上海立信会计金融学院
SHANGHAI LIXIN UNIVERSITY OF ACCOUNTING AND FINANCE

《Python金融数据分析》

Hong Cheng（程宏）
School of Statistics and Mathematics
Shanghai LiXin University of Accounting and Finance

April 2022

These two applications are **typical tasks of statistical inference** to infer the promptings of interesting targets.

03

Confidence Interval

How to estimate using confidence interval

we will explore **how to** estimate the **average return** using **confidence interval**.

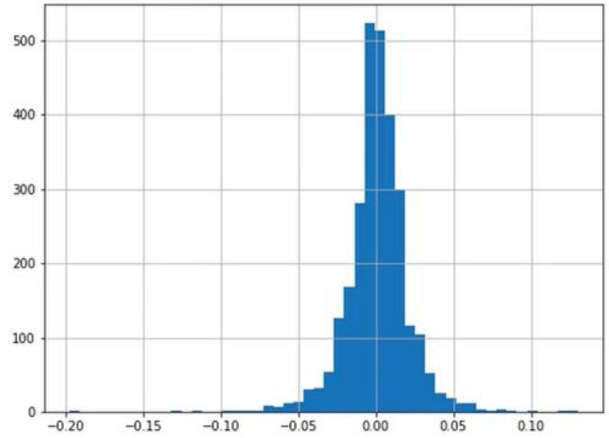
SHANGHAI LIXIN UNIVERSITY OF ACCOUNTING AND FINANCE

Here is a sample of the log return of a stock price of Apple. We can get average return in this sample.

Sample log return of Apple

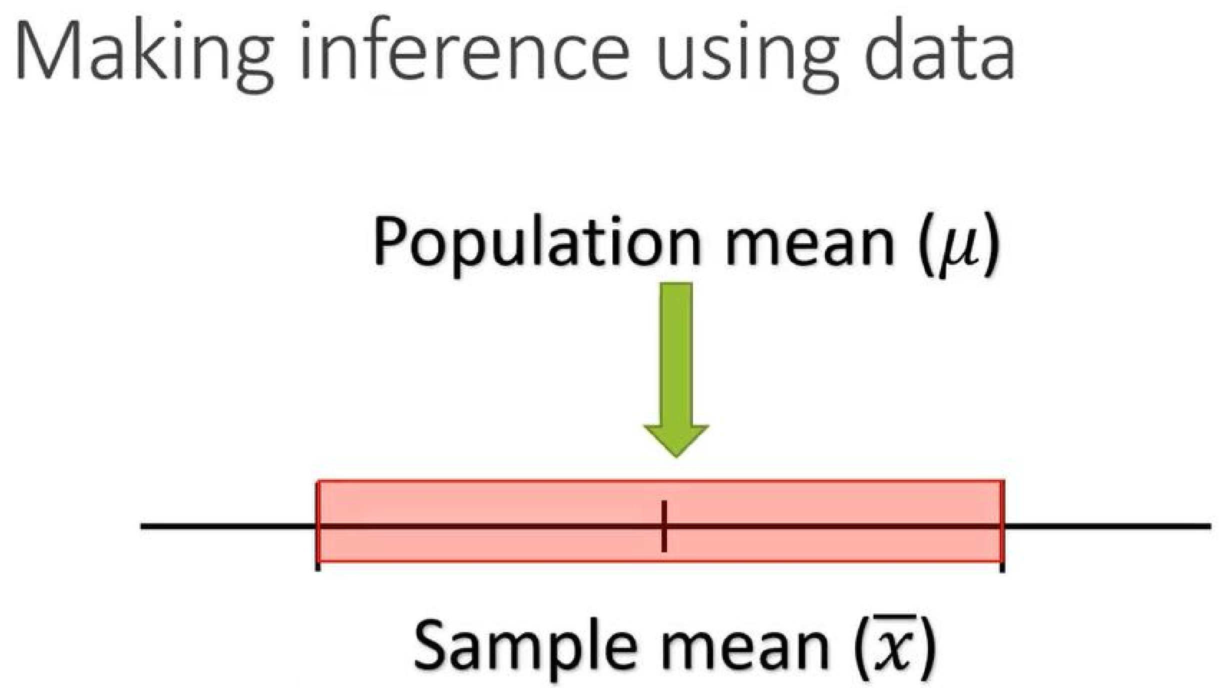
In [1]
```
aapl = pd.DataFrame.from_csv('data/apple.csv')
aapl['logReturn'] = np.log(aapl['Close'].shift(-1)) - np.log(aapl['Close'])
```

Out [2]

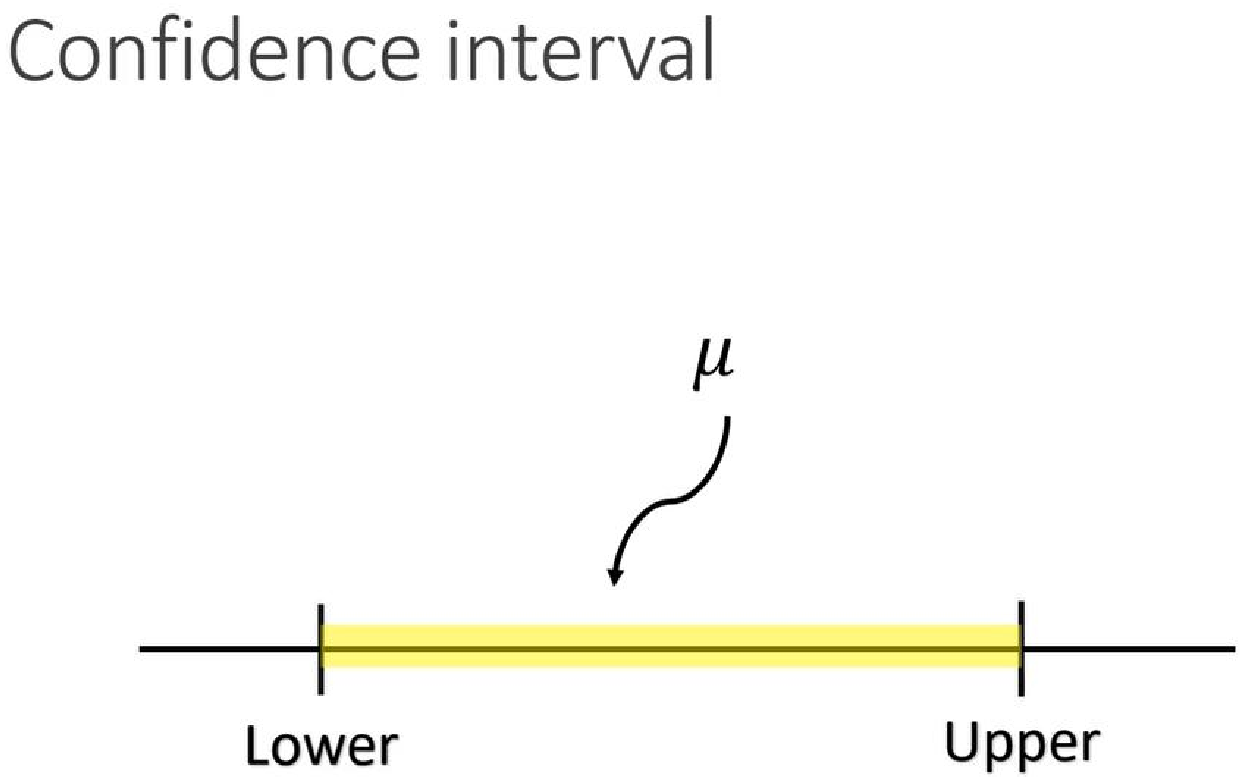


We can **<u>sample mean to estimate the real average return</u>**, which is population mean in our example.

