Untitled

Idriss

2024-07-14

Contents

PORTFOLIO FINANCIAL ANALYSIS **2**
1. **Modern Portfolio Theory (MPT)** 2
2. **Capital Asset Pricing Model (CAPM)** 2
3. **Arbitrage Pricing Theory (APT)** 2
4. **Fama-French Three-Factor Model** 2
5. **Black-Litterman Model** 2
6. **Value at Risk (VaR)** 2
7. **Monte Carlo Simulation** 3
8. **Sharpe Ratio** 3

getting the data **3**
Installing and importing thelibrary 3
importing data from yahoo 3
Calculating returns 4
Returns Summary 4

Vizualization **5**
Closing prices 5
candlestick vizualization 6
Volume plot 8
Daily returns plot 10
Moving averages plot 11
Returns Histogram 13
summary Statistics 15

PORTFOLIO FINANCIAL ANALYSIS

In portfolio financial analysis, several models and techniques are commonly used to assess risk, return, and overall performance. Here are some key models:

1. Modern Portfolio Theory (MPT)

- **Developed by:** Harry Markowitz
- **Concept:** Investors can construct portfolios to maximize expected return based on a given level of market risk, emphasizing diversification.
- **Key Metric:** Efficient Frontier, which represents the set of optimal portfolios offering the highest expected return for a defined level of risk.

2. Capital Asset Pricing Model (CAPM)

- **Concept:** Describes the relationship between systematic risk and expected return for assets, particularly stocks.
- **Formula:**

  $$E(R_i) = R_f + \beta_i(E(R_m) - R_f)$$

  Where $E(R_i)$ is the expected return on the capital asset, R_f is the risk-free rate, β_i is the beta of the security, and $E(R_m)$ is the expected market return.

3. Arbitrage Pricing Theory (APT)

- **Concept:** An alternative to CAPM, APT considers multiple factors that might affect an asset's return, instead of just market risk.
- **Formula:**

  $$E(R_i) = R_f + \beta_{i1}F_1 + \beta_{i2}F_2 + ... + \beta_{in}F_n$$

  Where F represents various factors.

4. Fama-French Three-Factor Model

- **Developed by:** Eugene Fama and Kenneth French
- **Concept:** Expands on CAPM by adding size risk (SMB, Small Minus Big) and value risk (HML, High Minus Low) factors.
- **Formula:**

  $$E(R_i) = R_f + \beta_{i1}(E(R_m) - R_f) + \beta_{i2}SMB + \beta_{i3}HML$$

5. Black-Litterman Model

- **Developed by:** Fischer Black and Robert Litterman
- **Concept:** Combines investor views with market equilibrium to estimate the expected returns of assets.
- **Use:** Often used in portfolio optimization to create a more stable and intuitive estimation of expected returns.

6. Value at Risk (VaR)

- **Concept:** Measures the potential loss in value of a portfolio over a defined period for a given confidence interval.
- **Usage:** Helps in risk management to understand the maximum potential loss.

7. Monte Carlo Simulation

- **Concept:** Uses repeated random sampling to simulate the performance of a portfolio under various conditions.
- **Usage:** Useful for estimating the impact of risk and uncertainty in financial and investment decisions.

8. Sharpe Ratio

- **Concept:** Measures the performance of an investment compared to a risk-free asset, after adjusting for its risk.
- **Formula:**
$$\text{Sharpe Ratio} = \frac{E(R_p) - R_f}{\sigma_p}$$
Where $E(R_p)$ is the expected portfolio return, R_f is the risk-free rate, and σ_p is the portfolio standard deviation.

These models and tools are essential for constructing, analyzing, and managing portfolios to achieve desired risk-return profiles.

getting the data

Installing and importing thelibrary

```
#install.packages("quantmod")
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
```

importing data from yahoo

```r
# Load the quantmod package
library(quantmod)

# Get S&P 500 (SPY) data
getSymbols("SPY", src = "yahoo", from = "2000-01-01", to = Sys.Date())
```

```
## [1] "SPY"
```

```r
spy_data <- SPY

# Get Dow Jones Industrial Average (DJI) data
getSymbols("^DJI", src = "yahoo", from = "2000-01-01", to = Sys.Date())
```

```
## [1] "DJI"
```

```r
dji_data <- DJI
colnames(spy_data)
```

```
## [1] "SPY.Open"     "SPY.High"     "SPY.Low"      "SPY.Close"    "SPY.Volume"
## [6] "SPY.Adjusted"
```

```r
colnames(dji_data)
```

```
## [1] "DJI.Open"     "DJI.High"     "DJI.Low"      "DJI.Close"    "DJI.Volume"
## [6] "DJI.Adjusted"
```

```r
#View(spy_data)
```

Calculating returns

```r
# Calculate daily returns for SPY
spy_returns <- dailyReturn(spy_data[, "SPY.Adjusted"])

# Calculate daily returns for DJI
dji_returns <- dailyReturn(dji_data[, "DJI.Adjusted"])
```

Returns Summary

```r
# Summary statistics for SPY returns
summary(spy_returns)
```

```
##      Index            daily.returns
##  Min.   :2000-01-03   Min.   :-0.1094234
##  1st Qu.:2006-02-22   1st Qu.:-0.0046920
##  Median :2012-04-07   Median : 0.0006732
##  Mean   :2012-04-07   Mean   : 0.0003658
##  3rd Qu.:2018-05-23   3rd Qu.: 0.0060115
##  Max.   :2024-07-12   Max.   : 0.1451972
```

