ANA 515 Assignment 2-SPY ETF Data Analysis

Shanmugapriya Mohan Kumar

2024-06-14

**Dataset Description**

The dataset measures the daily stock performance of the SPDR S&P 500 ETF (SPY). This dataset contains variables such as

1. **Open** – Open price of that trading day.
2. **High** - High price of that trading day.
3. **Low** - Low price of that trading day.
4. **Close** - Closing price of that trading day.
5. **Volume** – trading volume of that trading day.
6. **Date** - date of that trading day.

This data set is used to analyze historical price movements, identify trends and patterns, trading strategies, predict future prices, and make investment decisions.

This data is collected from stock exchanges or financial data provider services such as Yahoo Finance or Google Finance.

**Research Questions:**

1. What are the trends and patterns of stock price movements?
2. How was the performance of SPY over a specific period?
3. Can historical data be used to predict future prices?

**File Format:** File type is CSV and delimiter is Comma (,). The CSV files can be opened with Microsoft Excel, Google sheets, Statistical software such as R or Python’s Pandas library.

# Dataset URL  
spy\_data <- "https://raw.githubusercontent.com/ShanmugapriyaMohankumar/StockMarket/main/SPY.csv"  
  
# Read the dataset from the URL  
spy500\_data <- read\_csv(spy\_data,show\_col\_types = FALSE) #read\_csv is part of the readr package, which is included in the tidyverse collection of R packages  
  
#display the first few rows of the dataframe  
head(spy500\_data)

## # A tibble: 6 × 7  
## Date Open High Low Close `Adj Close` Volume  
## <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2023-06-02 424. 429. 424. 428. 422. 91366700  
## 2 2023-06-05 428. 430. 426. 427. 421. 65460200  
## 3 2023-06-06 427. 429. 426. 428. 422. 64022200  
## 4 2023-06-07 428. 430. 426. 427. 420. 85373300  
## 5 2023-06-08 427. 430. 426. 429. 423. 61952800  
## 6 2023-06-09 430. 432. 429. 430. 424. 85742800

#Number of rows in the original dataset  
spy500\_data\_row\_count <- nrow(spy500\_data)  
  
#Number of columns in the original dataset  
spy500\_data\_column\_count <- ncol(spy500\_data)

# Display the actual column names  
colnames(spy500\_data)

## [1] "Date" "Open" "High" "Low" "Close" "Adj Close"  
## [7] "Volume"

# Clean the data  
spy500\_data\_clean <- spy500\_data %>%  
 rename(  
 Open\_price = Open,  
 High\_price = High,  
 Low\_price = Low,  
 Close\_price = Close  
 ) %>%  
 mutate(  
 Date = ymd(Date) # Convert the date column to Date format  
 ) %>%  
 arrange(desc(Volume)) # Arrange the data by volume in descending order  
  
# Display the first few rows of the cleaned dataframe to verify the data is cleaned correctly  
head(spy500\_data\_clean)

## # A tibble: 6 × 7  
## Date Open\_price High\_price Low\_price Close\_price `Adj Close` Volume  
## <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2023-12-15 469. 471. 467. 469. 468. 141319300  
## 2 2024-01-31 489. 489. 483. 483. 481. 126011100  
## 3 2023-10-20 426. 427. 421. 421. 418. 123845800  
## 4 2024-01-02 472. 474. 470. 473. 471. 123623700  
## 5 2023-12-29 476. 477. 473. 475. 474. 122234100  
## 6 2023-10-19 431. 433. 426. 426. 423. 121323000

# Filter the dataset to include only the columns of interest  
spy500\_data\_filtered <- spy500\_data\_clean %>%  
 select(Date, Open\_price, High\_price, Close\_price, Volume)  
  
