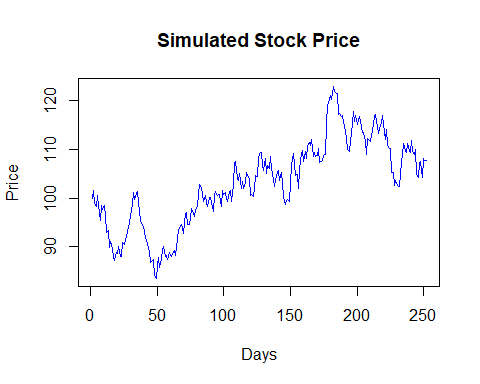
Week\_09

230701195

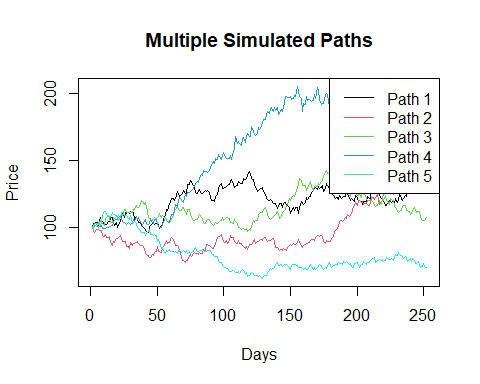
2025-04-19

# SIMULATION - MONTE CARLO METHOD

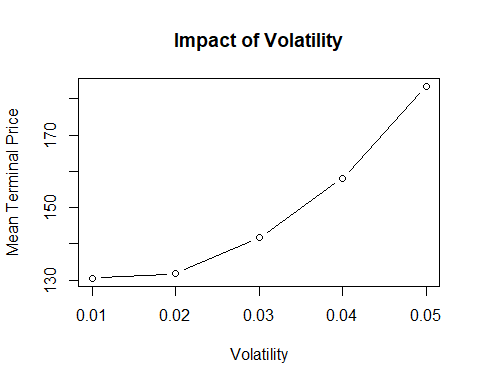
# Problem 1:  
   
# Simulate stock price with initial price S0 = 100, daily return mu = 0.001, volatility sigma = 0.02, and 252 trading days  
  
simulate\_stock <- function(S0, mu, sigma, T) {  
 prices <- numeric(T)  
 prices[1] <- S0  
 for (t in 2:T) {  
 prices[t] <- prices[t - 1] \* exp(rnorm(1, mean = mu, sd = sigma))  
 }  
 return(prices)  
}  
  
# Parameters  
S0 <- 100  
mu <- 0.001  
sigma <- 0.02  
T <- 252  
  
# Simulate and plot  
prices <- simulate\_stock(S0, mu, sigma, T)  
plot(prices, type = "l", col = "blue", main = "Simulated Stock Price", xlab = "Days", ylab = "Price")



# Problem 2:  
   
# Simulate 5 price paths for the stock price, based on the same parameters and simulate 252 days of trading  
  
simulate\_multiple <- function(S0, mu, sigma, T, n\_paths) {  
 paths <- matrix(NA, nrow = T, ncol = n\_paths)  
 for (i in 1:n\_paths) {  
 paths[, i] <- simulate\_stock(S0, mu, sigma, T)  
 }  
 return(paths)  
}  
  
# Parameters  
n\_paths <- 5  
  
# Simulate and plot  
paths <- simulate\_multiple(S0, mu, sigma, T, n\_paths)  
matplot(paths, type = "l", lty = 1, col = 1:n\_paths, main = "Multiple Simulated Paths", xlab = "Days", ylab = "Price")  
legend("topright", legend = paste("Path", 1:n\_paths), col = 1:n\_paths, lty = 1)



# Problem 3:  
# An analyst wants to understand how stock price volatility impacts the expected terminal price after 1 year.  
# Run simulations across different volatility levels and visualize the sensitivity.  
  
simulate\_stock <- function(S0, mu, sigma, T) {  
 prices <- numeric(T)  
 prices[1] <- S0  
 for (t in 2:T) {  
 prices[t] <- prices[t - 1] \* exp(rnorm(1, mean = mu, sd = sigma))  
 }  
 return(prices)  
}  
  
simulate\_terminal <- function(S0, mu, sigma, T, n\_sims) {  
 terminal\_prices <- numeric(n\_sims)  
 for (i in 1:n\_sims) {  
 prices <- simulate\_stock(S0, mu, sigma, T)  
 terminal\_prices[i] <- prices[T]  
 }  
 return(terminal\_prices)  
}  
  
