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

March 2022

Random variables and distribution

In the previous module, we **built a simple trading strategy base on Moving Average 10 and 50**, which are "random variables" in statistics.

In this module, we are going to **explore basic concepts of random variables.**

By understanding the frequency and distribution of random variables, we extend further to the discussion of probability.

In the later part of the module, **we apply the probability concept in measuring the risk of investing a stock by looking at the distribution of log daily return using python**.

Learners are **expected to have basic knowledge of probability** before taking this module.

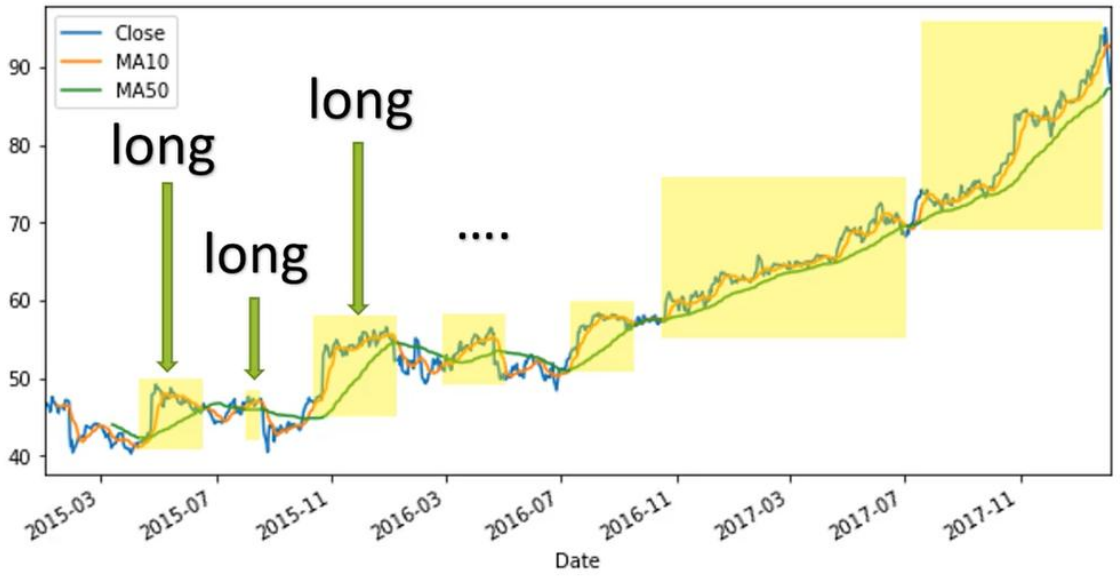
Learning Objectives

- Differentiate between outcome and variables by examples
- Categorize discrete and continuous random variables
- Explain the major reason of using "Relative Frequency" in comparing the distribution of random variables
- Conclude the distribution of random variables is close to the limit as number of trial increases
- Describe the probability distribution is similar to the random variable distribution of infinite trials
- Summarize mean and variable are used for describing the distribution of random variables
- Describe the main reason of using "Log Return" in measuing the risk of stock investment
- Use the distribution of log return to estimate the probability of losing a defined % of investment
- Evaluate the amount of an investment might lose at a certain probability by normal distribution quantiles
- Recall the kind of distribution of stock returns suggested by Fama and French

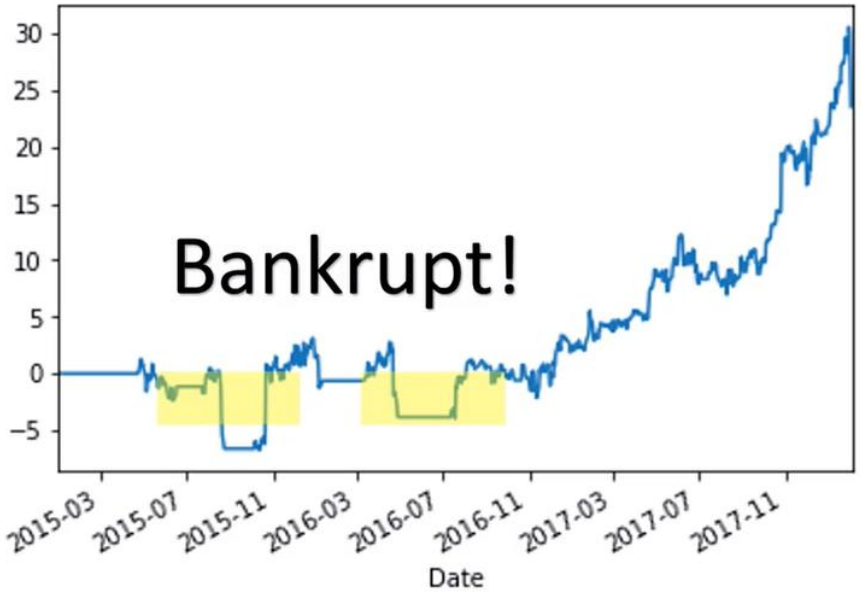
Module Introduction

In the previous topic, we primarily introduced the use of Python to import, read, and manipulate stock data by adding new features. **<u>One of the key new features</u>** we added is **a moving average**.

MA10 > MA50 "Long one share of stock"



There are **two points** whereby we lose money and financial analysis, **we will try our best to minimize the loss.**



What do we want to know is, **how to compute the chance of bankruptcy if I apply this strategy?**

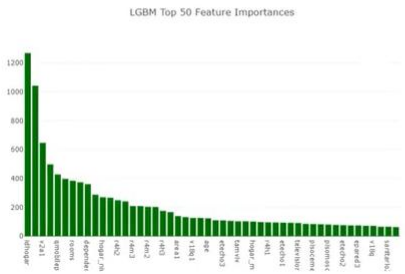
The simple trading strategy is built on two variables, moving average 10 and moving average 50.

In statistics, they are called random variables.

This is where we need to apply some statistical knowledge by asking, **what is a probability rule?** Or more formally speaking, **what is in distribution of these two random variables, MA10 and MA50?**

Identify important variables in other contexts

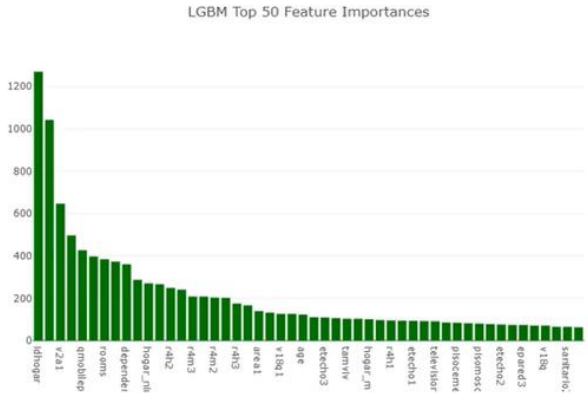
Helping us **making better prediction and decisions**, not just in financial, but in other contexts as well.

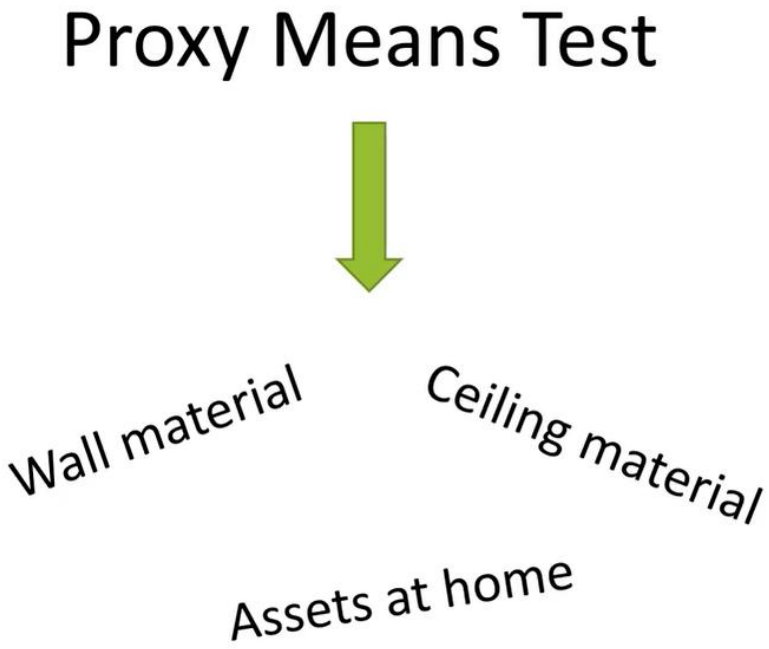


For example, many social programs have a hard time making sure that right people are given enough aid. It's especially tricky where in program focus on the poorest segment of the population. **The world's poorest typically cannot provide the necessary income and expense records to prove that they are qualified.**

Enough aid
for the right people?

In Latin America, **one popular method** to verify income qualification is called the **Proxy Means Test** or **PMT**.





PMT identifies new variables in the model, which are family observable **household attributes** like **the material of their walls**, and **the ceilings** or **the assets found in the home** to qualify them and predict their level of need.

There is also another success story in small lending industry.

Many people **struggle to get loans** due to insufficient and non-existing credit histories.

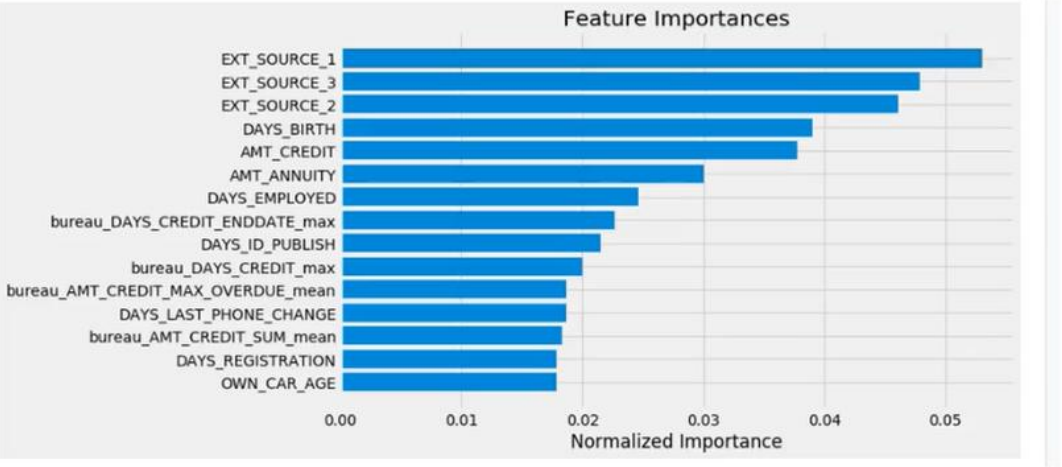
Unfortunately, this population is often taken advantage of by untrustworthy lenders.