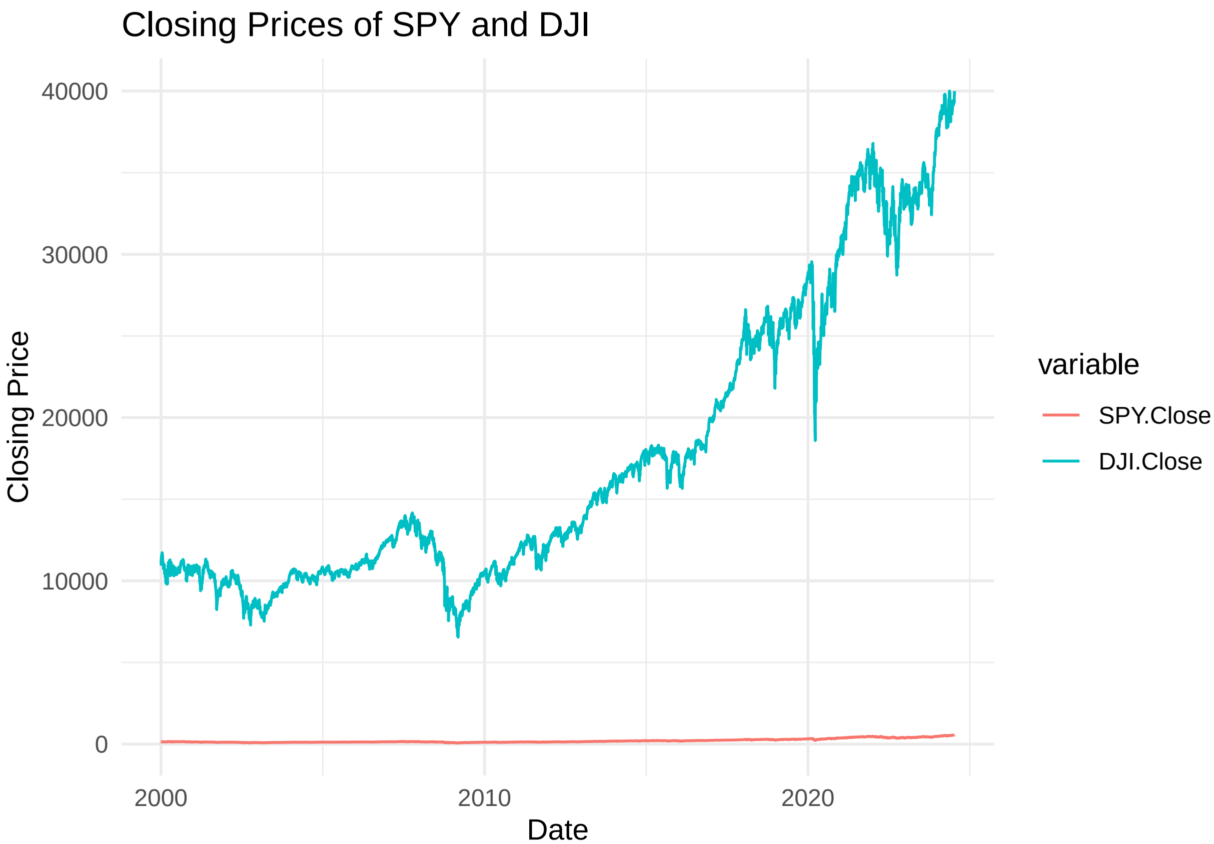
```r
# Summary statistics for DJI returns
summary(dji_returns)
```

```
##      Index             daily.returns
##  Min.   :2000-01-03   Min.   :-0.1292655
##  1st Qu.:2006-02-22   1st Qu.:-0.0045844
##  Median :2012-04-07   Median : 0.0004982
##  Mean   :2012-04-07   Mean   : 0.0002723
##  3rd Qu.:2018-05-23   3rd Qu.: 0.0055937
##  Max.   :2024-07-12   Max.   : 0.1136504
```

