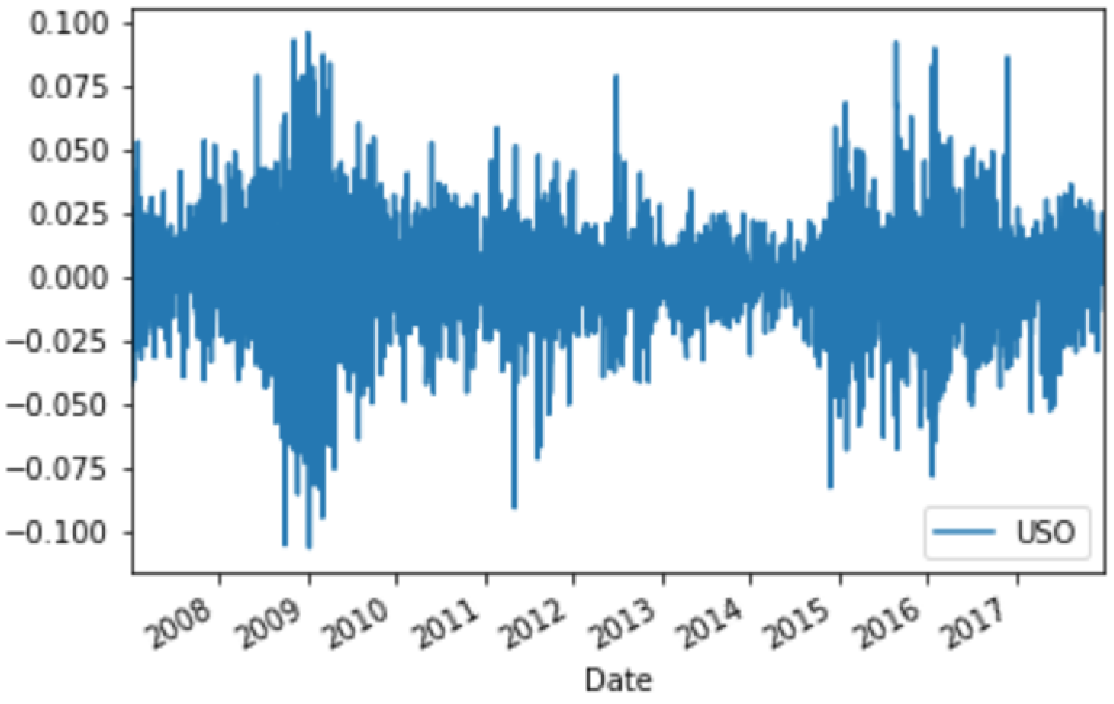
- Risk in financial markets is a measure of uncertainty
- Dispersion or variance of financial returns

How do you typically measure risk?

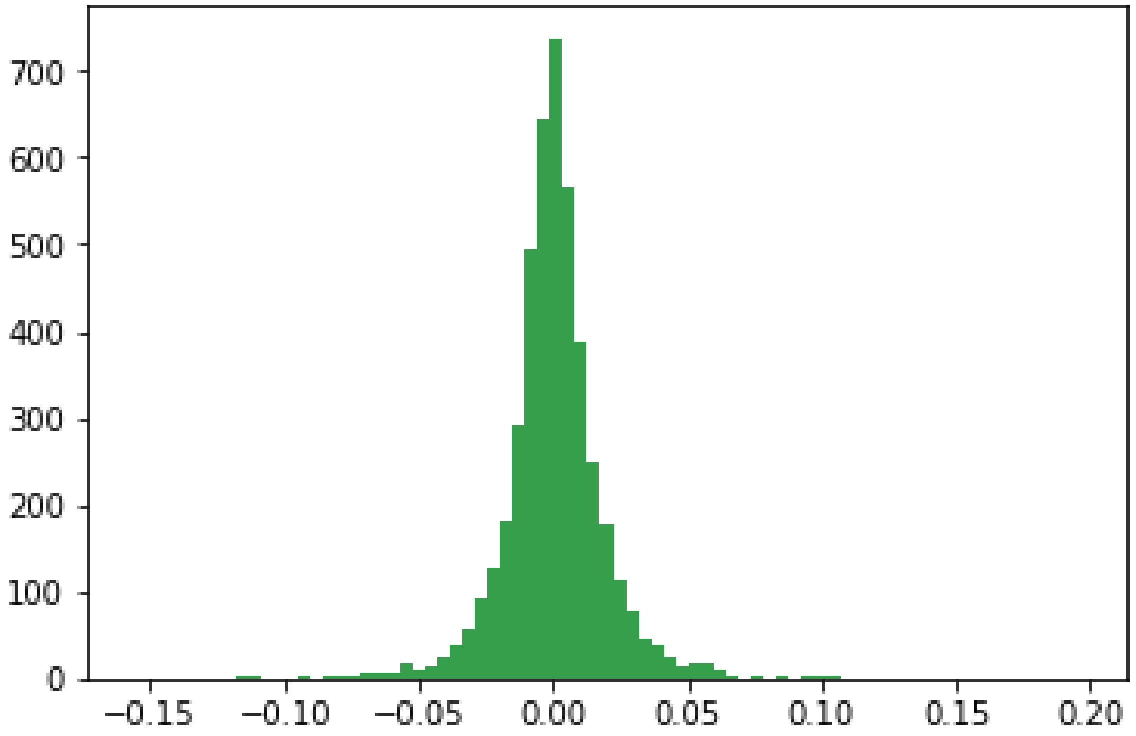
- Standard deviation or variance of daily returns
- Kurtosis of the daily returns distribution
- Skewness of the daily returns distribution
- Historical drawdown