Intuitionally, **if** a sample is a good representative of the population, **the population mean should be close to sample mean**. It is plausible to say that the population mean is in a range with sample mean centered.

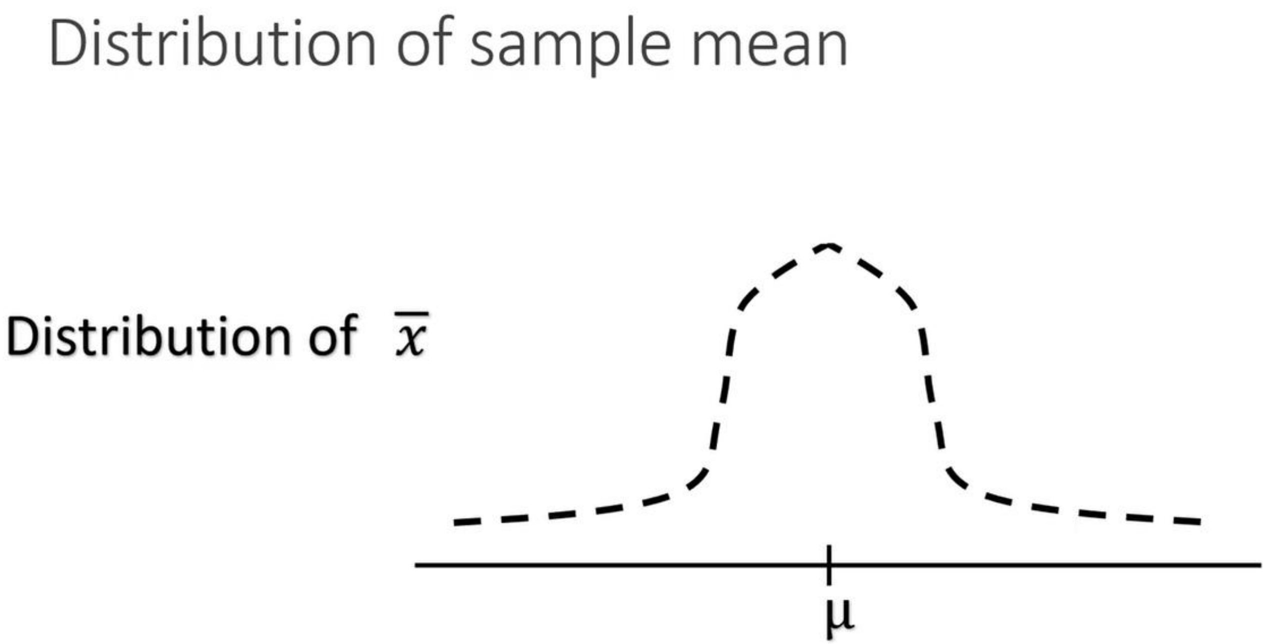


Hence, our task in this lecture is to estimate population mean using interval with lower and upper bound.



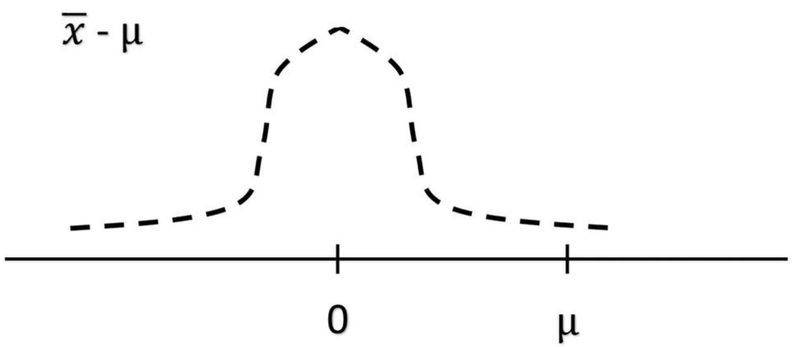
To start with, we need to **standardize sample mean** because different sample has different mean and a standard deviation.

We have learned that distribution of sample mean is normal in our last lecture.



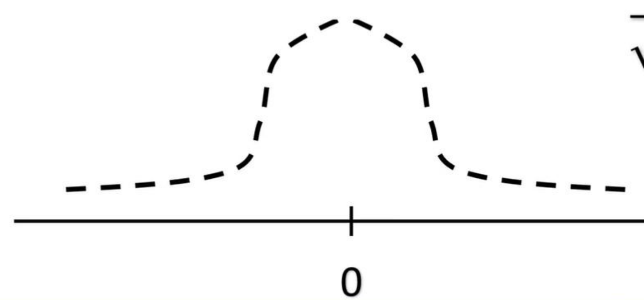
We can standardize sample mean by minus it's mean, **which is identical to population mean and then divided by its standard deviation**, which is the standard deviation of population divided by square root of sample size.

Standardize a normal random variable



Standardize a normal random variable

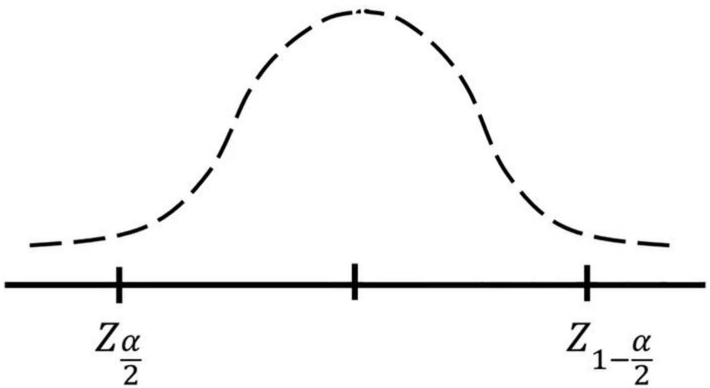
$$Z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$