Vizualization

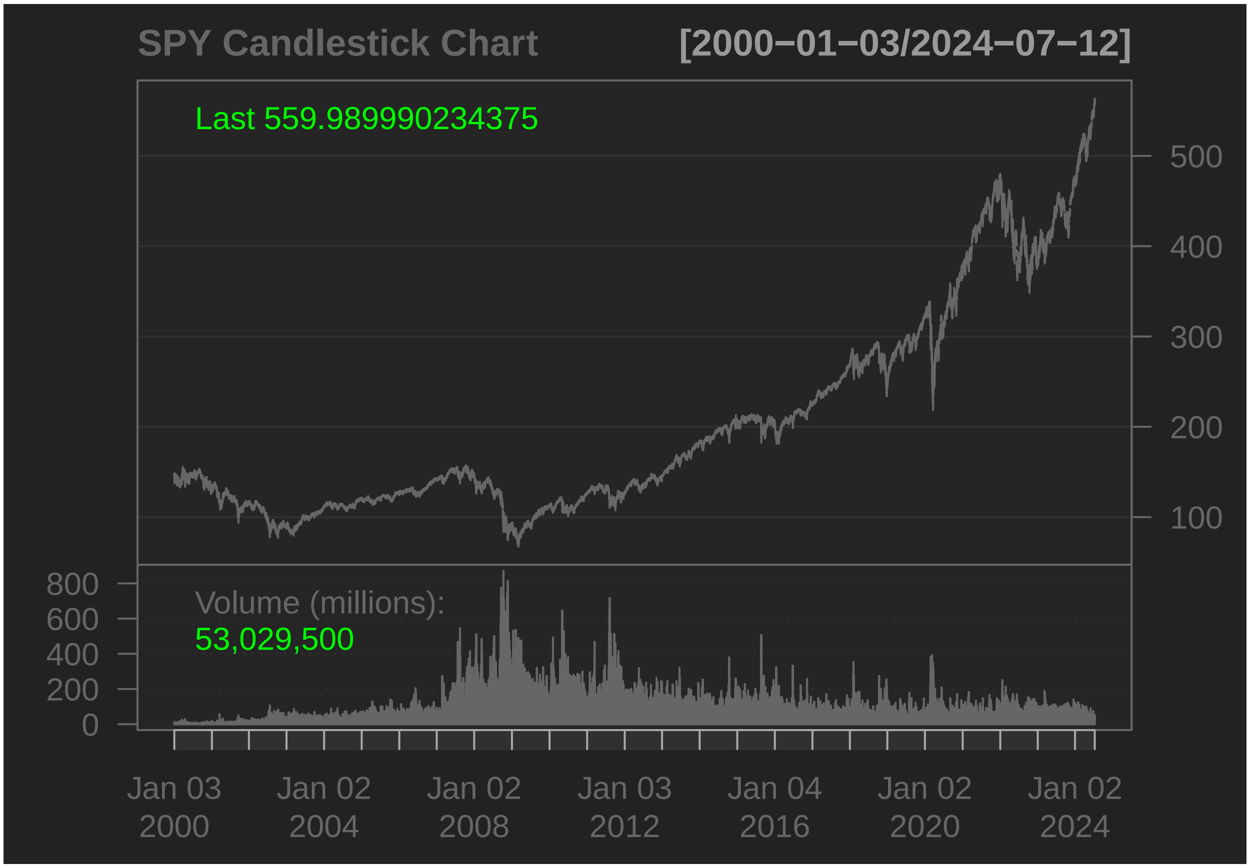
Closing prices

```r
# Load necessary libraries
library(ggplot2)
library(reshape2)

# Combine data into one data frame for plotting
spy_close <- data.frame(Date = index(spy_data), SPY.Close = spy_data$SPY.Close)
dji_close <- data.frame(Date = index(dji_data), DJI.Close = dji_data$DJI.Close)

# Merge data frames by Date
combined_close <- merge(spy_close, dji_close, by = "Date", all = TRUE)

# Melt the data for ggplot2
combined_close_melt <- melt(combined_close, id.vars = "Date")

# Line plot
ggplot(combined_close_melt, aes(x = Date, y = value, color = variable)) +
  geom_line() +
  labs(title = "Closing Prices of SPY and DJI", x = "Date", y = "Closing Price") +
  theme_minimal()
```

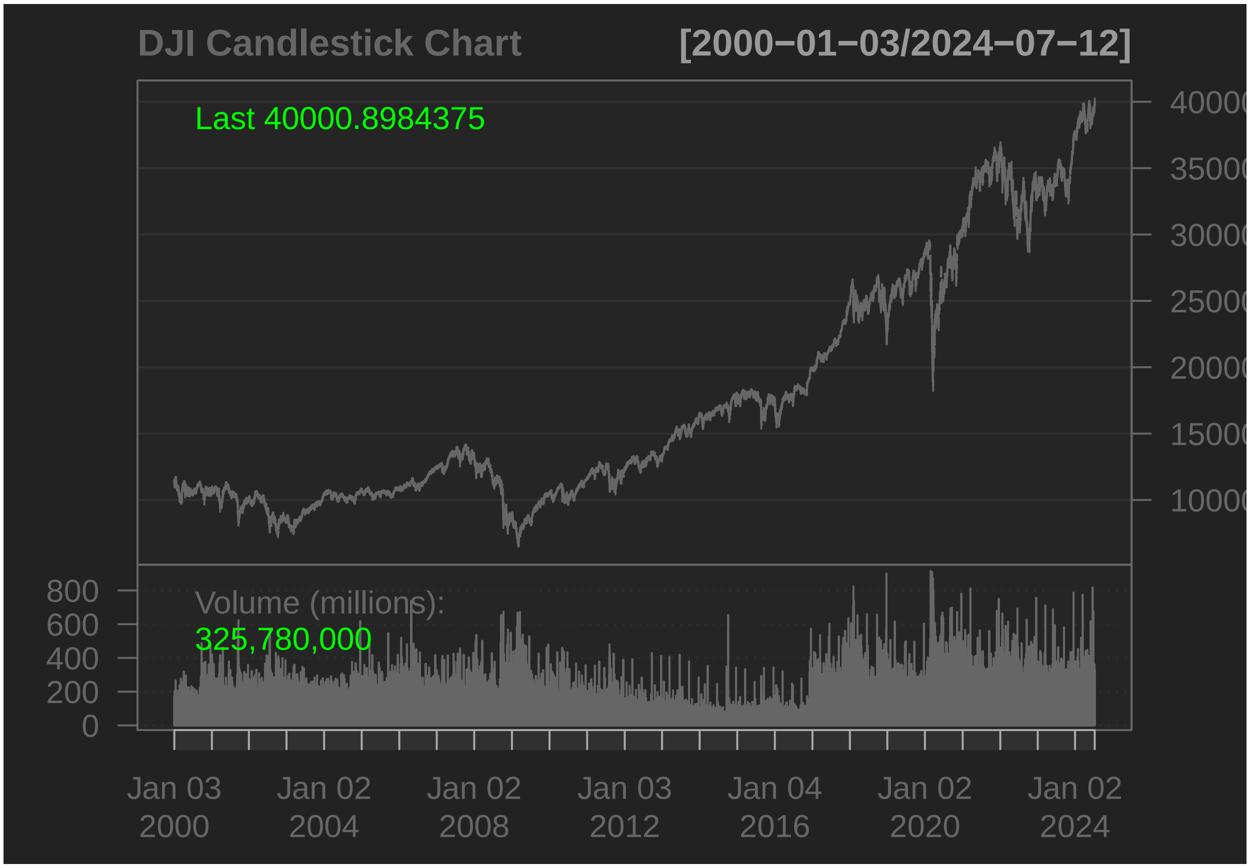


candlestick vizualization

```
# Load necessary library
library(quantmod)

# Candlestick chart for SPY
chartSeries(spy_data, type = "candlesticks", name = "SPY Candlestick Chart")
```



