An Introduction to Probability

Continuous Random Variables and Monte Carlo Simulation

Martin Summer

29 January, 2025

Continuous random variables and Monte Carlo Simulation

The Normal Distribution & Continuous Random Variables

In this lecture, we introduce one of the most important probability distributions, the **normal distribution**.

- So far, we have discussed **discrete random variables**, where the number of possible outcomes for X is **finite** or **countably infinite**.
- To discuss the **normal distribution**, we must consider cases where the number of outcomes for X is **uncountably infinite**, forming a **continuum**.
- This leads us to the concept of a **continuous random variable**.

Continuous Random Variables and Probability

Here, we discuss the most important concepts for practical work with continuous random variables. For a mathematically rigorous treatment, advanced techniques such as measure theory are required. Instead, we focus on applied and practical aspects.

Definition: Continuous Random Variable

A **continuous random variable** X can take on a continuum of possible values within a given range.

Why Continuous Random Variables?

- So far, we have studied **discrete random variables**, like coin flips.
- Many practical applications involve **continuous** variables.
- Examples:
  - **Asset prices or returns**: A stock's price could, in principle, take any value in $[0, \infty)$.
  - **Task completion times, lengths, weights**: These are often modeled as continuous.
- But are these *really* continuous? Or is this just a useful **modeling assumption**?

Are Stock Prices Really Continuous?

- In theory, stock prices can take any value in $[0, \infty)$.
- However, **prices are quoted in cents (or pennies)**—there is a smallest unit!
- We encountered a similar assumption in **Lecture 1** when discussing **sample spaces**.
- Despite this, treating prices as **continuous** makes modeling **easier** and **more powerful**.

Is Time Discrete or Continuous?

- You might argue: **“Time is measured in hours, minutes, or seconds.”**
- True, but we can refine our measurements **indefinitely**.
- The limitation is not in time itself but in our **measuring instruments**.
- **Unlike stock prices, time does not “jump”—it is truly continuous.**

The Fundamental Shift

- For **discrete** random variables, we assign probabilities to **specific outcomes**.
- Example: $P(X = 3)$ for a discrete variable might be **positive**.
- But for **continuous** random variables:

> **!** Important
>
> For every continuous random variable X,
> we have $P(X = x) = 0$ for all x.

- Why? Let's explore this with a **live demonstration in R**.

Generating a Continuous Random Variable

- In R, we can generate random numbers from a **uniform distribution** over $[0, 1]$.
- Example: Generate 10 random values from a **continuous** distribution.

```r
set.seed(123)
runif(10, 0, 1)
```

```
 [1] 0.2875775 0.7883051 0.4089769 0.8830174 0.9404673 0.04
 [8] 0.8924190 0.5514350 0.4566147
```

- These numbers are drawn from a continuum—**infinitely many possible values**.

What is the Probability of Picking a Specific Value?

- Suppose we pick one of these numbers: 0.4566147.
- What is the probability of selecting **exactly** this number?
- Let's investigate by simulating **one million** random values.

```
uniform_rv <- runif(10^6, 0, 1)
mean(uniform_rv == 0.4566147)
```

```
[1] 0
```

- What do we get? **Zero!**

Why is the Probability Zero?

- The interval $[0, 1]$ contains an **infinite number of points**.
- Any chosen number has **infinitely many values above and below it**.
- If we assigned a **positive probability** to any single number:
  - The **sum of all probabilities** would exceed 1.
  - This would **violate probability laws**.

! Important

A continuous random variable **never** assigns positive probability to a single value.
Instead, probability is defined over **intervals**.

Key Takeaways

- **Continuous random variables** differ fundamentally from discrete ones.
- For a continuous variable X:
  - $P(X = x) = 0$ for any single value x.
  - We can only assign probability to **intervals**.
- This concept is crucial for **understanding probability density functions (PDFs)**,
  which we will explore next.

Probability as an Area Under the Curve

- For **discrete random variables**, probabilities are assigned to **individual points**.

- For **continuous random variables**, probability is determined by **area under a curve**.

- Example: Consider a **uniform random variable** $X \sim U[0, 1]$.

- What is $P(0 \leq X \leq 1/4)$?

    - Using simulated numbers:

```r
mean(0 <= uniform_rv & uniform_rv <= 1/4)
```

```
[1] 0.250841
```

- The result is **25%**, as expected.

Confirming with the CDF

- We can use R's **cumulative distribution function (CDF)** for verification.

```r
punif(1/4)
```

```
[1] 0.25
```

- The CDF tells us:
  $P(X \leq x) = F(x)$
- So, $P(0 \leq X \leq 1/4) = F(1/4) - F(0) = 0.25 - 0 = 0.25$.

