Elements of Financial Engineering Course

Baruch-NSD Summer Camp 2019

Lecture 3 : Financial Engineering in a Nutshell

Tai-Ho Wang (王太和)

Coordinates

- Email: tai-ho.wang@baruch.cuny.edu

Aims of this course

- To be familiar with the financial products and the modeling of their prices
- To be familiar with well-known pricing models.
- To be familiar with the principle of no arbitrage and its applications

Agenda

- Financial market and financial product
- What is financial engineering?
- \( P \) quant v.s. \( Q \) quant
- Theory of pricing
  - Law of one price
  - Monotonicity
  - Linearity
- Principle of no arbitrage
- Brief introduction to programming in R

Online resources

- Investopedia (https://www.investopedia.com/)
- Wikipedia (https://www.wikipedia.org/)
- Quantopian (https://www.quantopian.com/)
- Advanced Risk and Portfolio Management (https://www.arpm.co/)
Financial market

Quotes from Investopedia (https://www.investopedia.com):

A financial market is a broad term describing any marketplace where buyers and sellers participate in the trade of assets such as equities, bonds, currencies and derivatives. Financial markets are typically defined by having transparent pricing, basic regulations on trading, costs and fees, and market forces determining the prices of securities that trade.

Types of financial markets and their roles

- Capital market
  - bond market
  - equity market
- Derivative market
- Forex (foreign exchange) and interbank market
- Primary vs secondary market
  - A primary market issues new securities on an exchange.
  - The secondary market is where investors purchase securities or assets from other investors, rather than from issuing companies themselves.
- OTC (over-the-counter) market
- Third and fourth market


We will only cover the equity and fixed income markets.

Financial products

- Primary or underlying
  - Equity
  - Fixed income
  - Commodity
  - Credit
- Secondary or derivatives
  - Forward and futures
  - Options
  - Swaps

We shall mostly focus on the pricing of derivatives based on the principle of no arbitrage.

As pricing is concerned, in financial engineering we usually consider the following types of problems:

- direct problem: under a given model for the underlying, calculate the prices of its derivatives
- inverse problem: for a given set of prices of liquidly derivatives, determine the parameters or even a model that generates the observed market prices.
What is financial engineering?

According to this article (https://www.iaqf.org/financial-engineering) in the website of Internation Association for Quantitative Finance (IAQF) (used to be named as IAFE), financial engineering is

- the application of mathematical methods to the solution of problems in finance
- also known as financial mathematics, mathematical finance, and computational finance

Therefore,

- Financial engineering draws on tools from applied mathematics, computer science, statistics, and economic theory.
- Investment banks, commercial banks, hedge funds, insurance companies, corporate treasuries, and regulatory agencies employ financial engineers.
- These businesses apply the methods of financial engineering to problems such as new product development, derivative securities valuation, portfolio structuring, risk management, and scenario simulation.

What is financial engineering for?

- Derivative pricing
- Portfolio construction
- Risk management
- Quantitative trading
- Market making
- Order implementation

What skills are required to be a financial engineer?

Quantitative finance is an interdisciplinary field, it requires hard skills in

- Math: calculus, linear algebra, probability, stochastic process, differential equation, optimization, etc
- Finance: derivative pricing theory, modern portfolio theory, market microstructure models
- Statistics: regression and classification, factor analysis, time series analysis
- Programming: R, Python, C++, Matlab, etc

Nowadays also requires

- machine learning techniques
- sentiment analysis
Quantitative analyst - Quant

Excerpt from Quants: The Rocket Scientists of Wall Street (https://www.investopedia.com/articles/financialcareers/08/quants-quantitative-analyst.asp) in Investopedia,

As financial securities become increasingly complex, demand has grown steadily for people who not only understand the complex mathematical models that price these securities, but who are able to enhance them to generate profits and reduce risk. These individuals are known as quantitative analysts, or simply "quants."

Quantitative analysts

- design and implement complex models that allow financial firms to price and trade securities
- front desk quants work directly with traders, providing them with pricing or trading tools
- back office quants validate the models, conduct research and create new strategies
- positions are found almost exclusively in major financial centers with trading operations

ℙ quant v.s. ℚ quant

Quants are briefly categorized as ℙ quants and ℚ quants.