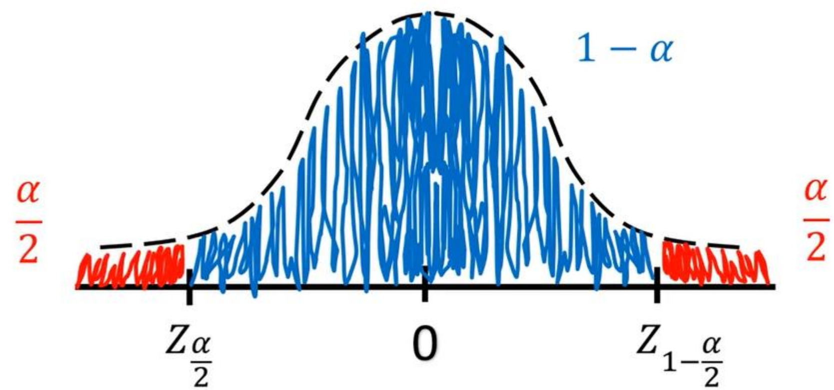
After standardization, it'll become standard normal, and follows **Z-distribution**.

For Z-distribution, it is not difficult to find the two quantities.

Quantiles of Z distribution



Quantiles of Z distribution



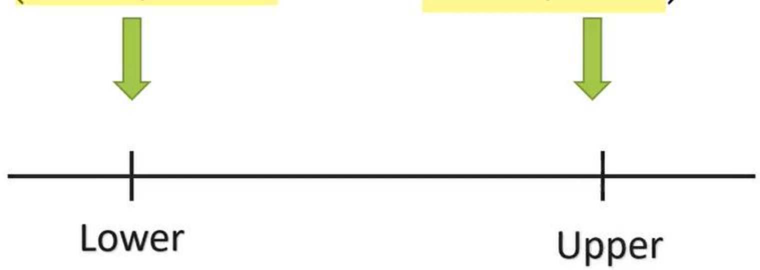
$$\boxed{Z_{1-\frac{\alpha}{2}} = -Z_{\frac{\alpha}{2}}}$$

Since the standardization form of sample mean is also Z, then we have this equation.

Confidence interval

$$P\left(Z_{\frac{\alpha}{2}} \le \frac{\bar{x}-\mu}{\frac{\sigma}{\sqrt{n}}} \le Z_{1-\frac{\alpha}{2}}\right) = 1-\alpha$$

$$P\left(\bar{X} - \frac{\sigma}{\sqrt{n}} Z_{\frac{\alpha}{2}} \le \mu \le \bar{X} + \frac{\sigma}{\sqrt{n}} Z_{\frac{\alpha}{2}}\right) = 1-\alpha$$



Notice that **Sigma is the population standard deviation**, which is usually **unknown**. In practice, we can **replace it using the sample standard deviation** if sample size is large enough.

Confidence interval

$$P\left(\bar{X} - \frac{\sigma}{\sqrt{n}} Z_{\frac{\alpha}{2}} \leq \mu \leq \bar{X} + \frac{\sigma}{\sqrt{n}} Z_{\frac{\alpha}{2}}\right) = 1 - \alpha$$

$\sigma \xrightarrow{\text{With large n}} S$

Confidence interval at the level of 1-α

In our problems, **to build the interval for the average return**, **we need to find quantiles of mean distribution** which has been discussed in topic two.

We can use the **norm.ppf** to get the quantiles.

Confidence interval for daily return

In [1]
```
aapl = pd.DataFrame.from_csv('data/apple.csv')
aapl['logReturn'] = np.log(aapl['Close'].shift(-1)) - np.log(aapl['Close'])

# values for calculating the 80% confidence interval
z_left = norm.ppf(0.1)
z_right = norm.ppf(0.9)
sample_mean = aapl['logReturn'].mean()
sample_std = aapl['logReturn'].std(ddof=1)/(aapl.shape[0])**0.5
```

The 80 percent of confidence interval is printed out. **Average return of Apple stocks falls in this interval with 80 percent chance.**

Confidence interval for daily return

In [2]
```
interval_left = sample_mean+z_left*sample_std
interval_right = sample_mean+z_right*sample_std
print("Sample Mean is ", sample_mean)
print("*****************************************************")
print("80% confidence interval is ")
print(interval_left,interval_right)
```

Out [2]
```
Sample Mean is 0.0009754677591500088
*****************************************************
80% confidence interval is
(0.00049273672549367546, 0.0014581987928065005)
```

Notice, **this interval is on the positive side. It implies that the average return is very likely to be positive**.

(0.00049273672549367546, 0.0014581987928065005) **Positive**

Lab1： Confidence Interval

Instructions

- You are going to practice the code of estimating the average stock return with a certain confidence level.

Confidence Interval

In [1]:
```
import pandas as pd
import numpy as np
from scipy.stats import norm
```

In [2]:
```
ms = pd.DataFrame.from_csv('../data/microsoft.csv')
ms.head()
```

Out[2]:

Date	Open	High	Low	Close	Adj Close	Volume
2014-12-31	46.730000	47.439999	46.450001	46.450001	42.848763	21552500
2015-01-02	46.660000	47.419998	46.540001	46.759998	43.134731	27913900
2015-01-05	46.369999	46.730000	46.250000	46.330002	42.738068	39673900
2015-01-06	46.380001	46.750000	45.540001	45.650002	42.110783	36447900
2015-01-07	45.980000	46.459999	45.490002	46.230000	42.645817	29114100

Estimate the average stock return with 90% Confidence Interval

In [3]:
```
# we will use log return for average stock return of Microsoft

ms['logReturn'] = np.log(ms['Close'].shift(-1)) - np.log(ms['Close'])
```

In [8]:
```
# Lets build 90% confidence interval for log return
sample_size = ms['logReturn'].shape[0]
sample_mean = ms['logReturn'].mean()
sample_std = ms['logReturn'].std(ddof=1) / sample_size**0.5

# left and right quantile
z_left = None
z_right = None

# upper and lower bound
interval_left = None
interval_right = None
```

In [9]:
```
# 90% confidence interval tells you that there will be 90% chance that the average stock return lies between "interval_left"
# and "interval_right".

print('90% confidence interval is ', (interval_left, interval_right))
```

```
90% confidence interval is  (-1.5603253899378836e-05, 0.001656066226145423)
```

Expected output: 90% confidence interval is (-1.5603253899378836e-05, 0.001656066226145423)

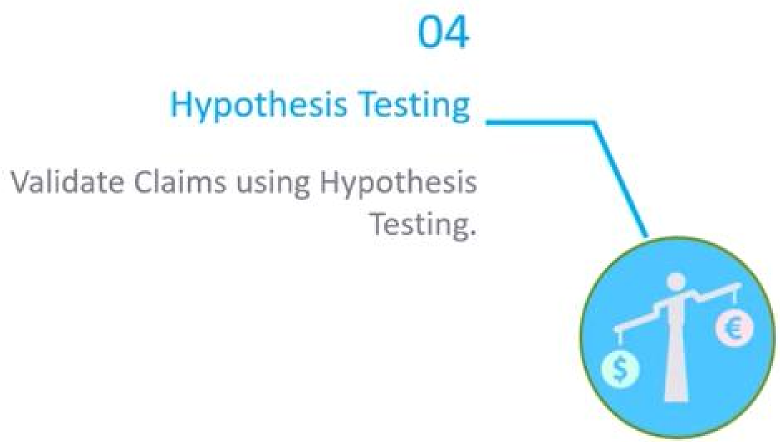
Confidence Interval.ipynb在Github中下载

https://github.com/cloudy-sfu/QUN-Data-Analysis-in-Finance/tree/main/Labs

Jupyternote Book课堂练习
二十分钟

In many situations, we need to demonstrate validity of assertions. **For example**, <u>you are a venture capitalist and is proposed a project running 36 months</u>.