Insufficient/Non-existing credit history

There is a company called **Home Credit**, who **makes use of variety and alternative external variables**, including **telecom company bills** and **other transnational information** to predict their clients' repayment abilities.

Home Credit

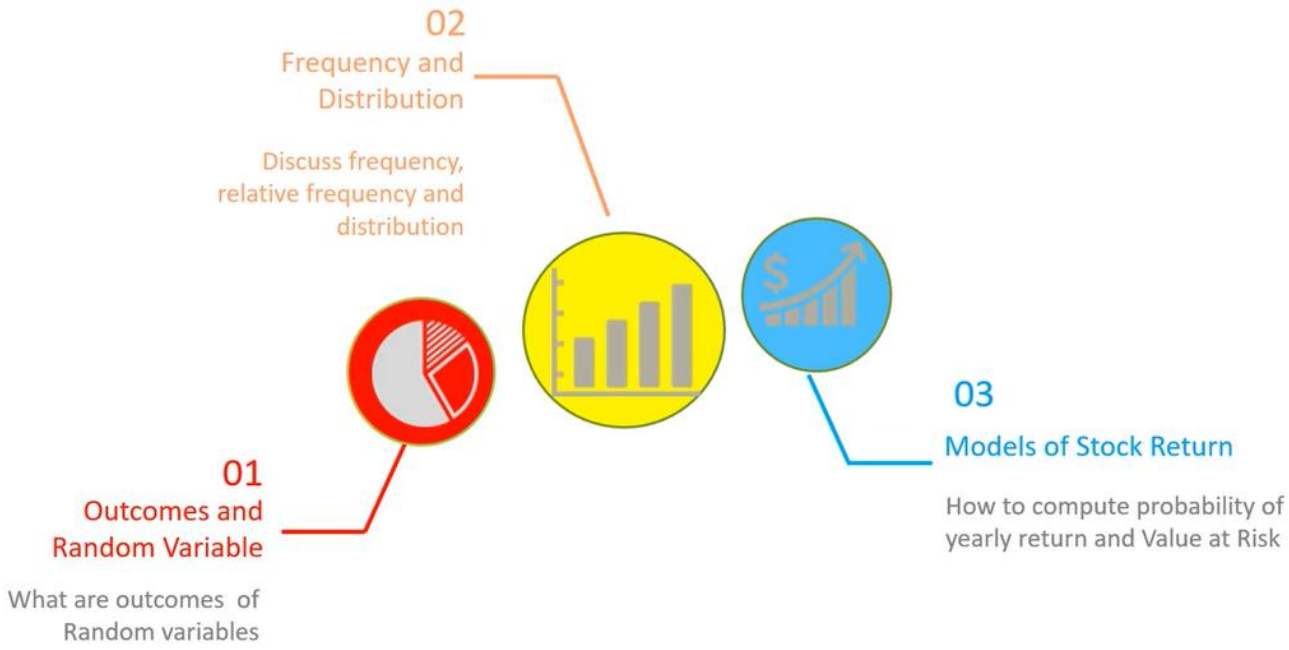
Variety of external variables



These new variables tend to be very important in new prediction model.

This example gave us enough incentives to explore **some basic concepts and facts about random variables**.

We will explore this topic in three parts. **First**, we will explain, *what random variable is*? In the **second part**, we will *describe the distribution of random variables*, distribution helps identify extreme values of events. **For example, it is used for risk management in financial context.** **After** knowing the distribution random variables, we will *apply this concept to measure the risk of investing money in Apple stock*.



01
Outcomes and Random Variable

What are outcomes of Random variables

In this part, we will **discuss** what is a random variable, the outcome and different types of random variable.

Let us start with a very simple game, we roll two dice.

In this game, we will compute and record **the sum of face values of two dice**.

Dice game

Sum of faces

Using Python, we can mimic this game.

Rolling dice in python

two means we are rolling two dice.

```
In [1]    die = pd.DataFrame([1,2,3,4,5,6])

In [2]    sum_of_dice = die.sample(2,replace=True).sum().loc[0]
          print('Sum of dice is ', sum_of_dice)

Out [2]   Sum of dice is 8
```

If we roll two dice **50 times**, we can get 50 observation or realize outcomes of sum

Rolling dice in python – 50 times

```
In [3]  trial = 50
        result = [die.sample(2,replace=True).sum().loc[0] for i in range(trial)]
        result[:10]
```

```
Out [3]  [5, 9, 5, 5, 9, 4, 5, 2, 7, 9]
```

Here, we print out first 10 observation of 50 trials.

Outcomes

In this game, **the sum**, which we **denote as X**, is **a random variable.**

Before we roll a die, we are not sure which outcome we can get. **That is called Randomness of variable.**

But we know the collection of outcomes, which ranged from 2 to 12.

The collection of outcomes on the left **is not a random variable**.

From 2 to 12....

[5, 9, 5, 5, 9, 4, 5, 2, 7, 9]

X: The sum of faces

Outcomes

Random Variable

In the example above, **the possible outcomes include integers from 2 to 12.** So, it is called **Discrete random variable**.

There is another type of variable, for example, **daily return of a stock price**. It can take **any real values**. Hence, it is called **continuous random variable**.

Random Variable

Discrete

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

Continuous

1.5% -4.558%
-8.8756% 13.48%
...

Lab1： Outcomes and Random Variables

Instructions

- In this Jupyter Notebook, you are going to mimic the roll dice game, and also the result of rolling the dice for any number of times.
- This game is to illustrate the concept of "randomness" of a variable.

Outcomes and Variables

In [1]:
```
#import numpy and pandas package
import numpy as np
import pandas as pd
```

Mimic the roll dice game

In [7]:
```
# roll two dice for multiple times
die = pd.DataFrame([1, 2, 3, 4, 5, 6])
sum_of_dice = die.sample(2, replace=True).sum().loc[0]
print('Sum of dice is', sum_of_dice)

# you may get different outcomes as we now mimic the result of rolling 2 dice, but the range must be limited between 2 and 12.
```

```
Sum of dice is 6
```

In [5]:
```
# It is your turn! let's replace the none with the code of rolling three dice, instead of two

np.random.seed(1)  # This is for checking answer, do NOT modify this line of code

#Modify the code, replace the None
sum_of_three_dice = None
```

In [4]:
```
print('Sum of three dice is', sum_of_three_dice)
```

```
Sum of three dice is  15
```

Expected output: Sum of three dice is 15

Mimic the roll dice game for multiple times ¶

In [5]:
```
# The following code mimics the roll dice game for 50 times. And the results are all stored into "Result"
# Lets try and get the results of 50 sum of faces.

trial = 50
result = [die.sample(2, replace=True).sum().loc[0] for i in range(trial)]
```

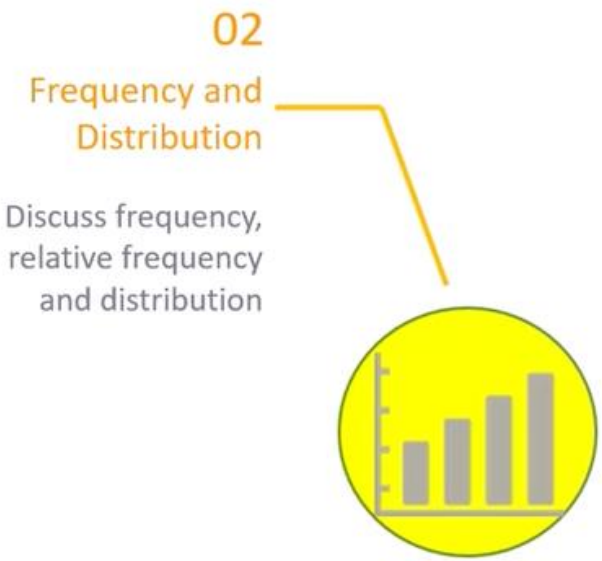
In [6]:
```
#print the first 10 results
print(result[:10])
```

```
[3, 10, 2, 7, 11, 5, 11, 8, 9, 8]
```

Outcomes and Random Variables.ipynb在Github中下载

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

Jupyternote Book课堂练习十五分钟



In this lecture, we will **discuss** **frequency**, **relative frequency of observed outcomes**, and introduce **the concept of distribution**.

To recall, this is a dice game from the last lecture. **Sum is a random variable**, the print out is the 10 observed outcomes. Totally, **we tried 50 times and get 50 realized outcomes of sum.**

Rolling dice in python

```
In [1]    die = pd.DataFrame([1,2,3,4,5,6])
          trial = 50
          results = [die.sample(2,replace=True).sum().loc[0] for i in range(trial)]
          results[:10]
```

```
Out [2]   [5, 9, 5, 5, 9, 4, 5, 2, 7, 9]
```

Outcomes

Now, we want to calculate the frequency in this collection of outcomes. There is a very useful **method of Data-Frame called value count**, it will **output pandas series**, which **has only one column if compared with dataFrame**.

Frequency

```
In [4]:  freq = pd.DataFrame(results)[0].value_counts() # count values
         sort_freq = freq.sort_index()  # sort index
         sort_freq
```

```
Out [4]:
2     1
3     4
4     4
5     8
6     7
7     5
8     8
9     6
10    4
11    3
Name: 0, dtype: int64
```

List of different outcomes

Frequency

Its index is a list of different outcomes and the value column, this is the frequency. But even can sort these series according to the index using sort index. For example, in our output, the first row says, the frequency for an outcome two is equal to one.

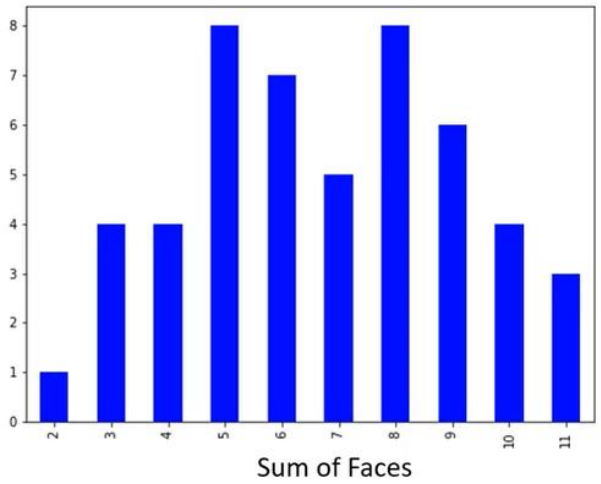
We can **plot frequency using bar chart**. And frequency will change as the number of trials changes.

