Nonlinear Modeling: Monte Carlo simulation

Introduction to Statistical Modelling

Prof. Joris Vankerschaver

Monte Carlo simulation technique

- Computational algorithm, based on
  - Large number of simulations (thousands to millions)
  - Repeated random sampling
- Used for modeling of
  - Climate and environment
  - Nuclear reactions
  - Financial systems
  - Molecular dynamics
  - ...
- In this course:
  - Global sensitivity analysis
  - Uncertainty analysis

Front view of Casino de Monte-Carlo

Principles behind Monte Carlo

Algorithm:

1. Draw random parameter samples
2. For each sample, run simulation
3. Aggregate results of simulations into quantity of interest (e.g. mean)

Advantages:

- Easy to implement/understand
- Applicable to wide range of problems (model-agnostic)

Disadvantages:

- Relatively slow convergence (many simulations needed)
- Requires information about parameter distributions

Example: computing your savings

You just started work and you want to compute your financial situation in 12 months from now.

Income 🚀:

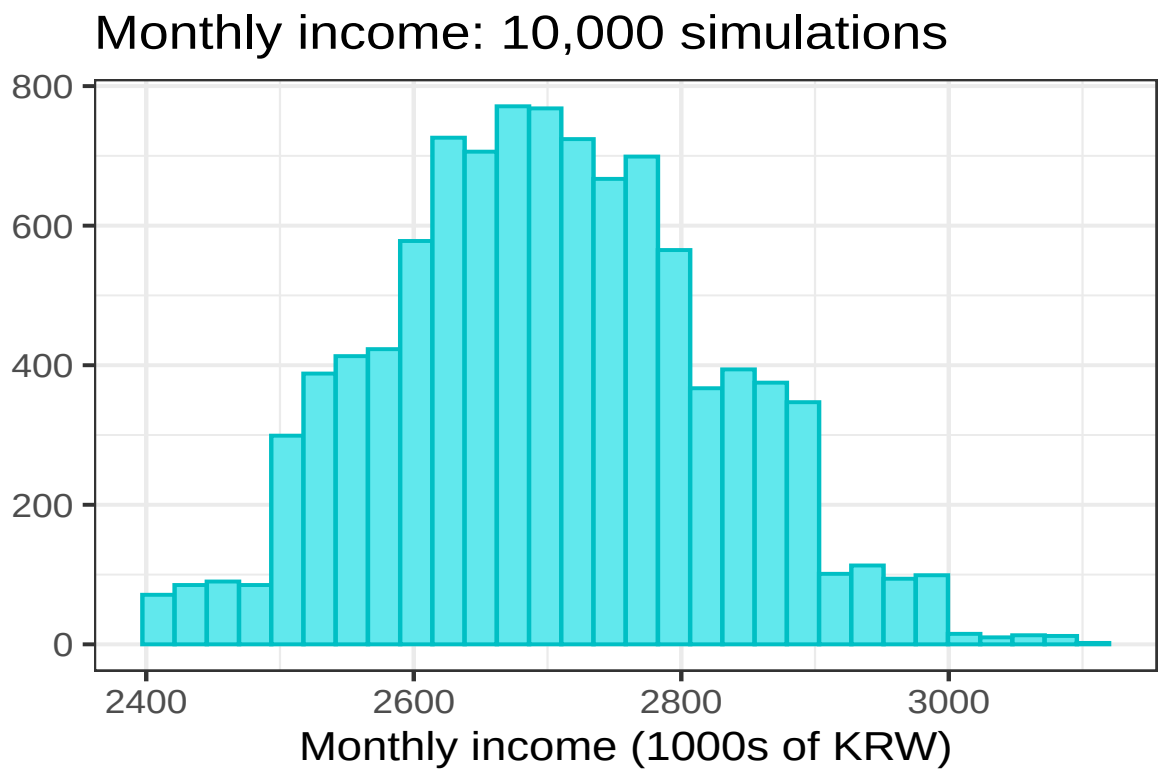
- Salary: Between 2,400,000 KRW and 2,600,000 per month (uniformly distributed).
- Tuition income:
  - Attendance: binomial distribution with $n = 5$ and $p = 0.4$
  - Each attending student pays you 100,000 KRW per month.