With 36 months data at hand, should you invest in this project? **Suppose you will invest if average monthly profit is over 20,000.** This question is not to ask you to estimate some parameters. Instead, you need to make a judgement whether the condition is satisfied. We need a new statistic tool, **hypothesis testing**.

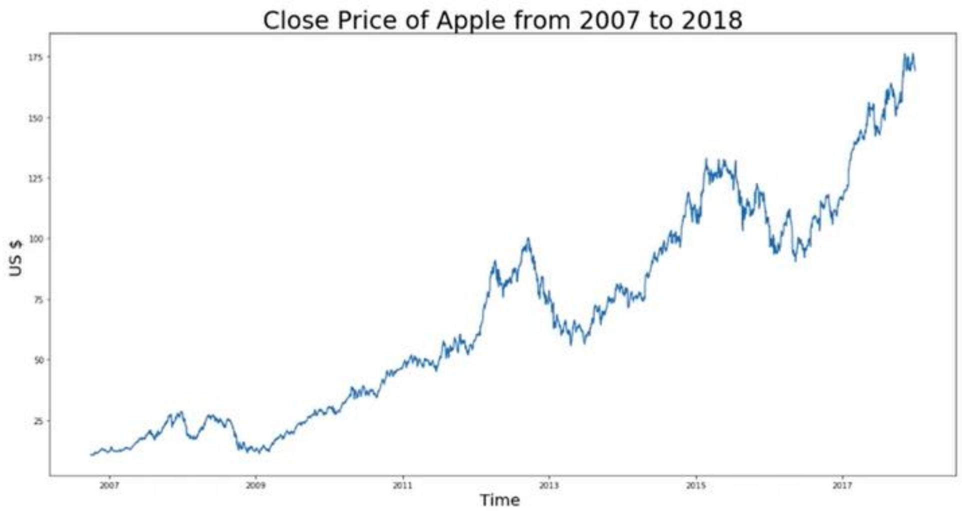


This is a daily close price of Apple from 2007 to 2018. **<u>It looks like the price is in an upward trend and we may guess the average of daily return is positive.</u>**

In [1]

```
plt.title("Close Price of Apple from 2007 to 2018",size=30)
plt.xlabel("Time", size=20)
plt.ylabel("US $", size=20)
plt.plot(aapl.loc[:,'Close'])
```

Out [1]

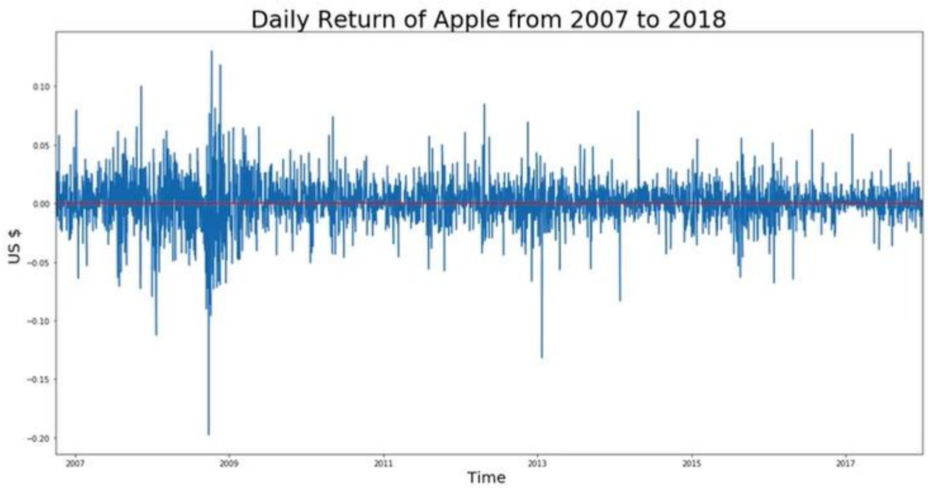


However, if we plot the daily return directly, the daily return goes positive, negative. And **our assertion that the average of daily return is positive is not obvious.**

In [2]

```
plt.title("Daily Return of Apple from 2007 to 2018",size=30)
plt.xlabel("Time", size=20)
plt.ylabel("US $", size=20)
plt.xlim(aapl.index[0], aapl.index[-1])
plt.plot(aapl.loc[:, 'logReturn'])
plt.axhline(0, color='red')
```

Out [2]



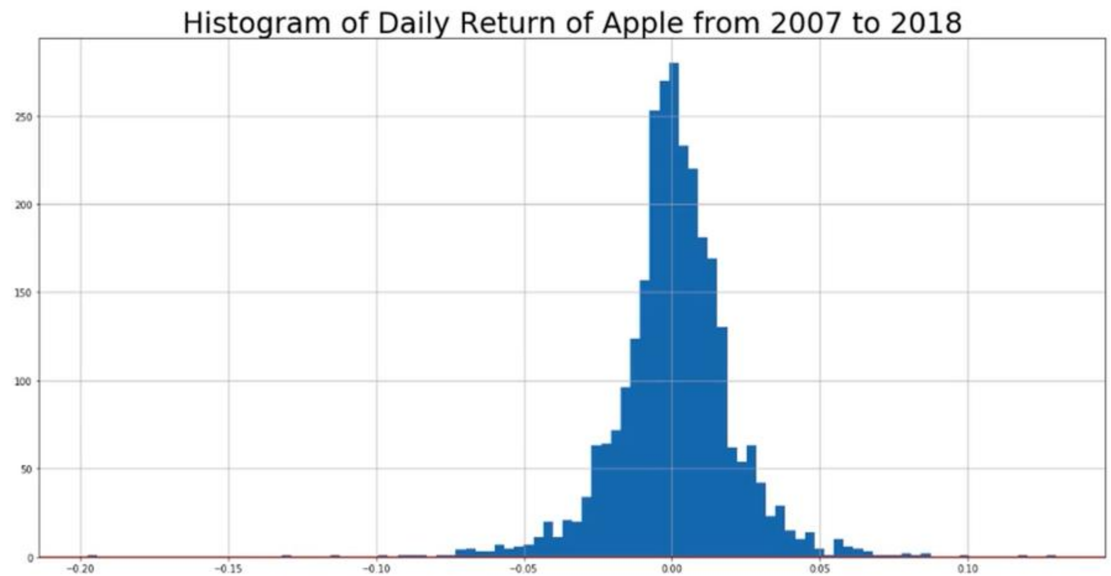
It is also not obvious whether the average of daily return is 0 or not.