- ℙ refers to the physical measure under which the financial assets presumably evolves.
- ℚ refers to the risk neutral measure for pricing under the principle of no arbitrage.

The mentality behind ℙ quant and the buy side is

- Regard the market as a whole, process historical data then forecast price movements
- Construct portfolio or investment strategy based on performance measures
- Decide the horizon of holding the portfolio
- Risk management

The mentality behind ℚ quant and the sell side is

- Construct model that could possibly explain the market
- Calibrate parameters of model to market data
- Apply the calibrated model to price new derivatives
- Hedging

Note

- In this course, we shall cover topics on both sides, though not evenly.
- Please refer to this link (https://www.arpm.co/lab/about-quantitative-finance.html#x4-130001) in ARPM.co (https://www.arpm.co/) for more details on the interplay between ℙ quants and ℚ quants.
Theory of asset pricing

Heuristically, pricing is a rule or a map that assigns a (unique, current) value to each (random) payoff/cashflow which can only be realized at the future investment horizon. The collection/universe of all the payoffs at the investment horizon is assumed to be equipped with a vector space structure. In other words, we would be able to add or subtract a payoff from another, and also be able to scale up or down by any (real, positive or negative) scalar. Thus, we can regard pricing as a function or a functional $\Pi$ from the space of (random) payoffs to real numbers:

$$\Pi : P \rightarrow \mathbb{R},$$

where $P$ is space of payoffs at investment horizon.

A pricing functional $\Pi$ presumably bears the following axioms

- Law of one price
- Monotonicity
- Linearity

By essentially the Riesz representation theorem, a pricing function/functional, should it exist, can be charactrized by the expectation of a (random) payoff weighted by a stochastic discount factor.

The price of any asset in the universe is given by the an expectation of discounted payoff discounted by a stochastic discount factor.

Note

In this course we shall explore the modeling of the pricing functional.

Law of one price

Two payoffs with equal values in all scenarios must have the same price.

Monotonicity

If the payoff $X$ dominates the payoff $Y$, i.e., $X > Y$ in any scenario, then the price of $X$ must be higher than that of $Y$.

Linearity

The price of linear combination of the payoffs $X$ and $Y$ is the same linear combination of the price of $X$ and $Y$. That is,

$$\Pi(\alpha X + \beta Y) = \alpha \Pi(X) + \beta \Pi(Y)$$

If the pricing function/functional satisfies the axioms, it can be represented as a discounted expectation. Precisely, for a given payoff $X$, its price is given by

$$\Pi(X) = \mathbb{E}[DX]$$

where $D$ is a (positive) random variable called (it stochastic discount factor).

Note

The $\mathbb{E}[D]$ is the price of zero coupon bond since, the payoff of zero coupon bond of face value 1 is $X \equiv 1$. 


Principle of no arbitrage

Arbitrage opportunity

Intuitively, an arbitrage opportunity is a trade or a trading strategy that have positive payoff at the investment horizon with zero cost currently. In other words, we are able to acquire a financial position, be statically or dynamically, without being required to pay initially for entering the position.

Any viable financial model should not permit arbitrage opportunity.

In reality, arbitrage opportunity does exist. However, it disappears very quickly even much quicker now due to the advent of technology, since once it is exploited, sophisticated market participants will quickly take the advantage of it, then the market reacts to it so as to reach a new status without arbitrage opportunity.

Principle of no arbitrage is the core behind the theory of derivative pricing.

Fundamental theorem of asset pricing


"The fundamental theorems of asset pricing (also: of arbitrage, of finance) provide necessary and sufficient conditions for a market to be arbitrage free and for a market to be complete."

To be more specific,

In a discrete (i.e. finite state) market, the following hold:

The First Fundamental Theorem of Asset Pricing

A discrete market, on a discrete probability space (\(\Omega, \mathcal{F}, \mathbb{P}\)), is arbitrage-free if and only if there exists at least one risk neutral probability measure that is equivalent to the original probability measure, \(\mathbb{P}\).