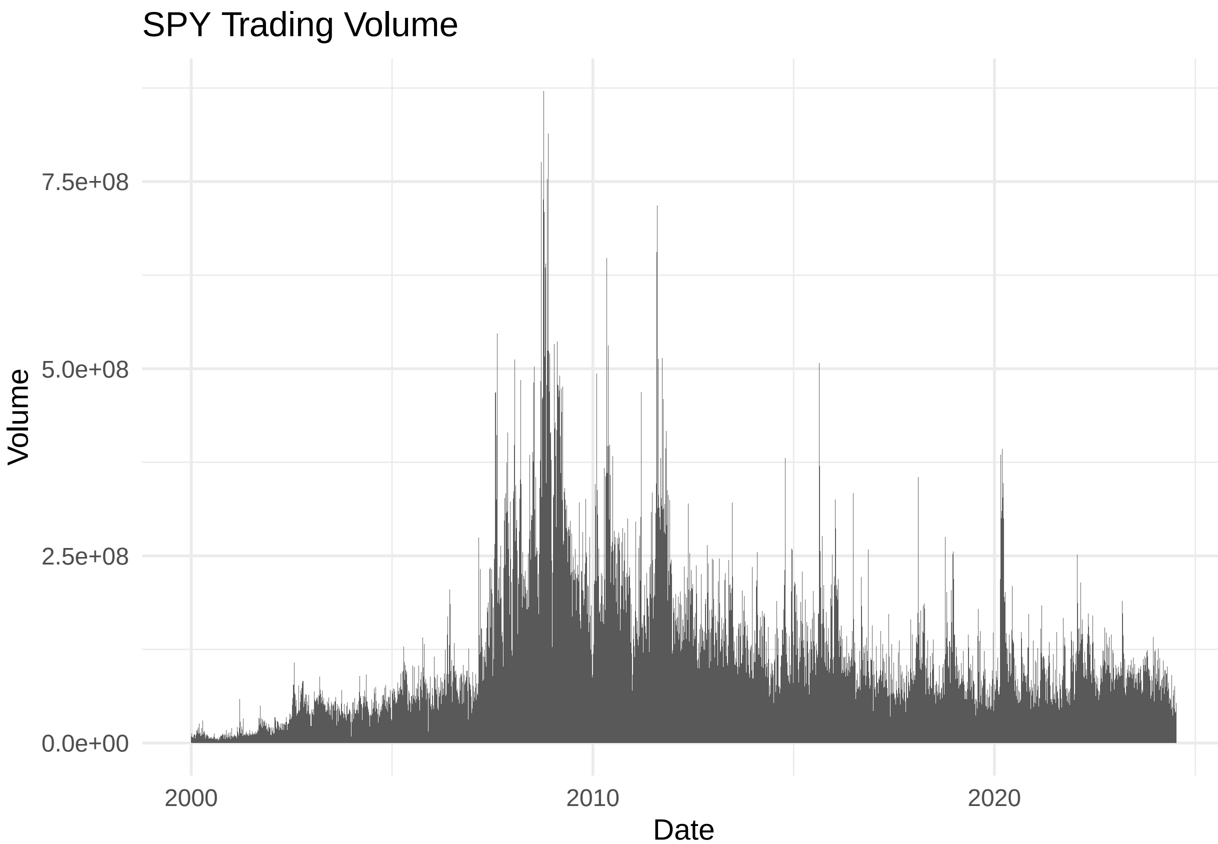

```r
# Candlestick chart for DJI
chartSeries(dji_data, type = "candlesticks", name = "DJI Candlestick Chart")
```