If we want to compare the frequency of different trials, we have to convert frequency into relative frequency. **Relative frequency equal to frequency divided by number of trials.**

Frequency

```
In [5]  sort_freq.plot(kind='bar', color='blue')
```

Out [5]



Relative frequency

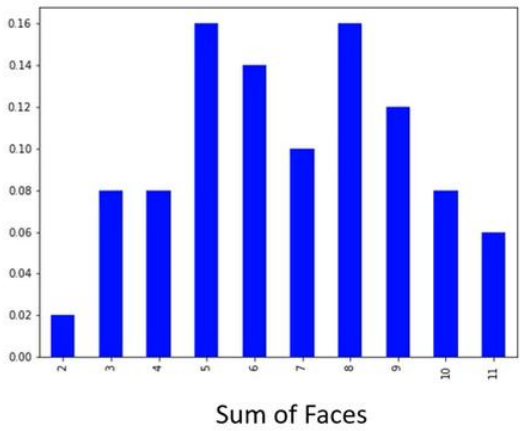
$$\frac{\textit{Frequency}}{\textit{No. of trials}}$$

With relative frequency, **the shape of bar chart does not change**. The scale of Y axis changes.

Relative frequency

In [6]
```
relative_freq=sort_freq/trial
relative_freq.plot(kind='bar', color='blue')
```

Out [6]



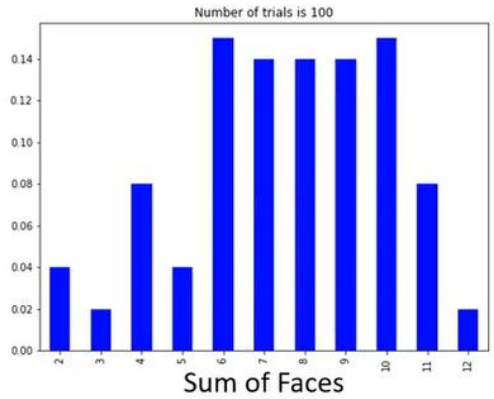
This is a bar chart showing the frequency for the outcomes of 100 trials.

Relative frequency

In [7]

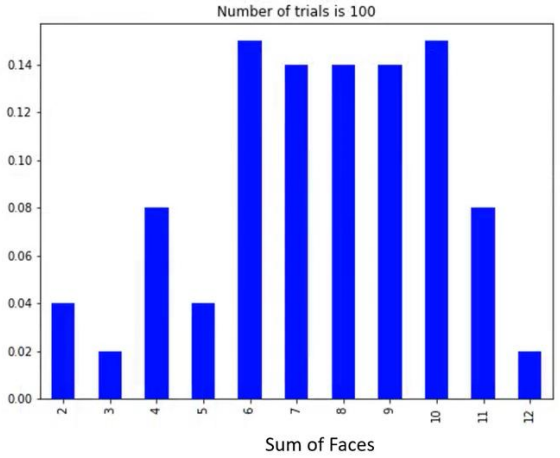
```
trial = 100
result = [die.sample(2,replace=True).sum().loc[0] for i in range(trial)]
freq = pd.DataFrame(results)[0].value_counts() # count values
sort_freq = freq.sort_index()  # sort index
relative_freq=sort_freq/sort_freq.sum()
relative_freq.plot(kind='bar', color='blue')
```

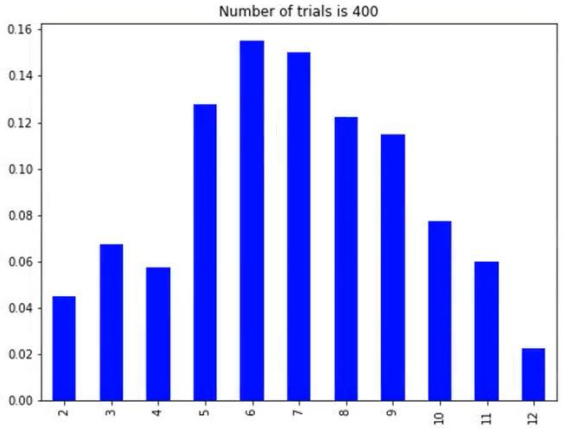
Out [7]

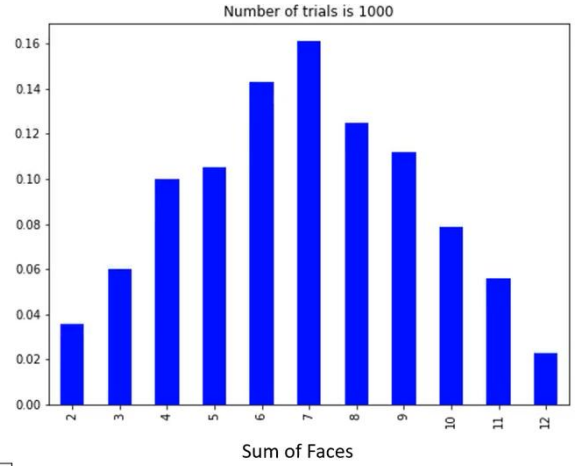


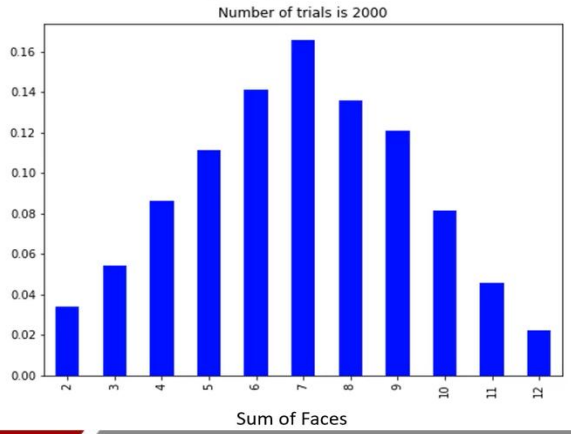
As we increase the numbers of trials for example, we start with 100 trials.

This is the one with the **400 trials**, **a 800 here**, **1000 trials** in this chart, **2000 trials** in this chart. The bar chart goes toward a limit. **The relative frequency become more and more stable as you increase the number of trials.**





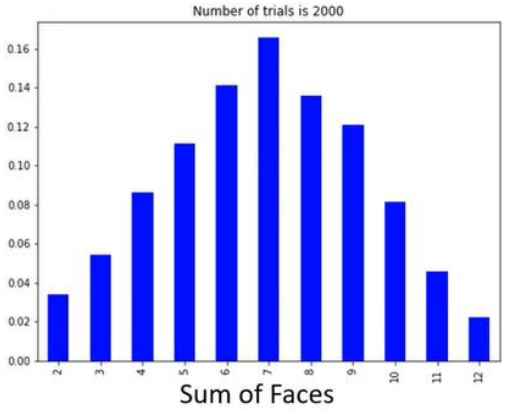




What could be the limit if we have an infinite number of trials? Distribution of a random variable is a table consists of two sets of values.

Infinite number of trials?

Limit: Distribution of sum of face X



One for different values of outcome, the other list the probability for each value.

Distribution table

X	2	3	. . .	12
Probability	P(X=2)	P(X=3)	. . .	P(X=12)

We can compute all probability for X using python here.

X=	2	3	4	5	6	7	8	9	10	11	12
P(X)	$1\times\left(\frac{1}{6}\right)^2$	$2\times\left(\frac{1}{6}\right)^2$	$3\times\left(\frac{1}{6}\right)^2$	$4\times\left(\frac{1}{6}\right)^2$	$5\times\left(\frac{1}{6}\right)^2$	$6\times\left(\frac{1}{6}\right)^2$	$5\times\left(\frac{1}{6}\right)^2$	$4\times\left(\frac{1}{6}\right)^2$	$3\times\left(\frac{1}{6}\right)^2$	$2\times\left(\frac{1}{6}\right)^2$	$1\times\left(\frac{1}{6}\right)^2$

```
In [6]  X_distri = pd.DataFrame(index=[2,3,4,5,6,7,8,9,10,11,12])
        X_distri['Prob'] = [1,2,3,4,5,6,5,4,3,2,1]
        X_distri['Prob'] = X_distri['Prob']/36
```

Usually, **mean** and the **variance** are two characteristics of the distribution of random variables. Mean of a random variable is also called **Expectation**.

Mean and Variance of a distribution

```
In [9]  Mean=(X_distri.index*X_distri['Prob']).sum()
        Var=(((X_distri.index-Mean)**2)*X_distri['Prob']).sum()
        print(Mean, Var)

Out [9] (6.999999999999998, 5.833333333333333)
```

$$Mean\ (or\ Expectation) = \sum_i p_i x_i$$

$$Variance = \sum_i (x_i - Mean)^2 p_i$$

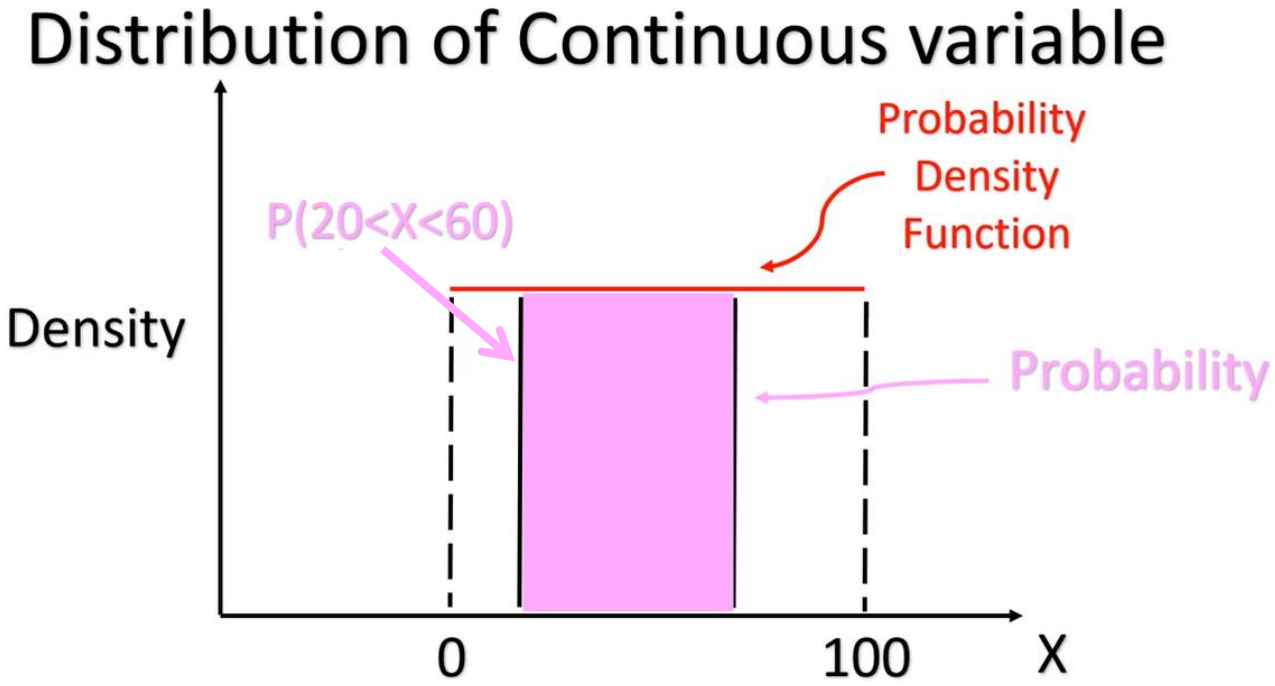
What is the distribution for continuous random variables? We'll compute the probability for continuous random variable.