Visualizing the Probability Density Function (PDF)

- **For continuous random variables, probability is an area under a curve**.
- This is **fundamentally different** from discrete random variables.
- Consider the **density function** for a uniform distribution:

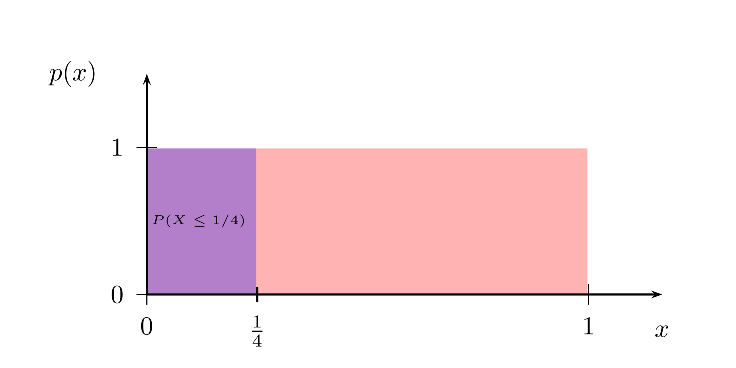


Figure 1: With continuous random variables, probabilities are areas

The Probability Density Function (PDF)

Definition

For a continuous random variable X with density function $f(x)$: 1. $f(x) \geq 0$, for all x. 2. $\int_{-\infty}^{\infty} f(x)\, dx = 1$ (Total probability is 1). 3. The probability that X falls in an interval (a, b) is:

$$P(a < X < b) = \int_a^b f(x)\, dx$$

- The **height of the PDF does not represent probability** directly.
- **Probability is always an area under the curve!**

The Cumulative Distribution Function (CDF)

Definition

The **cumulative distribution function (CDF)** shows the probability that X takes a value less than or equal to x:

$$F(x) = P(X \leq x)$$

For any $a < b$:

$$P(a < X < b) = F(b) - F(a) = \int_a^b f(x)\,dx$$

- The CDF gives the **total probability up to a certain point**.
- It **accumulates probability** as we move along the distribution.

Key Conceptual Shift: From Discrete to Continuous

- **In discrete probability**, $P(X = x)$ can be positive.
- **In continuous probability**, $P(X = x) = 0$ for any single value.
- Probability is determined by **integrating the density function** over an interval.
- The **PDF and CDF** allow us to compute probabilities effectively.

The Normal and the Log-Normal Distribution

The Normal Distribution

- The **normal distribution** is the most fundamental probability distribution.
- Its **bell-shaped curve** represents natural variability in many domains:
  - Heights of people
  - Measurement errors
  - Stock returns
  - The **Central Limit Theorem**
- Mathematically elegant, yet deeply connected to real-world data.

Why the Normal Distribution?

- The normal distribution is **simple and universal**.
- Defined by just **two parameters**:
  - **Mean** (μ) – controls the center.
  - **Variance** (σ^2) – controls the spread.
- Forms the basis for **statistical inference** and **stochastic modeling**.

Definition: The Normal Distribution

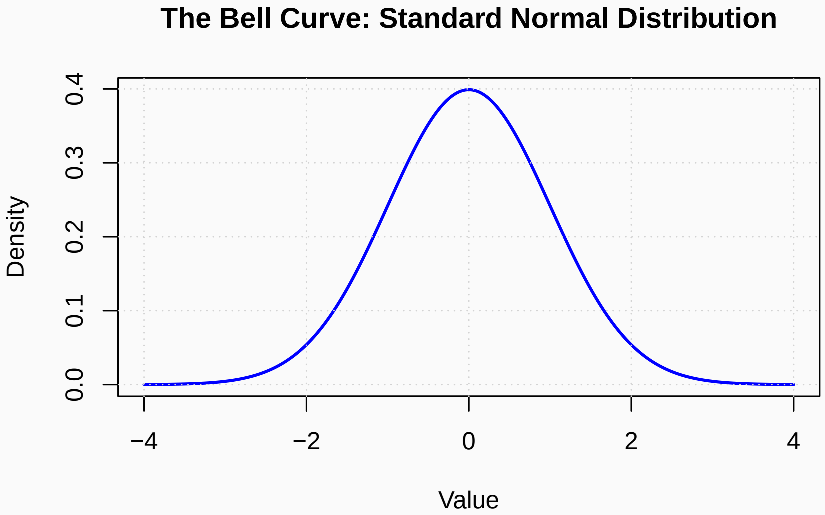
Probability Density Function (PDF)

For a normal random variable $X \sim N(\mu, \sigma^2)$:

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

- The **bell curve** is fully determined by μ and σ.
- It is **symmetric** around the mean.

Visualizing the Bell Curve



- The **height of the curve is not a probability—probabilities are areas** under it!

Standardizing the Normal Distribution

Standard Normal Transformation

The **standard normal distribution** is a special case with $\mu = 0$ and $\sigma^2 = 1$. Any normal random variable $X \sim N(\mu, \sigma^2)$ can be converted to the **standard normal**:

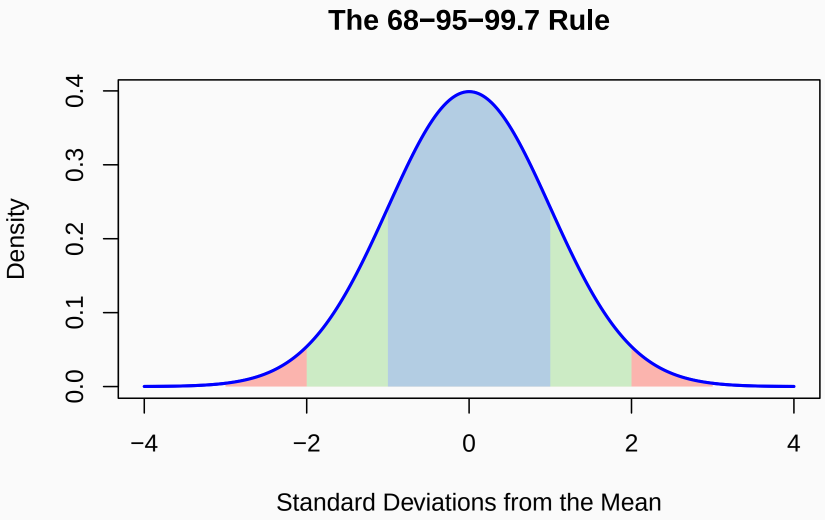
$$Z = \frac{X - \mu}{\sigma}, \quad Z \sim N(0, 1).$$

- Standardization allows us to **compare different normal distributions**.
- Probability tables and `pnorm()` in R use this transformation.

