Climate Risk Hedging

Thomas Lorans

May 17, 2024

Contents

1 Introduction **1**

2 Modelling Returns **3**
2.1 Returns as Random Variables 3
2.2 Expected Value and Variance of Returns 6
2.3 Returns Algebra 7
2.4 Sample Returns 9
2.5 Exercises 9
2.6 Solutions 9

3 Regression **11**
3.1 β estimation 11
3.2 Matrix Algebra 12
3.3 Matrix Form 12
3.4 OLS vs. GLS 12
3.5 Exercises 12
3.6 Solutions 12

4 Mimicking the Market Portfolio **13**

5 Time Series **15**
5.1 Unconditional and Conditional Expectations 15
5.2 White Noise 15
5.3 Means and Trends 15

6 Tracking Portfolio for News **17**

7 Climate Hedge Targets **19**
7.1 Climate News Series 19
7.2 Climate News Innovation 19
7.2.1 Climate News Shock 20
7.3 Portfolio Exposure to Climate News Innovations 20
7.3.1 Multifactor Regression 20
7.3.2 Climate News Innovations as a Risk Factor 20

8 Climate Risk Mimicking Portfolio **21**

Chapter 1

Introduction

We consider alternative approaches for climate risk hedging. All approaches share the same goal: to be long stocks that do well in periods with unexpectedly bad news about climate risks, and short stocks that do badly in those scenarios.

Chapter 2

Modelling Returns

2.1 Returns as Random Variables

We model stock returns as *random variables*. A random variable can take one of many values, with an associated probability. For example, the gross return on a stock can be one of four values as shown in Table 2.1.

R	π
1.1	0.6
1.2	0.1
0.7	0.25
0.0	0.05

Table 2.1: Example of a gross return distribution.

Each value is a possible *realization* of the random variable. You can experiment with this in Python using the following code:

```
import numpy as np

# Define the possible returns and their probabilities
returns = np.array([1.1, 1.2, 0.7, 0.0])
probabilities = np.array([0.6, 0.1, 0.25, 0.05])

# Generate a random return
print(np.random.choice(returns, size=1, p=probabilities))
```

Of course, stock returns can take on many more values than just four, but this is a simple example.

The *distribution* of the random variable is a listing of the values it can take, along with their associated probabilities. For example, the distribution of the random variable in Table 2.1 is:

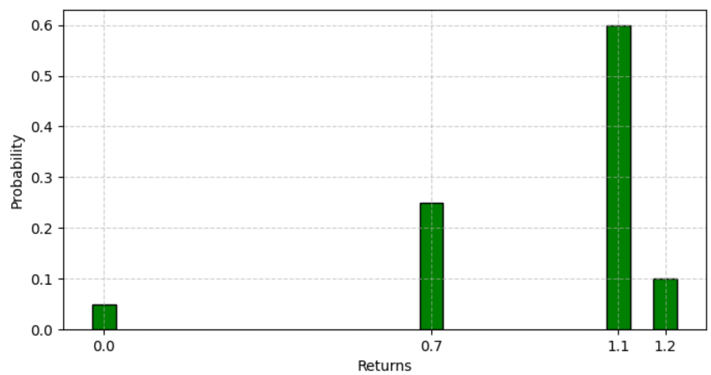

Figure 2.1: Example of a random variable distribution.

Another way to think of a random variable is as a *function* that maps "states of the world" to real numbers. For example, the random variable in Table 2.1 could be:

R	States of the world	π
1.1	New product works, competitor burns down	0.6
1.2	New product works, competitor ok.	0.1
0.7	Only old products work	0.25
0.0	Factory burns down, no insurance.	0.05

Table 2.2: Random variable as a function mapping states of the world to real numbers.

The probability really describes the external events that define the states of the world. Usually, we can't name those events, so we just use the probabilities that the stock return takes on various values.

In the end, all random variables have a discrete number of values, as in our example. Stock prices are only listed to 1/8 dollar, all payments are rounded to the nearest cent, etc. However, we often think of *continuous* random variables, that can be any real number. Corresponding to the discrete probabilities above, we now have a continuous probability *density*, usually denoted $f(R)$. The density tells you the probability per unit of R. For example, $f(R_0)\Delta R$ tells you the probability that the return is between R_0 and $R_0 + \Delta R$.

