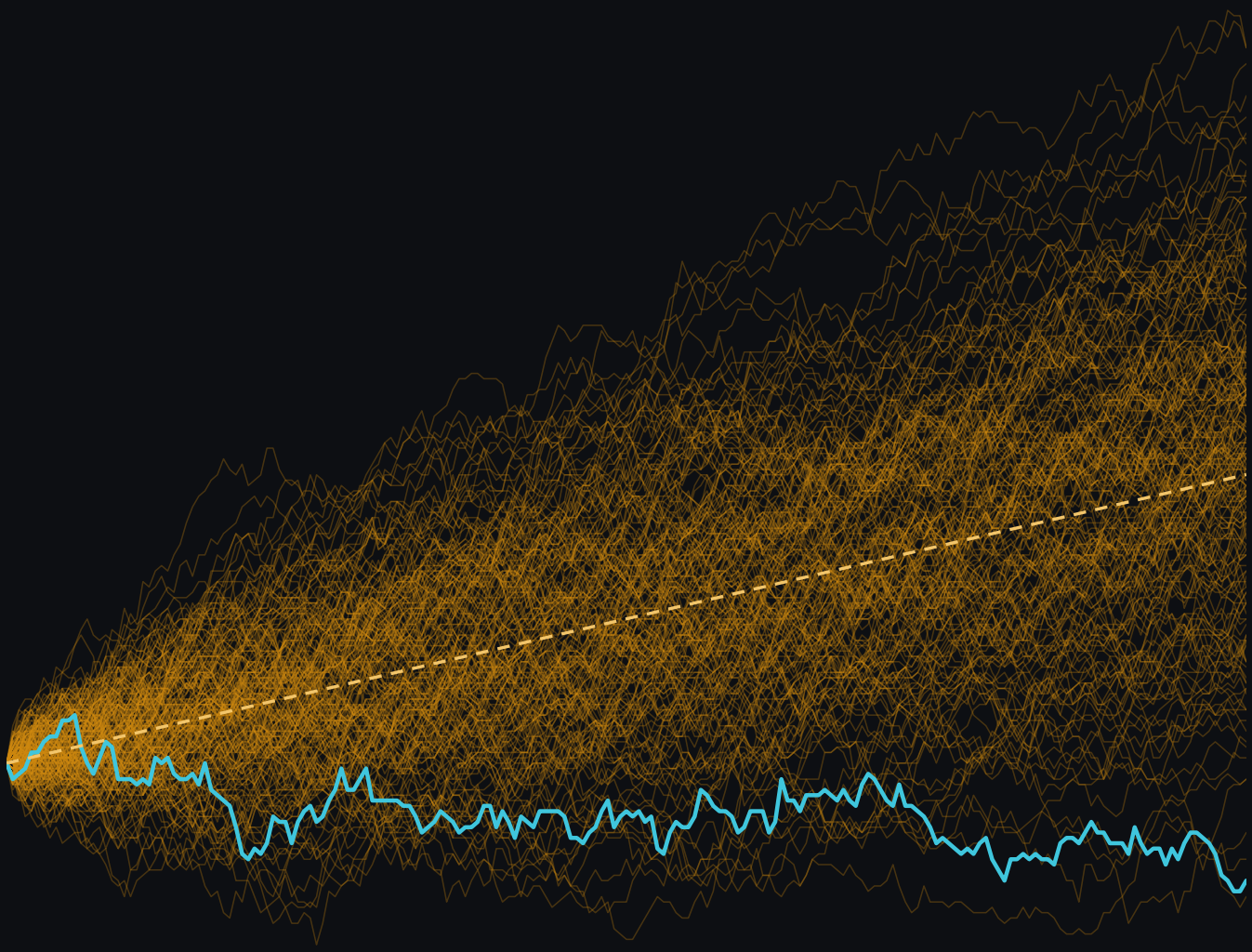
@QuantFinanceToGo



Quantitative Finance From Scratch

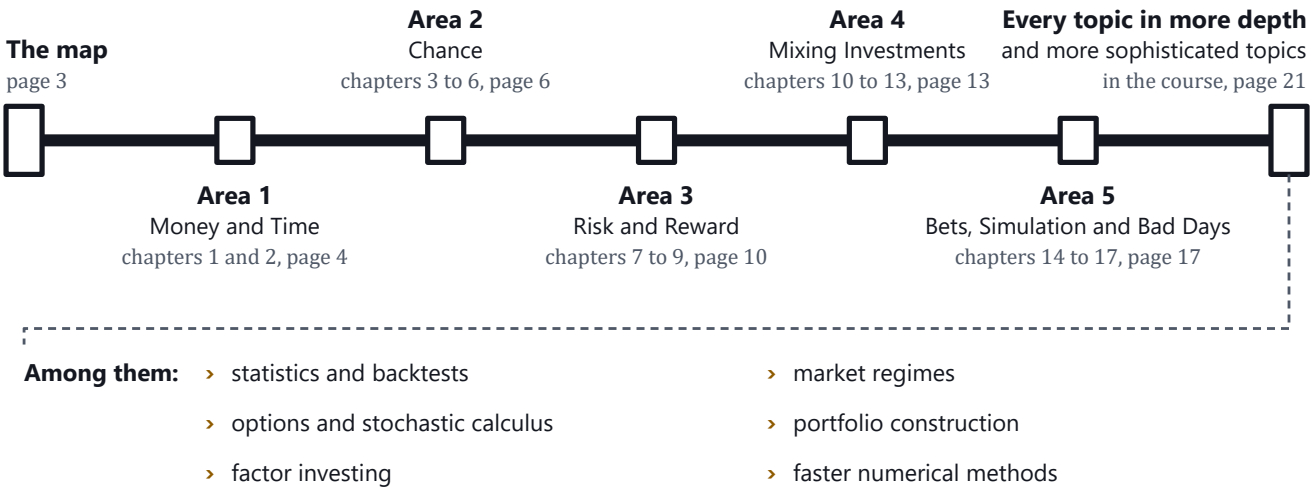
An intuitive first step into quantitative finance

All data in this book is simulated by my own programs. The book is for learning only and gives no financial advice.

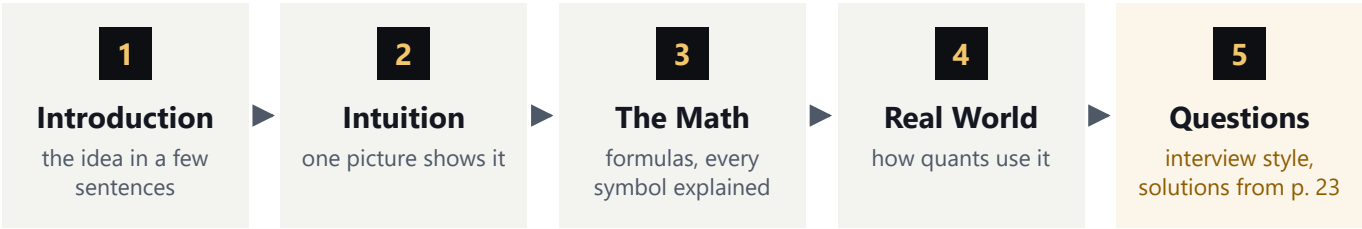
How This Book Works

This book takes you through the basics of **quantitative finance**, the use of math and data to understand prices and risk. Its seventeen short chapters follow the order professional courses teach. A word in **bold** is new and explained where it stands, and page 28 lists them all.

One route through the book



Every chapter follows the same order



After this book you can

- ✓ Measure returns, discount money and work with expected values.
- ✓ Compare investments fairly with volatility, the Sharpe ratio and correlation.
- ✓ Explain why the spread of a price grows with the square root of time, and where the bell curve fails.
- ✓ Use beta, the CAPM, Kelly, Monte Carlo and value at risk, and answer first interview questions on each.

The Map of Quant Finance

1 Money and Time

How a price change is measured, and why money later is worth less than money today.

1 Returns p. 4
2 Interest and Present Value p. 5

2 Chance

Why a small advantage wins in the long run, and how coin flips build a price chart.

3 Expected Value p. 6
4 The Random Walk p. 7
5 The Square Root Rule p. 8
6 The Bell Curve p. 9

3 Risk and Reward

How to put the swings of a price into one number, and what you are paid for them.

7 Volatility p. 10
8 The Sharpe Ratio p. 11
9 Volatility Drag p. 12

4 Mixing Investments

How several investments move together, and how mixing them calms the swings.

10 Correlation p. 13
11 Diversification p. 14
12 The Efficient Frontier p. 15
13 Beta and the CAPM p. 16

5 Bets, Simulation and Bad Days

How much to bet, how to try out ten thousand futures, and how to size up a very bad day.

14 The Kelly Criterion p. 17
15 Monte Carlo Simulation p. 18
16 Fat Tails p. 19
17 Value at Risk p. 20

Start gently · A bit deeper · Going further

After the chapters

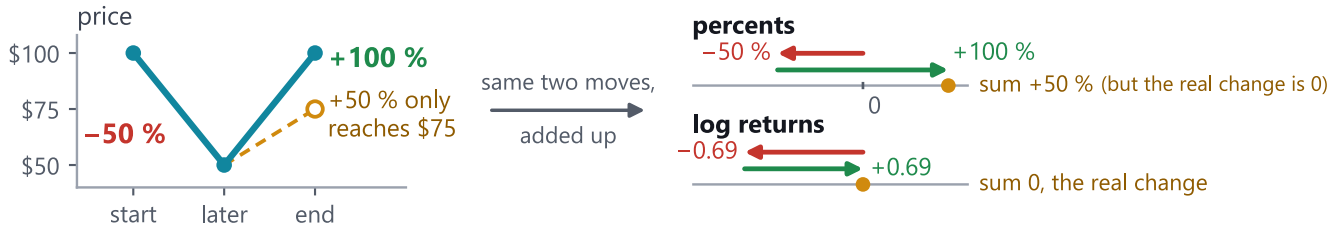
› Extra chapter: Everything in the Full Course p. 21 to 22
› Solutions p. 23 to 25
› Formula sheet p. 26 to 27
› Words in this book p. 28

1 Returns

Start gently

A **return** measures how much an investment gained or lost, as a share of what it cost at the start.

Returns might sound too simple, but they hide an insight that every later chapter builds on: percent returns of several periods cannot be added up, while **log returns** can.



*A **stock** falls −50 % and climbs back +100 %. Added up, the percents give a false +50 %, the log returns the true 0.*

$$R = \frac{S_1}{S_0} - 1$$

R = the return, as a decimal: 0.10 means +10 %
S_1 = the price at the end
S_0 = the price at the start

The return is the new price over the old price, minus one.

$$r = \ln \frac{S_1}{S_0}$$

r = the log return
ln = the natural logarithm, the "ln" button on a calculator

The log return is the natural logarithm of the new price over the old price.

$$1 + R_{\text{total}} = (1 + R_1)(1 + R_2) \qquad r_{\text{total}} = r_1 + r_2$$

R_{total} = the return over both periods
R_1 = the return of the first period
R_2 = the return of the second period
r_{total} = the log return over both periods
r_1 = the log return of the first period
r_2 = the log return of the second period

Over two periods, returns multiply, while log returns add up.

```
R = S1 / S0 - 1
r = math.log(S1 / S0)
```

In Python

Real World

- Fund documents show returns over one, five and ten years.
- Many risk models use log returns, because they add up over time.