The Second Fundamental Theorem of Asset Pricing

An arbitrage-free market \((S, B)\) consisting of a collection of stocks \(S\) and a risk-free bond \(B\) is complete if and only if there exists a unique risk-neutral measure that is equivalent to \(\mathbb{P}\) and has numeraire \(B\).

Note

We shall illustrate these concepts by using the binomial model.

Brief introduction to programming in R

Online resources for programming in R

- R documentation (https://www.rdocumentation.org)
- Nice R code (https://nicercode.github.io/)
This is a test for integral $\int \log x \, dx$

$$\int_{0}^{\infty} \frac{\sin(x)}{x} \, dx$$

可以试试中文字

也可以试试繁體字

$$\frac{df}{dx} = f'(x)$$
In [1]:

```r
# create a list in R
stock_list <- c(3.5, 5, 2, 8, 4.2)
stock_list

# present data as a data.frame
test <- data.frame(ex=stock_list)
test
test$ex

unname(test)
test$ex2 <- c(5, 4, 3, 2, 1)
test

unname(test)
```

```
3.5 5 2 8 4.2

<table>
<thead>
<tr>
<th>ex</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
</tr>
<tr>
<td>5.0</td>
</tr>
<tr>
<td>2.0</td>
</tr>
<tr>
<td>8.0</td>
</tr>
<tr>
<td>4.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ex</th>
<th>ex2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>5</td>
</tr>
<tr>
<td>5.0</td>
<td>4</td>
</tr>
<tr>
<td>2.0</td>
<td>3</td>
</tr>
<tr>
<td>8.0</td>
<td>2</td>
</tr>
<tr>
<td>4.2</td>
<td>1</td>
</tr>
</tbody>
</table>
```
In [2]: # matrix multiplication
   A <- diag(3)
   B <- 2*diag(3) + 1
   c <- 1:3
   A
   B
   c

   1 0 0
   0 1 0
   0 0 1
   3 1 1
   1 3 1
   1 1 3

   1 2 3

In [3]: A*B
   A%*%B

   3 0 0
   0 3 0
   0 0 3
   3 1 1
   1 3 1
   1 1 3
In [9]:
B*c
B%%c
# create a matrix of placeholders
x <- matrix(nrow=2, ncol=3)
x
x[,1] <- 1:2
x
x[,2] <- c(0.4, 2)
x
x[,3] <- c('a', 'n')
x

3 1 1
2 6 2
3 3 9

8
10
12
NA NA NA
NA NA NA
1 NA NA
2 NA NA
1 0.4 NA
2 2.0 NA
1 0.4 a
2 2 n

In [ ]:
dnorm # density
pnorm # cdf
qnorm # quantile function
rnorm # random number

exp
gamma
pois
t
In [4]:  # restructure in to a matrix
xx <- rnorm(10, mean=1, sd=2)
xx
mean(xx)
sd(xx)
matrix(xx, nrow = 2)

```
  -2.0740435 0.4423217 -0.03634995 -1.172697 1.8596282
  2.1213049 0.9509460 -0.82410150 1.0421606 -0.1678294
```

In [16]:  rep(rnorm(5), 3)

```
  -1.38346920888855 1.39944680007583 -0.670390893183419 -1.2575248215259
  -1.59513565430513 -1.38346920888855 1.39944680007583 -0.670390893183419
  -1.2575248215259 -1.59513565430513 -1.38346920888855 1.39944680007583
  -0.670390893183419 -1.2575248215259 -1.59513565430513
```

In [1]:  replicate(10, rnorm(6))

```
  -1.2331019 0.5358802 -0.7686204 -0.13611690 0.91525126 -1.04840873
  -1.0407616 1.3869342 -0.7879698 0.02576324 0.31171331 0.50273469
  -0.1574827 -0.7229757 -0.4398579 0.09643152 0.17458012 0.28813220
  -2.2443674 0.6394456 -0.1127722 -0.37072076 -0.33599530 -0.62793948
  -0.1912699 -0.2749901 -0.6185310 -0.77584672 0.14947946 -0.71391407
  -1.3196788 1.4048692 -0.3995725 -0.29868859 0.04198728 0.06246794
```