The 68-95-99.7 Rule

- The normal distribution has a **predictable concentration of probability**:
  - **68%** of values fall within **1 standard deviation** of the mean.
  - **95%** within **2 standard deviations**.
  - **99.7%** within **3 standard deviations**.
- This property allows **quick assessments of probabilities**.

The 68-95-99.7 Rule vizualized



Why is This Important?

1. **Universal Applicability**:
   - Heights, IQ scores, stock returns—all follow normal-like patterns.
2. **Quick Normality Check**:
   - The 68-95-99.7 rule helps diagnose normality in data.
3. **Decision-Making**:
   - Identifies **unusual** values in datasets.
   - Used in **quality control, finance, and risk management**.

When the Normal Distribution Doesn't Fit

- Some quantities—like **stock prices**—**cannot be negative**.
- The **lognormal distribution** is useful when data grows **multiplicatively**.

Definition: Lognormal Distribution

If $X \sim N(\mu, \sigma^2)$, then $Y = \exp(X)$ follows a **lognormal distribution**:

$$f(y, \mu, \sigma) = \frac{1}{y\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln y - \mu)^2}{2\sigma^2}\right), \quad y > 0.$$

Why the Lognormal Distribution for Stocks?

- **Stock prices can't be negative**.
- **Returns are typically modeled as normal, so prices follow a lognormal distribution**.
- This is the basis for **Geometric Brownian Motion (GBM)**, widely used in finance.

Geometric Brownian Motion (GBM)

A **stochastic process** modeling stock prices:

$$dS_t = \mu\, S_t\, dt + \sigma\, S_t\, dW_t$$

where: - μ = expected return - σ = volatility - W_t = Wiener process (Brownian motion)

Why Use Logarithmic Returns?

- Instead of simple returns:

$$R = \frac{S_t - S_0}{S_0}$$

- We use **log returns**:

$$r = \ln\left(\frac{S_t}{S_0}\right)$$

Advantages

1. **Additivity**: Log returns sum over time.
2. **Handles compounding effects naturally**.
3. **Better statistical properties** (often closer to normality).

Key Takeaways

- The **normal distribution** is central to probability and statistics.
- The **68-95-99.7 rule** provides a simple way to assess variability.
- When dealing with **positive-valued** quantities like stock prices, the **lognormal distribution** is more appropriate.
- **Geometric Brownian Motion (GBM)** is a key model for stock prices.

Next: Applying these distributions to **real-world financial data** in R.

Stock Returns vs. The Lognormal Distribution

- We now **compare real S&P 500 stock returns** to the **lognormal model**.
- **Objective**:
  - Retrieve **historical stock prices**.
  - Compute **log-returns**.
  - Overlay **empirical distribution** with a **fitted normal distribution**.
- **Why?**
  - To assess how well the normal/lognormal models describe financial returns.

Retrieving S&P 500 Stock Data

- We fetch **historical prices** using `tidyquant`:

```r
library(tidyquant)

# Retrieve S&P 500 historical data
sp500_data <- tq_get("^GSPC", from = "2015-01-01",
                     to = "2020-12-31",
                     get = "stock.prices")
```

- Compute **daily log-returns**:

```r
sp500_prices <- sp500_data$adjusted
log_returns <- diff(log(sp500_prices))
```

Stock Prices vs. Log-Returns: Which Model?

- **Stock Prices (S_t) follow a Lognormal Distribution**
  - Prices **can't be negative**.
  - Price changes **compound multiplicatively**.
  - If prices follow **Geometric Brownian Motion**:

$$S_t = S_0 \exp(X_t)$$

  where X_t is **normally distributed**.
- **Log-Returns (r_t) follow a Normal Distribution**
  - Log-returns are **additive** over time.
  - Defined as:

$$r_t = \ln\left(\frac{S_t}{S_{t-1}}\right)$$

  - **Mistake**: **Don't fit a lognormal model to log-returns!**

Empirical vs. Theoretical Normal Distribution

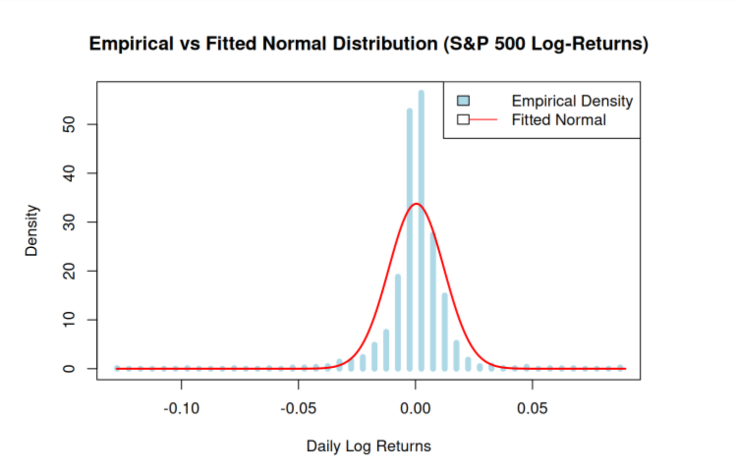


Figure 2: Fat Tails in the distribution of daily log returns

What do we observe?