In the histogram daily return, it is approximately symmetric above 0. **It is still not obvious whether the average daily return is different from 0**.

In [3]

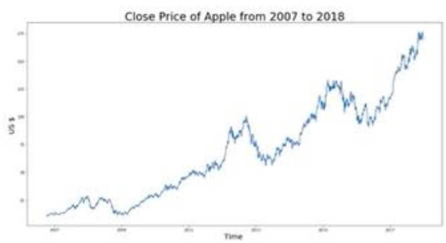
```
plt.title("Histogram of Daily Return of Apple from 2007 to 2018", size=30)
aapl.loc[:, 'logReturn'].dropna().hist(bins=100)
```

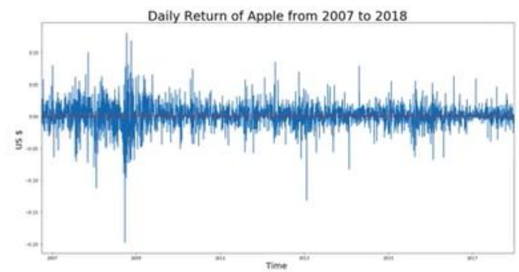
Out [3]

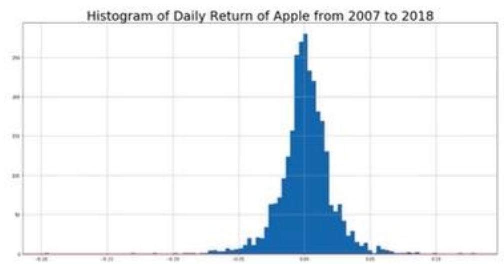


We want to use a **quantitative statistical tool** to **make judgement about the assertion that the average daily return is not 0**.

In statistics, **hypothesis testing** can use sample information to test the validity of conjectures about these parameters.







Positive ?

zero ?

Negative ?

Hypothesis Testing

The first step is to set hypothesis. We have null hypothesis and alternative hypothesis. Usually, **the null hypothesis is assertion we are against**. **Alternative hypothesis is a conclusion we accept whenever we reject the null**.

Setting hypothesis

Null hypothesis H_0: $\mu = 0$

Alternative hypothesis H_a: $\mu \neq 0$

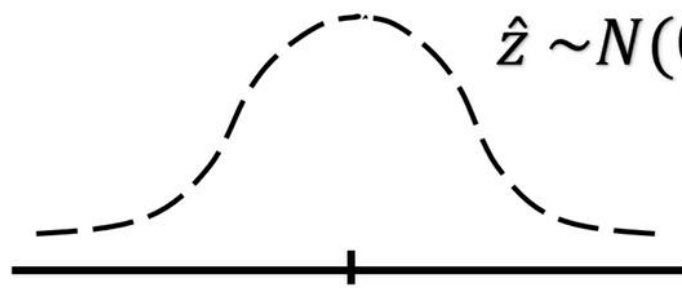
Intuitionally, given that the null is correct, the difference between sample statistic, $\bar{x}$, and the population parameter μ cannot be very large. **If it's significantly large**, **the null should be incorrect, and we should accept alternative.**

Given H_0 is correct

$$|\bar{x} - \mu| \longrightarrow \text{Not very large}$$

Standardization

$$\hat{z} = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$



$\hat{z} \sim N(0, 1)$

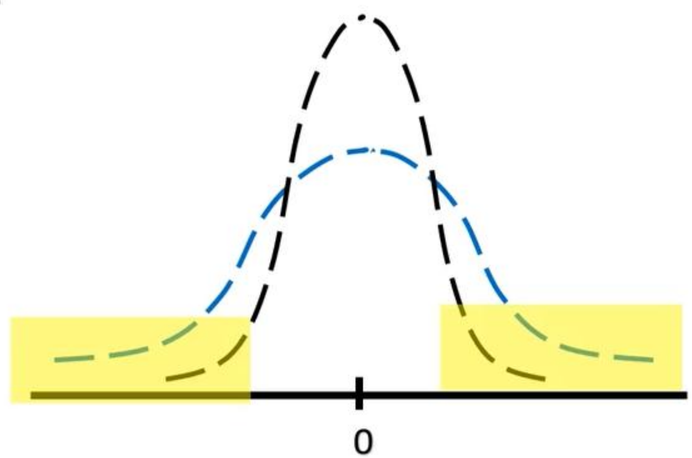
In hypothesis testing, we start with assumption that the null is correct. Hence, we know population mean is equal to 0. **But** in most situations, **population standard deviation is not known.** Then we can **replace population standard deviation with the sample standard deviation**. Then this new term denoted as t-hat, has a new distribution, t-distribution.

~~Population sd ?~~

Sample sd

Z-distribution

$$\hat{z} = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$



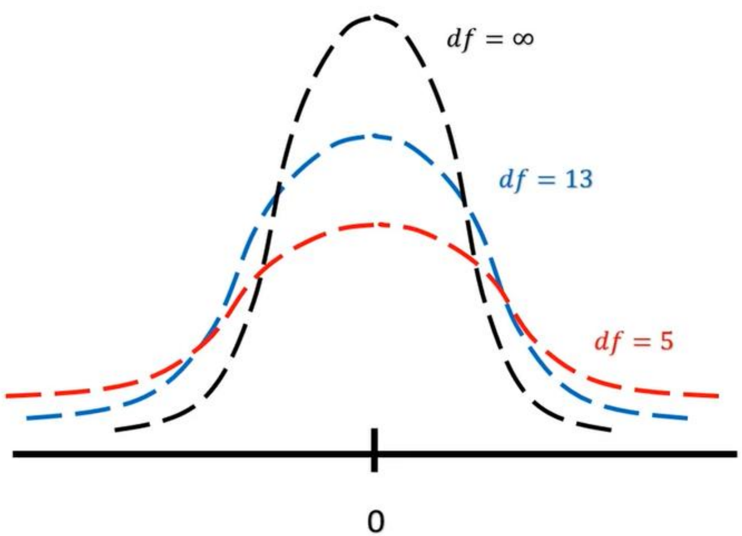
t-distribution

$$\hat{t} = \frac{\bar{x} - \mu}{s/\sqrt{n}}$$

The t-distribution is a dependent on the degree of freedom.