Continuous Random Variable

How to calculate the probability of stock return?

What is the distribution of continuous random variables?

We will start with the simplest continuous random variable, which has a **uniform distribution.**

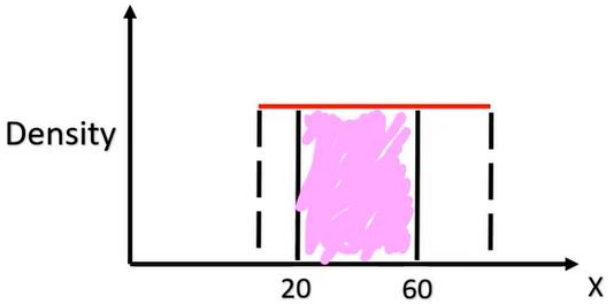


Summary

Discrete

X	2	3	. . .	12
Probability	P(X=2)	P(X=3)	. . .	P(X=12)

Continuous



Note that PDF is not probability.

Now, **let's come back to finance questions.** Why we need a continuous random variable? **Because the distribution of stock data return is continuous**. We know that real distribution stock return cannot be directly observed.

Approximated by
normal random variable

Do we have any good distributions that can describe the the data return reasonably good?

Most popular continuous random variable to approximate distribution of stock return.

Lab2:Frequency and Distributions

Instructions

In this Jupyter Notebook, we go further the distribution of the roll dice game by **plotting its frequency distribution.**

Using python, you can realize that with more trials, the result looks more and more stable, and this is very close to a probability distribution table.

And you will also realize that you can **use mean and variance to describe a distribution.**

Frequency and Distribution

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

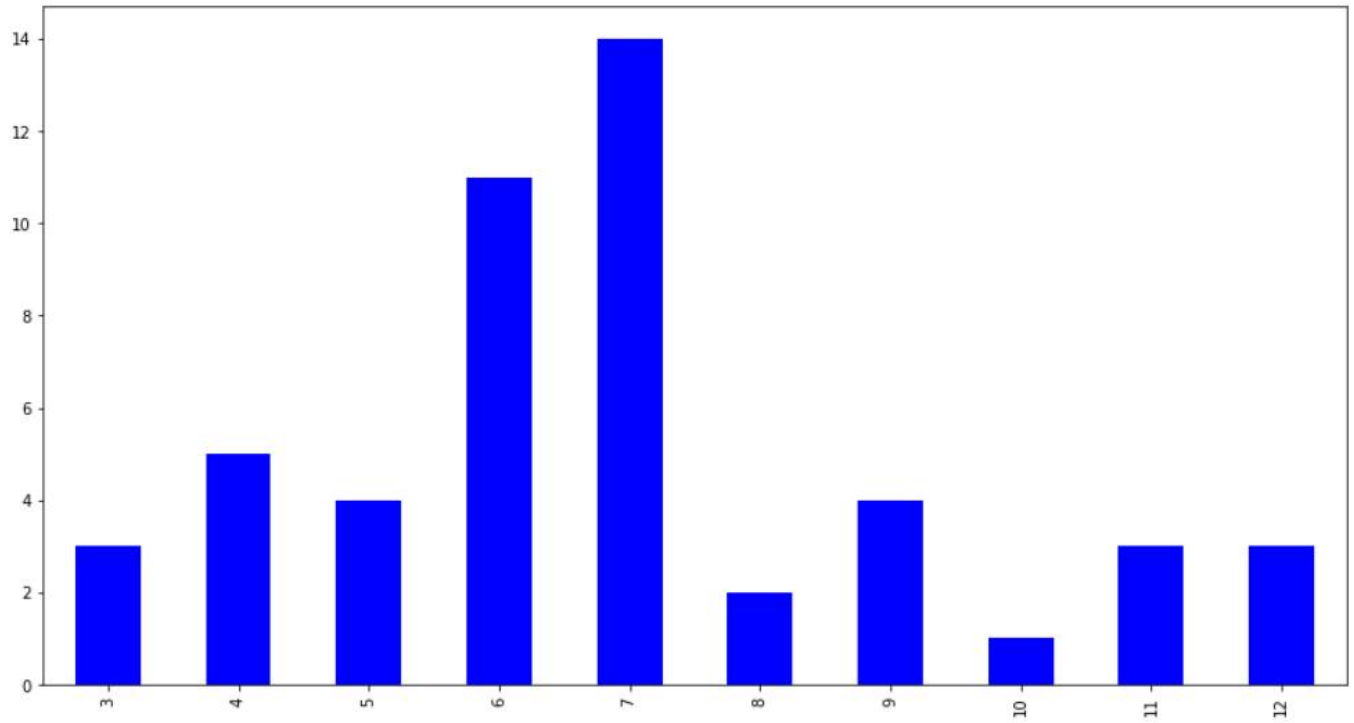
In [2]:
```
# To recall, this is the code to mimic the roll dice game for 50 times

die = pd.DataFrame([1, 2, 3, 4, 5, 6])
trial = 50
results = [die.sample(2, replace=True).sum().loc[0] for i in range(trial)]
```

In [3]:
```
# This is the code for summarizing the results of sum of faces by frequency

freq = pd.DataFrame(results)[0].value_counts()
sort_freq = freq.sort_index()
print(sort_freq)
```

```
3      3
4      5
5      4
6     11
7     14
8      2
9      4
10     1
11     3
12     3
Name: 0, dtype: int64
```

In [4]:
```
#plot the bar chart base on the result

sort_freq.plot(kind='bar', color='blue', figsize=(15, 8))
```

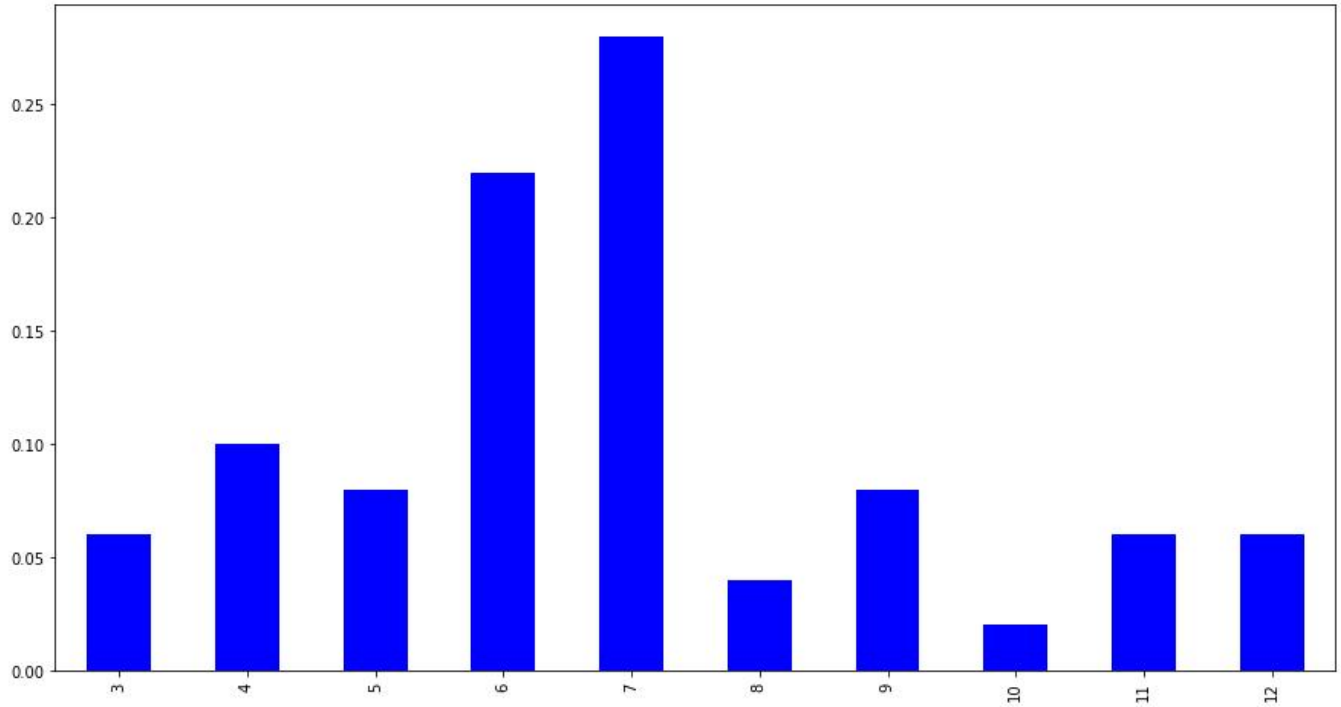
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7efd2df16e80>