A common assumption is that returns (or log returns) are normally distributed. This means that the density is given by the formula:

$$f(R) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(R-\mu)^2}{2\sigma^2}\right) \tag{2.1}$$

The graph of this function looks like this:

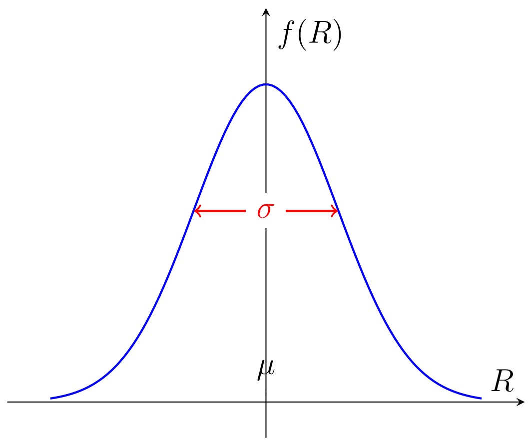

Figure 2.2: The probability density function of normally distributed returns

About 30% (31.73% to be exact) of the probability of a normal distribution is more than one standard deviation away from the mean. About 5% of the probability is more than two standard deviations away from the mean (in fact 4.55%, the 5% probability lines is at 1.96 standard deviation). That means that there is only one chance in 20 that the return will be more than two standard deviations away from the mean in a normally distributed world. In reality, stock returns have "fat tails", meaning that they are more likely to take on extreme values than a normal distribution would suggest.

You can experiment modelling returns with a normal distribution in Python:

```
import numpy as np

# Parameters for the normal distribution
mu = 1  # Mean
```

```
sigma = 0.05  # Standard deviation

print(np.random.normal(mu, sigma, 1))
```

2.2 Expected Value and Variance of Returns

Rather than plot the entire distribution, we usually summarize the behavior of a random variable with two numbers: the *mean* and the *variance*.

We denote the values that R can takes on a R_i, with associated probabilities π_i. The mean of R is given by:

$$E(R) = \sum_i R_i \pi_i \tag{2.2}$$

The mean is a measure of *central tendency*. It tells you where R is on average. A high mean stock return is obviously better than a low mean stock return.

The *variance* of R is given by:

$$\sigma^2(R) = E[(R - E(R))^2] = \sum_i \pi_i (R_i - E(R))^2 \tag{2.3}$$

Since squares are always positive, variance tells you how much R is far away from its mean. It measures the spread of the distribution. High variance is not a good thing. It will be our measure of *risk*.

Using our previous example, the Python code is:

```
import numpy as np

# Define the values that R can take (R_i) and their
                               associated probabilities (pi_i
                               )
values = np.array([1.1, 1.2, 0.7, 0.0])  # Example values R_i
probabilities = np.array([0.6, 0.1, 0.25, 0.05])  #
                               Corresponding probabilities
                               pi_i

# Calculate the expected value (mean) of R
expected_value = np.sum(values * probabilities)
print("Expected Value E(R):", expected_value)

# Calculate the variance of R
```

```
variance = np.sum((values - expected_value)**2 *
                           probabilities)
print("Variance of R:", variance)
```

The *covariance* is:

$$\mathrm{Cov}(R^a, R^b) = E[(R^a - E(R^a))(R^b - E(R^b))] = \sum_i \pi_i [(R^a_i - E(R^a))(R^b_i - E(R^b))] \tag{2.4}$$

It measures the tendency of two returns to move together. It's positive if they tend to move in the same direction, negative if one tends to go up when the other goes down. It's zero if they are independent.

The size of the covariance depends on the unit of measurement. For example, if we measure one return in cents, the covariance goes up by a factor of 100, even though the relationship between the two returns is the same. The *correlation coefficient* resolves this problem:

$$\mathrm{Corr}(R^a, R^b) = \frac{\mathrm{Cov}(R^a, R^b)}{\sigma(R^a)\sigma(R^b)} \tag{2.5}$$

The correlation coefficient is always between -1 and 1.