Volume plot

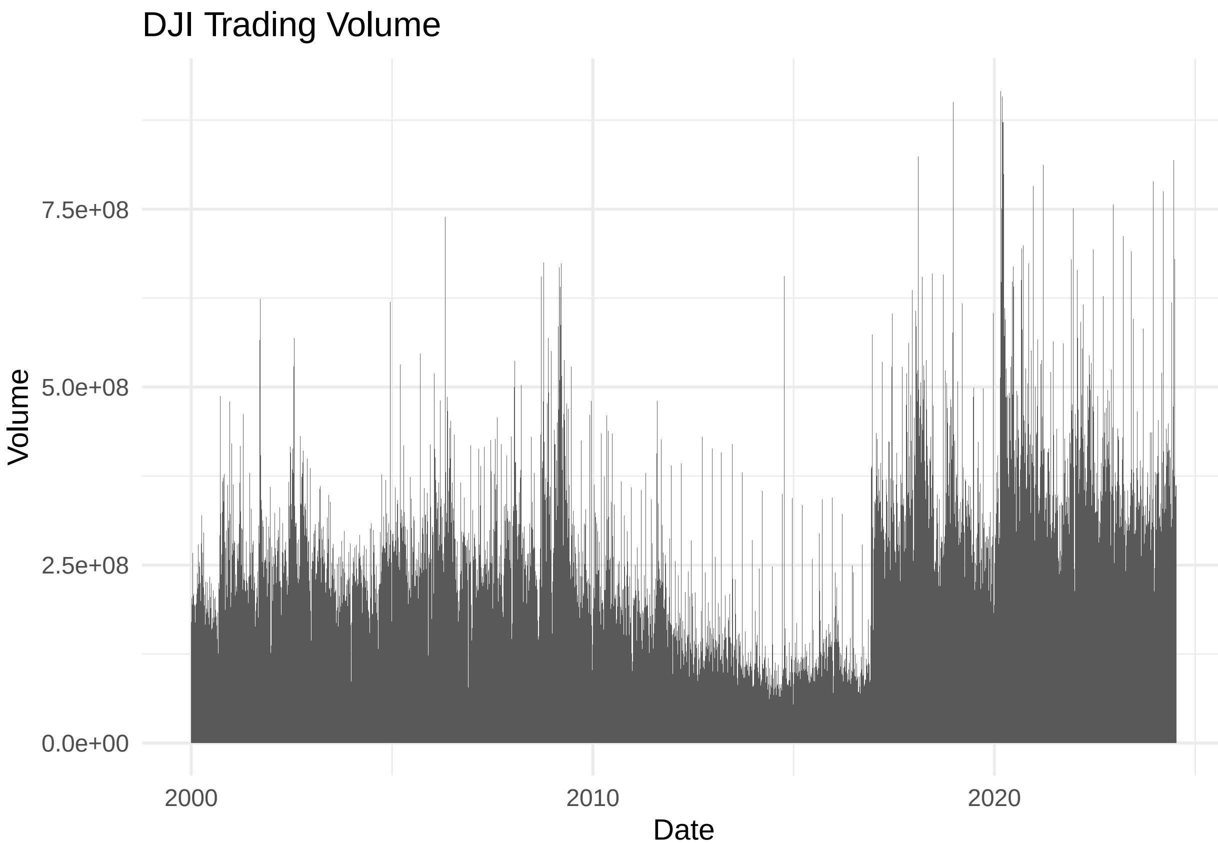
```r
# SPY Volume Plot
ggplot(spy_close, aes(x = Date, y = spy_data$SPY.Volume)) +
  geom_bar(stat = "identity") +
  labs(title = "SPY Trading Volume", x = "Date", y = "Volume") +
  theme_minimal()
```

```
## Don't know how to automatically pick scale for object of type <xts/zoo>.
## Defaulting to continuous.
```



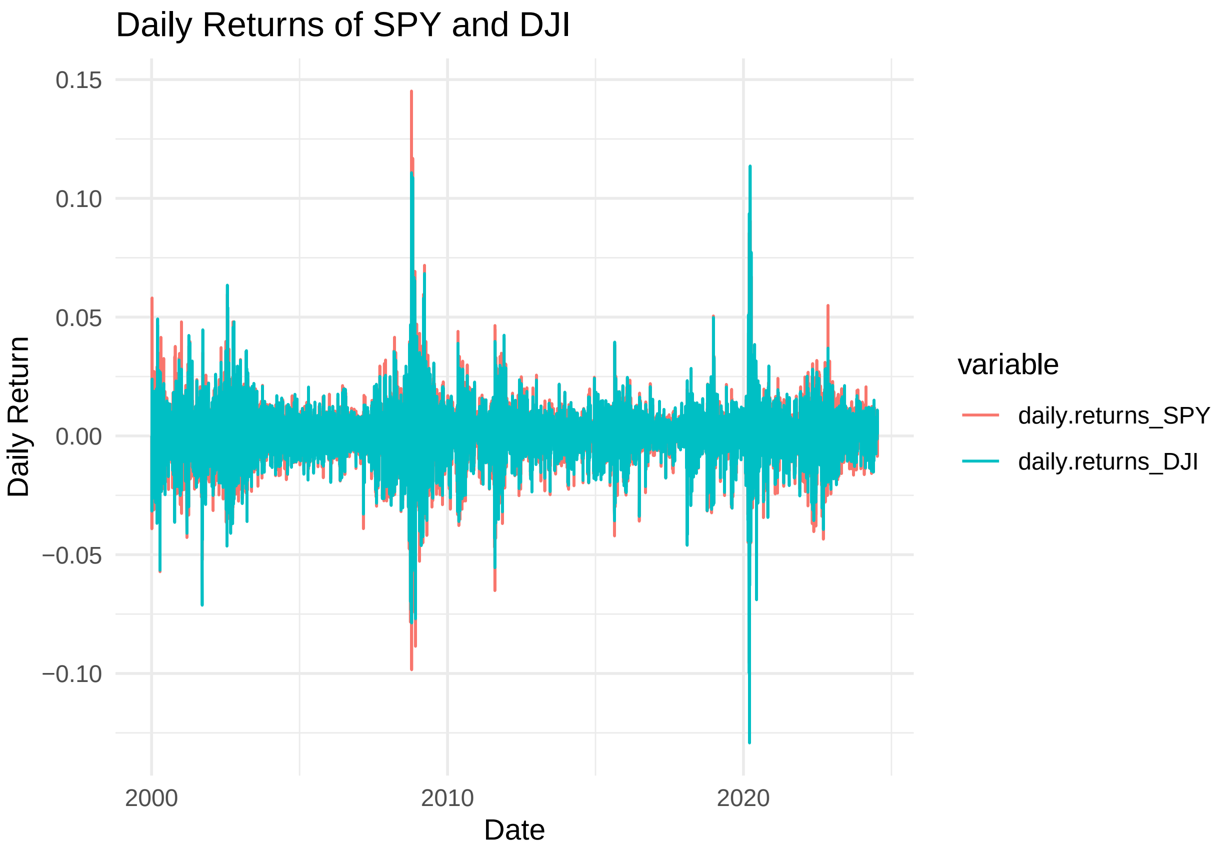

```r
# DJI Volume Plot
ggplot(dji_close, aes(x = Date, y = dji_data$DJI.Volume)) +
  geom_bar(stat = "identity") +
  labs(title = "DJI Trading Volume", x = "Date", y = "Volume") +
  theme_minimal()
```

```
## Don't know how to automatically pick scale for object of type <xts/zoo>.
## Defaulting to continuous.
```