Expenditure 😢:

- Variable costs: normally distributed with mean 2,000,000 KRW and standard deviation 500,000 KRW
- Occasional large expense: 10% chance of extra 1,000,000 KRW expense

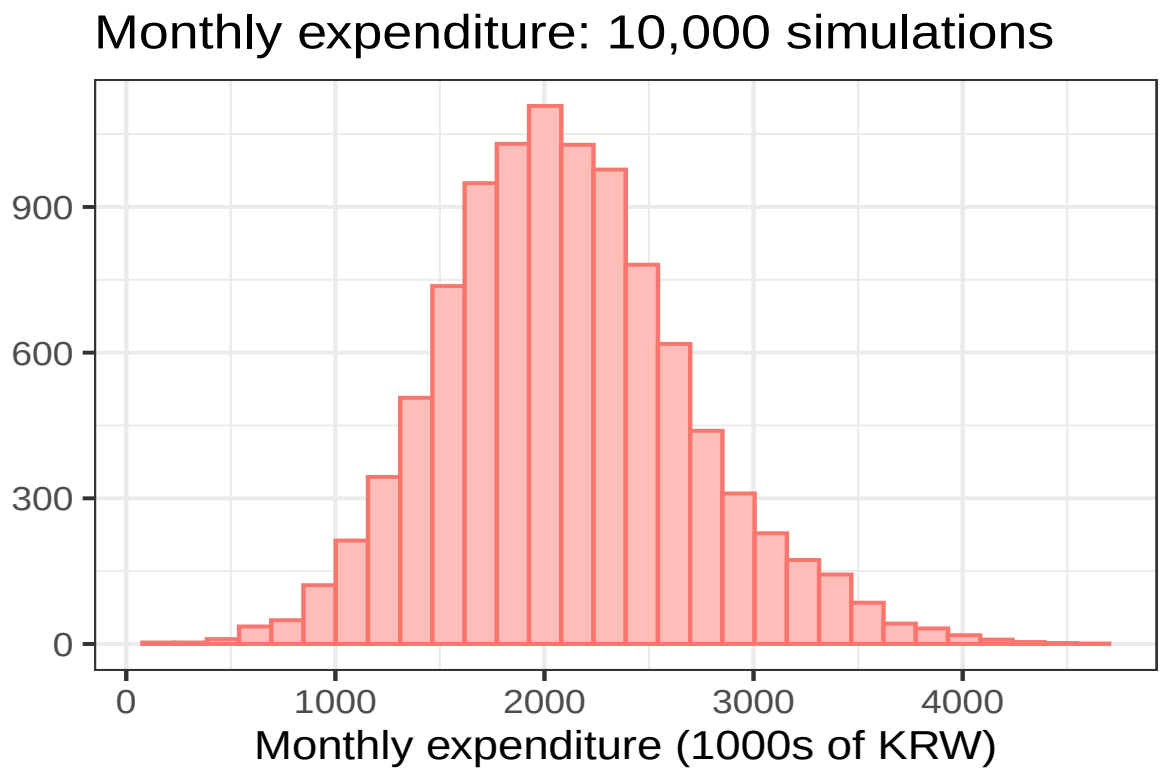
Simulating your income

```r
simulate_income <- function() {
  salary <- runif(1, min = 2400, max = 2600)
  tuition <- 100 * rbinom(1, size = 5, prob = 0.4)
  return(salary + tuition)
}
```