Keeping the same example as before but adding a second return, the Python code is:

```
import numpy as np

# Define the values and probabilities for two random
                           variables R^a and R^b
values_a = np.array([1.1, 1.2, 0.7, 0.0])  # Example values R
                           ^a_i
values_b = np.array([1.0, 1.3, 0.4, 0.1])  # Example values R
                           ^b_i
probabilities = np.array([0.6, 0.1, 0.25, 0.05])  #
                           Corresponding probabilities
                           pi_i

# Calculate the expected values (means) of R^a and R^b
expected_value_a = np.sum(values_a * probabilities)
expected_value_b = np.sum(values_b * probabilities)

# Calculate the covariance between R^a and R^b
```

```
covariance = np.sum(probabilities * (values_a -
                               expected_value_a) * (values_b
                               - expected_value_b))
print("Covariance between R^a and R^b:", covariance)

# Calculate the variances of R^a and R^b
variance_a = np.sum((values_a - expected_value_a)**2 *
                               probabilities)
variance_b = np.sum((values_b - expected_value_b)**2 *
                               probabilities)

# Calculate the standard deviations of R^a and R^b
std_dev_a = np.sqrt(variance_a)
std_dev_b = np.sqrt(variance_b)

# Calculate the correlation coefficient
correlation_coefficient = covariance / (std_dev_a * std_dev_b
                               )
print("Correlation Coefficient between R^a and R^b:",
                               correlation_coefficient)
```

2.3 Returns Algebra

We will have to do a lot of manipulation of random variables. For example, we will want to know what is the mean and standard deviation of a *portfolio* of two returns. To derive any of these rules, we have to keep in mind the definitions in the previous section.

First interesting fact: constants come out of expectations and expectations of sums are equals to sums of expectations. For example, if c and d are real numbers:

$$E(cR^a) = cE(R^a) \tag{2.6}$$

$$E(R^a + R^b) = E(R^a) + E(R^b) \tag{2.7}$$

$$E(cR^a + dR^b) = cE(R^a) + dE(R^b) \tag{2.8}$$

You can derive this by using the definition of expectations $E(R) = \sum_i R_i \pi_i$:

$$E(cR^a) = \sum_i \pi_i c R_i^a = c \sum_i \pi_i R_i^a = cE(R^a) \tag{2.9}$$

Variance of sums works like taking a square:

$$\sigma^2(R^a + R^b) = \sigma^2(R^a) + \sigma^2(R^b) + 2\mathrm{Cov}(R^a, R^b) \tag{2.10}$$

You can derive this with the definition of variance $\sigma^2(R) = E[(R - E(R))^2]$ and the definition of covariance $\mathrm{Cov}(R^a, R^b) = E[(R^a - E(R^a))(R^b - E(R^b))]$:

$$\begin{aligned}
\sigma^2(R^a + R^b) &= E[(R^a + R^b - E(R^a + R^b))^2] && (2.11)\\
&= E[(R^a + R^b - E(R^a) - E(R^b))^2] && (2.12)\\
&= E[(R^a - E(R^a) + R^b - E(R^b))^2] && (2.13)\\
&= E[(R^a - E(R^a))^2 + (R^b - E(R^b))^2 + 2(R^a - E(R^a))(R^b - E(R^b))] && (2.14)\\
&= \sigma^2(R^a) + \sigma^2(R^b) + 2\mathrm{Cov}(R^a, R^b) && (2.15)
\end{aligned}$$

With constants c and d:

$$\sigma^2(cR^a + dR^b) = c^2\sigma^2(R^a) + d^2\sigma^2(R^b) + 2cd\mathrm{Cov}(R^a, R^b) \tag{2.16}$$