1.1 *Calculate* ●●● A stock falls from \$80 to \$60. What is its return? And what return does it need to get back to \$80?

1.2 *Think it through* ●●● A price rises 10 % in one year and falls 10 % in the next. Is it back where it started? What do the two log returns add up to?

Solutions on page 23

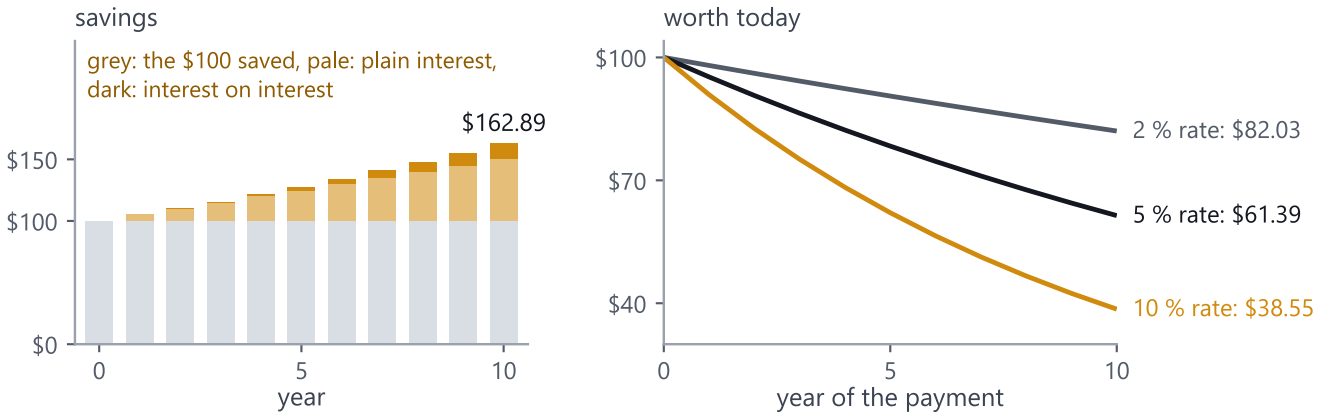
The course covers this topic in full depth, with Python code, exercises and a mini project.

2 Interest and Present Value

Start gently

Money today can earn interest, so a dollar today is worth more than a dollar later. This is the **time value of money**.

The reasons are simple: money you have now can be invested and grow, and prices usually rise over time, which is called inflation. Bringing all future payments or cash flows back to their **present value** today makes them comparable, the first step to pricing any loan or savings plan. The further away a payment lies and the higher the rate, the less it is worth today.



*Left: $100 at an **interest rate** of 5 % a year grows to $162.89 in ten years, and $12.89 of it is **compound interest**, interest on interest. Right: what $100 paid later is worth today, found by **discounting**.*

$$FV = PV\,(1+i)^{T}$$

FV	=	the future value, in *T* years
PV	=	the present value, worth today
i	=	the interest rate per year
T	=	the number of years

Money grows by the factor 1 + i, once every year.

$$PV = \frac{FV}{(1+i)^{T}}$$

The value today divides the future value by 1 + i once for every year.

```
PV = FV / (1 + i) ** T
```

In Python

Real World

- Bond desks price a bond as the present value of its future payments.
- EU insurers value future claim payments at their present value.

2.1 *Calculate* ●○○ You save $1,000 at an interest rate of 4 % a year for 3 years, with interest on interest. How much do you have at the end?

2.2 *Calculate* ●○○ Someone promises you $500 in 2 years. The interest rate is 5 %. What is the promise worth today? And at a rate of 8 %?

Solutions on page 23

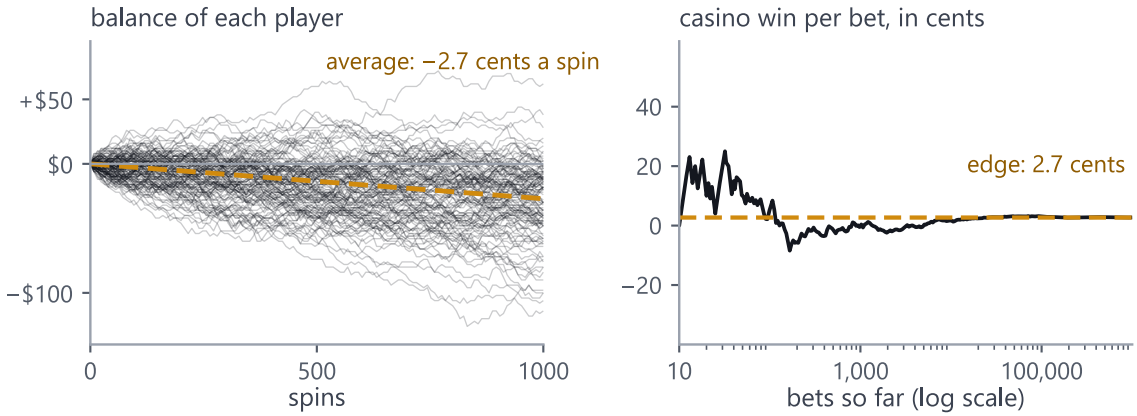
The course covers this topic in full depth, with Python code, exercises and a mini project.

3 Expected Value

Start gently

The **expected value** of a bet is what it pays on average per try, if you could repeat it very many times.

It tells you whether a bet pays off in the long run, even though a single try can go either way. Casinos and insurers live on this, because a small average gain per try becomes a reliable profit over millions of tries. Over many tries the average settles down like this, which is called the **law of large numbers**.



*Left: 120 of 1,000 players, each betting $1 on red 1,000 times. Red wins with a **probability** of 18/37, and only 181 of them end ahead. Right: the casino's win per bet settles near 2.7 cents, its **edge**.*

$$E[X] = \sum_{k} p_k \, x_k$$

$E[\,]$	=	the expected value
X	=	a random outcome, such as the win of one bet
Σ	=	add up over all outcomes
k	=	the number of an outcome
p_k	=	the probability of x_k
x_k	=	possible value number k of X

The expected value weighs every outcome by its probability and adds them up.

$$E[\text{red}] = \tfrac{18}{37} \cdot (+1) + \tfrac{19}{37} \cdot (-1) = -\tfrac{1}{37} \approx -0.027$$

$\approx$	=	about equal to

A $1 bet on red loses 2.7 cents on average.

```
EV = sum(p * x for p, x in zip(probs, values))
```

In Python

Real World

- An insurer sets premiums starting from the expected value of claims.
- Trading desks price many contracts as the discounted expected value of their payoffs.

3.1 *Calculate* ●●● A fair die pays you its number in dollars, from $1 for a one up to $6 for a six. What is the expected value of one roll? What is the most you would pay to play if only the average counts?

3.2 *Calculate* ●●● On a roulette wheel with 37 pockets, a bet on one single number pays 35 to 1: you win $35 or lose your $1. What is the expected value of this bet? Compare it with a $1 bet on red.

Solutions on page 23

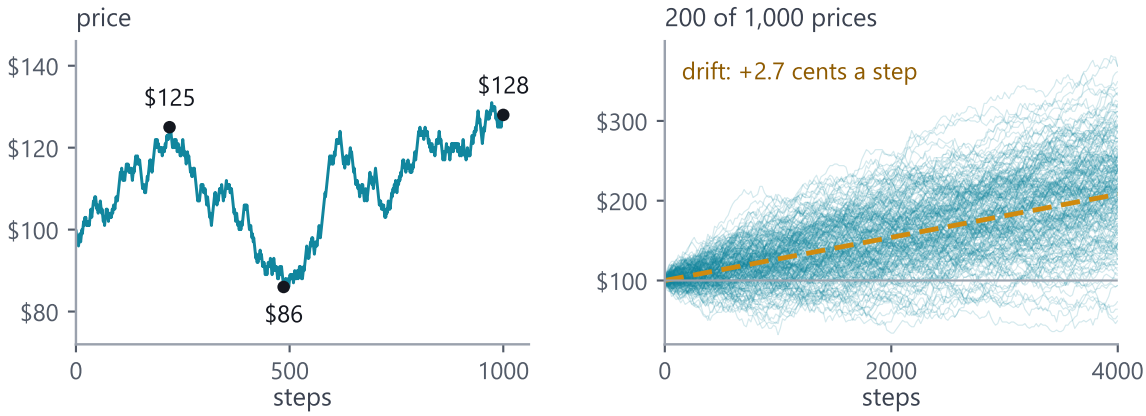
The course covers this topic in full depth, with Python code, exercises and a mini project.

4 The Random Walk

Start gently

A **random walk** is a price that moves up or down by a random step, again and again.

It is the simplest model of how prices move, and it shows how pure chance can draw what looks like a trend. Every step is a fresh coin flip, so a long rise on the chart says nothing about where the next step goes. A small push per step decides where the walk heads in the long run, while chance rules every short stretch.



*Left: one walk of 1,000 fair coin flips, chosen to show a rise, a fall and a comeback. Right: 200 of 1,000 walks whose coin lands up with probability 19/37, each line one **path**. They rise 2.7 cents a step on average, the **drift**.*

$$S_n = S_0 + X_1 + X_2 + \cdots + X_n$$

- S_n = the price after n steps
- S_0 = the price at the start
- X_1 = the first step, here +$1 or −$1
- X_2 = the second step
- X_n = the last step, number n

The price after n steps is the start price plus all n steps.

$$E[S_n] = S_0 + n\,\mu$$

- $E[\,]$ = the expected value, the average over many walks
- n = the number of steps
- μ = the average of one step, the drift

On average the walk moves by the drift in every step.

```
steps = np.where(rng.random(1000) < 0.5, 1, -1)
S = 100 + steps.cumsum()
```

In Python

Real World

- Standard pricing models let a stock price follow a random walk with drift.
- Researchers have tested real prices against a random walk since the 1960s.

4.1 *Calculate* ●●● A price starts at $100 and moves $1 up or $1 down each day, on the flip of a fair coin. After 4 days, which prices are possible, and which one is the most likely?

4.2 *Think it through* ●●● A chart made with a fair coin shows a long rise over hundreds of steps. Does that prove the coin is unfair?