Financial Risk

RETURNS

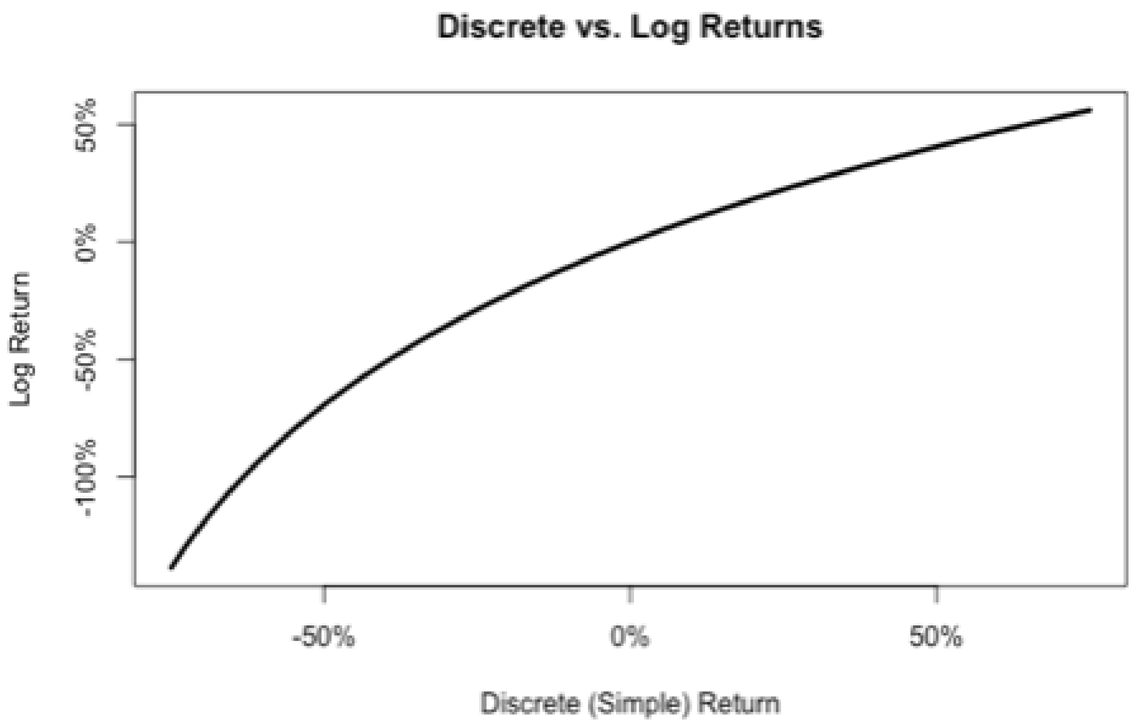


PROBABILITY



A Tale of Two Returns

- Returns are derived from stock prices
- **Discrete returns** (simple returns) are the most commonly used, and represent periodic (e.g. daily, weekly, monthly, etc.) price movements
- **Log returns** are often used in academic research and financial modeling. They assume continuous compounding.