The covariance works linearly with constants:

$$\mathrm{Cov}(cR^a, dR^b) = cd\mathrm{Cov}(R^a, R^b) \tag{2.17}$$

Normal distributions have an extra property. Linear combinations of normally distributed random variables are again normally distributed. Precisely, if R^a and R^b are normally distributed and:

$$R^p = cR^a + dR^b \tag{2.18}$$

then R^p is normally distributed. Its mean is:

$$E(R^p) = cE(R^a) + dE(R^b) \tag{2.19}$$

and its variance is:

$$\sigma^2(R^p) = c^2\sigma^2(R^a) + d^2\sigma^2(R^b) + 2cd\mathrm{Cov}(R^a, R^b) \tag{2.20}$$

2.4 Sample Returns

2.5 Exercises

2.6 Solutions

Chapter 3

Regression

We will run regression, for example of a return on the market return:

$$R_t = \alpha + \beta R_{m,t} + \epsilon_t \tag{3.1}$$

where R_t is the return on the asset, $R_{m,t}$ is the return on the market portfolio, α is the intercept, β is the slope coefficient and ϵ_t is the regression residual.

We may sometimes run multiple regressions of returns on the return of several portfolios, for example:

$$R_t = \alpha + \beta R_{m,t} + \gamma R_{p,t} + \epsilon_t \tag{3.2}$$

where R_p is the return on the portfolio of interest.

The generic form is:

$$y_t = \alpha + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \ldots + \beta_n x_{n,t} + \epsilon_t \tag{3.3}$$

3.1 β estimation

Starting with:

$$y_t = \alpha + \beta x_t + \epsilon_t \tag{3.4}$$

With the usual assumption that errors are uncorrelated, we have the right hand variables $E(\epsilon_t x_t) = 0$ and $E(\epsilon_t) = 0$.

Multiplying both sides by $x_t - E(x_t)$ and taking expectations:

$$\beta = \frac{Cov(y,x)}{Var(x)} \tag{3.5}$$

3.2 Matrix Algebra

3.3 Matrix Form

3.4 OLS vs. GLS

3.5 Exercises

3.6 Solutions

Chapter 4

Mimicking the Market Portfolio

Chapter 5

Time Series

5.1 Unconditional and Conditional Expectations

5.2 White Noise

5.3 Means and Trends

Chapter 6

Tracking Portfolio for News

Chapter 7

Climate Hedge Targets

One challenge with designing portfolios that hedge climate risks is that there is no unique way of choosing the hedge target. Climate change is a complex phenomenon and presents a variety of risks, including physical risks such as rising sea levels and transition risks such as the dangers to certain business models from regulations to curb emissions. Different risks may be relevant for different investors, and these risks are imperfectly correlated. In addition, climate change is a long-run threat, and we would thus ideally build portfolios that hedge the long-run realizations of climate risk, something difficult to produce in practice. To overcome these challenges, Engle *et al.* (2020) [**?**] argue that the objective of hedging long-run realizations of a given climate risk can be achieved by constructing a sequence of short-lived hedges against *news* (one-period innovation in expectations) about future realizations of the risk. Following the initial work of Engle *et al.* (2020), researchers have developped a variety of climate news series, capturing a variety of climate risks.

7.1 Climate News Series

Describe some climate new series.

7.2 Climate News Innovation

Building on the work of Engle *et al.* (2020), we use the $AR(1)$ innovations of each climate news series as the hedge targets. For a given climate news series

c, we denote these $AR(1)$ innovation in month t as $CC_{c,t}$.

7.2.1 Climate News Shock

7.3 Portfolio Exposure to Climate News Innovations

7.3.1 Multifactor Regression

7.3.2 Climate News Innovations as a Risk Factor

Chapter 8

Climate Risk Mimicking Portfolio

The mimicking portfolio approach combines a pre-determined set of assets into a portfolio that is maximally correlated with a given climate change shock, using historical data. To obtain the mimicking portfolios, we estimate the following regression model:

$$CC_t = wR_t + \epsilon_t \tag{8.1}$$

where CC_t denotes the (mean zero) climate hedge target in month t, w is a vector of N portfolio weights, R_t is the $N \times 1$ vector of demeaned excess returns and ϵ_t is the regression residual. The portfolio weights are estimated each month using a rolling window of T months of historical data.