Solutions on page 23

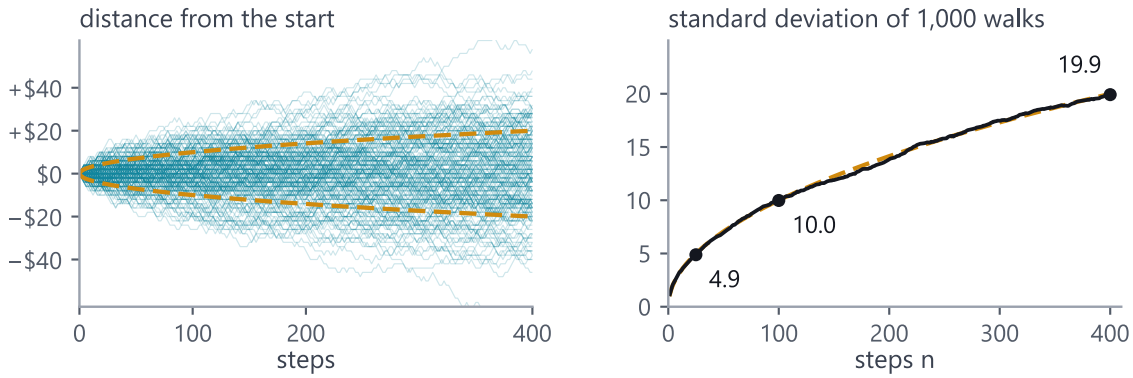
The course covers this topic in full depth, with Python code, exercises and a mini project.

5 The Square Root Rule

A bit deeper

The **square root rule** says that the spread of a random walk grows with the square root of time: four times the steps give twice the spread.

It tells you how much wider the range of possible prices gets the further ahead you look. Ups and downs partly cancel each other, so the spread grows more slowly than the number of steps.



*Left: 200 of 1,000 walks, each $1 step **independent** of the ones before. Right: the **variance** is the average squared distance, and its square root, the **standard deviation**, is 4.9 after 25 steps, 10.0 after 100 and 19.9 after 400, close to $\sqrt{n}$.*

$$\mathrm{Var}(X) = E\big[(X-\mu)^2\big] \qquad \sigma = \sqrt{\mathrm{Var}(X)}$$

Var = the variance
X = a random outcome
***E*[ ]** = the expected value
μ = the average of *X*, its expected value
σ = the standard deviation, the typical distance

Variance is the average squared distance from the average.

$$\mathrm{Var}(S_n - S_0) = n\,\sigma^2 \quad \Longrightarrow \quad \sigma_n = \sigma\sqrt{n}$$

S_n = the price after *n* steps
S_0 = the price at the start
n = the number of steps
⇒ = so it follows that
σ_n = the spread after *n* steps

For independent steps the variances add, so the spread grows with the square root of n.

```
sigma_n = sigma * math.sqrt(n)
```

In Python

Real World

- Bank rules let banks scale a one-day loss figure to ten days with this rule.
- Risk teams scale daily swings up to a month or a year the same way.

5.1 *Calculate* ●●● The daily moves of a stock have a standard deviation of 1 %, and the moves are independent. A year has 252 days with trading. What is the standard deviation over a year? And over 4 days?

5.2 *Think it through* ●●● Why does the spread of a random walk grow with the square root of time, and not in step with time?

Solutions on page 23

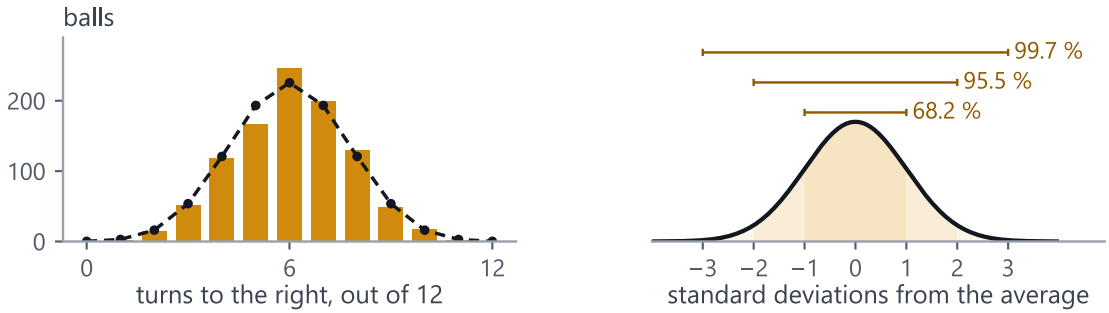
The course covers this topic in full depth, with Python code, exercises and a mini project.

6 The Bell Curve

A bit deeper

Add up many small independent random steps, and the sum almost always takes the same shape, the **bell curve**.

It is the distribution finance assumes most often, also called the normal distribution, because of the **central limit theorem**: sums of many small random effects take this shape. Most results land near the average, and rare extremes sit in the thin **tails**.



Left: 1,000 balls fall through 12 rows of pins and pile up close to the dashed bell that chance predicts. Right: of 100,000 sums of 100 random moves, 68.2 %, 95.5 % and 99.7 % lie within one, two and three standard deviations of the average.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- f = the height of the curve at the value x
- x = a value on the axis
- σ = the standard deviation
- π = the number 3.14159...
- exp = exp(y) is 2.718... to the power y
- μ = the average

This formula gives the height of the bell curve at the value x.

$$P(|X-\mu| \le \sigma) \approx 68\,\% \qquad 2\sigma\colon 95\,\% \qquad 3\sigma\colon 99.7\,\%$$

- $P()$ = the probability of what stands in the brackets
- $||$ = the size of a number without its sign
- X = a random outcome
- $\approx$ = about equal to

About 68 % of all values lie within one standard deviation of the average, 95 % within two and 99.7 % within three.

```
share = math.erf(k / math.sqrt(2))
```

In Python

Real World

- Classic risk models start by assuming that returns follow a bell curve.
- Risk teams use it to judge how rare a large daily loss should be.

6.1 *Calculate* ●●● Yearly returns follow a bell curve with an average of 8 % and a standard deviation of 16 %. In about what share of years is the return below −24 %?

6.2 *Think it through* ●●● Why does a yearly return look like a bell curve, even when the daily moves have some other shape?

Solutions on page 23

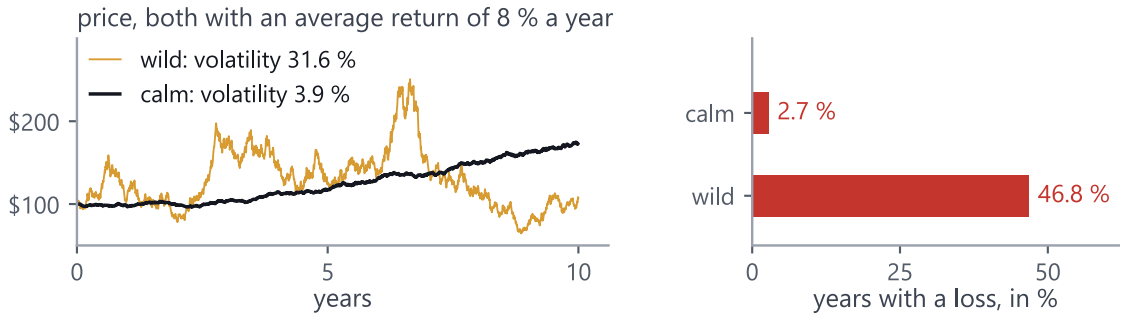
The course covers this topic in full depth, with Python code, exercises and a mini project.

7 Volatility

A bit deeper

Volatility is the standard deviation of the returns of an investment, usually given per year.

Quants often simply call it risk, because it puts the risk of an investment into one number. An average return alone hides how bumpy the ride is, and volatility adds that missing part. Turning daily numbers into yearly ones, over the **trading days** of a year, is called **annualizing**.



Left: a calm and a wild price, both built to average 8 % a year. A typical day moves 0.247 % and 1.99 %, over about 252 trading days 3.9 % and 31.6 % a year. Right: the calm one loses money in 2.7 % of 100,000 years, the wild one in 46.8 %.

$$\sigma_{\text{day}} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(r_t - \bar{r})^2}$$

σ_{day} = the daily volatility
N = the number of days
Σ = add up over all days t from 1 to N
t = the number of a day
r_t = the log return of day t
$\bar{r}$ = the average daily log return

The daily volatility is the typical size of a daily move. Many programs divide by N − 1 instead of N, which matters only for short series.

$$\sigma_{\text{year}} = \sigma_{\text{day}} \sqrt{252}$$

σ_{year} = the yearly volatility

The square root rule turns daily into yearly volatility.

```
vol_day = np.std(r)
vol_year = vol_day * math.sqrt(252)
```

In Python

Real World

- Key documents of EU funds rank risk on a scale from 1 to 7, built from volatility.
- Option traders often quote a price as a volatility number.

7.1 *Calculate* ●●● Over five days a stock has the daily log returns +1 %, −2 %, +3 %, 0 % and −2 %. What is the average? What is the daily volatility, dividing by 5 as in the formula?

7.2 *Think it through* ●●● Is volatility the same as the risk of losing money?

Solutions on page 24

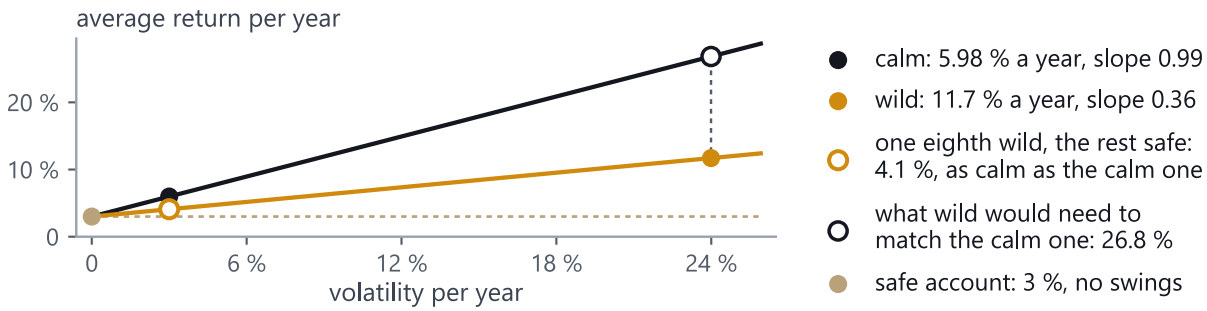
The course covers this topic in full depth, with Python code, exercises and a mini project.

8 The Sharpe Ratio

A bit deeper

The **Sharpe ratio** measures the extra return an investment pays for each unit of volatility, so calm and wild investments compare fairly.

A higher return alone proves little, because it may only come from taking bigger risks. The Sharpe ratio shows whether the extra return paid for the extra swings.



*A calm and a wild investment over 10,000 simulated years. The safe account pays the **risk-free rate** of 3 %, and the return above it is the **excess return**. The slope of the line from the safe account to an investment is its Sharpe ratio.*

$$\mathrm{SR} = \frac{\mu - r_f}{\sigma}$$

- **SR** = the Sharpe ratio
- μ = the average yearly return
- r_f = the risk-free rate, the interest of a safe account
- σ = the yearly volatility

The Sharpe ratio is the extra return over the safe account for each unit of volatility.

$$\mu_w - r_f = w\,(\mu - r_f) \qquad \sigma_w = w\,\sigma \qquad \Longrightarrow \qquad \mathrm{SR}_w = \mathrm{SR}$$

- μ_w = the average yearly return of the mix
- w = the share of your money in the investment, between 0 and 1
- σ_w = the yearly volatility of the mix
- ⇒ = so it follows that
- SR_w = the Sharpe ratio of the mix

Put a share w in the investment and the rest in the safe account, and the Sharpe ratio stays the same.

```
sharpe = (mu - rf) / sigma
```

In Python

Real World

- Analysts compare fund managers by the Sharpe ratio of their past returns.
- Risk teams use it to rank strategies with very different volatility on one scale.