Log returns are always smaller than discrete returns

Calculating Stock Returns

- Discrete returns are calculated as the change in price as a percentage of the previous period's price

Calculating Discrete Returns

$$R_{t_2} = \frac{(P_{t_2} - P_{t_1})}{P_{t_1}}$$

Calculating Log Returns

- Log returns are calculated as the difference between the log of two prices
- Log returns *aggregate across time*, while discrete returns *aggregate across assets*

Calculating Log Returns

$$\mathrm{Rl}_{t_2} = \frac{ln(P_{t_2})}{ln(P_{t_1})} = ln(P_{t_2}) - ln(P_{t_1})$$

Calculating Stock Returns in Python

STEP 1:

Load in stock prices data and store it as a pandas DataFrame organized by date:

```
In [1]: import pandas as pd
In [2]: StockPrices = pd.read_csv('StockData.csv', parse_dates=['Date'])
In [3]: StockPrices = StockPrices.sort_values(by='Date')
In [4]: StockPrices.set_index('Date', inplace=True)
```

Calculating Stock Returns in Python

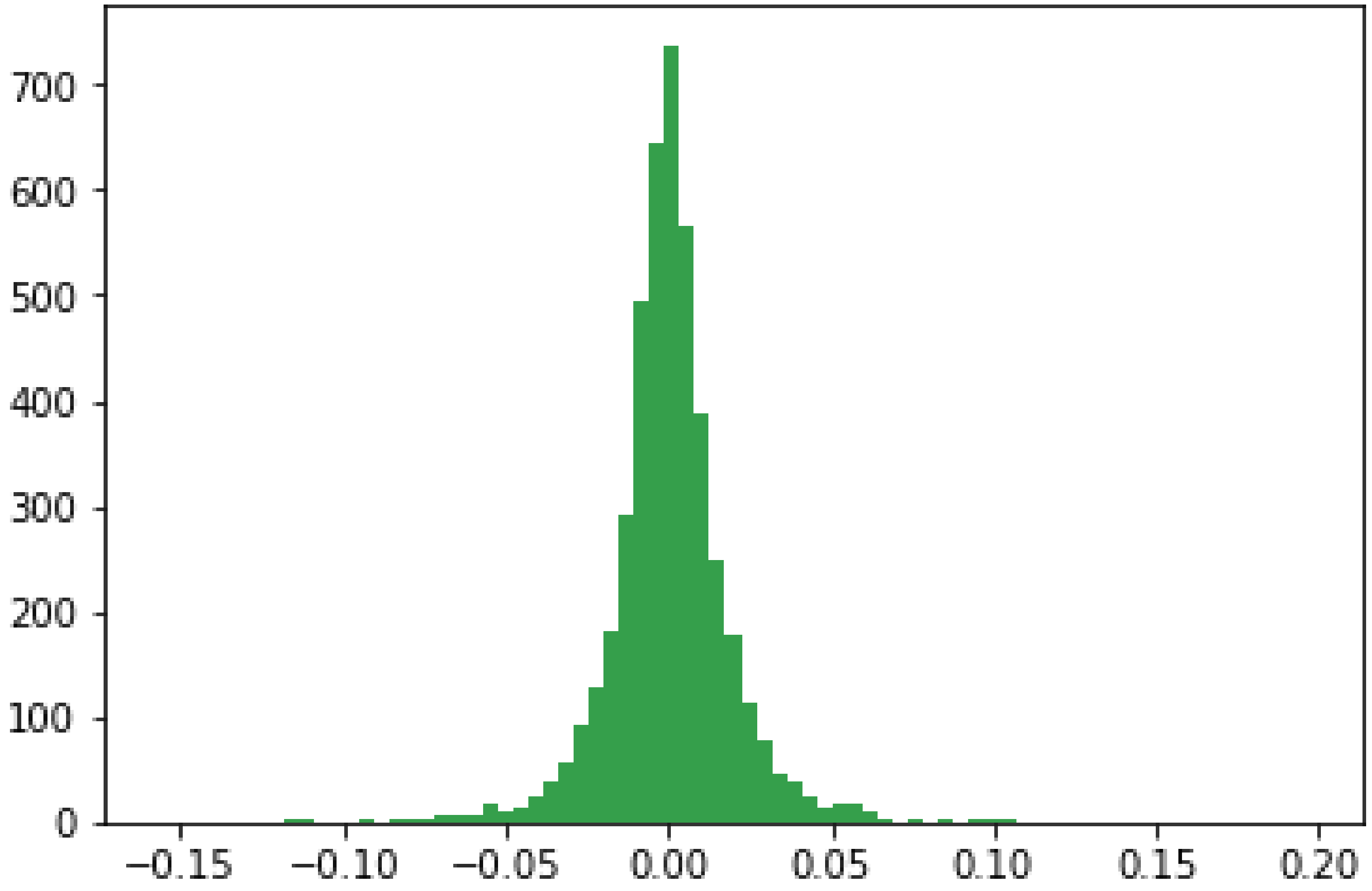
STEP 2:

Calculate daily returns of the adjusted close prices and append the returns as a new column in the DataFrame:

```
In [1]: StockPrices["Returns"] = StockPrices["Adj Close"].pct_change()
In [2]: StockPrices["Returns"].head()
```

Visualizing Return Distributions

```
In [1]: import matplotlib.pyplot as plt
In [2]: plt.hist(StockPrices["Returns"].dropna(), bins=75, density=False)
In [3]: plt.show()
```



INTRO TO PORTFOLIO RISK MANAGEMENT IN PYTHON

Let's practice!

INTRO TO PORTFOLIO RISK MANAGEMENT IN PYTHON

Mean, Variance, and Normal Distributions

Dakota Wixom
Quantitative Analyst | QuantCourse.com

Moments of Distributions

Probability distributions have the following moments:

- 1) Mean (μ)
- 2) Variance (σ^2)
- 3) Skewness
- 4) Kurtosis

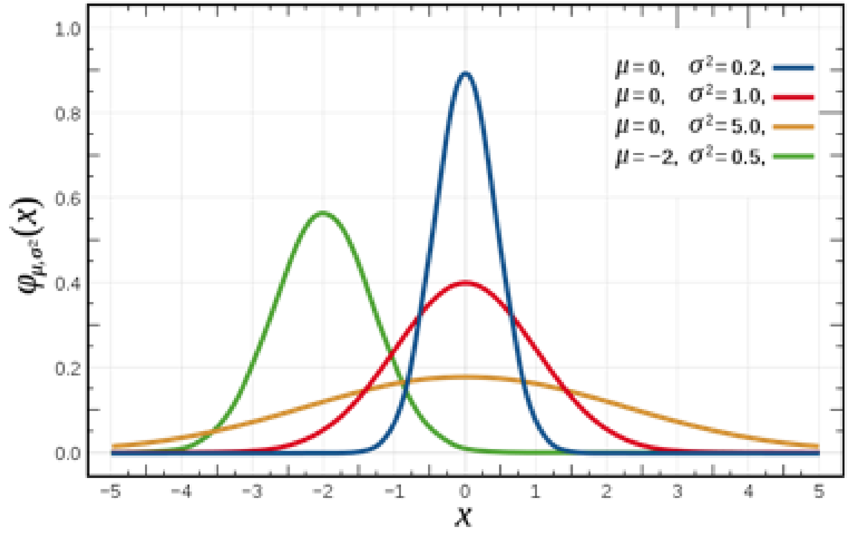
The Normal Distribution

There are many types of distributions. Some are normal and some are non-normal. A random variable with a **Gaussian distribution** is said to be *normally distributed*.

Normal Distributions have the following properties:

- Mean = μ
- Variance = σ^2
- Skewness = 0
- Kurtosis = 3

Probability Density Function of Normal Distributions



Probability Density Function Equation of a Standard Normal Distribution

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The Standard Normal Distribution