In our example, **<u>the degree of freedom is equal to the degree of freedom of the sample standard deviation, which is n-1.</u>** As the sample size increases, the degree of freedom increases, and the t is more and more like z-distribution. So with a large sample, we can treat t as if it is a z-distribution.

Treat $\hat{t}$ as if it is $\hat{z}$ distribution when n is large



$$degrees\ of\ freedom = n - 1$$

$$\hat{t} = \frac{\bar{x} - \mu}{s/\sqrt{n}}$$

上海立信会计金融学院
SHANGHAI LIXIN UNIVERSITY OF ACCOUNTING AND FINANCE

With large sample, t-hat follows z-distribution hence we denote this statistic **using z-hat too**. To emphasize that, it follows z-distribution.

Standardization

$$\boxed{\hat{z} = \frac{\bar{x} - \mu}{s/\sqrt{n}}}$$

In [4]

```
xbar= aapl['logReturn'].mean()
s= aapl['logReturn'].std(ddof=1)
n= aapl['logReturn'].shape[0]
zhat= (xbar-0)/(s/(n**0.5))
print(zhat)
```

Out [4]

```
2.5896661841029576
```

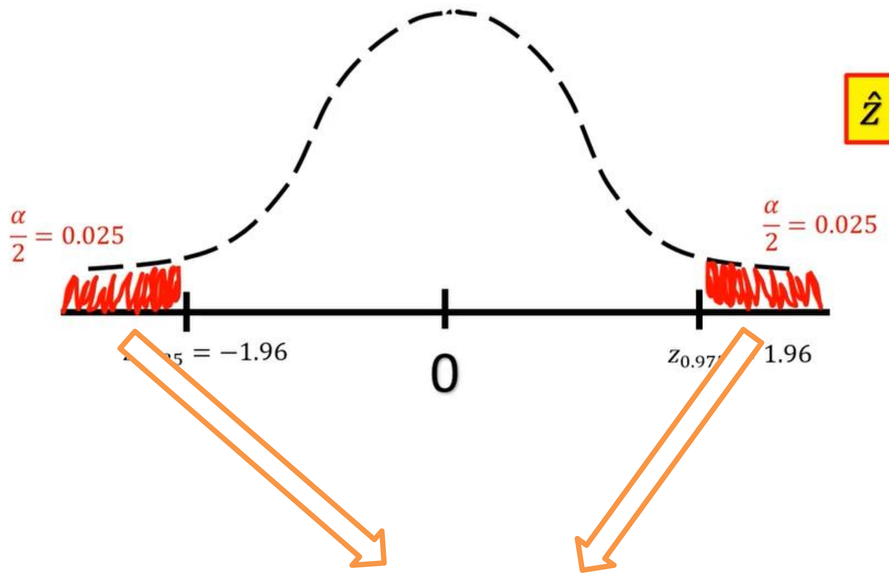
How do we find the significance level?

Set Decision Criteria

At $\alpha = 5\%$,

Reject H_0 if

$\boxed{\hat{z} > 1.96 \text{ or } \hat{z} < -1.96}$



This is called a **type 1 error,** **and the probability of a type 1 error is identical to the level of significance level.**

These two demands are called rejection regions.
This kind of test is called two-tailed test.

Here, we demonstrate how to get the quantiles which is also called **critical values**. Alpha equal to 5% is a given hence, **norm.ppf** can be applied to get the quantiles. In the print, we use a bold number to generate whether to reject or not directly.

Set Decision Criteria

In [5]

```
alpha=0.05
zleft= norm.ppf(alpha/2, 0, 1)
zright= -zleft
print(zleft, zright)
print('At the significance level of ', alpha)
print('Shall we reject?:', zhat>zright or zhat<zleft)
```

Out [5]

```
-1.9599639845400545 1.9599639845400545
At the significance level of 0.05
Shall we reject: True
```

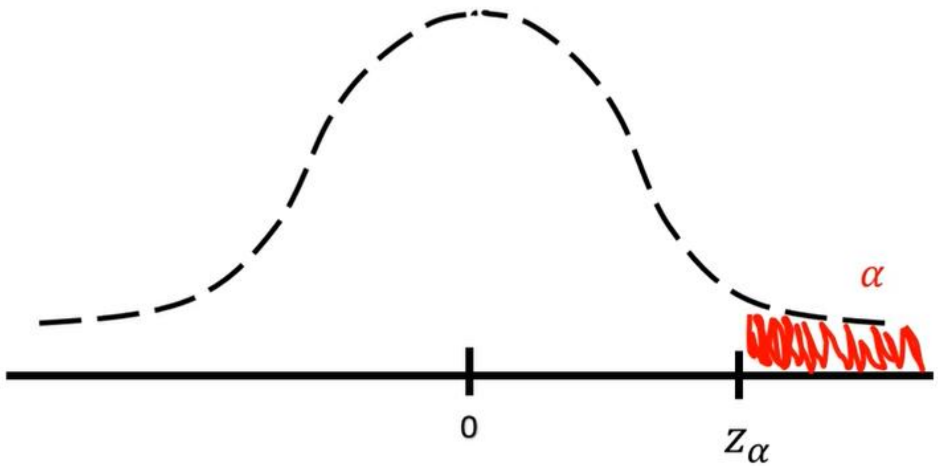
We may want to further demonstrate that the average return is in fact positive. We need another kind of test, **one-tail test**.

Hypothesis for One Tail Test

Null hypothesis	$H_0: \quad \mu \leq 0$
Alternative hypothesis	$H_a: \quad \mu > 0$

If z-hat is significantly large, which implies that sample mean is a positive comparing to mu equal to 0. Hence, **it is not likely to be sampled from population**, which may equal to 0. It is also **not likely to be sampled from population with negative μ**.

Set Decision Criteria



Reject H_0 if

$$\hat{z} > z_\alpha$$

Using Python, we can show that the null is rejected under 5%. It means that the average daily return of a population is indeed positive.