Baruch-NSD_Summer2019_Lec3-FinancialEngineering
http://localhost:8888/nbconvert/html/Google Drive/Lectures/Ba...
In [25]:

```r
# randomly create a matrix
set.seed(0)
A <- replicate(5, rt(10, df=1))
A
mean(A)
```

```
[1,]  0.72287339 -0.4313783 -0.2211401  1.2573112 -0.9018979 -16.8103321
[2,] 12.29973050  1.7676553  0.7624387  0.2268739  0.5875736  6.8388690
[3,] -1.44384361  0.4838340  2.9614228 -0.1936225 -0.4899404 -0.3567021
[4,] -0.00205066 -2.4255767  0.8058890  0.5875736  0.6870035 -2.4255767
[5,]  2.14975954 -0.5887315 -0.7722049  2.6919347  1.9326181 -0.0020507
[6,] -1.77521873  0.5853582 -4.5129805 -2.3002584 -0.2330814 -0.1104565
[7,] -0.40920665 -0.9979839  2.0828094 -0.7169420 -0.0905921  1.2117985
[8,]  0.67772018  1.2117985 -2.2773203 -0.9667084 -0.3696092  0.4500325
[9,]  1.03612989  0.4500325  0.1598536 -0.2992270 -0.5006909  0.6777202
[10,] -16.81033210  6.8388690 -0.0850384  1.6308382 -1.4617096 -0.1104565
```

-0.110456478644366
In [26]:
# for finding means columnwise and rowwise,
# we use the functions colMeans and rowMeans respectively

\[
colMeans(A)
\]
\[
rowMeans(A)
\]

# note that mean(A) returns the mean of all entries in A
mean(A)

# However, for standard deviations, we use "apply"
# apply a function to a matrix columnwise
apply(A, 2, sd)

# apply a function to a matrix columnwise
apply(A, 1, sd)

# of course, we can also use "apply" for columnwise and rowwise means
apply(A, 2, mean) # same as colMeans
apply(A, 1, mean) # same as rowMeans

-0.355510978927711  0.689387736044482  -0.109627071893797  -0.598071851264163
-0.178460227180643

-0.41777082805474  2.94000775371187  0.263570063355158  -0.0694322435010714
0.00590131577300863  -1.64723616721448  -0.026525855391177  -0.344823847362649
0.16921961698756  -1.97747459479921

-0.110456478644366

7.02921529497155  2.46761223066509  2.12000729027019  1.29844085590446
0.922293954208016

0.754838552955526  5.29034061665741  1.6597432268291  1.3533284191615
2.03382165965573  1.97734793355042  1.2268655827989  1.37754916674588
0.61210688625412  8.86860792616867

-0.355510978927711  0.689387736044482  -0.109627071893797  -0.598071851264163
-0.178460227180643

-0.41777082805474  2.94000775371187  0.263570063355158  -0.0694322435010714
0.00590131577300863  -1.64723616721448  -0.026525855391177  -0.344823847362649
0.16921961698756  -1.97747459479921

In [28]:
# rescale the plot
options(repr.plot.width=4, repr.plot.height=4)
In [5]: # plot graph of a function
f <- function(x) {sin(x)}
g <- function(x) {sin(x - pi/6)}
h <- function(x) {sin(x - pi/4)}
curve(f, from=0, to=2*pi, col='blue', las=1, main='Whatever the Title',
      xlab=expression(kappa), ylab=expression(sigma))
curve(g, from=0, to=2*pi, col='red', lty=2, lwd=3, add=TRUE)
curve(h, from=0, to=2*pi, lty=3, lwd=1, add=TRUE)
In [7]:

```r
# generate random samples from standard normal
sample <- rnorm(1e6)

# descriptive statistics of sample
summary(sample)

# histogram
# hist(sample, breaks=50)
hist(sample, breaks=50, prob=TRUE)

# superimpose the density
curve(dnorm, from=min(sample), to=max(sample), add=T, col='blue')
```

```
Min.    1st Qu.     Median       Mean   3rd Qu.      Max.
-5.136540 -0.673225 -0.000270  0.000552  0.675490  4.572936
```