- The **center** fits well.
- **Extreme returns** seem more frequent than the normal model predicts.

Do Stock Returns Have Fat Tails?

- **Fat tails** mean extreme returns happen **more often** than a normal model predicts.
- Let's **zoom into the left tail** to examine extreme negative returns.

Do Stock Returns have fat tails?

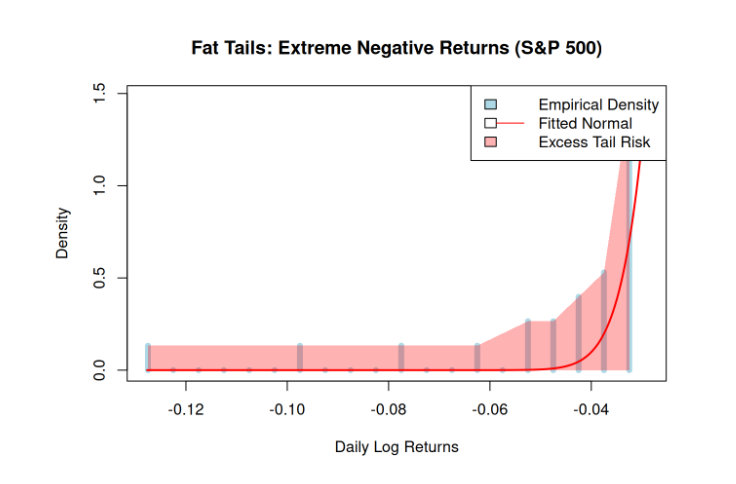


Figure 3: Zooming into the left tail

Key observation**:

- **More extreme losses than expected** under a normal model.
- The **normal assumption underestimates risk** in financial markets.

Normal Model vs. Reality: A Risk Management Perspective

- **The 68-95-99.7 Rule** predicts:
  - **99.7% of returns** should be within **3 standard deviations**.
  - **Only 0.15%** should be beyond **-3** .
- **Reality**:
  - In real stock data, **0.99%** of returns fall beyond **-3** .
  - This is **6× more than expected** under a normal model.

Implication

- Standard risk models (e.g., **Value at Risk (VaR)**) may **underestimate extreme losses**.
- **Tail risk must be explicitly accounted for** in risk management.

Asymmetry in Stock Returns

- **Stock returns are often negatively skewed**.
- This means **large crashes** happen more often than **large rallies**.
- Let's quantify skewness in our S&P 500 data:
- Here we have: 1.05.

! Important

- **A normal distribution has skewness = 0**.
- Our data shows a skewness of -1.05, confirming the **asymmetry**.

Summary & Implications for Financial Models

- **Log-returns** follow a **normal-like shape** in the center but have:
  - **Fat tails** – more extreme moves than predicted.
  - **Negative skewness** – crashes are more common than booms.
- **Risk Management Takeaways**:
  - Normal models **underestimate extreme downside risk**.
  - Alternative models like the **t-distribution** or **extreme value theory** may be more appropriate.

The inverse nomral and quantiles in risk management

Quantiles and Risk Management

- A fundamental question in risk management:

What is the worst-case loss I should expect, given a certain probability threshold?

- This is different from earlier probability questions:
  - Before: **Given a threshold, what is the probability of falling below it?**
  - Now: **Given a probability (e.g., 1%), what is the threshold where losses exceed this level?**

- This is called the **inverse problem** and is central to **Value at Risk (VaR)**.

Finding Quantiles: The Inverse Normal Function

- The **quantile function** (or **inverse CDF**) finds the **threshold** x for a given probability p:

$$P(X \leq x) = p$$

- If $X \sim N(\mu, \sigma^2)$, the quantile is:

$$x = F^{-1}(p)$$

- In R, we compute this using:

```
qnorm(0.01, mean = mean(log_returns),
      sd = sqrt(var(log_returns)))
```

Example:

- Finds the **1% quantile** for S&P 500 log returns. - Meaning: **Only 1% of days have worse losses** than this threshold.

Definition: p-Quantile

Definition: p-quantile

The p-**th quantile** of a distribution is the value x such that:

$$P(X \leq x) = p$$

For a normal distribution, this is:

$$x = F^{-1}(p)$$

where F^{-1} is the inverse **cumulative distribution function (CDF)**.

Value at Risk (VaR): A Key Risk Metric

- In portfolio management, we balance **risk and potential profit**.
- **Value at Risk (VaR)** quantifies the potential loss: **What is the maximum loss with a probability of h over a given time horizon?**
- Example: A bank wants to ensure the **probability of losing more than X%** of its capital is at most **1%**.
- **Key idea**: The **quantile of portfolio losses** is the **VaR**.