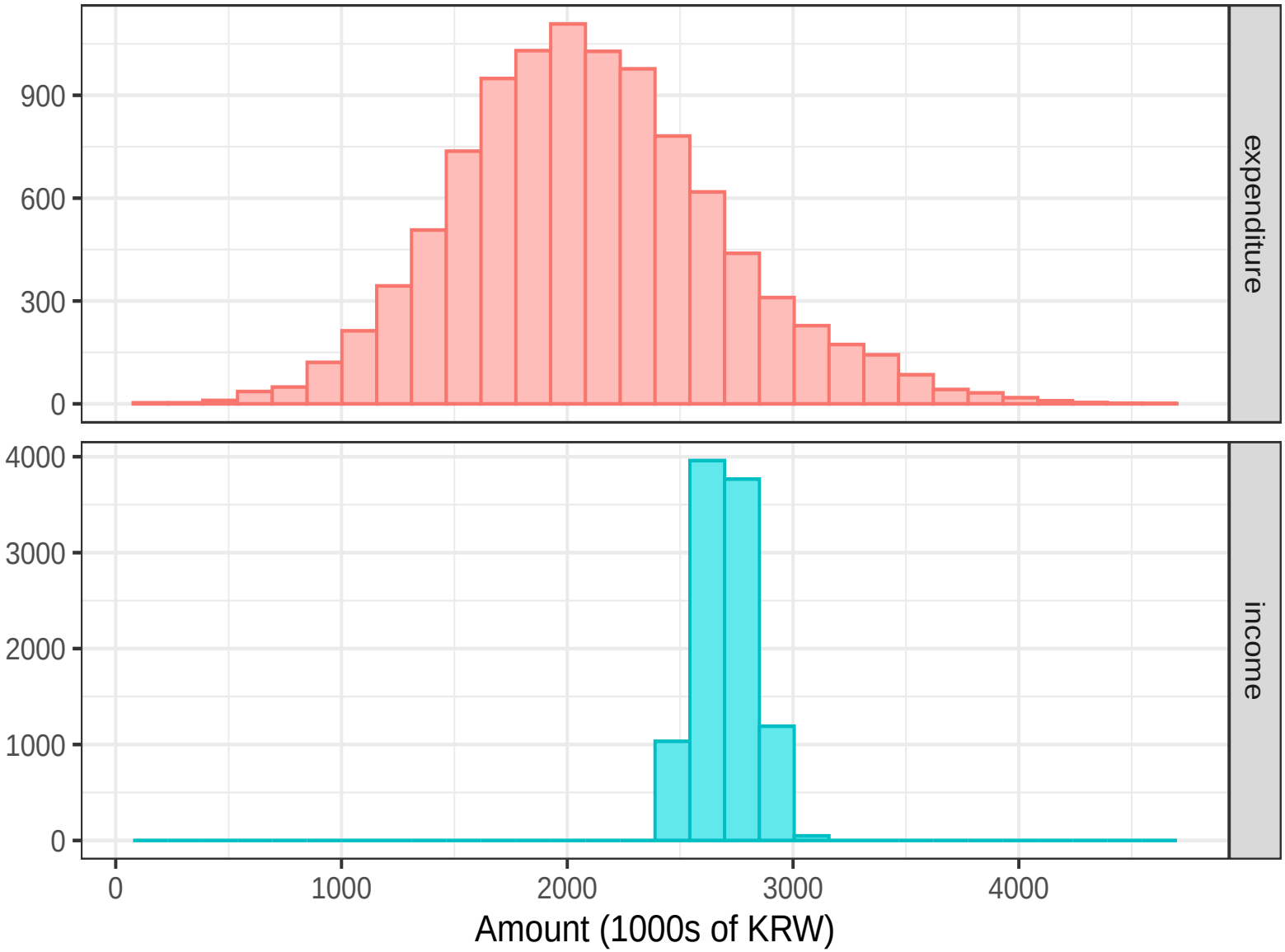


Simulating your expenditure

```r
simulate_expenditure <- function() {
  costs <- rnorm(1, mean = 2000, sd = 500)
  if (runif(1) < 0.10) {
    costs <- costs + 1000
  }
  return(costs)
}
```