8.1 *Calculate* ●●● Fund A returns 9 % a year with a volatility of 15 %. Fund B returns 6 % with a volatility of 6 %. The risk-free rate is 3 %. Which fund has the higher Sharpe ratio?

8.2 *Calculate* ●●● You put half of your money in fund A and the rest in the safe account at 3 %. What are the return, the volatility and the Sharpe ratio of this mix?

8.3 *Calculate* ●●● A strategy has a Sharpe ratio of 0.1, measured on daily returns with a daily risk-free rate. Roughly what is its yearly Sharpe ratio, with 252 trading days and independent days?

Solutions on page 24

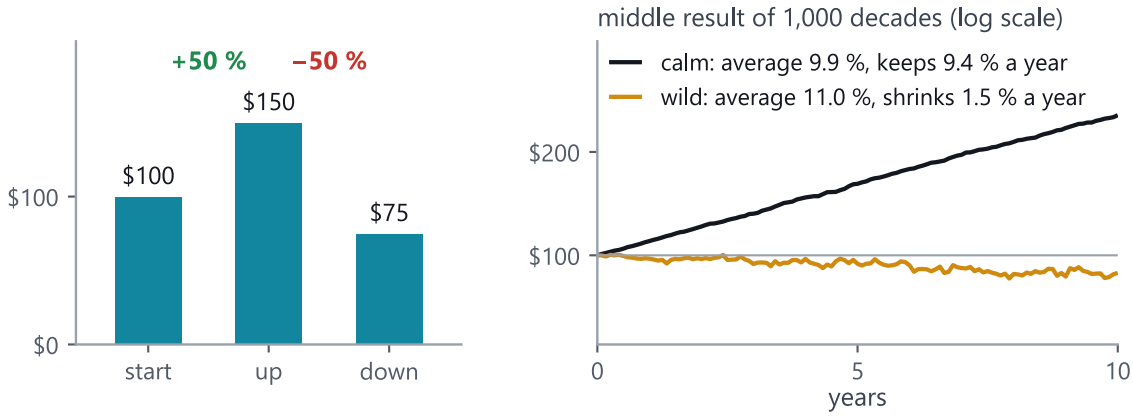
The course covers this topic in full depth, with Python code, exercises and a mini project.

9 Volatility Drag

A bit deeper

Gains and losses multiply, so the average return overstates what you keep. The gap is the **volatility drag**.

It explains why two investments with the same average return can leave you with very different amounts of money. A fall by half needs a doubling to win it back, so wild swings eat into your money even when the average looks fine. What you really keep is the **growth rate**, the average log return.



Left: +50 % and then −50 % average to zero, and still $100 ends at $75. Right: the middle of 1,000 decades for a calm and a wild investment, both averaging about 10 % a year. The calm one grows 9.4 %, the wild one shrinks 1.5 %.

$$(1+s)(1-s) = 1 - s^2$$

s = the size of one move, such as 0.10

Up by s and down by s always leaves you below the start.

$$g \approx \mu - \tfrac{1}{2}\,\sigma^2$$

g = the growth rate you keep each year in the end
$\approx$ = about equal to
μ = the average yearly return
σ = the yearly volatility

The growth rate you keep is the average minus half the variance. This approximation works best for small steps.

```
g = mu - sigma ** 2 / 2
```

In Python

Real World

- Regulators warn that funds which double each day's move fall behind in choppy markets.
- Fund documents report a compound yearly growth rate, which already includes the drag.

9.1 *Calculate* ●●● A price rises 50 % and then falls 50 %. Where does it end? And after +20 % and then −20 %?

9.2 *Calculate* ●●● An investment has an average yearly return of 10 % and a volatility of 20 %. Roughly what growth rate do you keep over many years? And with a volatility of 40 %?

Solutions on page 24

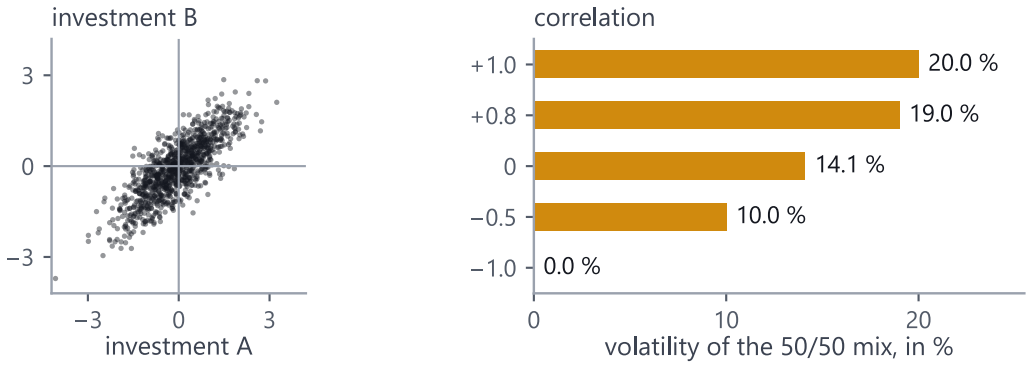
The course covers this topic in full depth, with Python code, exercises and a mini project.

10 Correlation

A bit deeper

Correlation is one number between minus one and one that says how closely the daily moves of two investments go together.

It tells you whether two investments tend to fall at the same time, which decides how much mixing them calms your money. Two investments that fall on the same days give little protection, while two that move apart soften each other's losses. Correlation comes from the **covariance**, which grows on days when both move the same way, scaled by both standard deviations.



Left: 1,000 days of two investments that share a market push, with a correlation of +0.80. Right: a half and half mix of two investments that swing 20 % a year gets calmer the lower their correlation.

$$\mathrm{Cov}(X, Y) = E\big[(X - \mu_X)(Y - \mu_Y)\big]$$

Cov = the covariance
X = a first random outcome, such as a daily return
Y = a second random outcome
$E[\,]$ = the expected value
μ_X = the average of X
μ_Y = the average of Y

Covariance measures whether two investments move away from their averages at the same time.

$$\rho = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \, \sigma_Y} \qquad -1 \le \rho \le 1$$

ρ = the correlation (rho)
σ_X = the standard deviation of X
σ_Y = the standard deviation of Y

Correlation is the covariance, scaled to a number between minus one and one.

```
rho = cov / (sigma_x * sigma_y)
```

In Python

Real World

- Risk teams estimate correlations between assets to see which losses tend to come together.
- Studies find that correlations between stock markets rise in falling markets.

10.1 *Calculate* ●●● Two stocks each have a yearly volatility of 20 %, and their covariance is 0.02. What is their correlation?

10.2 *Think it through* ●●● Does a correlation of zero mean that two things have nothing to do with each other?

Solutions on page 24

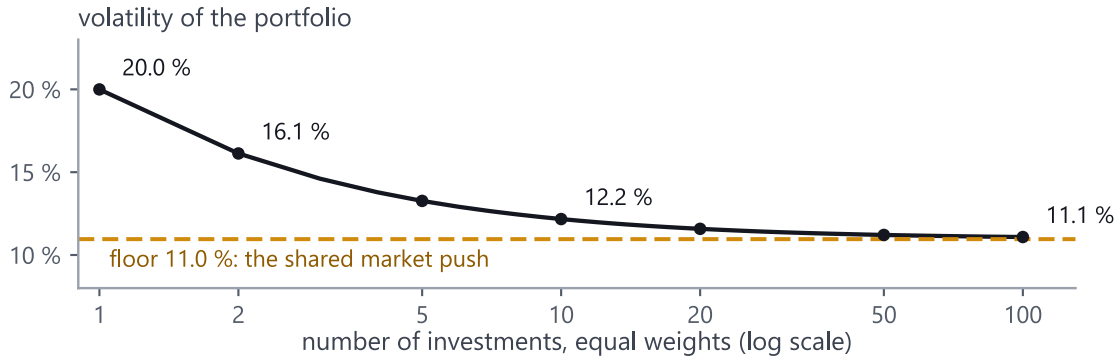
The course covers this topic in full depth, with Python code, exercises and a mini project.

11 Diversification

A bit deeper

Diversification spreads money over many investments, so that their own ups and downs partly cancel.

It lowers the swings of your money without lowering the average return, as long as the investments do not all move together. Each investment has its own surprises, and in a large mix they mostly cancel out, while the shared market moves remain as a floor that no mix can remove. Such a mix is a **portfolio**, and each investment's share of the money is its **weight**.



Up to 100 investments with the same weight, each swinging 20 % a year, any two with a correlation of 0.3. The volatility of the mix falls from 20.0 % to 12.2 % with 10 of them and to 11.1 % with 100, near the floor of 11.0 %.

$$\sigma_p^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2\,w_1 w_2\,\rho\,\sigma_1\sigma_2$$

- σ_p = the volatility of the portfolio
- w_1 = the weight, the share of the money in the first investment
- σ_1 = the volatility of the first investment
- w_2 = the weight of the second investment
- σ_2 = the volatility of the second investment
- ρ = the correlation between the investments

This is the variance of a mix of two investments.

$$\sigma_p^2 = \sigma^2\left(\frac{1}{N} + \left(1 - \frac{1}{N}\right)\rho\right) \;\longrightarrow\; \sigma_p = \sigma\sqrt{\rho}$$

- σ = the volatility of each single investment
- N = the number of investments
- → = becomes, as N grows very large

With N equal investments, the own part shrinks and the shared part stays.

```
vol_p = sigma * math.sqrt(1 / N + (1 - 1 / N) * rho)
```

In Python

Real World

- US pension law requires plans to diversify to minimize the risk of large losses.
- EU rules limit how much of a retail fund one company's securities may make up.

11.1 *Calculate* ●●● You split your money equally over 4 independent investments, each with a volatility of 20 %. What is the volatility of this portfolio?

11.2 *Calculate* ●●● Now every pair of these investments has a correlation of 0.25. What is the volatility with 4 of them? What is the lowest you can reach with very many?