Definition: Value at Risk (VaR)

Definition: Value at Risk

For a position X and loss tolerance h, **VaR** is the smallest number V such that:

$$P(-X > V) \leq h$$

Equivalently:

$$P(-X \leq V) > 1 - h$$

Key interpretation:

- **VaR at 95% confidence ($h = 5\%$):**
  In 95% of cases, the loss will be less than V.
- **VaR at 99% confidence ($h = 1\%$):**
  Only in 1% of cases do we expect a loss worse than V.

Visualizing Value at Risk (VaR)

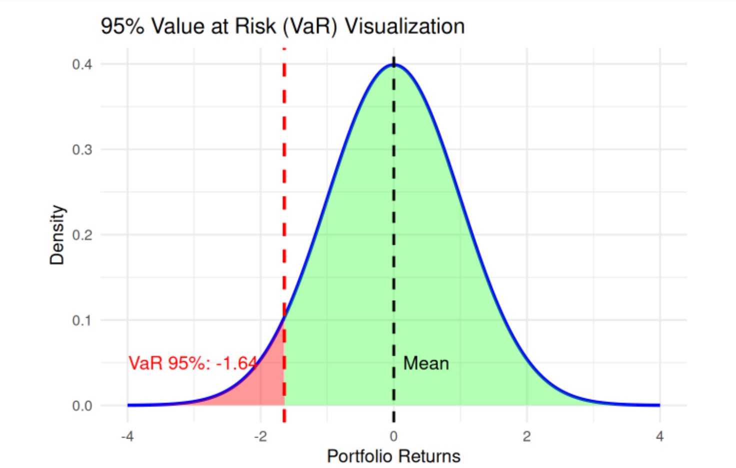


Figure 4: Zooming into the left tail

VaR for the Normal Distribution

Proposition: VaR for a Normal Distribution

If X follows a normal distribution with mean μ and standard deviation σ:

$$VaR_h(X) = -\sigma \cdot F_N^{-1}(h) - \mu$$

where F_N is the **CDF of the standard normal variable** (mean 0, standard deviation 1).

Example 1: Highly Liquid Portfolio

- Assets that can be **easily bought/sold** with minimal price impact.
- Examples:
  - Short-term U.S. Treasury bills
  - Large-cap ETFs (SPY, QQQ)
  - Highly traded forex pairs (EUR/USD)

Highly Liquid Portfolio

Short-term risk depends mainly on volatility: - Mean returns are **negligible** over a short horizon. - **Approximation**:

$$VaR_{95\%} \approx 1.65 \times \sigma$$

Let's check using R:

```
qnorm(0.05) * (-1)  # Should be ~1.65
```

```
[1] 1.644854
```

Computing VaR: Highly Liquid Portfolio

Assumptions

- **Daily volatility**: 1.5%
- **Mean return**: ~0%
- **95% confidence level**

R-example

```r
# Define parameters
sigma_liquid <- 0.015  # 1.5% daily volatility
mu_liquid <- 0         # Negligible mean return
alpha <- 0.05          # 95% confidence level

# Compute 1-day VaR
VaR_liquid <- qnorm(alpha,
                    mean = mu_liquid,
                    sd = sigma_liquid)

# Output result
sprintf("1-day 95%% VaR for a highly liquid portfolio: %.4f
        VaR_liquid, VaR_liquid * 100)
```

```
[1] "1-day 95% VaR for a highly liquid portfolio: -0.0247
```

Example 2: 10-Day VaR for a Pension Fund

- **Diversified investment**: stocks, bonds, alternatives.
- **Portfolio Value**: $500 million.
- **Scaling VaR**: We assume **normal returns** and apply:

$$VaR_{N\text{-day}} = VaR_{1\text{-day}} \times \sqrt{N}$$

Computing 10-Day VaR for a Pension Fund

```r
# Define parameters
sigma_fund <- 0.02    # 2% daily volatility
mu_fund <- 0.0002     # 0.02% daily return (assumed)
N <- 10               # 10-day horizon
portfolio_value <- 500 # $500 million
# Compute 1-day VaR
VaR_fund_1d <- qnorm(alpha,
                     mean = mu_fund,
                     sd = sigma_fund)
# Compute 10-day VaR using square-root scaling
VaR_fund_10d <- VaR_fund_1d * sqrt(N) * portfolio_value
# Output result
sprintf("10-day 95%% VaR for a $500M pension fund: $%.2f mi
```

```
[1] "10-day 95% VaR for a $500M pension fund: $-51.70 milli
```

Example 3: Diversification & VaR

Portfolio Composition - 50% in **stocks** (2.5% volatility). - 50% in **bonds** (1.0% volatility). - **Negative correlation**: -0.3.

Key Property of VaR:

- VaR satisfies **subadditivity**:
- $$VaR(A+B) \leq VaR(A) + VaR(B)$$
- Diversification **reduces overall risk**.