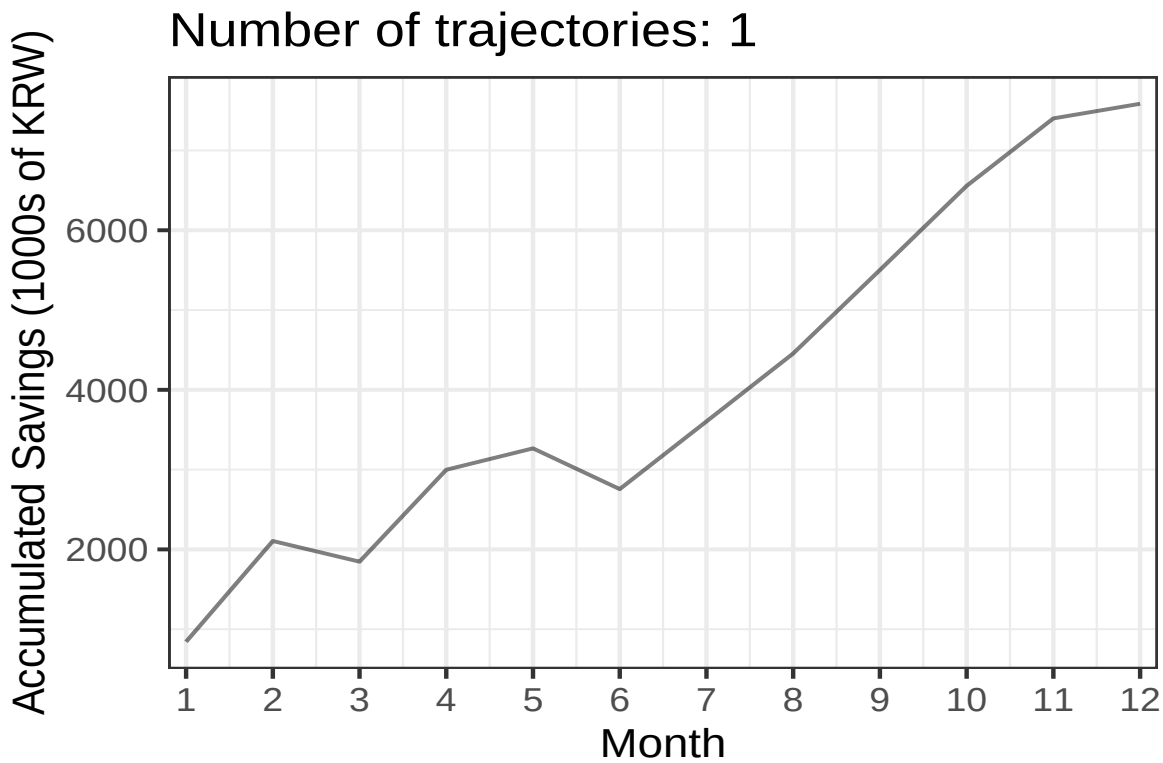


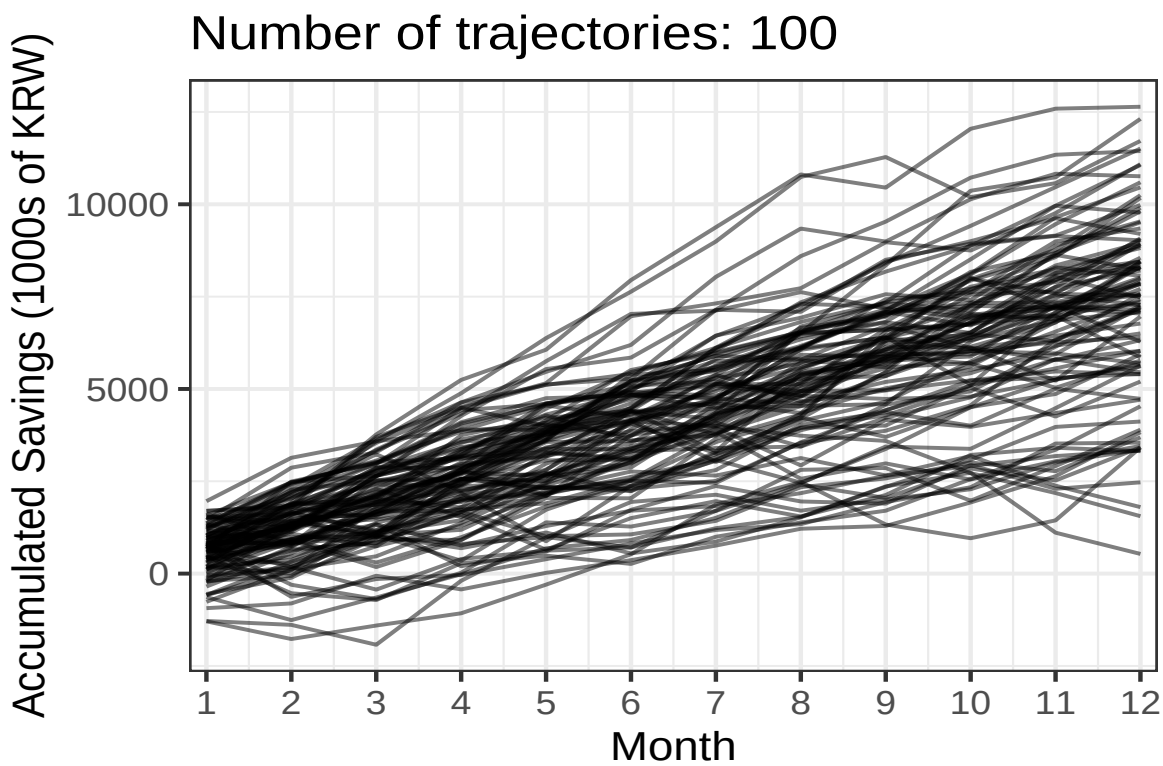
Comparing income and expenditure

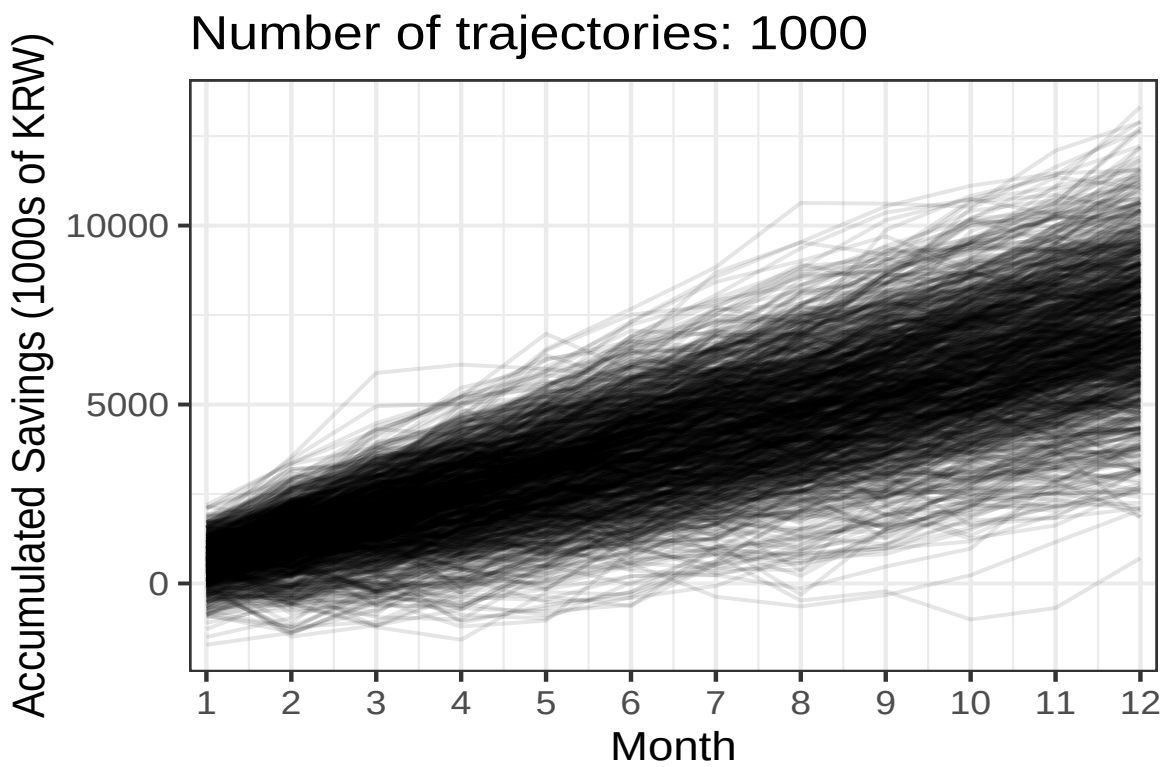


Simulating your savings over 12 months

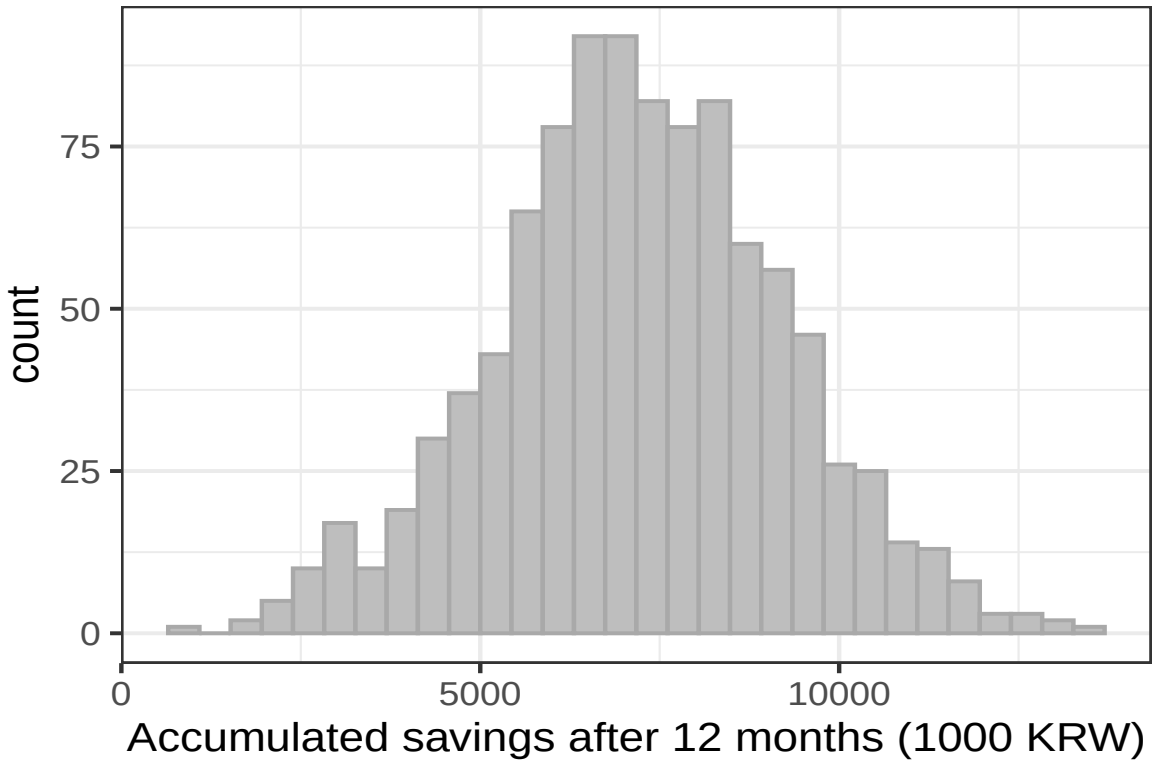
```
simulate_monthly_savings <- function() {
  return(simulate_income() - simulate_expenditure())
}
simulate_yearly_savings <- function() {
  return(cumsum(replicate(12, simulate_monthly_savings())))
}
```


Number of trajectories: 1
Accumulated Savings (1000s of KRW)
6000
4000
2000
1
2
3
4
5
6
7
8
9
10
11
12
Month


Number of trajectories: 100
Accumulated Savings (1000s of KRW)
10000
5000
0
1
2
3
4
5
6
7
8
9
10
11
12
Month


Number of trajectories: 1000
Accumulated Savings (1000s of KRW)
10000
5000
0
1
2
3
4
5
6
7
8
9
10
11
12
Month

Savings after 12 months



Probability of bankruptcy

Bankruptcy: your savings go below zero for at least one month.

```
simulate_bankruptcy <- function() {
  savings <- cumsum(replicate(12, simulate_monthly_savings()))
  return(any(savings < 0))
}

n <- 1000
bankruptcies <- replicate(n, simulate_bankruptcy())
```

- Out of 1000 simulations, 205 end in bankruptcy.
- Probability of bankruptcy: 20.5%

Summary

Monte Carlo simulations involve:

- Repeatedly simulating the process for various initial conditions or parameters
- Aggregating the results

Monte Carlo simulations are:

- Easy to implement but computationally intensive
- Very powerful