C58_Project

Gursimar Singh

2025-03-29

```
library(quantmod)
library(corpcor)
library(plotly)
library(forecast)
library(smooth)
library(quadprog)
library(ggplot2)
```

```
# This loads the stock price data and calculates log returns

symbols = c('AMZN', 'INTC', 'JNJ', 'JPM', 'KO', 'META', 'NFLX', 'PG', 'TSLA', 'XOM')
start_date <- "2024-01-01"
end_date <- "2025-01-01"

assets <- getSymbols(symbols, src = "yahoo", from = start_date, to = end_date,
                     periodicity = "daily", auto.assign = TRUE)

prices <- lapply(symbols, function(sym) get(sym)[, paste0(sym, ".Adjusted")])
returns <- lapply(prices, function(pr) na.omit(diff(log(pr))))

pTab <- do.call(data.frame, prices)
rTab <- do.call(data.frame, returns)
rTab_appended <- rTab
colnames(pTab) <- paste(symbols, "Price")
colnames(rTab) <- paste(symbols, "")
colnames(rTab_appended) <- paste(symbols, "Return")
```

```
# Histogram Plot for log returns

library(ggplot2)
library(patchwork)

# Generate the list of histogram plots
plot_list <- lapply(1:ncol(rTab_appended), function(i) {
  df <- data.frame(Return = rTab_appended[[i]])

  ggplot(df, aes(x = Return)) +
    geom_histogram(aes(y = ..density..), bins = 40,
                   fill = "skyblue", color = "white", alpha = 0.6) +
    geom_density(color = "blue", size = 1) +
    labs(
```

```
      title = paste("Histogram of", colnames(rTab_appended)[i]),
      x = "Daily Return", y = "Density"
    ) +
    theme_minimal(base_size = 10) +
    theme(
      plot.title = element_text(hjust = 0.5, size = 10),
      axis.title = element_text(size = 9),
      axis.text = element_text(size = 8)
    )
})



# Combine the plots in 2 rows x 5 columns
combined_plot <- wrap_plots(plotlist = plot_list, ncol = 5, nrow = 2) &
  theme(plot.background = element_rect(fill = "white", color = NA))

# Add caption under the whole figure
final_plot <- combined_plot +
  plot_annotation(
    caption = "Histograms of log returns",
    theme = theme(
      plot.caption = element_text(hjust = 0.5, size = 12,
              face = "plain", margin = margin(t = 10))
    )
  )

# Display
print(final_plot)
```

```
# Summary table for mean

mean_returns <- apply(rTab, 2, mean)          # Mean return for each column
min_returns  <- apply(rTab, 2, min)           # Min return for each column
max_returns  <- apply(rTab, 2, max)           # Max return for each column
var_returns  <- apply(rTab, 2, var)           # Variance for each column

summary_stats <- data.frame(
  Mean_Return = round(mean_returns, 5),
  Variance    = round(var_returns, 5),
  Min_Return  = round(min_returns, 5),
  Max_Return  = round(max_returns, 5)
)

print(summary_stats)
```

```
# Sample covariance matrix

library(reshape2)
library(ggplot2)

# Step 1: Get covariance matrix
cov_matrix <- cov(rTab)
```

```r
# Step 2: Melt it to long format
cov_df <- melt(cov_matrix)
colnames(cov_df) <- c("Ticker1", "Ticker2", "Covariance")

# Step 3: Define the correct order of tickers
ticker_order <- colnames(cov_matrix)

# Step 4: Apply factor levels to match axes (Y reversed to show matrix-style)
cov_df$Ticker1 <- factor(cov_df$Ticker1, levels = rev(ticker_order))
cov_df$Ticker2 <- factor(cov_df$Ticker2, levels = ticker_order)

# Step 5: Plot
ggplot(cov_df, aes(x = Ticker2, y = Ticker1, fill = Covariance)) +
  geom_tile(color = "white", linewidth = 0.2) +
  geom_text(aes(label = sprintf("%.4f", Covariance)), size = 3, color = "black") +
  scale_fill_gradientn(
    colors = c("#D1E5F0", "#FEE08B", "#B2182B"),
    name = "Covariance",
    limits = c(min(cov_df$Covariance), max(cov_df$Covariance))
  ) +
  labs(
    title = "Covariance Matrix of Daily Returns",
    x = "Ticker",
    y = "Ticker"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 0.5),
    axis.text.y = element_text(size = 9),
    plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    legend.position = "right",
    legend.key.height = unit(1.5, "cm"),
    plot.margin = margin(t = 20, r = 10, b = 10, l = 10),
    axis.title = element_text(face = "bold")
  ) +
  coord_fixed()
```

```r
# Confidence intervals for Mean

alpha <- 0.05
z_value <- qnorm(1 - alpha / 2)

ci_df <- data.frame(
  Mean = apply(rTab, 2, mean),
  SD = apply(rTab, 2, sd),
  N = nrow(rTab)
)

# Add CI bounds
ci_df$Lower_Bound <- ci_df$Mean - z_value * ci_df$SD / sqrt(ci_df$N)
ci_df$Mean_Estimate <- ci_df$Mean
```

```r
ci_df$Upper_Bound <- ci_df$Mean + z_value * ci_df$SD / sqrt(ci_df$N)

# Round for readability
ci_df <- round(ci_df, 5)

print(ci_df[, -1:-3])
```