Solutions on page 24

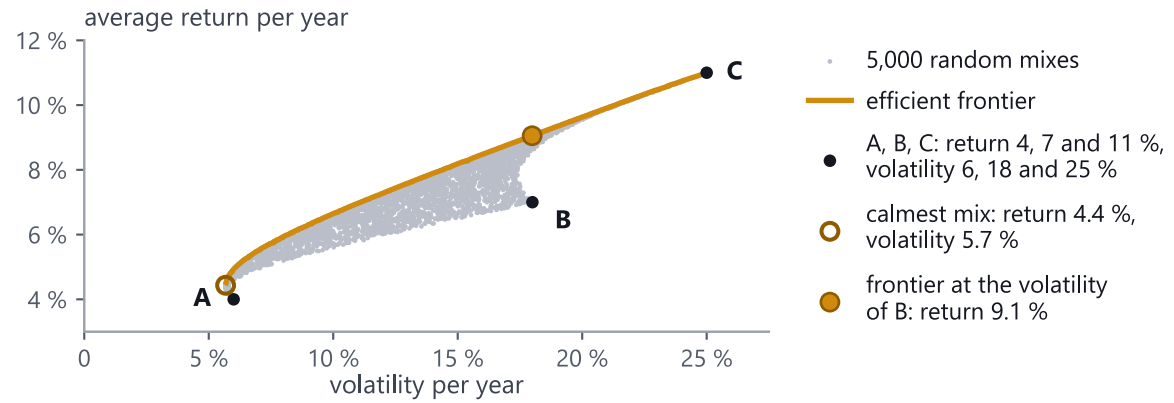
The course covers this topic in full depth, with Python code, exercises and a mini project.

12 The Efficient Frontier

Going further

The **efficient frontier** is the set of portfolios with the highest average return for each level of volatility.

It shows which mixes are worth a look at all, because every mix below the frontier gives up return for nothing. In the end, an investor picks the level of risk they accept, and the frontier gives the mix with the highest return for it.



*Each grey point is one of 5,000 random mixes of the assumed investments A, B and C. The upper edge is the efficient frontier, its left tip the **minimum variance** portfolio. At B's volatility the frontier earns 9.1 %, B alone 7.0 %.*

$$\mu_p = w_1\mu_1 + w_2\mu_2$$

- μ_p = the average return of the portfolio
- w_1 = the weight of the first investment
- μ_1 = the average return of the first investment
- w_2 = the weight of the second investment
- μ_2 = the average return of the second investment

The return of a mix is the weighted average of the returns of its parts.

$$w_1^* = \frac{\sigma_2^2 - \rho\,\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2 - 2\rho\,\sigma_1\sigma_2}$$

- w_1^* = the weight of the first investment in the calmest mix
- σ_2 = the volatility of the second investment
- ρ = the correlation of the two investments
- σ_1 = the volatility of the first investment

This weight gives the calmest mix of two investments. It can lie outside 0 to 1, which means borrowing one of the two and selling it, called selling short.

```
w1 = (s2 ** 2 - rho * s1 * s2) / (s1 ** 2 + s2 ** 2 - 2 * rho * s1 * s2)
```

In Python

Real World

- Asset allocation teams often start from the efficient frontier to pick a mix of stocks and bonds.
- Index providers publish minimum variance versions of stock indexes that funds track.

12.1 *Calculate* ●●● Two investments have volatilities of 10 % and 30 % and a correlation of −1. Which weights give a portfolio with zero volatility?

12.2 *Think it through* ●●● What makes a portfolio "efficient"?

Solutions on page 24

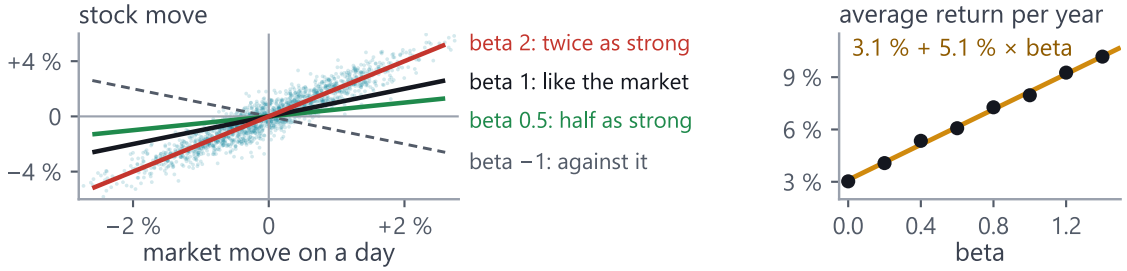
The course covers this topic in full depth, with Python code, exercises and a mini project.

13 Beta and the CAPM

Going further

Beta is a stock's sensitivity to market risk, and the **CAPM**, the capital asset pricing model, says only this risk earns extra return.

A beta of 1 means the stock moves like the **market portfolio**, all stocks of a market together, 2 twice as strongly, 0.5 half and a negative beta against it. The CAPM says only this factor earns a reward, since anyone can mix away the rest, each stock's own **specific risk**. Real prices follow it only roughly.



*Left: daily moves of three stocks against the market, each line's slope its beta. Right: eight stocks over 10,000 simulated years, with a risk-free rate of 3 % and a **market risk premium** of 5 %, lie close to the **security market line**.*

$$\beta_i = \frac{\mathrm{Cov}(R_i, R_m)}{\sigma_m^2} = \rho \frac{\sigma_i}{\sigma_m}$$

- β_i = the beta of stock *i*
- Cov = the covariance
- R_i = the return of stock *i*
- R_m = the return of the market
- σ_m = the volatility of the market
- ρ = the correlation of stock and market
- σ_i = the volatility of stock *i*

Beta says how many percent a stock moves when the market moves one percent.

$$E[R_i] = r_f + \beta_i \left(E[R_m] - r_f \right)$$

- $E[\,]$ = the expected value
- r_f = the risk-free rate

The CAPM expects the risk-free rate plus beta times the market risk premium.

```
expected = rf + beta * (market - rf)
```

In Python

Real World

- Many company finance chiefs use the CAPM to estimate the return their owners require.
- Utility regulators use the CAPM to set the return a water or power firm may earn.

13.1 *Calculate* ●●● A stock has a correlation of 0.6 with the market and a volatility of 30 %, and the market has a volatility of 15 %. What is its beta, and what move do you expect on a day the market falls 4 %?

13.2 *Calculate* ●●● The risk-free rate is 3 % and the market is expected to return 8 %. What return does the CAPM expect from a stock with a beta of 1.4?

13.3 *Think it through* ●●● Why does the CAPM pay for beta and nothing for the specific risk of a stock?

Solutions on page 25

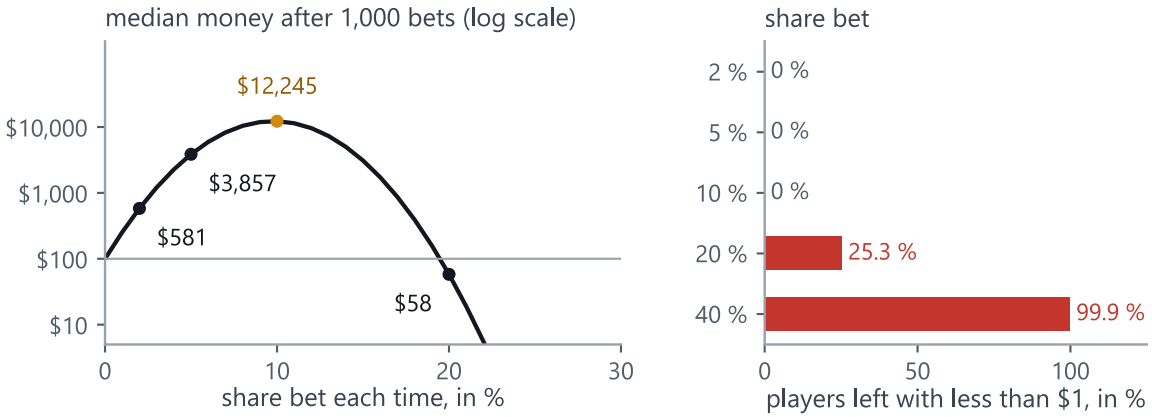
The course covers this topic in full depth, with Python code, exercises and a mini project.

14 The Kelly Criterion

Going further

The **Kelly criterion** gives the optimal bet size when the odds are known, the share of your money that makes it grow fastest in the long run.

It guards against two mistakes, betting so little that your money barely grows and so much that a few losses wipe it out. Even a bet in your favor can leave you with almost nothing if you stake too much, because each loss shrinks the money for all later bets. The fair yardstick is the **median** player, the middle result, because a few lucky winners pull the average up.



*One thousand players with $100 bet a fixed share 1,000 times on a coin winning 55 % at even money. Left: the median player does best at the Kelly share of 10 %. Right: larger shares bring **ruin**, a quarter below $1 at 20 %.*

$$g(f) = p\,\ln(1 + bf) + (1 - p)\,\ln(1 - f)$$

- g = the average log growth per bet
- f = the share of your money you bet
- p = the probability to win a bet
- **ln** = the natural logarithm
- b = what one dollar wins, here 1 for even money

This is the average log growth per bet when you bet the share f.

$$f^* = p - \frac{1 - p}{b} \qquad (b = 1\colon\ f^* = 2p - 1)$$

- f^* = the Kelly share, with the largest growth

The Kelly share f* is the share that makes this growth largest.

```
kelly = p - (1 - p) / b
```

In Python

Real World

- Many practitioners bet a fraction such as half Kelly, giving up some growth for calmer swings.
- Some investment managers size positions with Kelly when they trust their edge.

14.1 *Calculate* ●●● A bet wins with a probability of 60 % and pays even money. What share of your money does Kelly bet?

14.2 *Calculate* ●●● A bet pays 2 to 1 (b = 2) and wins with a probability of 40 %. What is the Kelly share? Why do many traders bet only half of it?

Solutions on page 25

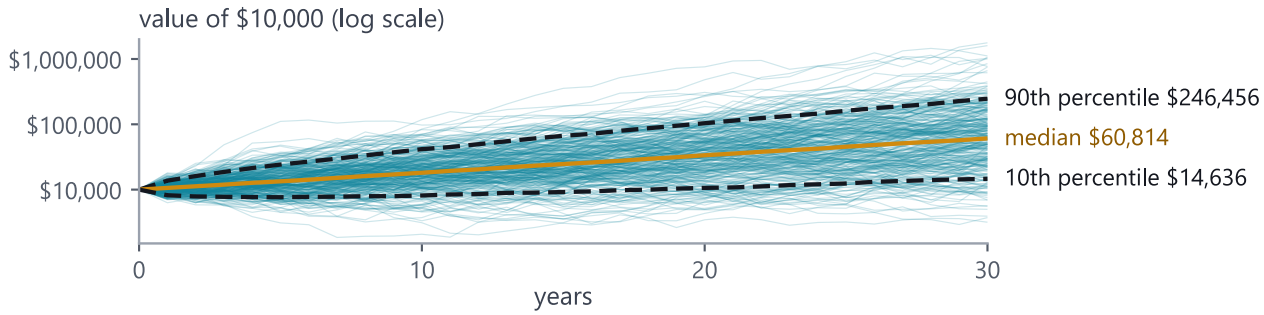
The course covers this topic in full depth, with Python code, exercises and a mini project.

15 Monte Carlo Simulation

Going further

A **Monte Carlo simulation** tries out thousands of possible futures with random numbers and counts how often each outcome comes up.

Each simulated path strings together random returns into one possible course of events under the chosen assumptions, so thousands of paths cover the many futures that could happen. From them you read the median, the range and how likely a goal is, which no simple formula can tell you. A fixed **seed** gives the same paths in every run, and each number read off them has a **standard error** that shrinks with the square root of the number of paths.