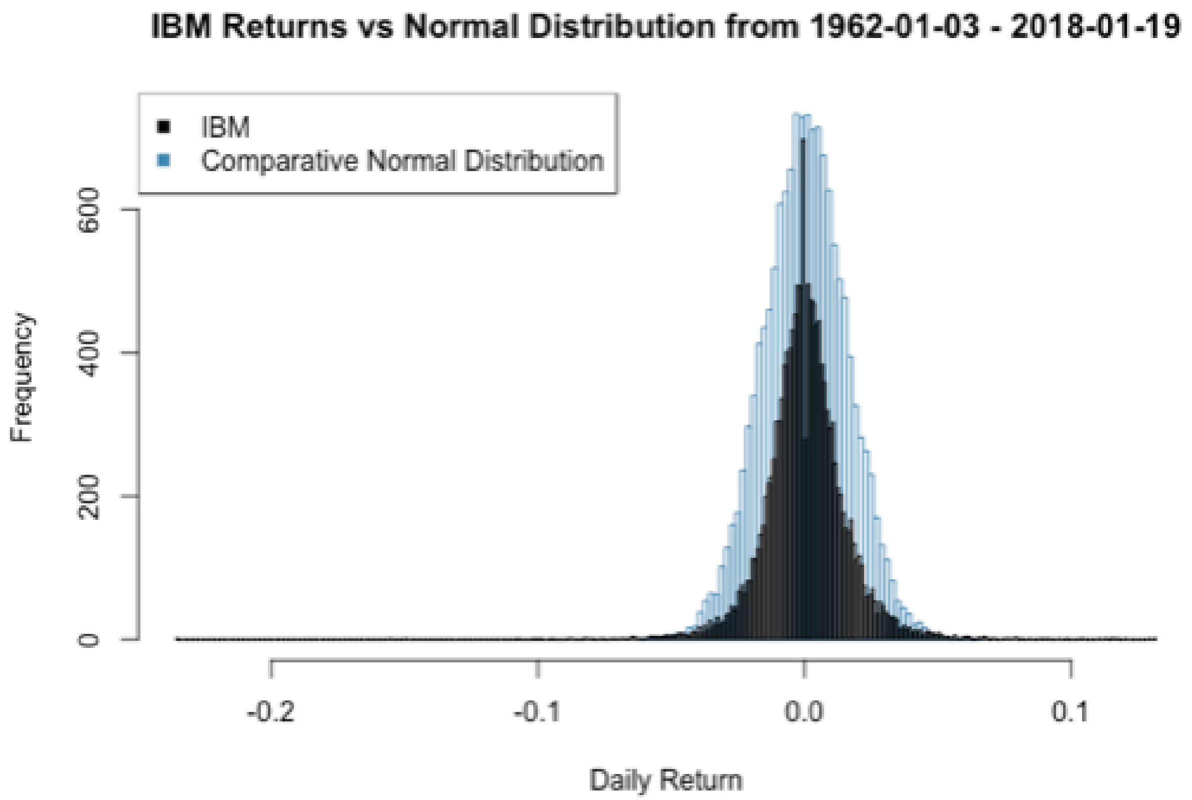
The **Standard Normal** is a special case of the Normal Distribution when:

- $\sigma = 1$
- $\mu = 0$

Comparing Against a Normal Distribution

- Normal distributions have a skewness near 0 and a kurtosis near 3.
- Financial returns tend not to be normally distributed
- Financial returns can have high kurtosis

Comparing Against a Normal Distribution



Distribution	Mu	Sigma	Max	Min	Skewness	Kurtosis
Normal Distribution	4e-04	0.0158	0.0530	-0.0610	-0.02	2.96
IBM	4e-04	0.0158	0.1316	-0.2352	0.02	13.20

Calculating Mean Returns in Python

To calculate the average daily return, use the `np.mean()` function:

```
In [1]: import numpy as np
In [2]: np.mean(StockPrices["Returns"])
Out [2]: 0.0003
```

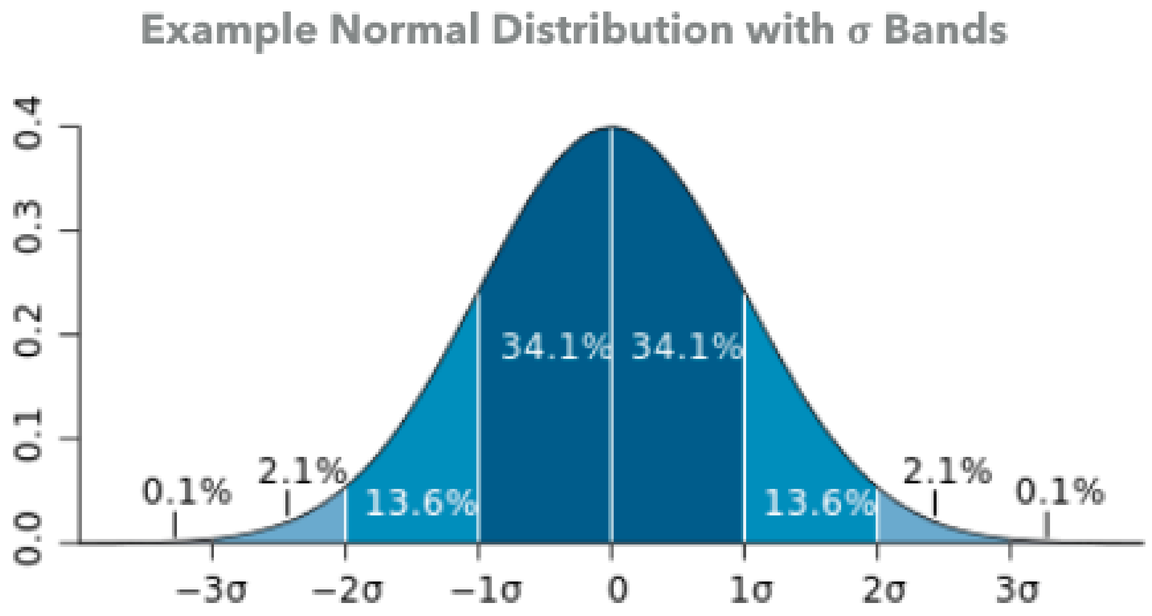
To calculate the average annualized return assuming 252 trading days in a year:

```
In [1]: import numpy as np
In [2]: ((1+np.mean(StockPrices["Returns"]))**252)-1
Out [2]: 0.0785
```