Set Decision Criteria of One Tail Test

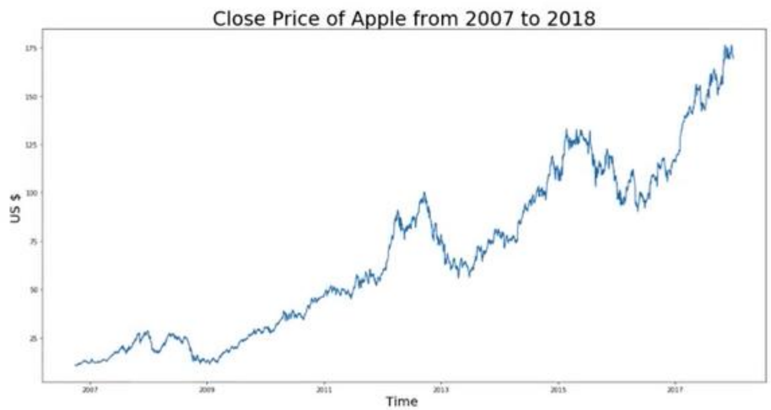
In [6]

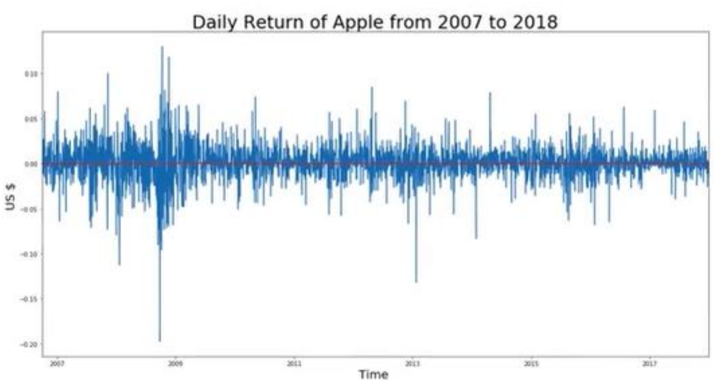
```
alpha=0.05
zright= norm.ppf(1- alpha, 0, 1)
print(zleft, zright)
print('At the significance level of ', alpha)
print('Shall we reject?:', zhat>zright)
```

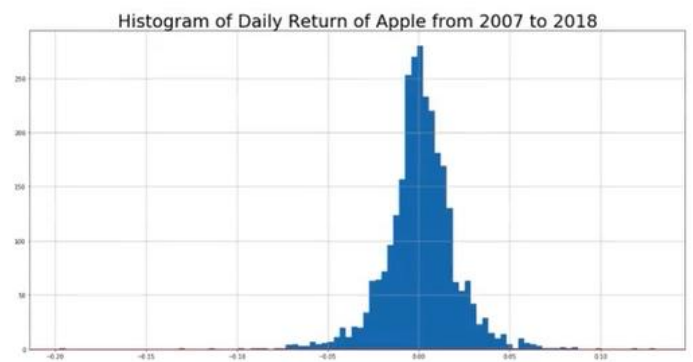
Out [6]

```
1.6448536269514722
At the significance level of 0.05
Shall we reject: True
```

From this result, we do need a quantitative statistic tool to validate our assertion in addition to visualize the data.







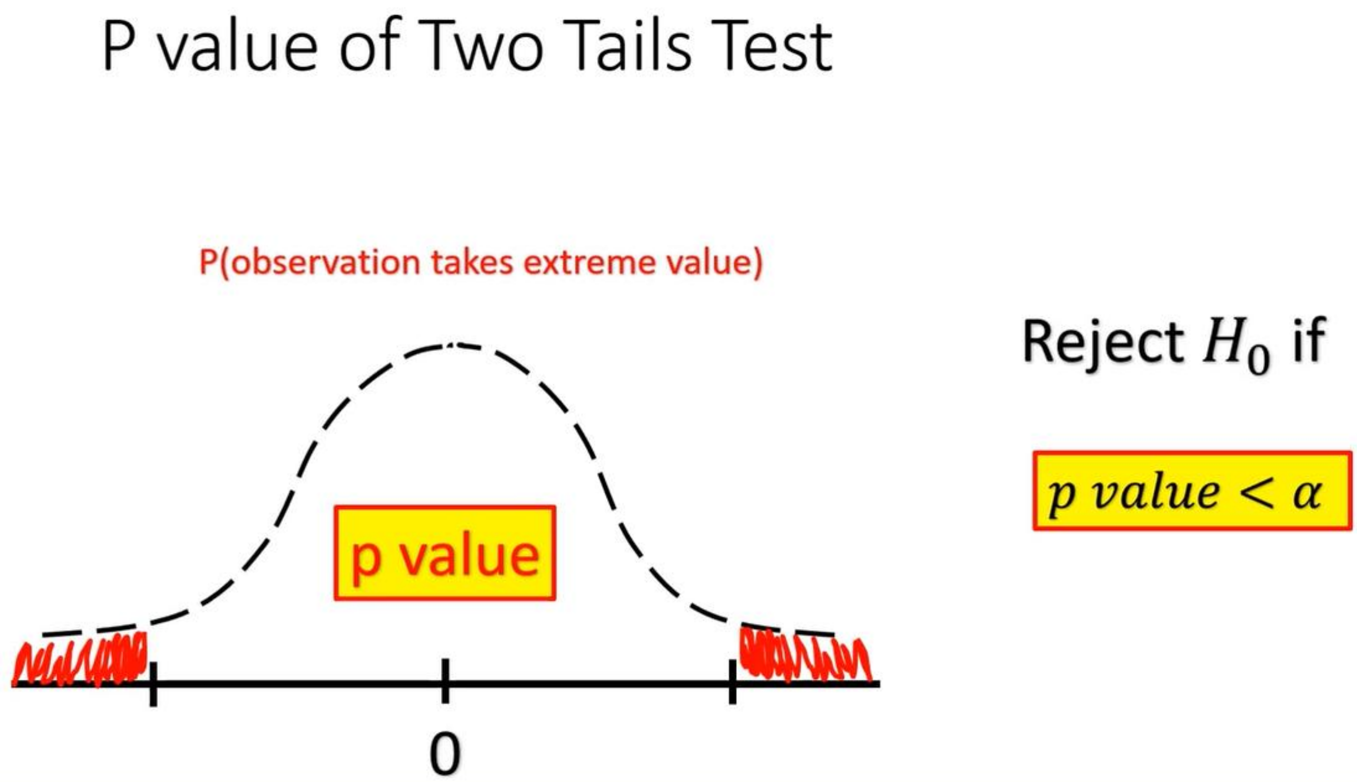
For population mean, we have these three kinds of hypothesis in the regression criteria.

Two Tails Test
H_0: $\mu = 0$
Reject if $\hat{z} > z_{\frac{\alpha}{2}}, or\ \hat{z} < -z_{\frac{\alpha}{2}}$

One Tail Test	
H_0: $\mu \leq 0$	H_0: $\mu \geq 0$
Reject if $\hat{z} > z_{\alpha}$	Reject if $\hat{z} < -z_{\alpha}$

p value

What is the probability for this distribution to take a more extreme value than our observation in given sample? This is a **p-value**, if p is less than alpha which is a threshold, it means that the null is unlikely to be true.



Here's a demonstration of p-value approach, in two-tailed test, **abs is to compute the absolute value**. We use **norm.cdf** to compute cumulative probability.