Ten thousand paths over 30 years, each year's log return drawn around 6 %, the growth rate of an 8 % average with 20 % volatility (chapter 9). Half end below $60,814, the median, one in ten below $14,636, the 10th ***percentile****.*

$$\hat{X} = \frac{1}{M} \sum_{m=1}^{M} X_m$$

- $\hat{X}$ = the estimate
- M = the number of simulated paths
- Σ = add up over all paths m from 1 to M
- m = the number of a path
- X_m = the result of path m

The estimate is the average over M simulated paths.

$$\mathrm{SE} = \frac{\sigma}{\sqrt{M}}$$

- **SE** = the standard error of the estimate
- σ = the standard deviation of one simulated result

Four times as many paths halve the standard error.

```
final = 10_000 * np.exp(rng.normal(0.06, 0.2, (10_000, 30)).sum(axis=1))
low, mid, high = np.percentile(final, [10, 50, 90])
```

In Python

Real World

- Banks price complex contracts by Monte Carlo when no formula exists.
- Risk teams simulate thousands of paths to see the spread of a portfolio's future value.

15.1 *Calculate* ●●● A Monte Carlo estimate from 10,000 paths has a standard error of 0.4. How many paths do you need for a standard error of 0.1?

15.2 *Think it through* ●●● Why does a simulation fix its seed?

Solutions on page 25

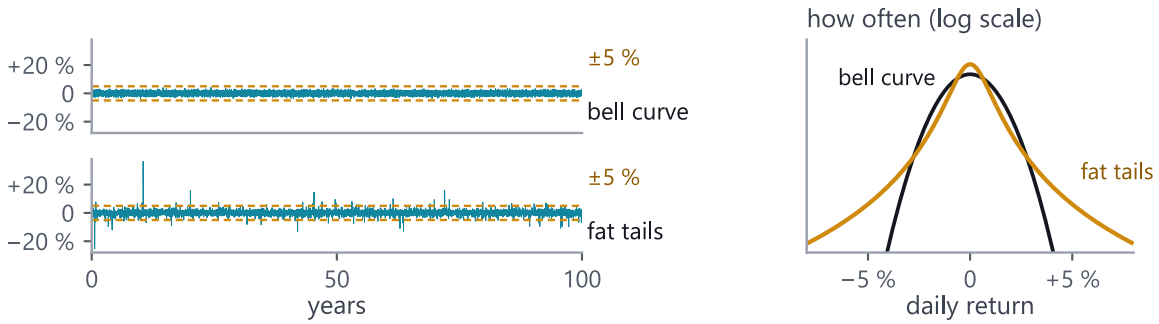
The course covers this topic in full depth, with Python code, exercises and a mini project.

16 Fat Tails

Going further

Real returns have **fat tails**, far more very large moves than the bell curve allows.

The bell curve expects a very large drop once in many thousands of years. Real stock returns bring such days again and again, so the normal distribution does not fit them.



*One hundred simulated years of daily returns, both with a volatility of 1 % a day. Top: the bell curve has no day beyond 5 %, its worst −4.1 %. Bottom: the fat-tailed model has 74 such days, its worst −25.5 %, and a **kurtosis** of 107 against 3.0.*

$$\kappa = \frac{E\left[(X-\mu)^4\right]}{\sigma^4} \qquad \kappa_{\text{bell}} = 3$$

κ	=	the kurtosis (kappa); many programs report $\kappa - 3$, the excess kurtosis, which is 0 for the bell curve
$E[\,]$	=	the expected value
X	=	a daily return
μ	=	the average daily return
σ	=	the standard deviation of the daily returns
κ_{bell}	=	the kurtosis of the bell curve

Kurtosis measures how heavy the tails are.

$$P\left(X < \mu - 5\sigma\right) \approx 2.9 \times 10^{-7}$$

$P()$	=	the probability of what stands in the brackets
$\approx$	=	about equal to
10^{-7}	=	one ten-millionth, 0.0000001

For daily returns on the bell curve, with 252 trading days a year, a drop of more than five standard deviations comes once in about 14,000 years.

```
p = 0.5 * math.erfc(5 / math.sqrt(2))
years = 1 / (p * 252)
```

In Python

Real World

- Researchers treat fat tails as a basic fact of daily returns in most markets.
- Newer bank rules measure trading risk deeper in the tails, under stress.

16.1 *Calculate* ●●● Daily returns follow a bell curve. How often should a drop of more than five standard deviations happen, with 252 trading days a year?

16.2 *Think it through* ●●● Two return series have the same average and the same volatility. One has a kurtosis of 3, the other of 9. Which one would worry you more, and why?

Solutions on page 25

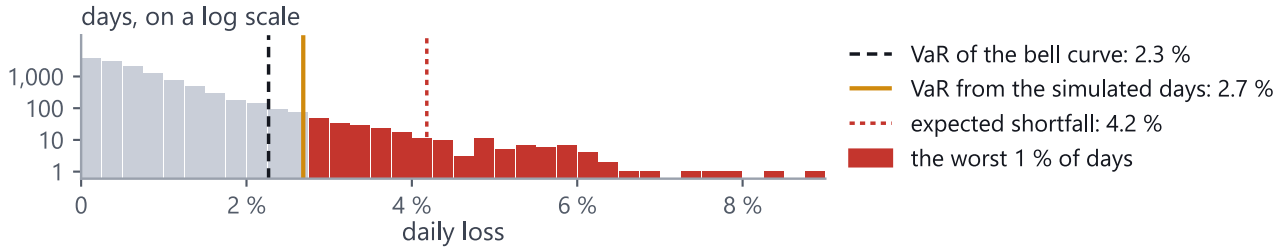
The course covers this topic in full depth, with Python code, exercises and a mini project.

17 Value at Risk

Going further

Value at risk tells you the maximum expected loss at a certain **confidence level** within a specified time frame.

A one-year VaR at 99 % means that within one year the loss stays below this amount with 99 % probability. It says nothing about the losses beyond it, and the **expected shortfall** fills this gap by averaging them.



One hundred years of daily losses from the fat-tailed model of chapter 16. The bell curve's 99 % VaR of 2.3 % is crossed on 396 days, where 252 were expected. The worst 1 % of days lose 4.2 % on average, the expected shortfall.

$$P\big(L > \mathrm{VaR}_c\big) = 1 - c$$

$P()$ = the probability of what stands in the brackets
L = the loss of one day
VaR_c = the value at risk at the confidence level c, as a share of the portfolio or an amount of money
c = the confidence level, such as 0.99

Losses exceed the VaR only on the share 1 − c of all days. A 99 % VaR over one year is a loss size that the yearly loss stays below in 99 of 100 years, so only about one year in a hundred brings a larger loss.

$$\mathrm{VaR}_c = z_c\,\sigma\,V \qquad \mathrm{ES}_c = \frac{\varphi(z_c)}{1-c}\,\sigma\,V$$

z_c = how many standard deviations the cutoff lies from the average
σ = the daily volatility
V = the value of the portfolio
ES_c = the expected shortfall at the confidence level c
φ = the height of the standard bell curve, with average 0 and standard deviation 1

For a bell curve with average 0 and c = 99 %, z is 2.33, and the expected shortfall is 2.67 standard deviations.

```
var, es = z * sigma * V, NormalDist().pdf(z) / (1 - c) * sigma * V
```

In Python

Real World

- Bank rules long used a ten-day value at risk at a 99 % confidence level.
- EU insurers must hold capital for a one-year value at risk at 99.5 %.

17.1 *Calculate* ●●● A portfolio is worth $1,000,000. Its daily returns follow a bell curve with an average of 0 and a volatility of 1 % a day. What is its one-day VaR at a confidence level of 99 %?

17.2 *Calculate* ●●● What is the expected shortfall at 99 % for the same portfolio?

17.3 *Think it through* ●●● Two portfolios have the same VaR at 99 %. Why can one of them still be much riskier?

Solutions on page 25

The course covers this topic in full depth, with Python code, exercises and a mini project.

Everything in the Full Course

If you are serious about learning quantitative finance, and maybe about working in the field one day, the full course takes you further, one level at a time. It is science based and covers everything, from the concepts and the math to the code and its use in the real world.

The course in four levels

Every module explains one topic in plain words first and then in full depth, with Python code, exercises with worked solutions and a mini project.

1	**Quant Beginner**	18 modules
2	**Quant Intermediate**	15 modules
3	**Quant Advanced**	16 modules
4	**Quant Expert**	13 modules

Projects for Your CV

Each project is a complete tool with all of its code. These four are ready now, and more projects will follow.

1. **Option Pricing Tool**
2. **Backtesting Environment**
3. **Market Regime Detection**
4. **Portfolio Optimizer and Risk Report**

Interview Prep

Practice the way quant interviews ask, with a worked solution for every question.

155	9	4	3
questions	categories	roles	levels, easy to hard

By role

Quant Researcher	Quant Trader	Quant Developer	Risk Quant
ideas tested in data	fast math and betting games	code first, then puzzles	risk numbers and model checks

More questions are added over time.

Portfolio Optimizer and Risk Report

This is a preview of one of the four projects. The program below is the start of the project's tool. It simulates twenty years of monthly returns of five asset classes from a known truth, estimates the mean of each with its error bar and measures a mix with equal weights as a first benchmark.

```python
# Portfolio Optimizer and Risk Report, the start of the tool
import math
import numpy as np

# Settings: five asset classes, their true means and swings a year
NAMES = ["Stocks", "Bonds", "Credit", "Real estate", "Commodities"]
TRUE_MU = np.array([0.070, 0.025, 0.035, 0.060, 0.040])
TRUE_VOL = np.array([0.16, 0.05, 0.07, 0.18, 0.20])
TRUE_CORR = np.array([[1.0, -0.1, 0.5, 0.7, 0.3],
                      [-0.1, 1.0, 0.6, 0.1, -0.1],
                      [0.5, 0.6, 1.0, 0.5, 0.2],
                      [0.7, 0.1, 0.5, 1.0, 0.3],
                      [0.3, -0.1, 0.2, 0.3, 1.0]])
TAIL_DF = 5           # fat tails of the simulated history
MONTHS = 240          # 20 years of monthly history
RF = 0.02             # risk free rate a year

def simulate_history(rng):
    # monthly returns: true means and correlations, fat tails
    cov = TRUE_CORR * np.outer(TRUE_VOL, TRUE_VOL) / 12
    z = (rng.standard_normal((MONTHS, len(NAMES)))
         @ np.linalg.cholesky(cov).T)
    scale = np.sqrt((TAIL_DF - 2)
                    / rng.chisquare(TAIL_DF, (MONTHS, 1)))
    return TRUE_MU / 12 + z * scale
```

The second half of the program is part of the course.