Relative Frequency

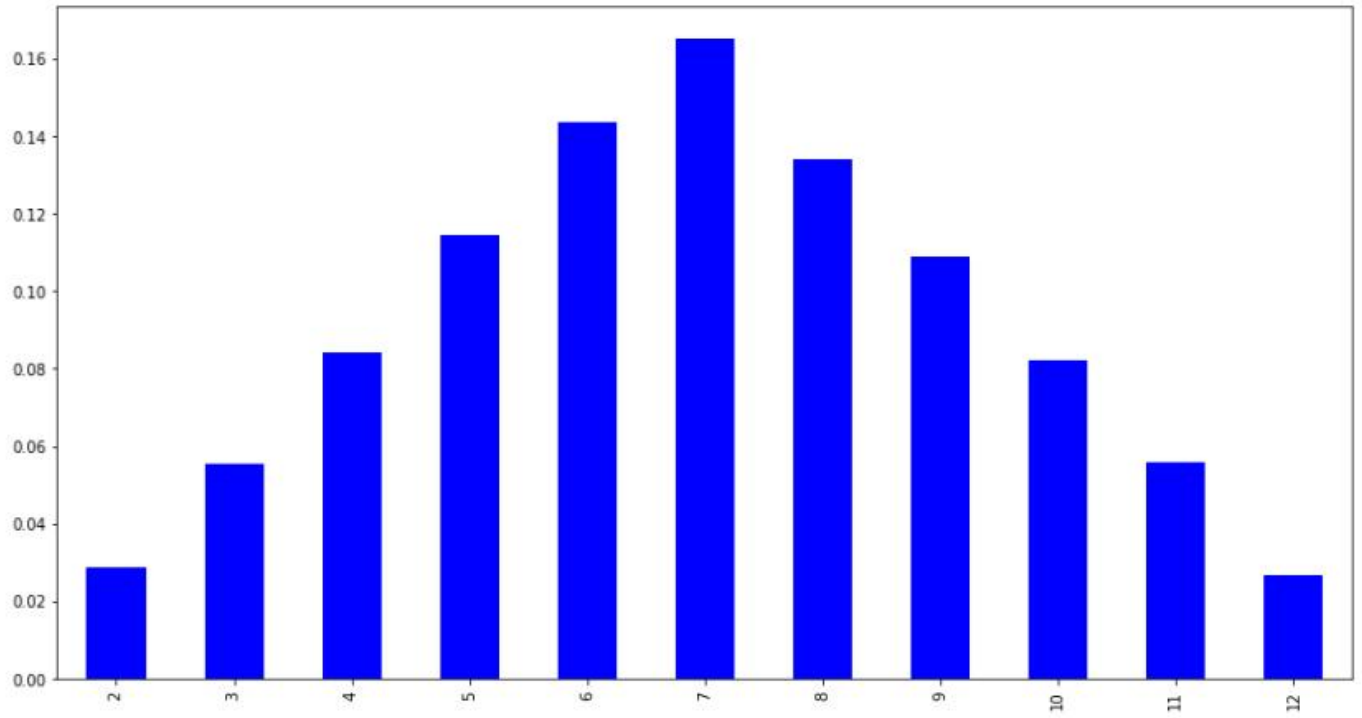
In [5]:
```
# Using relative frequency, we can rescale the frequency so that we can compare results from different number of trials
relative_freq = sort_freq/trial
relative_freq.plot(kind='bar', color='blue', figsize=(15, 8))
```

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x7efd2dbdabe0>



In [6]:
```
# Let us try to increase the number of trials to 10000, and see what will happen...
trial = 10000
results = [die.sample(2, replace=True).sum().loc[0] for i in range(trial)]
freq = pd.DataFrame(results)[0].value_counts()
sort_freq = freq.sort_index()
relative_freq = sort_freq/trial
relative_freq.plot(kind='bar', color='blue', figsize=(15, 8))
```

Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x7efd2dc84828>



We can see that with more trials, the result looks more and more stable, and this is very close to a probability distribution. Try increasing the number of "trial" further (but it may take some time for Jupyter Notebook to output the result)

Expectation and Variance of a distribution

In [15]:
```
# assume that we have fair dice, which means all faces will be shown with equal probability
# then we can say we know the 'Distribtuion' of the random variable - sum_of_dice

X_distri = pd.DataFrame(index=[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
X_distri['Prob'] = [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1]
X_distri['Prob'] = X_distri['Prob']/36
X_distri
```

Out[15]:

	Prob
2	0.027778
3	0.055556
4	0.083333
5	0.111111
6	0.138889
7	0.166667
8	0.138889
9	0.111111
10	0.083333
11	0.055556
12	0.027778

In [20]:
```
mean = pd.Series(X_distri.index * X_distri['Prob']).sum()
var = pd.Series(((X_distri.index - mean)**2)*X_distri['Prob']).sum()
```

In [21]:
```
#Output the mean and variance of the distribution. Mean and variance can be used to describe a distribution
print(mean, var)
```

```
6.999999999999999 5.833333333333334
```

Empirical mean and variance

In [22]:
```
# if we calculate mean and variance of outcomes (with high enough number of trials, eg 20000)...
trial = 20000
results = [die.sample(2, replace=True).sum().loc[0] for i in range(trial)]
```

In [23]:
```
#print the mean and variance of the 20000 trials
results = pd.Series(results)
print(results.mean(), results.var())
```

```
6.99505 5.864618728436524
```

Frequency and Distribution.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

03

Models of Stock Return

How to compute probability of yearly return and Value at Risk

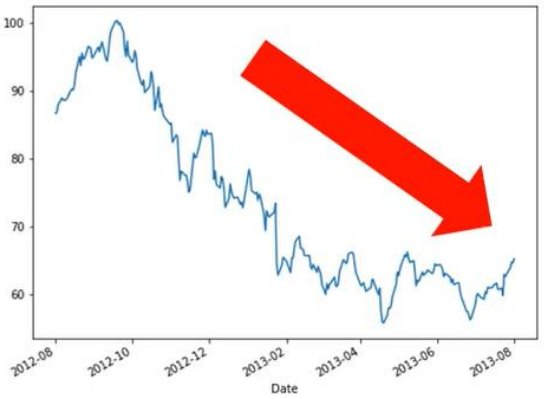
we will model stock return using **normal random variable** and **demonstrate the importance of distribution in identifying financial risk**.

Why it is **important** to know **the distribution or model for stock return**. It is really **crucial** in risk management.

Big drop in Apple's stock price

In [1]
```
aapl =pd.DataFrame.from_csv("data/apple.csv")
aapl.loc['2012-8-01':'2013-8-01','Close'].plot()
```

Out [1]



Probability of dropping over 40%?

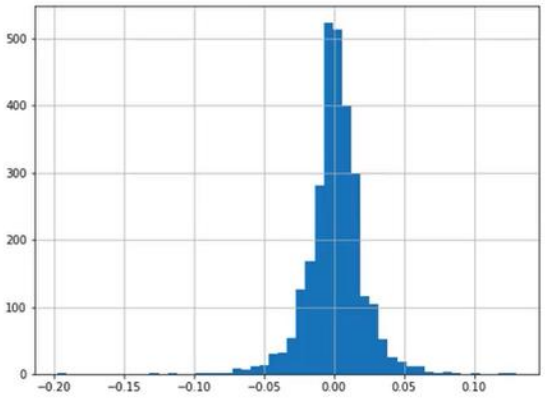
We need to compute what's the chance that the **yearly return can be less than negative 40%**. **Is that possible, or just an extreme case like black swan.**

We compute **log daily return for stock price of Apple**. The histogram of a return is symmetric and bell shaped, which is very similar to normal distribution.

Log daily return of Apple

In [1]
```
aapl['LogReturn'] = np.log(aapl['Close']).shift(-1) - np.log(aapl['Close'])
aapl['LogReturn'].hist(bins=50)
```

Out [1]



≈Normal distribution

Using scipy which is a scientific computation package of python, we can **get density function and a cumulative distribution function.**

norm.pdf where a given density for each possible value of a normal random variable.

This normal random variable is also called **a standard normal random variable**, and its distribution is also called **z-distribution**.

```
In [3]  from scipy.stats import norm

In [4]  density =pd.DataFrame()
        density['x'] =np.arange(-4,4,0.001)
        density['pdf'] =norm.pdf(density['x'],0,1)   # get pdf
        density['cdf'] =norm.cdf(density['x'],0,1)   # get cdf
```

Probability Density Function

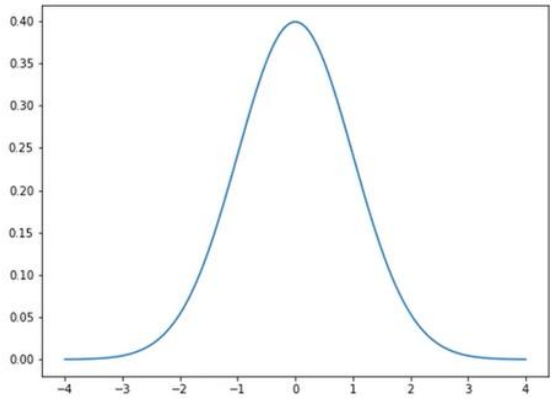
Cumulative Distribtuion Function

We also can get a cumulative distribution function, or in short a CDF which outputs the probability for the area, and lower side of each possible value.

PDF

```
In [3] plt.plot(density['x'],density['pdf'])
```

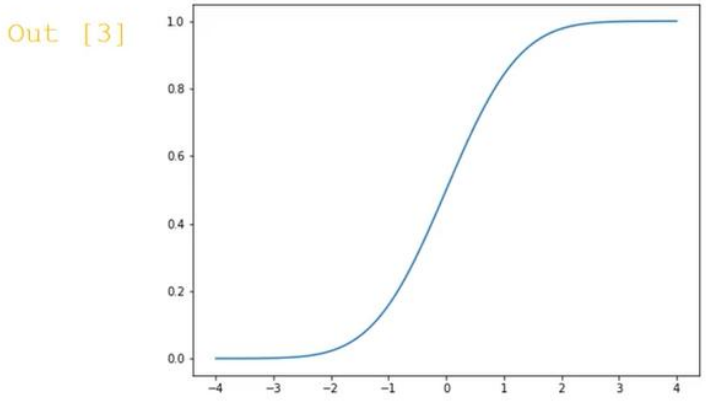
Out [3]



CDF

```
In [3] plt.plot(density['x'],density['cdf'])
```

Out [3]



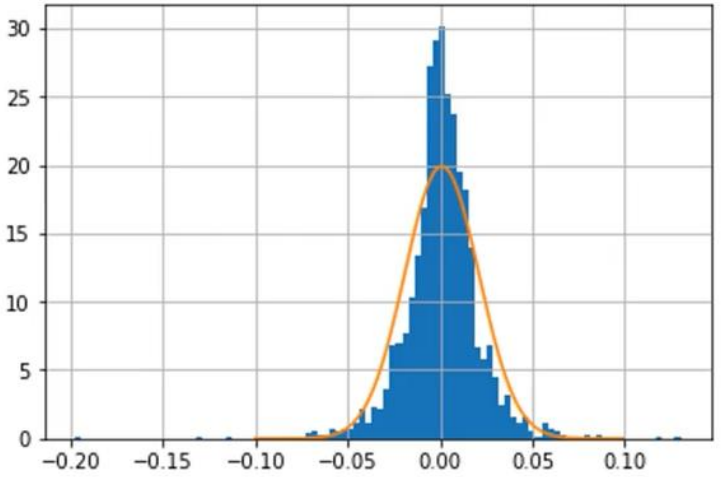
We can model daily stock return using normal distribution.

We have a large collection of data return from historic data. **We can compute the mean and the standard deviation in this collection.**

Approximate mean and variance of the log daily return

```
mu = aapl['LogReturn'].mean()
sigma = aapl['LogReturn'].std(ddof=1)
print(mu, sigma)
```

```
0.00097546775 0.0200454476
```



Then, **what is the chance data loss can be more than 5%?**

What is the chance of losing over 5% in a day?