Calculate p value for Two tails Test in python

In [7]

```
alpha= 0.05
p= 1- (norm.cdf(abs(zhat),0, 1))
print('At the significance level of ', alpha)
print('Shall we reject: ', p< alpha)
```

Out [7]

```
At the significance level of 0.05
Shall we reject: True
```

P-value

- If Ha: mu not equal 0, it is two tail test and p-value=2(1-norm.cdf(np.abs(z), 0, 1))
- if Ha: mu>0, it is upper tail test and p-value=1-norm.cdf(z,0,1)
- if Ha:mu<0, it is lower tail test and p-value=norm.cdf(z,0,1)

上海立信会计金融学院
SHANGHAI LIXIN UNIVERSITY OF ACCOUNTING AND FINANCE

In next topic, we will use all knowledge we learned in topic two and three to explore relationship among different variables to **build a prediction model in stock market**. And finally, evaluate the performance of our models.

Return on investment

上海立信会计金融学院
SHANGHAI LIXIN UNIVERSITY OF ACCOUNTING AND FINANCE

Lab2：Hypothesis Testing

Instructions

- This Jupyter Notebook **testifies the claim** **whether the average daily return of Microsoft's stock is 0 or not**, base on the years of historical data available.

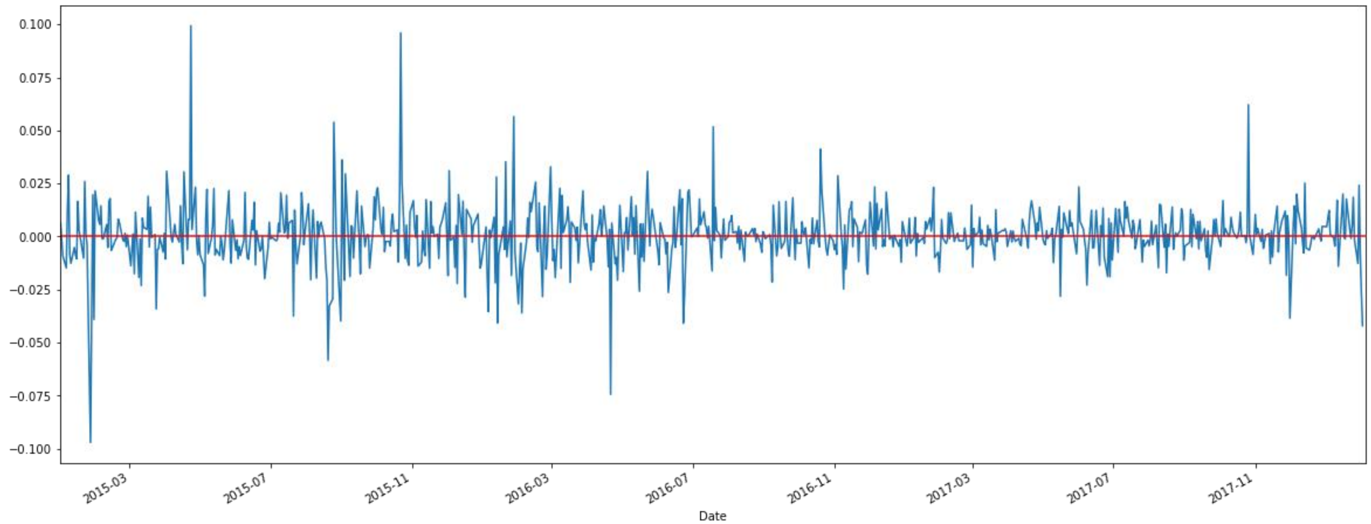
上海立信会计金融学院
SHANGHAI LIXIN UNIVERSITY OF ACCOUNTING AND FINANCE

Hypothesis testing

In [6]:
```
import pandas as pd
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt
% matplotlib inline
```

In [3]:
```
# import microsoft.csv, and add a new feature - logreturn
ms = pd.DataFrame.from_csv('../data/microsoft.csv')
ms['logReturn'] = np.log(ms['Close'].shift(-1)) - np.log(ms['Close'])
```

In [7]:
```
# Log return goes up and down during the period
ms['logReturn'].plot(figsize=(20, 8))
plt.axhline(0, color='red')
plt.show()
```



Steps involved in testing a claim by hypothesis testing

Step 1: Set hypothesis

$H_0 : \mu = 0$ $H_a : \mu \neq 0$

H0 means the average stock return is 0 H1 means the average stock return is not equal to 0

Step 2: Calculate test statistic

In [8]:
```
sample_mean = ms['logReturn'].mean()
sample_std = ms['logReturn'].std(ddof=1)
n = ms['logReturn'].shape[0]

# if sample size n is large enough, we can use z-distribution, instead of t-distribtuion
# mu = 0 under the null hypothesis
zhat = (sample_mean - 0)/(sample_std/n**0.5)
print(zhat)
```

```
1.6141477140003675
```

Step 3: Set desicion criteria

In [9]:
```
# confidence level
alpha = 0.05

zleft = norm.ppf(alpha/2, 0, 1)
zright = -zleft  # z-distribution is symmetric
print(zleft, zright)
```

```
-1.95996398454 1.95996398454
```

Step 4: Make decision - shall we reject H0?

In [10]:
```
print('At significant level of {}, shall we reject: {}'.format(alpha, zhat>zright or zhat<zleft))
```

```
At significant level of 0.05, shall we reject: False
```

Try one tail test by yourself !

$H_0 : \mu \leq 0 \; H_a : \mu > 0$

In [11]:
```
# step 2
sample_mean = ms['logReturn'].mean()
sample_std = ms['logReturn'].std(ddof=1)
n = ms['logReturn'].shape[0]

# if sample size n is large enough, we can use z-distribution, instead of t-distribtuion
# mu = 0 under the null hypothesis
zhat = None
print(zhat)
```

```
1.6141477140003675
```

Expected output: 1.6141477140003675

In [12]:
```
# step 3
alpha = 0.05

zright = norm.ppf(1-alpha, 0, 1)
print(zright)
```

```
1.64485362695
```

Expected output: 1.64485362695

In [13]:
```
# step 4
print('At significant level of {}, shall we reject: {}'.format(alpha, zhat>zright))
```

```
At significant level of 0.05, shall we reject: False
```

Expected output: At significant level of 0.05, shall we reject: False

An alternative method: p-value

In [14]:
```
# step 3 (p-value)
p = 1 - norm.cdf(zhat, 0, 1)
print(p)
```

```
0.053247694997
```

In [15]:
```
# step 4
print('At significant level of {}, shall we reject: {}'.format(alpha, p < alpha))
```

```
At significant level of 0.05, shall we reject: False
```

Hypothesis testing.ipynb在Github中下载

https://github.com/cloudy-sfu/QUN-Data-Analysis-in-Finance/tree/main/Labs

Jupyternote Book课堂练习

二十分钟

Thank You