What it prints

```
MEAN A YEAR, WITH ITS ERROR
Stocks       6.54% +/- 3.60%
Bonds        3.41% +/- 1.09%
Credit       3.89% +/- 1.61%
Real estate  6.90% +/- 3.85%
Commodities  3.49% +/- 4.77%
EQUAL WEIGHTS, 20 % EACH
expected return     4.84%
volatility          9.40%
Sharpe ratio        0.303
beta to stocks      0.468
```

The five steps

Inputs from return series

chapters 1, 7 and 10

means, volatilities and correlations of five asset classes, each mean with its error bar

An optimizer with constraints

chapters 11 and 12

a Markowitz optimizer: the calmest mix for each target return and the one with the highest Sharpe ratio, fully invested, no short positions and at most 40 % in each class

Monte Carlo scenarios for the future

chapters 4, 5, 9 and 15

Value at risk and expected shortfall

chapters 16 and 17

The risk report

chapters 8, 13 and 17

The whole project, all five steps with all of their code, is part of the course.

Solutions to the Chapter Questions

Chapters 1 to 6

Try each question on its chapter page first, with paper and pen. Each solution gives the answer in bold and then the way to it, step by step.

1 Returns

1.1 −25 %, then +33.3 %.

$R = 60/80 - 1 = -0.25$. The way back starts from the lower price: $R = 80/60 - 1 = 0.333$. A fall always needs a bigger rise to undo it.

1.2 It is 1 % lower than at the start. The log returns add up to ln 0.99 = −0.0101.

$(1 + 0.10)(1 - 0.10) = 0.99$, a total return of −1 %. $\ln 1.1 = 0.0953$ and $\ln 0.9 = -0.1054$ add up to −0.0101, which is $\ln 0.99$. Adding the log returns gives the true total, while adding the percents gives a wrong 0 %.

2 Interest and Present Value

2.1 You end with $1,124.86.

$FV = 1{,}000 \times 1.04^3 = 1{,}000 \times 1.124864 = 1{,}124.86$. Plain interest would give 1,120, so 4.86 dollars come from interest on interest.

2.2 $453.51 today, and only $428.67 at 8 %.

$PV = 500 / 1.05^2 = 500 / 1.1025 = 453.51$. At 8 %: $500 / 1.1664 = 428.67$. A higher rate means a bigger factor to divide by, so a fixed payment later is worth less today.

3 Expected Value

3.1 The expected value is $3.50, so you would pay at most $3.50.

Each face has the probability 1/6. $E[X] = (1 + 2 + 3 + 4 + 5 + 6)/6 = 21/6 = 3.5$. At a price of $3.50 the game is fair, and at any lower price the edge is yours.

3.2 −2.7 cents, the same as a bet on red.

$E[X] = 1/37 \times 35 + 36/37 \times (-1) = -1/37 = -0.027$. On red: $18/37 - 19/37 = -1/37$ as well. The two bets swing very differently, while the average loss per dollar is the same.

4 The Random Walk

4.1 $96, $98, $100, $102 and $104. The most likely is $100, with a probability of 6/16 = 37.5 %.

Four flips give $2^4 = 16$ equally likely paths. Two ups and two downs happen on 6 of them, three ups on 4, four ups on 1, and the same for downs: 1, 4, 6, 4, 1 out of 16.

4.2 No, long rises and falls are common in a random walk.

In chapter 4, a walk with a fair coin climbed from $100 to $125, fell to $86 and ended at $128 after 1,000 steps, and that walk was picked to show it. Without any picking, long runs are just as common: in a long fair walk, about 41 % of all walks spend more than 90 % of the time on one side of the start. A single trend proves little. You need many separate tries, or a proper test, before you trust a pattern.

5 The Square Root Rule

5.1 About 15.9 % over a year, and 2 % over 4 days.

The variances of the days add up. Over n days the variance is n times as large, so $\sigma_n = 1\ \% \times \sqrt{n}$. $\sqrt{252} = 15.87$ and $\sqrt{4} = 2$.

5.2 Independent ups and downs partly cancel each other.

The variance of a sum of independent steps is the sum of their variances, so it grows like n. The standard deviation is its square root and grows like $\sqrt{n}$. If every step went the same way, the spread would grow like n.

6 The Bell Curve

6.1 About 2.3 % of all years, roughly one year in 44.

−24 % is $8 - 2 \times 16$, two standard deviations below the average. About 95 % lie within two of them, so 5 % lie outside, half of them below: 2.5 %. The exact value of the bell curve is 2.28 %.

6.2 The central limit theorem: a sum of many small independent steps follows the bell curve.

A yearly log return is the sum of 252 daily log returns. Whatever the shape of one day, the sum of many independent days comes close to the bell. Real daily moves are less independent and have more big jumps than the model, so the real tails are heavier than the bell (chapter 16).

Solutions to the Chapter Questions

Chapters 7 to 12

7 Volatility

7.1 The average is 0 %, and the volatility is 1.90 % a day.

The five returns add up to 0, so $\bar{r} = 0$. The squared distances are 1, 4, 9, 0 and 4, together 18. Then 18/5 = 3.6 and $\sqrt{3.6} = 1.90$.

7.2 No, it measures swings up and down alike.

A price that often jumps up has a high volatility and may still never lose. Volatility also says little about rare crashes, which the bell curve underrates (chapter 16). Volatility is still a useful first number for risk.

8 The Sharpe Ratio

8.1 Fund B, with 0.50 against 0.40.

$SR_A = (9 - 3)/15 = 0.40$ and $SR_B = (6 - 3)/6 = 0.50$. Fund A earns more, while fund B pays more excess return for each unit of volatility.

8.2 The mix earns 6 % with a volatility of 7.5 %, and its Sharpe ratio is 0.40.

$\mu_w = 0.5 \times 9 + 0.5 \times 3 = 6$ % and $\sigma_w = 0.5 \times 15 = 7.5$ %. SR = (6 − 3)/7.5 = 0.40, the same as fund A. Mixing with the safe account moves along the same line and keeps the ratio.

8.3 The yearly Sharpe ratio is about 1.59.

Over a year the average excess return grows 252 times, while the volatility grows only $\sqrt{252}$ times by the square root rule. So the ratio grows by $252/\sqrt{252} = \sqrt{252} = 15.87$, and $0.1 \times 15.87 = 1.59$.

9 Volatility Drag

9.1 At −25 %, and at −4 %.

$(1 + 0.5)(1 - 0.5) = 1 - 0.25 = 0.75$, and $(1.2)(0.8) = 1 - 0.04 = 0.96$. The loss is s^2, so it grows fast with the size of the swing.

9.2 About 8 % a year, and about 2 % a year (as log growth rates).

$g \approx 0.10 - 0.5 \times 0.20^2 = 0.10 - 0.02 = 0.08$. With 40 %: $0.10 - 0.5 \times 0.16 = 0.02$. Twice the volatility gives four times the drag.

10 Correlation

10.1 The correlation is 0.5.

$\rho = 0.02 / (0.20 \times 0.20) = 0.02 / 0.04 = 0.5$.

10.2 No, correlation only sees links along a straight line.

Let X be −1, 0 or +1, each with probability 1/3, and let $Y = X^2$. Then Y depends fully on X. The averages are $\mu_X = 0$ and $\mu_Y = 2/3$, so with the formula of chapter 10, $\mathrm{Cov}(X, Y) = E[X(Y - 2/3)] = E[X^3] - 2/3 \times E[X]$. Here X^3 is again −1, 0 or +1, so both averages are 0, Cov = 0 and $\rho = 0$.

11 Diversification

11.1 The portfolio swings 10 % a year.

With $\rho = 0$ only the own part is left: $\sigma_p^2 = 0.04/4 = 0.01$, so $\sigma_p = 0.10$.

11.2 13.2 % with 4 of them, and 10 % as the limit.

$\sigma_p^2 = 0.04 \times (1/4 + 3/4 \times 0.25) = 0.04 \times 0.4375 = 0.0175$, so $\sigma_p = 13.2$ %. With very many: $\sigma\sqrt{\rho} = 0.20 \times 0.5 = 10$ %. The shared part of the swings never goes away.

12 The Efficient Frontier

12.1 75 % in the first and 25 % in the second.

With $\rho = -1$ the formula gives $w_1^* = (0.09 + 0.03)/(0.01 + 0.09 + 0.06) = 0.12/0.16 = 0.75$. Check: 0.75×10 % $- 0.25 \times 30$ % $= 0$, the two swings cancel.

12.2 No other portfolio has a higher average return for the same volatility.

Any point below the efficient frontier can be improved, with more return for the same volatility or less volatility for the same return. Which point on the frontier suits you depends on how much risk you want to carry, so no single mix is right for everyone.

Solutions to the Chapter Questions

Chapters 13 to 17

13 Beta and the CAPM

13.1 Beta is 1.2, and the expected move is about −4.8 %.

$\beta = \rho\,\sigma_i/\sigma_m = 0.6 \times 30/15 = 1.2$. The expected move is $1.2 \times (-4\ \%) = -4.8\ \%$.

13.2 The CAPM expects 10 % a year.

$E[R] = 3\ \% + 1.4 \times (8\ \% - 3\ \%) = 3\ \% + 7\ \% = 10\ \%$.

13.3 Specific risk can be mixed away at no cost.

In a large portfolio the own parts of many stocks cancel out. Investors who hold such a portfolio carry only the market part, so in the model only that part earns a premium.

14 The Kelly Criterion

14.1 Kelly bets 20 % of the money.

With $b = 1$: $f^* = 2p - 1 = 2 \times 0.6 - 1 = 0.2$.

14.2 The Kelly share is 10 %. Half Kelly keeps about three quarters of the growth with much smaller swings.

$f^* = 0.4 - 0.6/2 = 0.1$. The growth per bet is $g(0.10) = 0.0097$ at full Kelly and $g(0.05) = 0.0073$ at half Kelly, about three quarters of it. Real probabilities are only estimates, and betting too much hurts far more than betting too little.

15 Monte Carlo Simulation

15.1 You need 160,000 paths.

$\mathrm{SE} = \sigma/\sqrt{M}$. An error four times smaller needs $\sqrt{M}$ four times larger, so M grows 16 times: $16 \times 10{,}000 = 160{,}000$.

15.2 So that every run gives the same numbers, and anyone can check them.

With the same seed the program draws the same random numbers, so a result can be repeated, tested and compared. To see how much a result depends on luck, run it again with other seeds.

16 Fat Tails

16.1 About once in 13,800 years, roughly 14,000.

$P(X < \mu - 5\sigma) = 2.87 \times 10^{-7}$ per day. Per year: $252 \times 2.87 \times 10^{-7} = 7.2 \times 10^{-5}$, so about once in 13,800 years. Real markets have seen several such days within a few decades, a sign of fat tails.

16.2 The one with kurtosis 9: more of its swings come as rare, very large moves.

Kurtosis weighs the fourth power of the distance from the average, so big moves count far more. Same volatility with a higher kurtosis means many calm days and a few violent ones, the fat tails that a VaR from the bell curve (chapter 17) underrates.

17 Value at Risk

17.1 The VaR is about $23,300.

$\mathrm{VaR} = z\,\sigma\,V = 2.3263 \times 0.01 \times 1{,}000{,}000 = 23{,}263$. On one day in 100 the loss should be larger.

17.2 The expected shortfall is about $26,700.

$\mathrm{ES} = \varphi(z)/0.01 \times \sigma\,V = 0.026652/0.01 \times 10{,}000 = 26{,}652$. On the worst 1 % of days the loss is about $26,700 on average.