# Parameters  
S0 <- 100  
mu <- 0.001  
T <- 252  
n\_sims <- 1000  
  
volatilities <- seq(0.01, 0.05, by = 0.01)  
results <- sapply(volatilities, function(sigma) mean(simulate\_terminal(S0, mu, sigma, T, n\_sims)))  
  
plot(volatilities, results, type = "b", main = "Impact of Volatility",   
 xlab = "Volatility", ylab = "Mean Terminal Price")



# Problem 4:  
   
# Simulate portfolio returns for 3 assets with different return and volatility parameters, and calculate optimal allocation  
  
simulate\_portfolio <- function(weights, mus, sigmas, T, n\_sims) {  
 portfolio\_returns <- replicate(n\_sims, {  
 prices <- sapply(1:length(mus), function(i) simulate\_stock(100, mus[i], sigmas[i], T))  
 portfolio\_value <- rowSums(prices \* weights)  
 (portfolio\_value[T] - portfolio\_value[1]) / portfolio\_value[1]  
 })  
 return(portfolio\_returns)  
}  
  
# Define parameters  
weights <- c(0.4, 0.4, 0.2)  
mus <- c(0.001, 0.002, 0.0008)  
sigmas <- c(0.02, 0.03, 0.015)  
  
# Simulate and analyze  
portfolio\_returns <- simulate\_portfolio(weights, mus, sigmas, T, n\_sims)  
mean(portfolio\_returns) # Expected return

## [1] -0.2637623

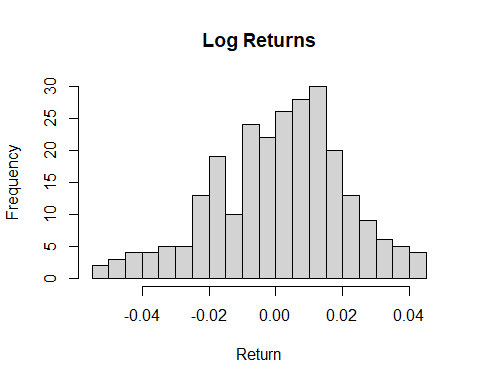
sd(portfolio\_returns) # Portfolio risk

## [1] 0.1686095

# Problem 5:  
   
# Estimate the 5% Value-at-Risk (VaR) for the portfolio by simulating terminal prices over 1,000 simulations  
  
# Define simulate\_terminal before using it  
simulate\_terminal <- function(S0, mu, sigma, T, n\_sims) {  
 terminal\_prices <- numeric(n\_sims)  
 for (i in 1:n\_sims) {  
 path <- simulate\_stock(S0, mu, sigma, T)  
 terminal\_prices[i] <- path[T]  
 }  
 return(terminal\_prices)  
}  
  
# Define VaR function  
VaR <- function(S0, mu, sigma, T, n\_sims, alpha = 0.05) {  
 terminal\_prices <- simulate\_terminal(S0, mu, sigma, T, n\_sims)  
 quantile(terminal\_prices, alpha)  
}  
  
# Parameters  
S0 <- 100  
mu <- 0.001  
sigma <- 0.02  
T <- 252  
n\_sims <- 1000  
  
# Estimate 5% VaR  
VaR\_5 <- VaR(S0, mu, sigma, T, n\_sims)  
cat("5% Value-at-Risk (VaR):", VaR\_5, "\n")

## 5% Value-at-Risk (VaR): 75.53933

# Problem 6:  
   
# Simulate daily log returns and analyze their distribution. The returns should be normally distributed with mean = mu and sd = sigma.  
  
simulate\_log\_returns <- function(mu, sigma, T) {  
 rnorm(T, mean = mu, sd = sigma)  
}  
  
# Simulate and analyze  
log\_returns <- simulate\_log\_returns(mu, sigma, T)  
hist(log\_returns, breaks = 30, main = "Log Returns", xlab = "Return")



mean\_log\_return <- mean(log\_returns) # Should approximate mu  
sd\_log\_return <- sd(log\_returns) # Should approximate sigma  
  
cat("Mean of log returns: ", mean\_log\_return, "\n")

## Mean of log returns: 0.0009233027

cat("Standard deviation of log returns: ", sd\_log\_return, "\n")

## Standard deviation of log returns: 0.01937098