Standard Deviation and Variance

Standard Deviation (Volatility)

- Variance = σ^2
- Often represented in mathematical notation as σ, or referred to as *volatility*
- An investment with higher σ is viewed as a higher risk investment
- Measures the dispersion of returns



Note that 2 σ from the mean contains approximately 95.4% of all data in a normal distribution

Standard Deviation and Variance in Python

Assume you have pre-loaded stock returns data in the `StockData` object. To calculate the periodic standard deviation of returns:

```
In [1]: import numpy as np
In [2]: np.std(StockPrices["Returns"])
Out [2]: 0.0256
```

To calculate variance, simply square the standard deviation:

```
In [1]: np.std(StockPrices["Returns"])**2
Out [2]: 0.000655
```

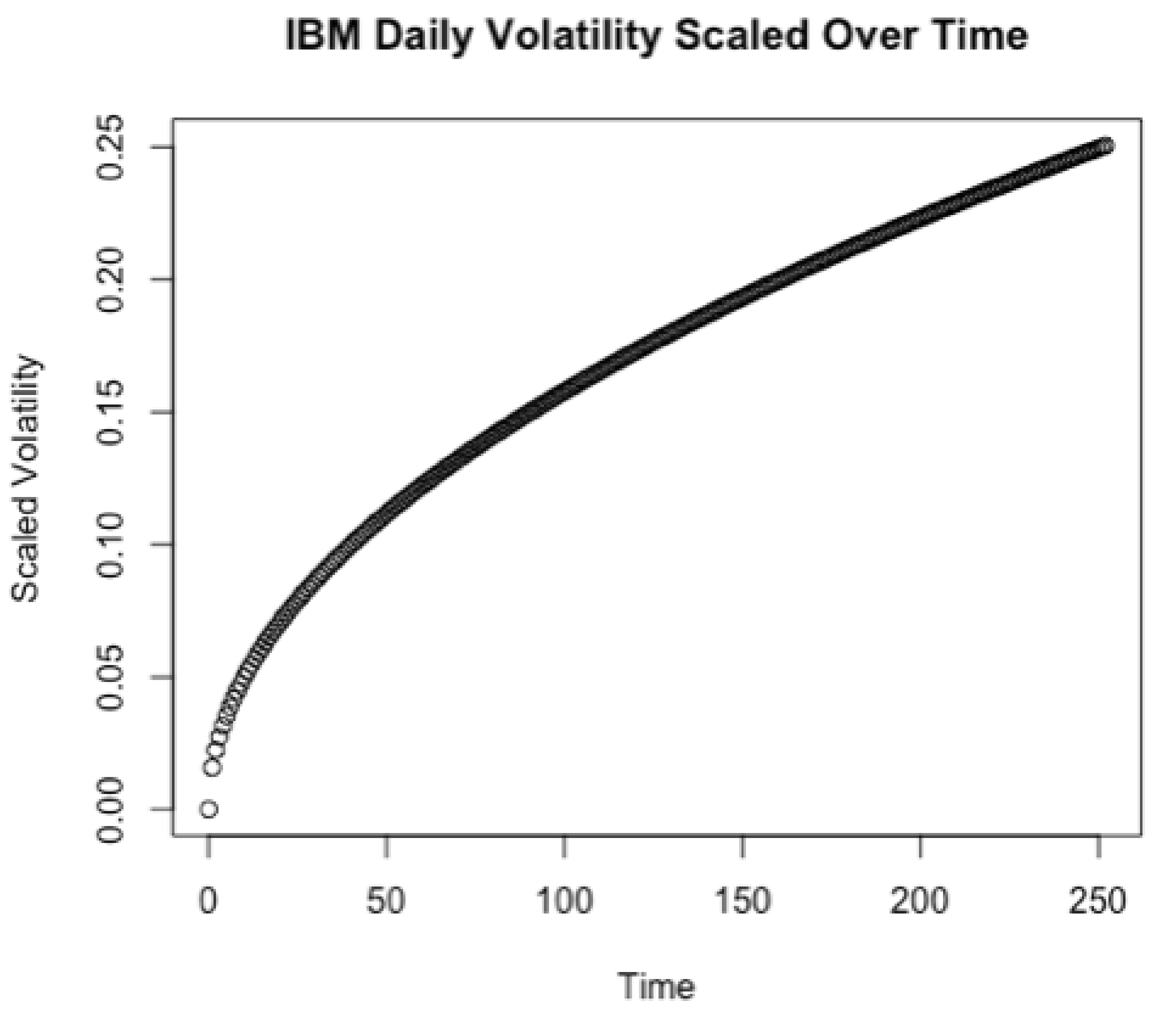
Scaling Volatility

- Volatility scales with the square root of time
- You can normally assume 252 trading days in a given year, and 21 trading days in a given month

Example Volatility Scaling Equations

$$\sigma_{Annual} = \sigma_{Daily} * \sqrt{(252)}$$