Daily returns plot

```r
# Calculate daily returns for SPY and DJI
spy_returns <- dailyReturn(spy_data[, "SPY.Adjusted"])
dji_returns <- dailyReturn(dji_data[, "DJI.Adjusted"])

# Convert to data frames
spy_returns_df <- data.frame(Date = index(spy_returns), Return = coredata(spy_returns))
dji_returns_df <- data.frame(Date = index(dji_returns), Return = coredata(dji_returns))

# Combine returns for plotting
combined_returns <- merge(spy_returns_df, dji_returns_df, by = "Date", suffixes = c("_SPY", "_DJI"))
combined_returns_melt <- melt(combined_returns, id.vars = "Date")

# Line plot of daily returns
ggplot(combined_returns_melt, aes(x = Date, y = value, color = variable)) +
  geom_line() +
  labs(title = "Daily Returns of SPY and DJI", x = "Date", y = "Daily Return") +
  theme_minimal()
```

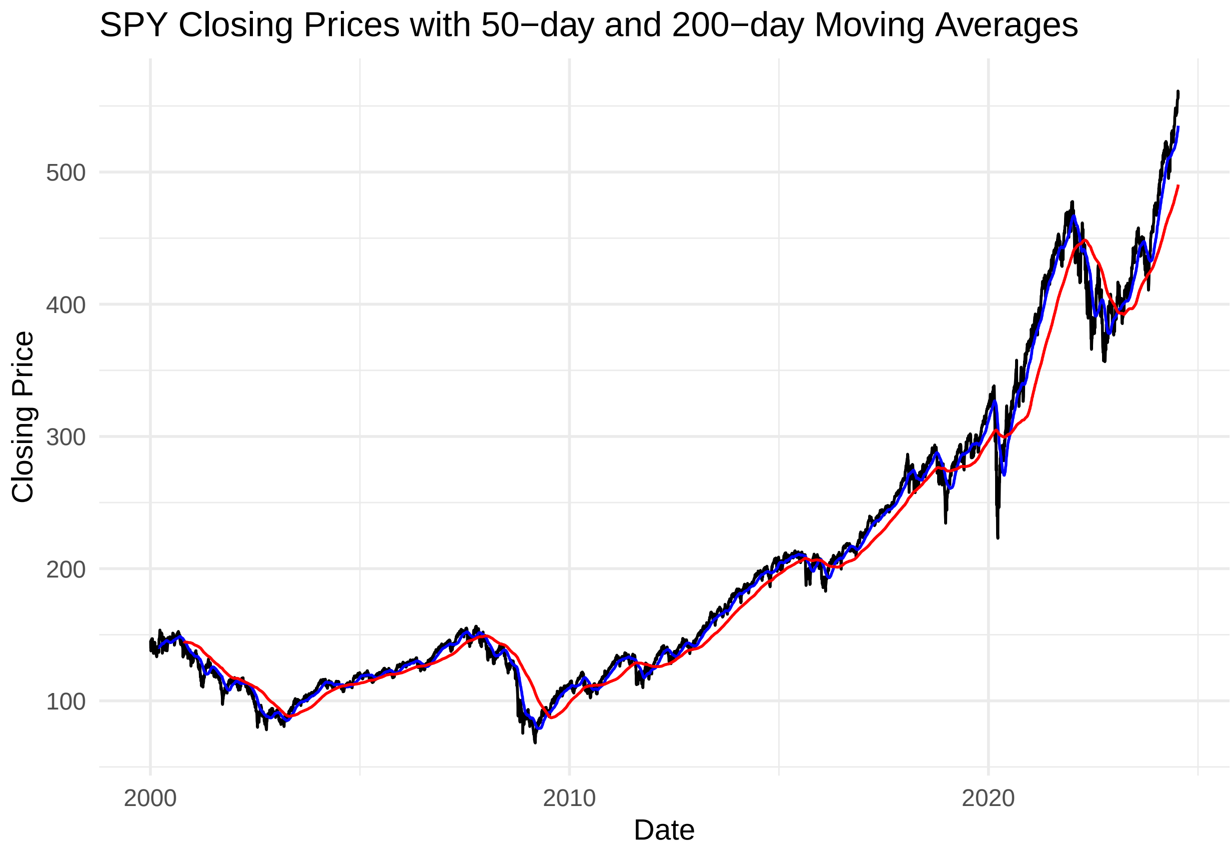


Moving averages plot

```
# Calculate 50-day and 200-day moving averages for SPY
spy_data$SPY.MA50 <- SMA(spy_data[, "SPY.Close"], n = 50)
spy_data$SPY.MA200 <- SMA(spy_data[, "SPY.Close"], n = 200)

# Plot SPY with moving averages
ggplot(spy_close, aes(x = Date, y = SPY.Close)) +
  geom_line() +
  geom_line(aes(y = spy_data$SPY.MA50), color = "blue") +
  geom_line(aes(y = spy_data$SPY.MA200), color = "red") +
  labs(title = "SPY Closing Prices with 50-day and 200-day Moving Averages", x = "Date", y = "Closing Pr
  theme_minimal()
```