VaR for a Diversified Portfolio

```
# Define portfolio components
sigma_stocks <- 0.025  # 2.5% daily volatility
sigma_bonds <- 0.01    # 1.0% daily volatility
w_stocks <- 0.5        # 50% allocation to stocks
w_bonds <- 0.5         # 50% allocation to bonds
correlation <- -0.3    # Negative correlation
# Compute portfolio volatility
portfolio_volatility <- sqrt(
  (w_stocks * sigma_stocks)^2 +
  (w_bonds * sigma_bonds)^2 +
  2 * w_stocks * w_bonds * sigma_stocks * sigma_bonds * cor
# Compute portfolio VaR
VaR_portfolio <- qnorm(alpha, mean = 0, sd = portfolio_vola
# Compute individual VaRs
VaR_stocks <- qnorm(alpha, mean = 0, sd = sigma_stocks)
```

Output results

```
sprintf("VaR without diversification: $%.2f million", VaR_s
```

```
[1] "VaR without diversification: $-14.39 million"
```

```
sprintf("VaR with diversification: $%.2f million", VaR_port
```

```
[1] "VaR with diversification: $-9.86 million"
```

Summary: VaR in Practice

Key Lessons 1. **VaR measures extreme losses at a confidence level (e.g., 95%).** 2. **Higher volatility = Higher VaR** for a given portfolio. 3. **VaR scales with time** using the square-root rule. 4. **Diversification reduces risk** (subadditivity property). 5. **VaR underestimates risk when fat tails & skewness exist**.

Limitations

- **Ignores tail risk beyond the threshold** (e.g., extreme crashes).
- **Assumes normality** (which may not hold in real markets).
- **Doesn't capture worst-case scenarios like Black Swan events.**

Empirical Value at Risk (VaR)

Why use empirical VaR?

- **So far**, we assumed **normality** for VaR calculations.
- But **real-world returns** may **not** be normally distributed.
- **Empirical VaR** estimates risk **directly from historical data**.

Empirical Value at Risk

How?

1. **Sort historical returns** (smallest to largest).
2. **Find the 5% quantile** $\rightarrow$ Empirical **95% VaR**.
3. **No normality assumption** needed.

Example: Compute **10-day empirical VaR** using **weekly returns**.

Computing Empirical 10-Day VaR

```
# Convert daily log returns to 5-day log returns
log_returns_10d <- colSums(matrix(log_returns,
                                       nrow = 5, byrow = TRUE),
# Sort returns in ascending order
sorted_returns <- sort(na.omit(log_returns_10d))
# Compute empirical cumulative distribution function (ECDF
n <- length(sorted_returns)
ecdf_values <- seq(1, n) / n
# Identify empirical quantiles for 95% and 99% VaR
VaR_95_empirical <- sorted_returns[min(which(ecdf_values >=
VaR_99_empirical <- sorted_returns[min(which(ecdf_values >=
# Output results
sprintf("Empirical 95%% VaR: %.4f (10-day horizon)", VaR_95
sprintf("Empirical 99%% VaR: %.4f (10-day horizon)", VaR_99
```

Empirical VaR: Visualizing the CDF

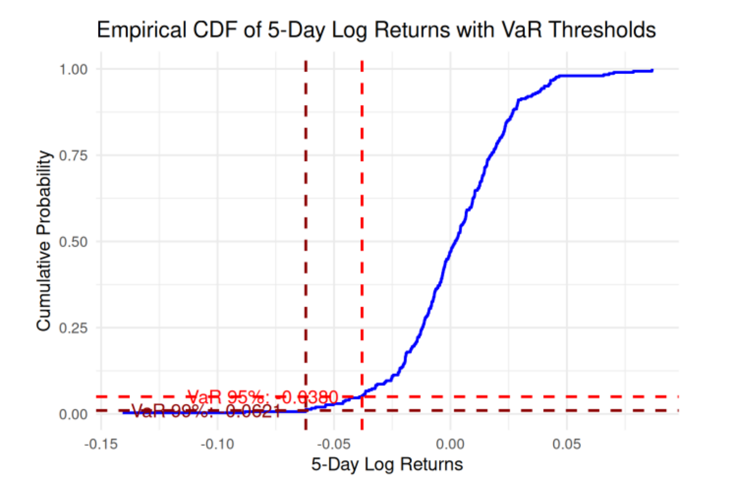


Figure 5: Empirical value at risk

Limitations of Empirical VaR

Potential Pitfalls - Limited historical data

- A few extreme weeks **dominate VaR estimates**.
- **Small sample sizes** = **unreliable tail estimates**.
- **99% VaR is especially problematic**
  - Requires **even fewer observations**.
  - Highly **sensitive to individual extreme weeks**.
- **Ignores extreme risks that haven't happened yet**
  - If an extreme event **hasn't occurred in the dataset**, **VaR misses it**.
  - **Example**: Long-Term Capital Management (LTCM) crisis.

Empirical vs. Parametric VaR: When to Use Which?

Feature	**Parametric VaR** (Normal)	**Empirical VaR**
Assumption	Normal distribution	No assumption
Computation	Uses mean/variance	Uses historical data
Captures Fat Tails?	No	Yes (if in data)
Works in Small Samples?	Yes (assumes normality)	No (unstable)
Use Case	Large, stable portfolios	Risk-sensitive trading

Data and Statistics

Data and Statistics in Finance

Key Question:
- So far, we have estimated **moments of log-returns** and plugged them into software. - Now, let's step back: **How reliable are these estimates?**

Probability vs. Statistics
- In **probability**, we **assume** a random process and derive consequences. - In **statistics**, we **observe data** and **infer** the underlying process.

Challenge:
- Some parameters (e.g., **volatility**) are **well-estimated** from data.
- Others (e.g., **expected return**) are **extremely unreliable**!