In [1]

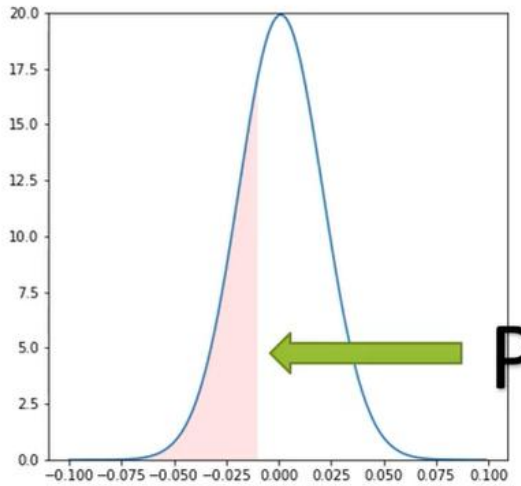
```
denApp =pd.DataFrame()
denApp['x'] =np.arange(-0.1,0.1,0.001)
denApp['pdf'] =norm.pdf(denApp['x'],mu, sigma)
```

mean, standard deviation

This pink area is the probability of losing more than 5% in one day.

In [1]

```
plt.ylim(0,20)
plt.plot(denApp['x'],denApp['pdf'])
plt.fill_between(x=np.arange(-0.1,-0.01,0.0001),
                 y2=0,
                 y1= norm.pdf(np.arange(-0.1, 0.05,0.0001),mu,sigma),
                 facecolor='pink',
                 alpha=0.5)
```



P(Daily return < -5%)

Hence we have 0.5% chance to have a daily loss more than 5%.

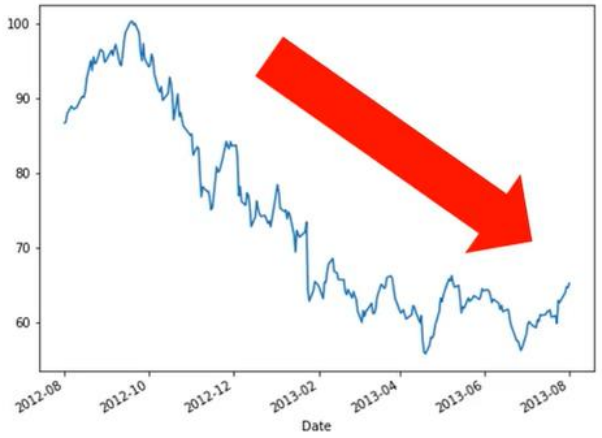
In [1]
```
prob_return1 =norm.cdf(-0.05,mu,sigma)
print('The probability  is ', prob_return1)
```

Out [1]
```
The probability is 0.00549534425096
```

P(Daily return < -5%)

Our **goal** is **to find out how likely that the stock price of Apple were dropped over 40%, 1 year which has 220 trading days.** We will use another normal distribution to model yearly return. We need to figure out the mean and the standard deviation of a yearly return.

How about probability of dropping over 40% in 1 year (220 trading days)?



P(Drop over 40% in 220 days)

We **make assumption that the daily returns are independent**, **which is quite wrong.** But it can simplify our discussion here, to get the mean and variance of a year return. **We have formulas for sum of variables**. We need the independence when we compute the variance. ***If the data returns independent the variance of a unit return is equal to the sum of a variance of 220 daily return.***

Sum of independent normal random variables

mean $\mu_{X_1+X_2+X_3+\ldots+X_n} = \mu_{X_1} + \mu_{X_2} + \mu_{X_3} + \ldots + \mu_{X_n}$

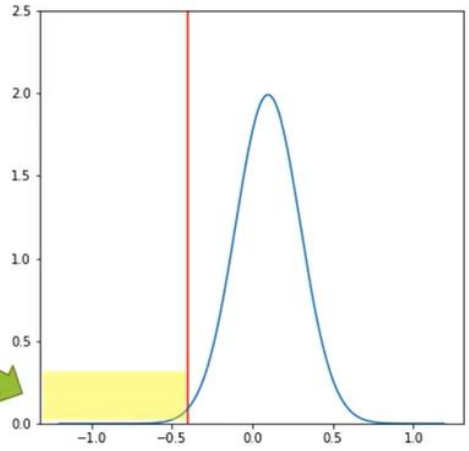
variance $\sigma^2_{X_1+X_2+X_3\ldots+X_n} = \sigma^2_{X_1} + \sigma^2_{X_2} + \sigma^2_{X_3} \ldots\ldots + \sigma^2_{X_n}$

Again, with no CDF we compute a probability, and it is less than 2%. That implies, **there's only less than 2% chance to have yearly loss to be more than 40%.**

```
mu220 = 220*mu
sigma220 = 220**0.5*sigma
print(mu220, sigma220)

print('The probability of dropping over 40% in 220 days is ',
norm.cdf(-0.4, mu220, sigma220)
```

```
0.21460290701301937, 0.29732203656371786
The probability of dropping over 40% in 220 days is
0.019361015454142632
```

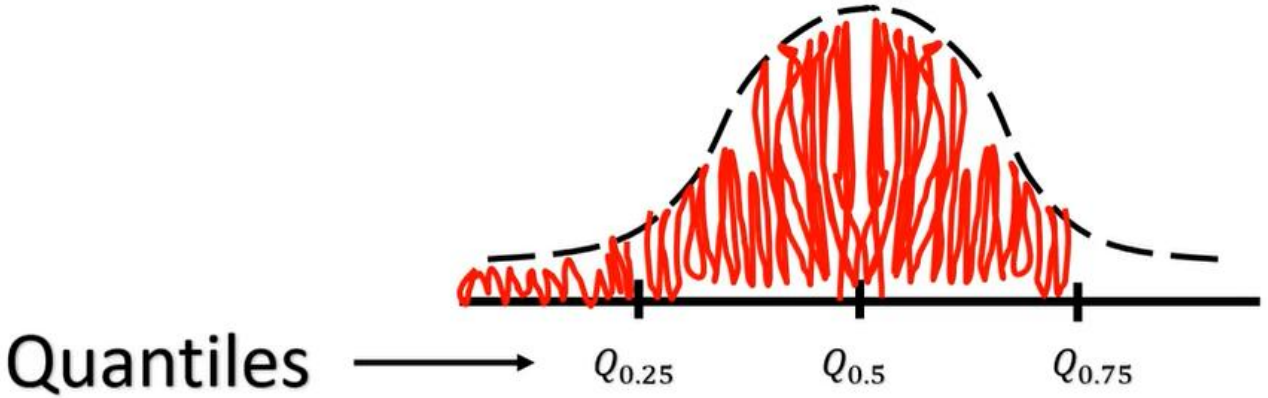
Probability of dropping over 40%



What happened to Apple in 2012 and 2013? Is not consistent with its overall performance. In many circumstances, we need to solve a different type of problem with distribution.

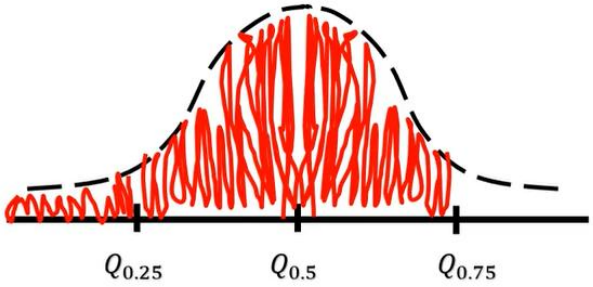
For example, **finding quantiles of a normal distribution is a common task when performing statistical test in the financial risk management.**

Normal distribution quantiles can obtain using **norm.ppf**. Ppf stands for percent point function.



In finance related to the quantile there is **an important risk measure value of risk** or **VaR.**

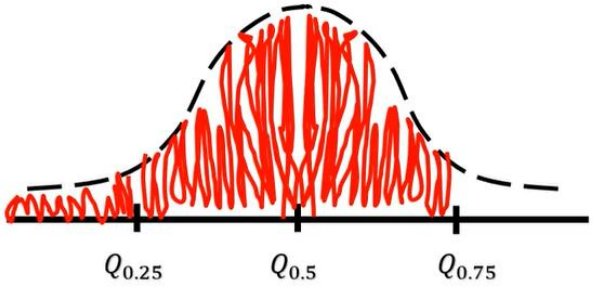
It estimates how much a set of investments might lose with a given probability. **VaR is typically used by firms and the regulators in the financial industry to gauge the amount of assess needed to cover possible loss.**



Value at Risk

Measure how much a set of investment might lose

VaR是指在某个置信区间下投资组合在未来某段时间的最大可能损失，从数学上看它衡量的是投资组合损益分布的分位数，假设C是我们选择的置信区间，则VaR对应于损益分布的下尾处。例如置信区间为95%，则**VaR等于损益分布函数的5%处分位数**。



Value at Risk

Measure how much a set of investment might lose

For example, **5% of quantile of daily return is called a 95% VaR or VaR at the level of 95%.** We can use a **ppf** to get a 5% of quantile which is negative 0.03.

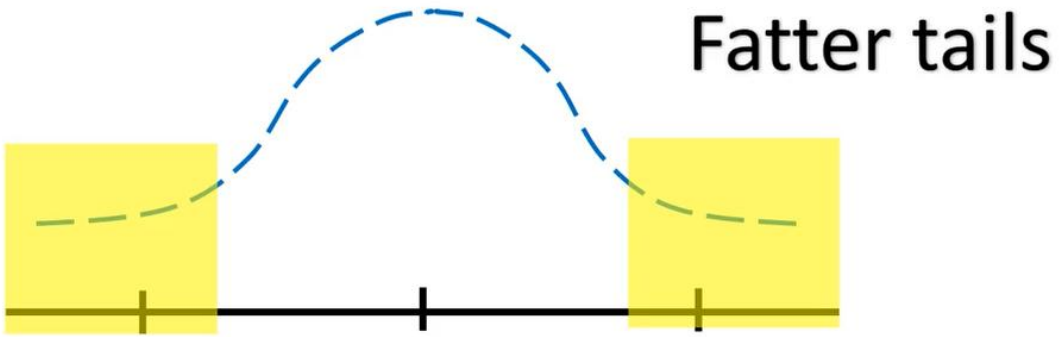
Value at Risk (VaR)

```
In [2]   norm.ppf(0.05,mu,sigma)

Out [2]  -0.031996359455654697
```

Is it safe to use normal description to model stock return? Two famous professors in the field asset pricing, Fama and French responds in this way.

Is it safe to use Normal Distribution?