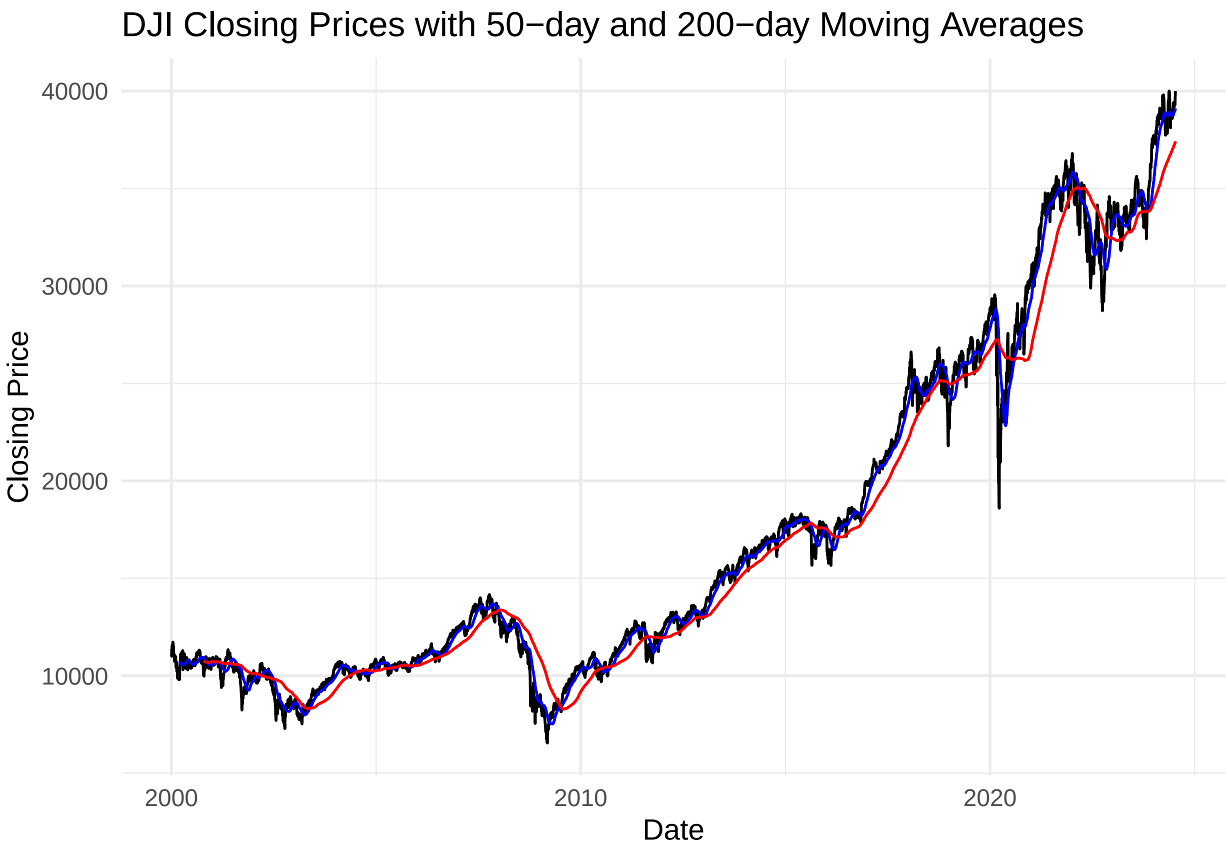
```
## Warning: Removed 49 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```

```
## Warning: Removed 199 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```



