stock_return_risk_analysis

Maowen Deng

2025-12-23

Stock Return & Risk Analysis

- Overview: This project develops an **interactive tool to compare the return and risk profile of two equity assets**. Users can:
  - Select tickers and date ranges
  - Choose return frequency (daily, weekly, monthly)
  - Explore return time series, scatterplots, and distributions
  - Evaluate risk metrics like Sharpe ratio and max drawdown
  - Examine rolling volatility and correlation
- Methodology: Daily, weekly, or monthly returns are calculated from adjusted closing prices using `quantmod`. Risk metrics are computed using base R. Rolling statistics are calculated using a 20-day window.
- Interpretation:
  - Stock A may exhibit higher average returns but higher volatility.
  - Stock B may be more stable with lower drawdowns.
  - Rolling correlation highlights periods of co-movement, especially during market stress.

Installation

```
# Install the packages you need
# install.packages(c("shiny", "bslib", "quantmod", "ggplot2", "TTR"))
```

UI & Server

```
library(shiny)
```

```
## Warning: package 'shiny' was built under R version 4.5.2
```

```
library(bslib)
```

```
##
## Attaching package: 'bslib'
```

```
## The following object is masked from 'package:utils':
##
##     page
```

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.5.2
```

```
library(TTR)

# ------ UI ------
ui <- page_sidebar(
  title = "Stock Return & Risk Comparison",

  # ---- Side Bar ----
  sidebar = sidebar(
    helpText("Compare the return and risk profile of two stocks. Feel free to change tickers, date rang

    textInput("stock1", "Ticker 1", value = "AAPL"),
    textInput("stock2", "Ticker 2", value = "MSFT"),
    dateInput("from", "Start Date", value = Sys.Date() - 365),
    dateInput("to",   "End Date",   value = Sys.Date()),

    selectInput("freq", "Return Frequency:",
                choices = c("Daily" = "daily",
                            "Weekly" = "weekly",
                            "Monthly" = "monthly"),
                selected = "daily"),

    radioButtons(
      "Relationship Guess",
      "Your Guess About Relationship:",
      choices = c(
        "Very Strong Positive (> 0.9)",
        "Strong Positive (0.7 - 0.9)",
        "Moderate Positive (0.5 - 0.7)",
        "Weak Positive (0.2 - 0.5)",
        "Very Weak Positive (0 - 0.2)",
        "Negative (< 0)"
      )
    ),
    actionButton("go", "Run")
  ),

  # ---- Main Panel ----
  mainPanel(
    tabsetPanel(
      tabPanel("Return Time Series", plotOutput("line", height = "300px", width = "600px")),
```

```
      tabPanel("Return Scatter", plotOutput("scatter", height = "300px", width = "600px")),
      tabPanel("Return Distribution",
               plotOutput("hist1", height = "300px"),
               plotOutput("hist2", height = "300px")),
      tabPanel("Risk & Correlation",
               textOutput("t_test"),
               textOutput("corr"),
               textOutput("corr_text"),
               textOutput("risk_text")
      ),
      tabPanel("Rolling Stats",
               plotOutput("roll_vol_plot", height = "350px"),
               plotOutput("roll_corr_plot", height = "350px")
      ),
      tabPanel("Interpretation", textOutput("interpret"))
    )
  )
)

# ------ Server ------
server <- function(input, output){

  observeEvent(input$go,{

    # Fetch stock prices
    price1 <- getSymbols(input$stock1, src = "yahoo", from = input$from, to = input$to, auto.assign = F
    price2 <- getSymbols(input$stock2, src = "yahoo", from = input$from, to = input$to, auto.assign = F

    # Compute returns based on selected frequency
    ret_fun <- switch(input$freq,
                      "daily"   = dailyReturn,
                      "weekly"  = weeklyReturn,
                      "monthly" = monthlyReturn)

    ret1 <- ret_fun(Ad(price1))
    ret2 <- ret_fun(Ad(price2))

    # ---- Return Time Series Plot ----
    df1 <- data.frame(date = index(ret1), ret = as.numeric(ret1), ticker = input$stock1)
    df2 <- data.frame(date = index(ret2), ret = as.numeric(ret2), ticker = input$stock2)
    df_ret <- rbind(df1, df2)

    output$line <- renderPlot({
      ggplot(df_ret, aes(date, ret, color = ticker)) +
        geom_line(size = 1) +
        labs(x = "Date", y = paste(input$freq, "Return"), title = "Returns Over Time")
    })

    # ---- Scatter Plot ----
    df_scatter <- data.frame(ret1 = as.numeric(ret1), ret2 = as.numeric(ret2))

    output$scatter <- renderPlot({
      ggplot(df_scatter, aes(x = ret1, y = ret2)) +
```