> '' Distributions of daily and monthly stock returns are rather symmetric about their means, but the tails are fatter (i.e., there are more outliers) than would be expected with normal distributions. ''
>
> Fama and French



It means that, if tail returns negative, as well as positive, may occur more often than we expect. If we use normal distribution, this is debatable, at least for the returns of some assets with different time window size. **To modal a fat tail, people proposed modal return using t-distributions with low degree of freedom.**

Lab3： Models of stock return

Instructions

- In this Jupyter Notebook, you are going to **apply the concept of probability to measure the probability that the stock price drops a certain percentage in a day, and in a year.**

- We also demonstrate **how to calculate the value at risk (VaR) using python**.

Models of Stock Return

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
% matplotlib inline
```

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

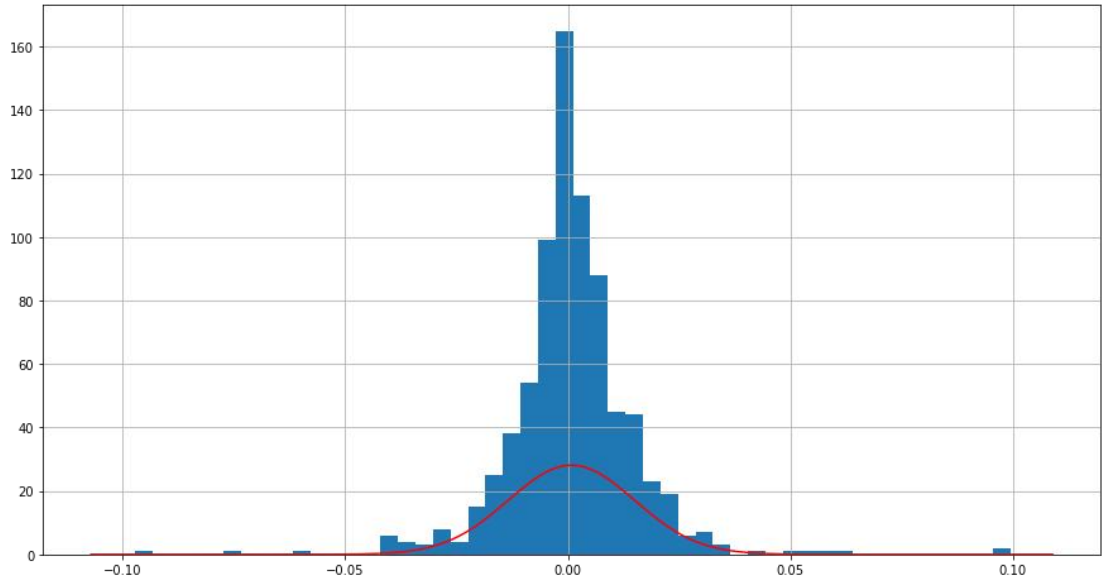
Out[2]:

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

Distribution of Log return

In [3]:
```
# let play around with ms data by calculating the log daily return
ms['LogReturn'] = np.log(ms['Close']).shift(-1) - np.log(ms['Close'])
```

In [4]:
```
# Plot a histogram to show the distribution of log return of Microsoft's stock.
# You can see it is very close to a normal distribution
from scipy.stats import norm
mu = ms['LogReturn'].mean()
sigma = ms['LogReturn'].std(ddof=1)

density = pd.DataFrame()
density['x'] = np.arange(ms['LogReturn'].min()-0.01, ms['LogReturn'].max()+0.01, 0.001)
density['pdf'] = norm.pdf(density['x'], mu, sigma)

ms['LogReturn'].hist(bins=50, figsize=(15, 8))
plt.plot(density['x'], density['pdf'], color='red')
plt.show()
```



Calculate the probability of the stock price will drop over a certain percentage in a day

In [5]:
```
# probability that the stock price of microsoft will drop over 5% in a day
prob_return1 = norm.cdf(-0.05, mu, sigma)
print('The Probability is ', prob_return1)
```

```
The Probability is  0.000171184826087
```

In [6]:
```
# Now is your turn, calculate the probability that the stock price of microsoft will drop over 10% in a day
prob_return1 = None
print('The Probability is ', prob_return1)
```

```
The Probability is  6.05677563486e-13
```

Expected Output: The Probability is 6.05677563486e-13

Calculate the probability of the stock price will drop over a certain percentage in a year

In [7]:
```
# drop over 40% in 220 days
mu220 = 220*mu
sigma220 = (220**0.5) * sigma
print('The probability of dropping over 40% in 220 days is ', norm.cdf(-0.4, mu220, sigma220))
```

```
The probability of dropping over 40% in 220 days is  0.00291236331333
```

In [9]:
```
# drop over 20% in 220 days
mu220 = 220*mu
sigma220 = (220**0.5) * sigma
drop20 = None
print('The probability of dropping over 20% in 220 days is ', drop20)
```

```
The probability of dropping over 20% in 220 days is  0.0353523772749
```

Expected Output: The probability of dropping over 20% in 220 days is 0.0353523772749

Calculate Value at risk (VaR)

In [12]:
```
# Value at risk(VaR)
VaR = norm.ppf(0.05, mu, sigma)
print('Single day value at risk ', VaR)
```

```
Single day value at risk  -0.0225233624071
```

In [13]:
```
# Quatile
# 5% quantile
print('5% quantile ', norm.ppf(0.05, mu, sigma))
# 95% quantile
print('95% quantile ', norm.ppf(0.95, mu, sigma))
```

```
5% quantile  -0.0225233624071
95% quantile  0.0241638253793
```

In [14]:
```
# This is your turn to calcuate the 25% and 75% Quantile of the return
# 25% quantile
q25 = None
print('25% quantile ', q25)
# 75% quantile
q75 = None
print('75% quantile ', q75)
```