$$\sigma_{Monthly} = \sigma_{Daily} * \sqrt{(21)}$$



Scaling Volatility in Python

Assume you have pre-loaded stock returns data in the `StockData` object. To calculate the annualized volatility of returns:

```
In [1]: import numpy as np
In [2]: np.std(StockPrices["Returns"]) * np.sqrt(252)
Out [2]: 0.3071
```

INTRO TO PORTFOLIO RISK MANAGEMENT IN PYTHON

Let's practice!

INTRO TO PORTFOLIO RISK MANAGEMENT IN PYTHON

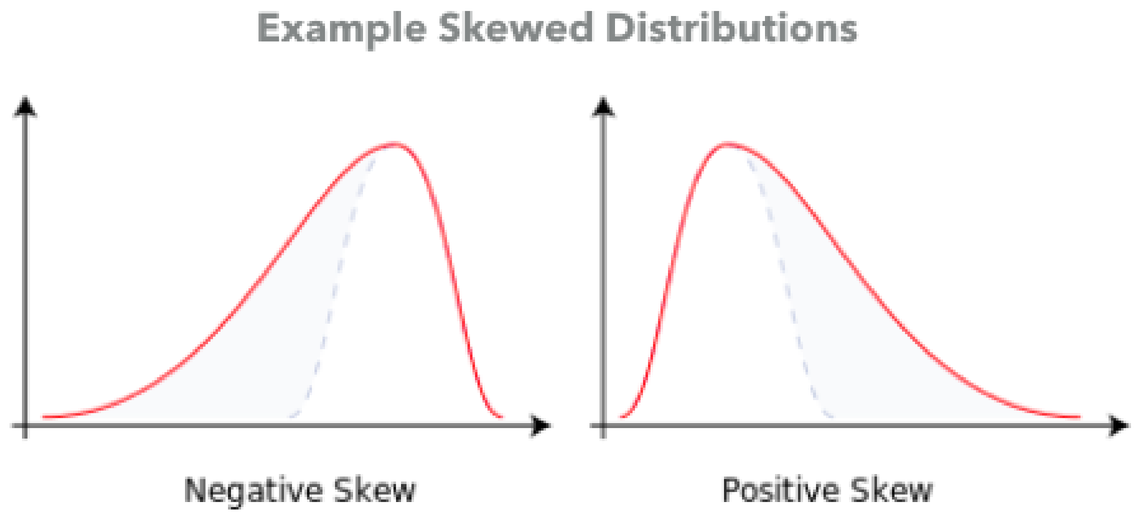
Skewness and Kurtosis

Dakota Wixom
Quantitative Analyst | QuantCourse.com

Skewness

Skewness is the third moment of a distribution.

- **Negative Skew:** The mass of the distribution is concentrated on the right. Usually a right-leaning curve
- **Positive Skew:** The mass of the distribution is concentrated on the left. Usually a left-leaning curve
- In finance, you would tend to want positive skewness



Note the difference in the length and fatness of the tails depending on the skewness

Skewness in Python

Assume you have pre-loaded stock returns data in the `StockData` object.