```
    geom_point(color = "darkred") +
    geom_smooth(method = "lm", se = F, color = "dodgerblue") +
    labs(x = input$stock1, y = input$stock2, title = "Return Scatter Plot")
})

# ---- Return Histograms ----
output$hist1 <- renderPlot({
  ggplot(data.frame(ret = as.numeric(ret1)), aes(x = ret)) +
    geom_histogram(bins = 30, fill = "skyblue", color = "black") +
    labs(title = paste("Histogram of", input$stock1, input$freq, "Returns"), x = "Return", y = "Cou
})

output$hist2 <- renderPlot({
  ggplot(data.frame(ret = as.numeric(ret2)), aes(x = ret)) +
    geom_histogram(bins = 30, fill = "salmon", color = "black") +
    labs(title = paste("Histogram of", input$stock2, input$freq, "Returns"), x = "Return", y = "Cou
})

# ---- Paired t-test ----
output$t_test <- renderText({
  t_res <- t.test(df_scatter$ret1, df_scatter$ret2, paired = TRUE)
  t_val <- round(t_res$statistic, 3)
  p_val <- ifelse(t_res$p.value < 0.001, "< 0.001", round(t_res$p.value, 3))
  df_val <- round(t_res$parameter, 0)
  mean_diff <- round(t_res$estimate, 6)

  paste("t value =", t_val,
        ", p value =", p_val,
        ", degrees of freedom =", df_val,
        ", mean diff =", mean_diff)
})

# ---- Correlation ----
corr_val <- cor(df_scatter$ret1, df_scatter$ret2)

output$corr <- renderText({
  paste("Correlation:", round(corr_val, 3))
})

output$corr_text <- renderText({
  if (abs(corr_val) < 0.2) {
    "Very weak linear relationship - the stocks move independently most of the time."
  } else if (abs(corr_val) < 0.5) {
    "Weak correlation - the stocks move together somewhat but not strongly."
  } else if (abs(corr_val) < 0.7) {
    "Moderate correlation - the stocks show clear co-movement."
  } else if (abs(corr_val) < 0.9) {
    "Strong correlation - the stocks move together most of the time."
  } else {
    "Very strong correlation - the stocks are almost perfectly linked."
  }
})
```

```
    # ---- Risk Metrics (base R version) ----
    sharpe1 <- round(mean(ret1) / sd(ret1) * sqrt(252), 3)  # Annualized Sharpe
    sharpe2 <- round(mean(ret2) / sd(ret2) * sqrt(252), 3)

    mdd1 <- round(min(cummax(Ad(price1)) - Ad(price1)) / max(Ad(price1)), 3)
    mdd2 <- round(min(cummax(Ad(price2)) - Ad(price2)) / max(Ad(price2)), 3)

    output$risk_text <- renderText({
      paste0(input$stock1, " - Sharpe: ", sharpe1, ", Max Drawdown: ", mdd1, "\n",
             input$stock2, " - Sharpe: ", sharpe2, ", Max Drawdown: ", mdd2)
    })

    # ---- Rolling Volatility ----
    roll_vol1 <- runSD(ret1, n = 20)
    roll_vol2 <- runSD(ret2, n = 20)

    df_roll_vol <- data.frame(
      date = index(ret1),
      roll_vol1 = roll_vol1,
      roll_vol2 = roll_vol2
    )

    output$roll_vol_plot <- renderPlot({
      ggplot(df_roll_vol, aes(x = date)) +
        geom_line(aes(y = roll_vol1, color = input$stock1)) +
        geom_line(aes(y = roll_vol2, color = input$stock2)) +
        labs(title = "20-day Rolling Volatility", y = "Volatility")
    })

    # ---- Rolling Correlation ----
    roll_corr <- runCor(as.numeric(ret1), as.numeric(ret2), n = 20)
    df_roll_corr <- data.frame(date = index(ret1), roll_corr = roll_corr)

    output$roll_corr_plot <- renderPlot({
      ggplot(df_roll_corr, aes(x = date, y = roll_corr)) +
        geom_line(color = "purple", size = 1) +
        labs(title = "20-day Rolling Correlation", y = "Correlation")
    })

    # ---- Interpretation Text ----
    output$interpret <- renderText({
      paste0(
        input$stock1, " has higher average returns but also higher volatility.\n",
        input$stock2, " is more stable with lower drawdowns.\n",
        "Rolling correlation indicates periods of higher co-movement during market stress.\n",
        "This analysis helps understand relative return-risk tradeoffs over time."
      )
    })
  })
}

# ------ Run App ------
shinyApp(ui = ui, server = server)
```