17.3 VaR says where the worst 1 % of days begin, and nothing about how bad they get.

On its bad days one portfolio may lose a little more than the VaR, while the other loses ten times as much. Expected shortfall averages the losses beyond the VaR and shows the difference.

Formula Sheet

The main formulas of all seventeen chapters, each with its line in plain words and, below it, every symbol on its own. Chapters 10 to 17 follow on page 27.

1 Returns p. 4

$$R = \frac{S_1}{S_0} - 1$$

The return is the new price over the old price, minus one.

- R = the return, as a decimal: 0.10 means +10 %
- S_1 = the price at the end
- S_0 = the price at the start

$$r = \ln \frac{S_1}{S_0}$$

The log return is the natural logarithm of the new price over the old price.

- r = the log return
- ln = the natural logarithm, the “ln” button on a calculator

2 Interest and Present Value p. 5

$$PV = \frac{FV}{(1+i)^T}$$

The value today divides the future value by 1 + i once for every year.

- PV = the present value, worth today
- FV = the future value, in T years
- i = the interest rate per year
- T = the number of years

3 Expected Value p. 6

$$E[X] = \sum_k p_k \, x_k$$

The expected value weighs every outcome by its probability and adds them up.

- $E[\,]$ = the expected value
- X = a random outcome, such as the win of one bet
- Σ = add up over all outcomes
- k = the number of an outcome
- p_k = the probability of x_k
- x_k = possible value number k of X

4 The Random Walk p. 7

$$E[S_n] = S_0 + n\,\mu$$

On average the walk moves by the drift in every step.

- $E[\,]$ = the expected value, the average over many walks
- S_n = the price after n steps
- S_0 = the price at the start
- n = the number of steps
- μ = the average of one step, the drift

5 The Square Root Rule p. 8

$$\mathrm{Var}(S_n - S_0) = n\,\sigma^2 \quad \Longrightarrow \quad \sigma_n = \sigma\sqrt{n}$$

For independent steps the variances add, so the spread grows with the square root of n.

- Var = the variance
- S_n = the price after n steps
- S_0 = the price at the start
- n = the number of steps
- σ = the standard deviation of one step
- $\Rightarrow$ = so it follows that
- σ_n = the spread after n steps

6 The Bell Curve p. 9

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

This formula gives the height of the bell curve at the value x.

- f = the height of the curve at the value x
- x = a value on the axis
- σ = the standard deviation
- π = the number 3.14159…
- exp = exp(y) is 2.718… to the power y
- μ = the average

7 Volatility p. 10

$$\sigma_{\text{year}} = \sigma_{\text{day}}\,\sqrt{252}$$

The square root rule turns daily into yearly volatility.

- σ_{year} = the yearly volatility
- σ_{day} = the daily volatility

8 The Sharpe Ratio p. 11

$$\mathrm{SR} = \frac{\mu - r_f}{\sigma}$$

The Sharpe ratio is the extra return over the safe account for each unit of volatility.

- SR = the Sharpe ratio
- μ = the average yearly return
- r_f = the risk-free rate, the interest of a safe account
- σ = the yearly volatility

9 Volatility Drag p. 12

$$g \approx \mu - \tfrac{1}{2}\,\sigma^2$$

The growth rate you keep is the average minus half the variance. This approximation works best for small steps.

- g = the growth rate you keep each year in the end
- $\approx$ = about equal to
- μ = the average yearly return
- σ = the yearly volatility

Formula Sheet, Continued

The main formulas of chapters 10 to 17, in the same form as on page 26.

10 Correlation p. 13

$$\rho = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \, \sigma_Y} \qquad -1 \le \rho \le 1$$

Correlation is the covariance, scaled to a number between minus one and one.

- ρ = the correlation (rho)
- Cov = the covariance
- X = a first random outcome, such as a daily return
- Y = a second random outcome
- σ_X = the standard deviation of X
- σ_Y = the standard deviation of Y

11 Diversification p. 14

$$\sigma_p^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2\, w_1 w_2\, \rho\, \sigma_1\sigma_2$$

This is the variance of a mix of two investments.

- σ_p = the volatility of the portfolio
- w_1 = the weight, the share of the money in the first investment
- σ_1 = the volatility of the first investment
- w_2 = the weight of the second investment
- σ_2 = the volatility of the second investment
- ρ = the correlation between the investments

12 The Efficient Frontier p. 15

$$w_1^* = \frac{\sigma_2^2 - \rho\,\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2 - 2\rho\,\sigma_1\sigma_2}$$

This weight gives the calmest mix of two investments. It can lie outside 0 to 1, which means borrowing one of the two and selling it, called selling short.

- w_1^* = the weight of the first investment in the calmest mix
- σ_2 = the volatility of the second investment
- ρ = the correlation of the two investments
- σ_1 = the volatility of the first investment

13 Beta and the CAPM p. 16

$$\beta_i = \frac{\mathrm{Cov}(R_i, R_m)}{\sigma_m^2} = \rho\,\frac{\sigma_i}{\sigma_m}$$

Beta says how many percent a stock moves when the market moves one percent.

- β_i = the beta of stock i
- Cov = the covariance
- R_i = the return of stock i
- R_m = the return of the market
- σ_m = the volatility of the market
- ρ = the correlation of stock and market
- σ_i = the volatility of stock i

$$E[R_i] = r_f + \beta_i \left(E[R_m] - r_f\right)$$

The CAPM expects the risk-free rate plus beta times the market risk premium.

- $E[\,]$ = the expected value
- r_f = the risk-free rate

14 The Kelly Criterion p. 17

$$f^* = p - \frac{1-p}{b} \qquad (b = 1\colon\ f^* = 2p - 1)$$

The Kelly share f^* is the share that makes this growth largest.

- f^* = the Kelly share, with the largest growth
- p = the probability to win a bet
- b = what one dollar wins, here 1 for even money

15 Monte Carlo Simulation p. 18

$$\mathrm{SE} = \frac{\sigma}{\sqrt{M}}$$

Four times as many paths halve the standard error.

- SE = the standard error of the estimate
- σ = the standard deviation of one simulated result
- M = the number of simulated paths

16 Fat Tails p. 19

$$\kappa = \frac{E\left[(X-\mu)^4\right]}{\sigma^4} \qquad \kappa_{\text{bell}} = 3$$

Kurtosis measures how heavy the tails are.

- κ = the kurtosis (kappa); many programs report $\kappa - 3$, the excess kurtosis, which is 0 for the bell curve
- $E[\,]$ = the expected value
- X = a daily return
- μ = the average daily return
- σ = the standard deviation of the daily returns
- κ_{bell} = the kurtosis of the bell curve

17 Value at Risk p. 20

$$\mathrm{VaR}_c = z_c\,\sigma\,V \qquad \mathrm{ES}_c = \frac{\varphi(z_c)}{1-c}\,\sigma\,V$$

For a bell curve with average 0 and c = 99 %, z is 2.33, and the expected shortfall is 2.67 standard deviations. A 99 % VaR over one year is exceeded in about one year in a hundred.

- VaR_c = the value at risk at the confidence level c, as a share of the portfolio or an amount of money
- z_c = how many standard deviations the cutoff lies from the average
- σ = the daily volatility
- V = the value of the portfolio
- ES_c = the expected shortfall at the confidence level c
- φ = the height of the standard bell curve, with average 0 and standard deviation 1
- c = the confidence level, such as 0.99

Words in This Book

Every new word appears here page by page, in the order the book explains it, next to the page where it stands in bold.

quantitative finance math and code for money and markets 2

return gain or loss as a share of the start 4

log return ln of new over old price 4

stock a piece of a company with a price 4

time value of money a dollar today is worth more than later 5

present value a future payment, worth today 5

interest rate yearly payment on money, in % 5

compound interest interest on interest 5

discounting going back from later to today 5

expected value average result over many tries 6

law of large numbers many tries approach the average 6

probability how likely, from 0 to 1 6

edge small advantage per bet 6

random walk random steps added up 7

path one possible history 7

drift average move per step 7

square root rule spread grows like $\sqrt{n}$ 8

independent one step says nothing about the next 8

variance average squared distance 8

standard deviation typical distance, written σ 8

bell curve shape of many small random steps 9

central limit theorem many small steps add up to a bell curve 9

tail the far ends of the curve 9

volatility standard deviation of returns 10

trading day a day with trading, about 252 a year 10

annualize turn a daily number into a yearly one 10

Sharpe ratio extra return per unit of volatility 11

risk-free rate interest of a safe account 11

excess return return above the risk-free rate 11

volatility drag gap between average and kept growth 12

growth rate what you keep per year 12

correlation how closely two move, from −1 to 1 13

covariance how two move together, unscaled 13

diversification spreading money to calm swings 14

portfolio a mix of investments 14

weight share of money in one part 14

efficient frontier best return for each risk 15

minimum variance portfolio the calmest mix 15

beta how strongly a stock follows the market 16

CAPM expected return rises with beta 16

market portfolio all stocks together 16

specific risk risk you can mix away 16

market risk premium market return above the risk-free rate 16

security market line the CAPM line 16

Kelly criterion best share to bet 17

median the middle result 17

ruin almost all lost (here under $1 of $100) 17

Monte Carlo simulation many simulated futures 18

seed start number for random numbers 18

standard error typical error of an estimate 18

percentile value below a given share 18

fat tails more big moves than the bell curve 19

kurtosis how heavy the tails are 19

value at risk loss exceeded on few days, such as 1 in 100 20

confidence level share of days VaR holds 20

expected shortfall average loss beyond VaR 20

Where to Go from Here

You have walked through the basics of quant finance, from one return to the risk of a whole portfolio.

Now you can

- ✓ Measure returns, discount money and work with expected values.
- ✓ Compare investments fairly with volatility, the Sharpe ratio and correlation.
- ✓ Explain why the spread of a price grows with the square root of time, and where the bell curve fails.
- ✓ Use beta, the CAPM, Kelly, Monte Carlo and value at risk, and answer first interview questions on each.

Your next steps

1 Try a full module for free

Module 16 of the course, Monte Carlo Simulation, is open to everyone. It takes chapter 15 of this book to its full depth, with every step derived, the complete Python code and exercises with worked solutions.

quantfinancetogo.com/free

2 See what the full course adds

This book	The course
17 chapters, one page each	62 modules and growing, about 20 pages each, code included
formulas in plain words	every formula derived step by step
one line of Python per chapter	the complete Python code, exercises with worked solutions and a mini project in every module
questions to test yourself	interview prep by role and by category, each question with a worked solution
the first half of one project, page 22	the complete project, plus three more tools for your CV
the basics, from returns to value at risk	four levels, up to options, factor investing and faster numerical methods

The full course

All four levels, the projects for your CV and the interview prep are on the website.

quantfinancetogo.com

© 2026 QuantFinanceToGo. Text, formulas, code and pictures in this book are my own. All data is made up or simulated by my own programs, and no number in this book describes a real investment. All rights reserved. This book is for learning only and gives no financial, investment or other professional advice.