```
25% quantile  -0.00875205783841
75% quantile  0.0103925208107
```

Expected Output: 25% quantile -0.00875205783841 75% quantile 0.0103925208107

Data和Models of Stock Return.ipynb在Github中下载

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

Jupyternote Book课堂练习三十分钟

附：风险价值VaR系列一:模型简介（来源：知乎）

风险是金融的本质，经营金融就是经营风险。我国资本市场有40万亿的股票、80万亿规模的债券，投资者涵盖商业银行、券商、保险、基金及散户。专业机构投资者持有了大量的股票、债券及期货、期权等其他衍生品，承担了巨大风险，**如何管理风险是各家机构投资者不得不面对的问题**。**风险价值（Value at Risk），简称VaR模型**，兴起于上世纪90年代，JP Morgan将其发扬，创立了RiskMetrics系统。目前VaR模型已被广泛运用于各金融机构的市场风险计量和管理。

VaR是指在某个置信区间下投资组合在未来某段时间的最大可能损失，从数学上看它衡量的是投资组合损益分布的分位数，假设C是我们选择的置信区间，则VaR对应于损益分布的下尾处。**例如**置信区间为95%，则**VaR等于损益分布函数的5%处分位数**。

如何计算单个股票的VaR值，我们以上市公司西南证券为例

假设当前持有100万市值的西南证券股票，那么明天的投资组合最大亏损可能是多少？

从图1可以看出，西南证券在2019年下半年一共126个交易日的**日收益率变动区间在基本在-6%与6%之间**，其收益率分布的柱形图如图2所示，我们去**从小到大排列的第5%、大概在6/126的位置**，得出该**分位数的值为-2.35%**，即今天100万市值的西南证券明天95%的概率下最大亏损2.35万。**该2.35万即为95%置信区间下的一天VaR值**。

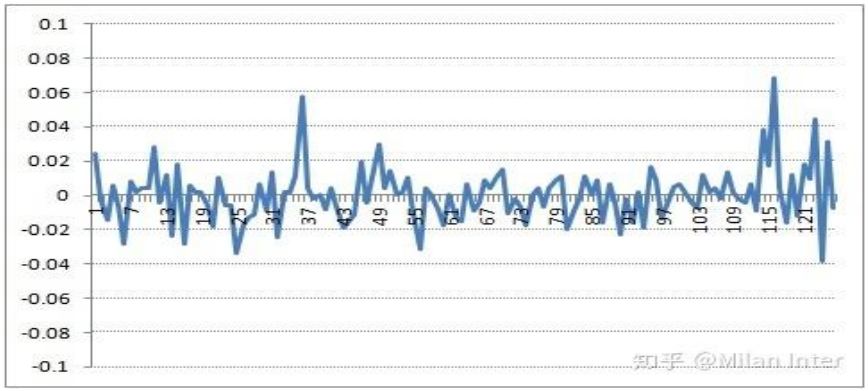


图1. 西南证券（600369）2019年下半年的日收盘价变动情况

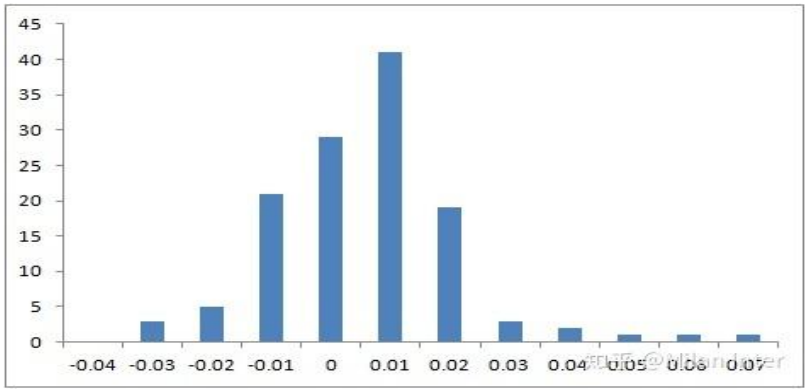


图2. 西南证券损益分布柱形图

前面计算VaR的方法是通过股票历史收益率数据求得其分位数，称之为**历史模拟法**。

另一种计算VaR值得方法为参数法（也称模型法），该方法不直接通过历史收益率求得分位数，而是需要假设收益率的概率分布函数，进而通过收益率的概率密度函数求得对应置信区间的分位点。

以股票西南证券为例，假设西南证券日收益率的分布函数为均值0的正态分布，其标准差σ 可以通过历史数据求得结果为1.60%，而**标准正态分布的5%分位点的随机变量值为1.645**，因此在正态分布假设下95%置信区间下的一天VaR值为1.645×0.016 ,即2.64%，约大于历史模拟法的结果2.35%。相应的，n天的VaR值为$1.645 \times \sqrt{n} \times \sigma$ 。

在**实际的风险管理过程中，总规模限制**是最常见的风险控制指标，限制总规模一定程度上**限制了组合的最大亏损，但无法衡量损失的概率问题**。

例如同样的100万市值的两个投资组合，组合A持有100万市值的西南证券股票，组合B持有100万市值的星期六股票，星期六的损益分布图如图3所示。将**图3和图1相比较**，很明显**星期六的波动幅度大于西南证券**。

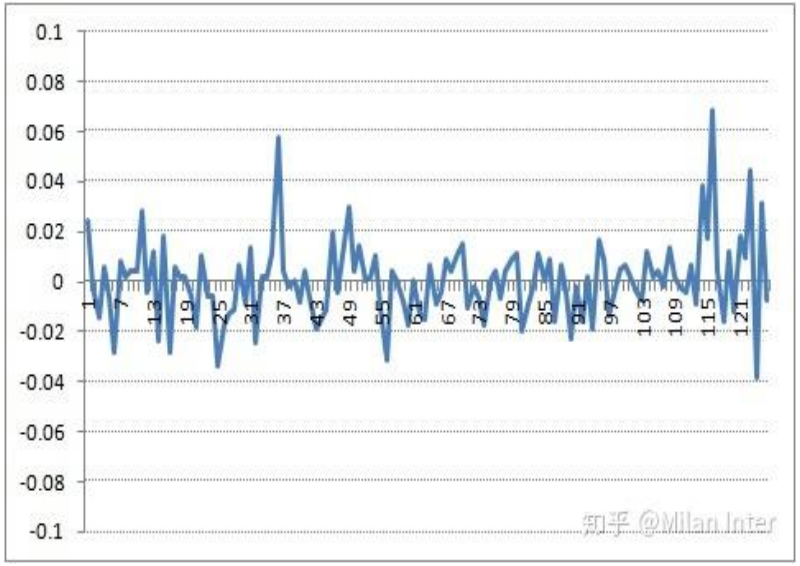


图1. 西南证券（600369）2019年下半年的日收盘价变动情况

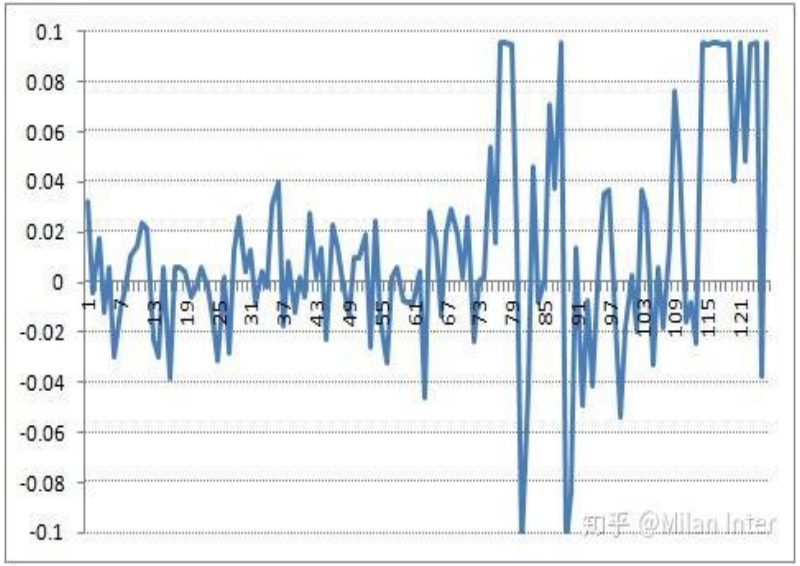


图3. 星期六（002291）2019年下半年的日收盘价变动情况

事实上，在同样的正态分布假设下，星期六的收益率标准差为4.10%，是西南证券波动的两倍有余，其VaR值为6.73%，因此，从VaR角度看，持有股票星期六的风险远远大于持有同样市值的西南证券，持有西南证券有95%的概率最大亏损不超过2.64%，而持有星期六只能保证95%的概率最大损失不超过6.73%。

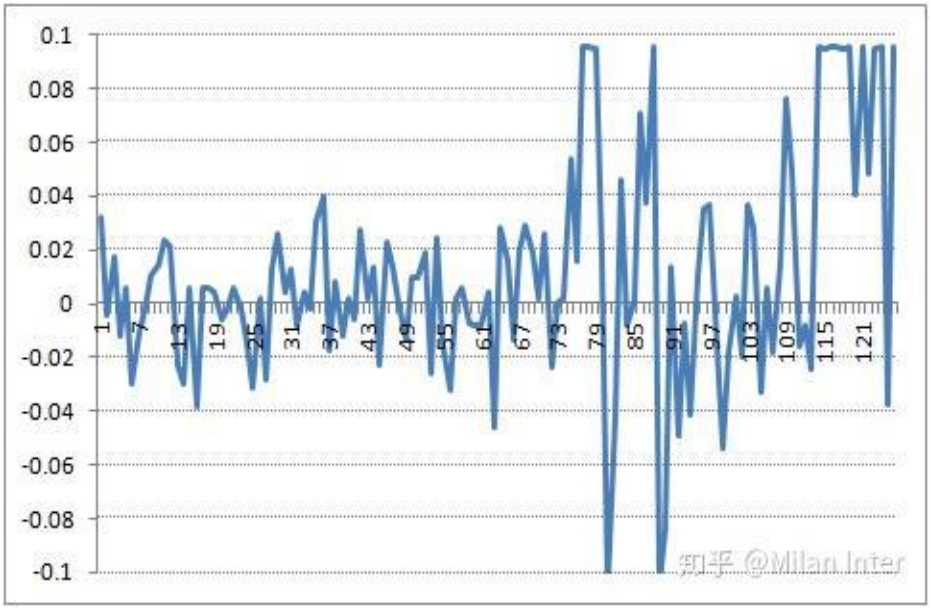


图3. 星期六（002291）2019年下半年的日收盘价变动情况

同样的逻辑也适合债券投资组合，我们常用**久期、凸性、基点价值**等指标衡量债券对利率变动的敏感性，例如某债券投资组合的基点价值为100万元，则在利率上升100bp的情况下，该组合将亏损1亿元。**如果进一步引入该组合的VaR值，则可以回答该投资组合在某概率下的最大亏损**，因为利率变动存在某概率分布。总之，**VaR将敞口类指标与概率统计相结合，给出了组合潜在损失的概率边界，可以更加直观的理解投资组合的可能损失情况**。

VaR衡量的是**市场正常波动下的尾部损失**，即市场正常波动下的极端损失，它解释的是市场正常波动下的最大可能损失，而不是市场极端情形下的损失。

VaR的主要缺陷有两个方面：

- 历史模拟法计算VaR太过**依赖尾部数据**，参数法下的正态分布假设不准确，因为几乎不存在刚好符合正态分布的金融时间序列，且大部分序列都是厚尾的，因此**低估了尾部风险。**

- **对极端损失的度量不足**，这就是典型的期望过高，因为VaR度量的是市场正常波动下的极端损失。

1. Roll two dice and X is the sum of faces values. If we roll them 5 times and get 2,3,4,5,6. Which of the following is/are true about X?

☐ X is a random variable

☐ The mean of X is 4.

☐ X can only take values 2,3,4,5,6

2. Roll two dice and X is the sum of faces values. If we roll them 5 times and get 2,3,4,5,6. What do we know about X?

☐ The dice is fair.

☐ We have 5 observations of X

☐ The most likely value of X is 6

☐ Range of X is 6-2=4

3. Roll two dice and X is the sum of faces values. If we roll them 5 times and get 2,3,4,5,6. X is a _____ random variable.

☐ continuous

☐ discrete

☐ None of the above

4. Why do we use relative frequency instead of frequency?

☐ Relative frequency is easier to compute

☐ Frequency cannot show the number of appearance of outcomes

☐ Relative frequency can be used to compare the ratio of values between difference collections with difference number of values

☐ Relative frequency is easier to compute when the number of observations increases

5. What can we say about relative frequency when we have large number of trials?

☐ Relative frequency becomes approximately the distribution of the corresponding random variable

☐ The relative frequency of each possible outcome will be the same

☐ The relative frequency stays constant after a very large number of trials, eg. n=10000

☐ None of the above

6. What is the notion of "95% Value at Risk" ?

☐ 95% VaR measures how much you can win at most

☐ 95% Value at Risk is 95% quantile

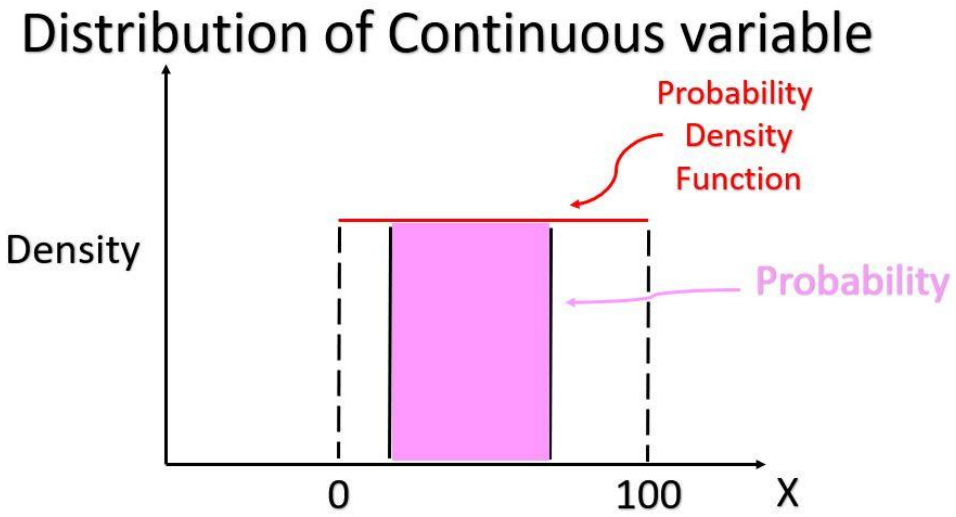
☐ 95% VaR measures the amount of investment you can lose, at the worst 5% scenario

☐ 95% VaR measures how much you can lose at most

7. In the lecture video, we mentioned the calculation of continuous random variable is based on the probability density function.

Given a probability density function, f(x) = 1/100, what is the probability P(10<X<20), where X~Uniform[0, 100]?

- ☐ f(20) –f(10)
- ☐ f(10)
- ☐ f(20)
- ☐ (20- 10) * 1/100



8. What methods should we use to get the cdf and pdf of normal distribution?

☐ norm.cdf() and norm.pdf() from scipy.stats

☐ cdf() and pdf() from pandas

☐ cdf() and pdf() form numpy

☐ norm.cdf() and norm.pdf() from statsmodels

9. Which additional library should we import when we want to calculate log daily return specifically?

☐ Numpy

☐ Pandas

☐ Matplotlib

☐ Statsmodels

10. What is the distribution of stock returns suggested by Fama and French in general?

☐ Close to normal distribution but with fat tail

☐ Left-skewed distribution

☐ A perfect normal distribution

☐ Arbitrary distribution

Thank You