Period-Length Effects in Returns

Returns over time periods are multiplicative:

$$1 + r_y = (1 + r_1)(1 + r_2) \dots (1 + r_{12})$$

For **small** returns, we approximate:

$$1 + r_y \approx 1 + r_1 + r_2 + \cdots + r_{12}$$

Assume: - Returns r_i are **uncorrelated**. - All have **same expected return** $\bar{r}$ and variance σ^2.

Implication:

Implications: - Expected return grows **linearly**:

$$\bar{r_y} = 12\bar{r}$$

- **Variance grows linearly**:

$$\sigma_y^2 = 12\sigma^2$$

- **Standard deviation grows as $\sqrt{12}$**:

$$\sigma_y = \sqrt{12}\sigma$$

Scaling Returns Across Time

For a period length p (fraction of a year), we generalize:

$$\bar{r_p} = p\bar{r}_y, \quad \sigma_p = \sqrt{p}\sigma_y$$

What does this mean?

- **Expected return decreases linearly** as period shrinks.
- **Standard deviation grows slower** (only as $\sqrt{p}$).

Scaling Returns Across Time

The ratio $\frac{\sigma_p}{\bar{r}_p}$ skyrockets for short periods:

- At **yearly scale**: $\frac{\sigma_y}{\bar{r}_y} \approx 1.25$
- At **monthly scale**: $\frac{\sigma_{1/12}}{\bar{r}_{1/12}} \approx 4.3$
- At **daily scale**: $\frac{\sigma_{1/250}}{\bar{r}_{1/250}} \approx 19.8$

Short periods = Unstable return estimates!

Translating Annual Returns to Monthly & Daily Values

Assume:

- **Annual mean return**: $\bar{r}_y = 12\%$
- **Annual standard deviation**: $\sigma_y = 15\%$

Compute for **monthly returns** ($p = 1/12$):

$$\bar{r}_{1/12} = \frac{12\%}{12} = 1\%$$

$$\sigma_{1/12} = \frac{15\%}{\sqrt{12}} \approx 4.33\%$$

Ratio: $\frac{\sigma}{\bar{r}} = 4.3$

Translating Annual Returns to Monthly & Daily Values

Compute for **daily returns** ($p = 1/250$):

$$\bar{r}_{1/250} = \frac{12\%}{250} = 0.048\%$$

$$\sigma_{1/250} = \frac{15\%}{\sqrt{250}} \approx 0.95\%$$

Ratio: $\frac{\sigma}{\bar{r}} = 19.8$

Key Takeaway:
As **period length shrinks**, return estimates become much **less reliable**.

Estimating Expected Returns: A Fundamental Problem

We want to estimate the mean return $\bar{r}$. - Suppose we observe n **independent samples** of period returns.

Best estimate for the mean:

$$\hat{\bar{r}} = \frac{1}{n}\sum_{i=1}^{n} r_i$$

How accurate is this estimate? - Expected value of the estimate:

$$\mathbb{E}(\hat{\bar{r}}) = \bar{r}$$

- Standard deviation of the estimate:

$$\sigma_{\hat{\bar{r}}} = \frac{\sigma}{\sqrt{n}}$$

More data → Lower estimation error
Short periods → High estimation error

Why Expected Returns Are Nearly Impossible to Estimate

Let's compute the estimation error for different periods: - **For monthly returns ($p = 1/12$)** - $\bar{r} = 1\%$, $\sigma = 4.33\%$ - With **12 months of data**:

$$\sigma_{\hat{\bar{r}}} = \frac{4.33\%}{\sqrt{12}} = 1.25\%$$

Error $>$ True Mean! (Unusable estimate)

Why Expected Returns Are nearly Impossible to Estimate

- **For 4 years of data ($n = 48$)**

$$\sigma_{\hat{\hat{r}}} = \frac{4.33\%}{\sqrt{48}} = 0.625\%$$

Still high error (Not a reliable estimate)

Why Expected Returns Are Nearly Impossible to Estimate

- **For a good estimate (error $<$ 10% of mean)**:
  - We need $\sigma_{\hat{\bar{r}}} < 0.1\%$
  - Requires $n \approx 1875$ **samples**
  - **156 years of data!**

This is called the *historical blur* problem.

The Historical Blur Problem

Why can't we estimate expected returns accurately?

- **Volatility is large relative to the mean.**
- **More data helps, but not enough** – Even with **100+ years**, error remains large.
- **Using shorter periods doesn't help** – More samples, but each sample is worse.

The Historical Blur Problem

Implication:

- **We cannot estimate expected returns reliably using historical data alone!**
- Variance and covariances are **more stable**, but **expected returns are NOT**.

The Historical Blur Problem

Takeaways:

1. **Historical return data is useful for estimating RISK (volatility, covariance).**
2. **Expected returns are nearly impossible to measure.**
3. **Alternative approaches (e.g., factor models) are needed to estimate $\bar{r}$.**

Monte Carlo Simulation in Finance

Introduction to Monte Carlo Simulation in Finance

What is Monte Carlo Simulation?
- A computational method using **random sampling** to estimate uncertain outcomes.
- Essential for **complex probability models** where analytical solutions are difficult.

Why Monte Carlo?

- Used extensively in **risk estimation**, **option pricing**, and **portfolio modeling**.
- Enables **large-scale simulations** to model **market uncertainty**.
- Ecellent **application for performance optimization** in R.