To calculate the skewness of returns:

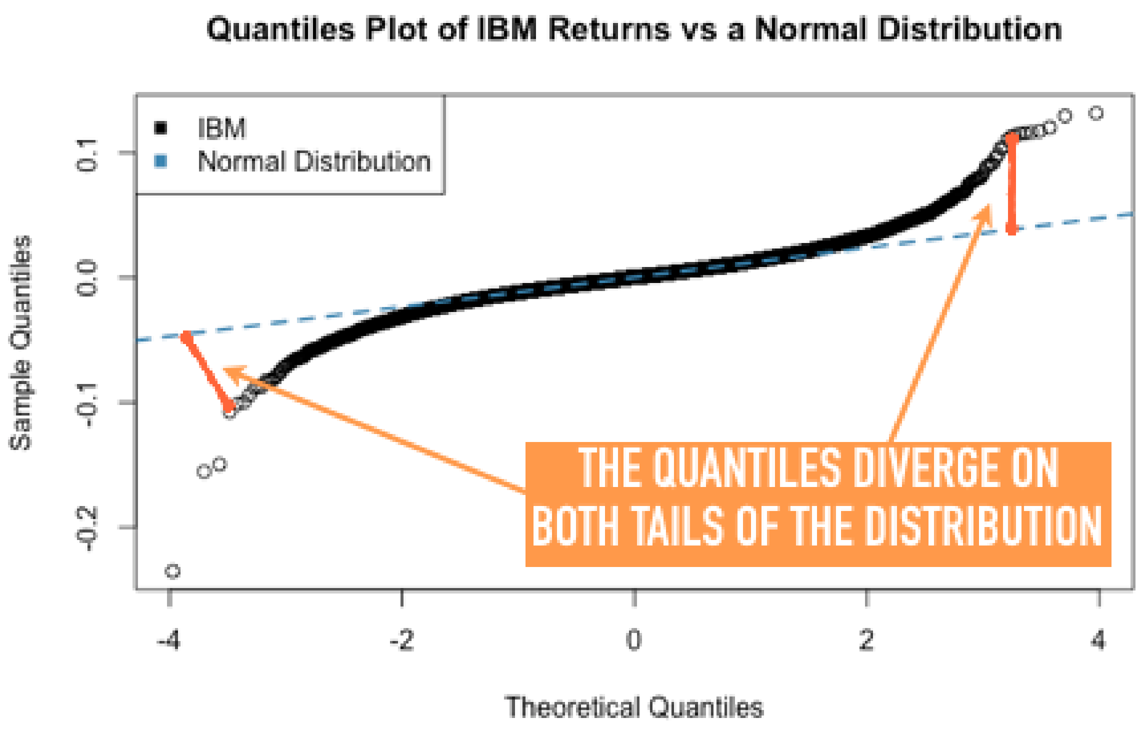
```
In [1]: from scipy.stats import skew
In [2]: skew(StockData["Returns"].dropna())
Out [2]: 0.225
```

Note that the skewness is higher than 0 in this example, suggesting non-normality.

Kurtosis

Kurtosis is a measure of the thickness of the tails of a distribution

- Most financial returns are leptokurtic
- **Leptokurtic:** When a distribution has positive excess kurtosis (kurtosis greater than 3)
- **Excess Kurtosis:** Subtract 3 from the sample kurtosis to calculate "Excess Kurtosis"



Note the divergence near the tails? That's an example of kurtosis

Excess Kurtosis in Python

Assume you have pre-loaded stock returns data in the `StockData` object. To calculate the **excess kurtosis** of returns:

```
In [1]: from scipy.stats import kurtosis
In [2]: kurtosis(StockData["Returns"].dropna())
Out [2]: 2.44
```

Note the excess kurtosis greater than 0 in this example, suggesting non-normality.

Testing for Normality in Python

How do you perform a statistical test for normality?

The null hypothesis of the **Shapiro-Wilk test** is that the data are normally distributed.

To run the Shapiro-Wilk normality test in Python:

```
In [1]: from scipy import stats
In [2]: p_value = stats.shapiro(StockData["Returns"].dropna())[1]
In [3]: if p_value <= 0.05:
In [4]:     print("Null hypothesis of normality is rejected.")
In [5]: else:
In [6]:     print("Null hypothesis of normality is accepted.")
```

The p-value is the second variable returned in the list. If the p-value is less than 0.05, the null hypothesis is rejected because the data are most likely non-normal.

INTRO TO PORTFOLIO RISK MANAGEMENT IN PYTHON

Let's practice!