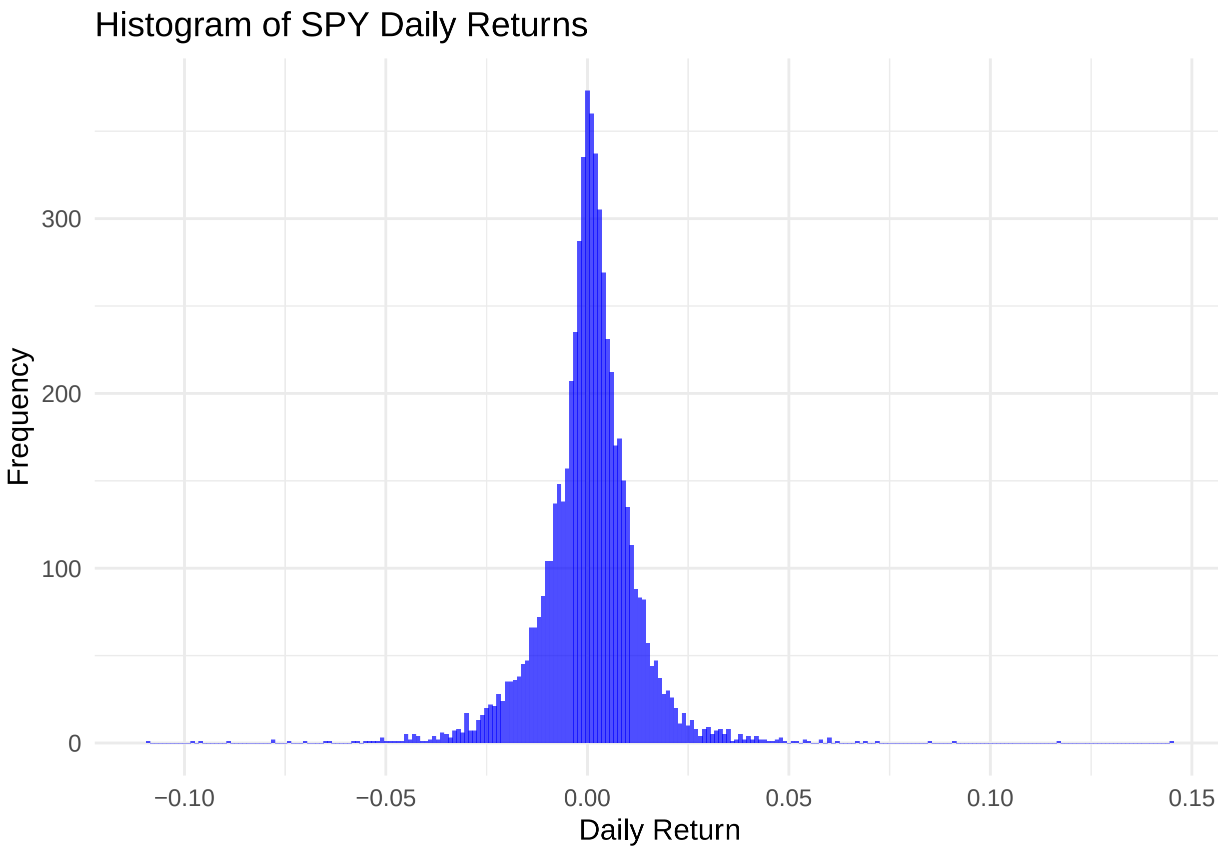

```r
# Calculate 50-day and 200-day moving averages for DJI
dji_data$DJI.MA50 <- SMA(dji_data[, "DJI.Close"], n = 50)
dji_data$DJI.MA200 <- SMA(dji_data[, "DJI.Close"], n = 200)

# Plot DJI with moving averages
ggplot(dji_close, aes(x = Date, y = DJI.Close)) +
  geom_line() +
  geom_line(aes(y = dji_data$DJI.MA50), color = "blue") +
  geom_line(aes(y = dji_data$DJI.MA200), color = "red") +
  labs(title = "DJI Closing Prices with 50-day and 200-day Moving Averages", x = "Date", y = "Closing Pr
  theme_minimal()
```

```
## Warning: Removed 49 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
## Removed 199 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```

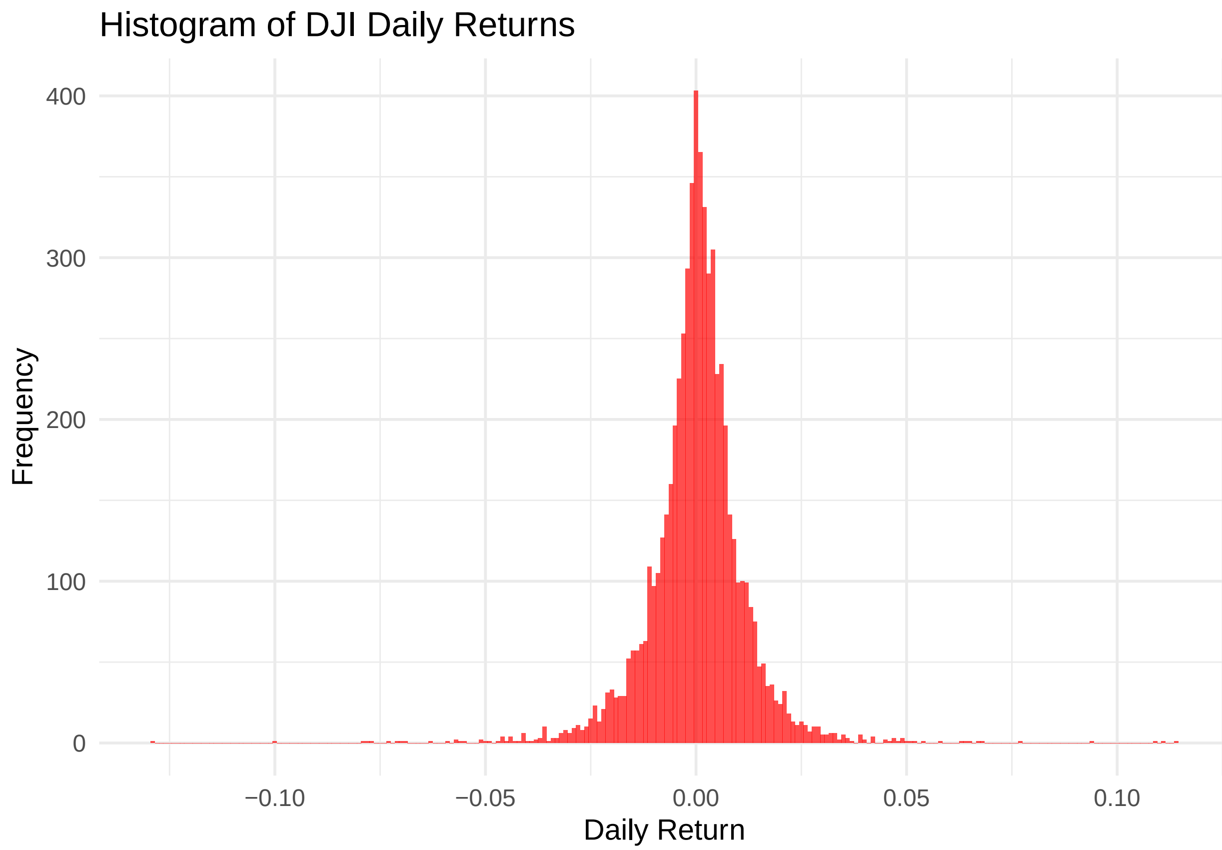


Returns Histogram

```
# Histogram of SPY returns
ggplot(spy_returns_df, aes(x = daily.returns)) +
  geom_histogram(binwidth = 0.001, fill = "blue", alpha = 0.7) +
  labs(title = "Histogram of SPY Daily Returns", x = "Daily Return", y = "Frequency") +
  theme_minimal()
```




```r
# Histogram of DJI returns
ggplot(dji_returns_df, aes(x = daily.returns)) +
  geom_histogram(binwidth = 0.001, fill = "red", alpha = 0.7) +
  labs(title = "Histogram of DJI Daily Returns", x = "Daily Return", y = "Frequency") +
  theme_minimal()
```



summary Statistics