Historical Origin

- Developed during the **Manhattan Project** (WWII).
- Named after the **Monte Carlo Casino**, highlighting its reliance on randomness.

This lecture:

- Simulating **future stock returns**.
- Estimating **Value at Risk (VaR)** using Monte Carlo.
- Optimizing computational performance in R.

Why Monte Carlo?

- Models **future risks**, not just historical losses.
- Can handle **non-normal return distributions**.
- Useful for **complex portfolios and derivative pricing**.

Step 1: Simulating Portfolio Returns

- **Assumption:** Log-returns follow a **normal distribution**
- Estimate **mean** (μ) and **standard deviation** (σ) from historical data
- Simulate **10,000 future return scenarios**

Simulating Portfolio Returns

```
# Load necessary packages
library(ggplot2)

# Set seed for reproducibility
set.seed(123)

# Simulated historical daily log-returns (e.g., stock index
historical_returns <- rnorm(250, mean = 0.0005, sd = 0.01)
```

Step 2: Estimating Value at Risk (VaR)

- **Extracting VaR from Monte Carlo simulation**
- *VaR is the quantile of the loss distribution**
- Take **left-tail quantile** (5% or 1%) to estimate worst-case losses

Estimating VaR

```r
# Compute portfolio losses
simulated_losses <- -simulated_returns

# Compute VaR at 95% and 99% confidence levels
VaR_95 <- quantile(simulated_losses, probs = 0.95)
VaR_99 <- quantile(simulated_losses, probs = 0.99)
```

Step 2: Interpreting VaR in Monetary Terms

Example:** Managing a **$10 billion** portfolio

- **95% VaR = 1.51%**
- **99% VaR = 2.14%**

Daily Loss Interpretation

- 95% chance **loss does not exceed $151 million** in a single day.
- 5% chance losses **exceed $151 million**.

Yearly Exceedance Frequency

- 5% of 250 trading days = **12–13 days per year** exceeding $151M loss.

Step 3: Optimizing Monte Carlo Simulation

- **Challenges in Monte Carlo Simulations**
- **Large-scale** simulations (millions of iterations) are **slow**
- Naive loops in R can **bottleneck performance**

Optimizing Performance

- **Vectorization**: Use efficient matrix operations
- **Parallel Computing (`future.apply`)**: Distribute tasks across CPU cores
- **Efficient Data Handling (`data.table`)**: Reduce memory overhead

Example:

```r
# Load parallel processing package
library(future.apply)

# Use parallel computation
plan(multisession)

# Monte Carlo function
monte_carlo_var <- function(n_sim, mu, sigma, confidence =
  simulated_losses <- -rnorm(n_sim, mean = mu, sd = sigma)
  return(quantile(simulated_losses, probs = confidence))
}

# Run parallel Monte Carlo simulation
n_sim <- 1e6
VaR_95_par <- future_sapply(1:10, function(x)
```

Comparing Parallel vs. Sequential Execution

- **Does parallelization improve performance?**
- **For small-scale simulations**: **No benefit**, overhead dominates.
- **For large-scale simulations**: **Parallel computing speeds up execution!**

Comparing Parallel vs. Sequential Execution

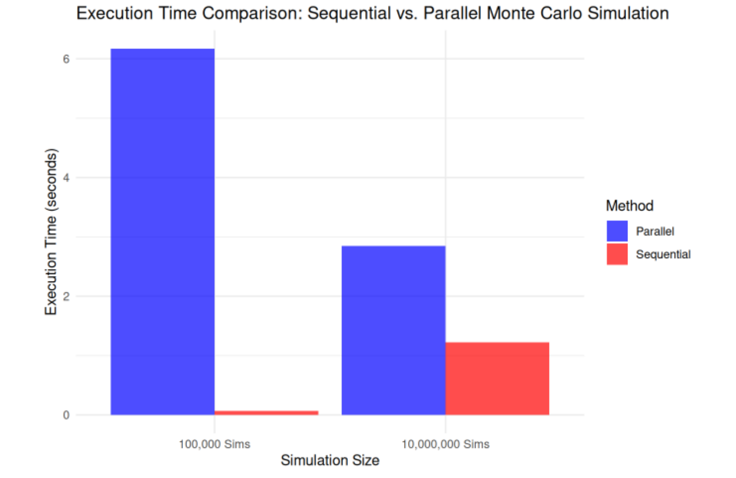


Figure 6: Empirical value at risk

Efficient Data Handling with `data.table`

Why use `data.table`?

-**Faster than `data.frame`** for large datasets -**Efficient memory usage**

Example:

```r
# Load package
library(data.table)

# Generate large dataset
n_sim <- 1e6
simulated_losses <- -rnorm(n_sim, mean = mu, sd = sigma)

# Store as data.table
dt_losses <- data.table(losses = simulated_losses)

# Compute quantile (VaR)
VaR_95_dt <- dt_losses[, quantile(losses, probs = 0.95)]
```

Result: `data.table` can be **5-10x faster** than `data.frame` for large datasets!

Summary

- **Monte Carlo simulation** models market uncertainty.
- **Value at Risk (VaR)** estimates potential portfolio losses.
- **Performance optimization** is key for large-scale simulations.
- **Parallel computing (`future.apply`)** accelerates Monte Carlo.
- **Efficient data handling (`data.table`)** improves speed.