# Display the first few rows of the filtered dataframe  
head(spy500\_data\_filtered)

## # A tibble: 6 × 5  
## Date Open\_price High\_price Close\_price Volume  
## <date> <dbl> <dbl> <dbl> <dbl>  
## 1 2023-12-15 469. 471. 469. 141319300  
## 2 2024-01-31 489. 489. 483. 126011100  
## 3 2023-10-20 426. 427. 421. 123845800  
## 4 2024-01-02 472. 474. 473. 123623700  
## 5 2023-12-29 476. 477. 475. 122234100  
## 6 2023-10-19 431. 433. 426. 121323000

# Find the date with the highest volume  
highest\_volume <- spy500\_data\_clean %>%  
 slice\_max(Volume, n=1)   
  
# Extract the date and volume  
highest\_volume\_date <- highest\_volume$Date  
highest\_volume\_value <- highest\_volume$Volume

### Date with the Highest Trading Volume:

The date with the highest trading volume is 2023-12-15, with a volume of 1.413193^{8}.

### Data Frame Information

This data set has 251 rows and 7 columns.

The names of the columns and a brief description of each are in the table below:

# Create a dataframe for column descriptions  
column\_descriptions <- data.frame(  
 Column\_number = c("1", "2", "3", "4", "5", "6"),  
 Column = c("Date", "Open", "High", "Low", "Close", "Volume"),  
 Description = c(  
 "The date of the trading day",  
 "The opening price of the trading day",  
 "The highest price of the trading day",  
 "The lowest price of the trading day",  
 "The closing price of the trading day",  
 "The trading volume of the trading day"  
 )  
)  
  
  
# Display the table  
kable(column\_descriptions, col.names = c("\*\*Column Number\*\*", "\*\*Column Name\*\*", "\*\*Description\*\*"))

| **Column Number** | **Column Name** | **Description** |
| --- | --- | --- |
| 1 | Date | The date of the trading day |
| 2 | Open | The opening price of the trading day |
| 3 | High | The highest price of the trading day |
| 4 | Low | The lowest price of the trading day |
| 5 | Close | The closing price of the trading day |
| 6 | Volume | The trading volume of the trading day |

# Create a subset of the dataframe with selected columns  
spy500\_data\_subset <- spy500\_data\_filtered %>%  
 select(Open\_price, High\_price, Close\_price)  
  
# Calculate summary statistics  
summary\_statistics <- spy500\_data\_subset %>%  
 summarise(  
 Open\_price\_min = min(Open\_price, na.rm = TRUE),  
 Open\_price\_max = max(Open\_price, na.rm = TRUE),  
 Open\_price\_mean = mean(Open\_price, na.rm = TRUE),  
 Open\_price\_na\_count = sum(is.na(Open\_price)),  
 High\_price\_min = min(High\_price, na.rm = TRUE),  
 High\_price\_max = max(High\_price, na.rm = TRUE),  
 High\_price\_mean = mean(High\_price, na.rm = TRUE),  
 High\_price\_na\_count = sum(is.na(High\_price)),  
 Close\_price\_min = min(Close\_price, na.rm = TRUE),  
 Close\_price\_max = max(Close\_price, na.rm = TRUE),  
 Close\_price\_mean = mean(Close\_price, na.rm = TRUE),  
 Close\_price\_na\_count = sum(is.na(Close\_price))  
 )  
  
# Display the summary statistics in a table  
# Create a dataframe for the summary statistics  
summary\_df <- data.frame(  
 Statistic = c("\*\*Minimum\*\*", "\*\*Maximum\*\*", "\*\*Mean\*\*", "\*\*Missing Values\*\*"),  
 Open = c(summary\_statistics$Open\_price\_min, summary\_statistics$Open\_price\_max, summary\_statistics$Open\_price\_mean, summary\_statistics$Open\_price\_na\_count),  
 High = c(summary\_statistics$High\_price\_min, summary\_statistics$High\_price\_max, summary\_statistics$High\_price\_mean, summary\_statistics$High\_price\_na\_count),  
 Close = c(summary\_statistics$Close\_price\_min, summary\_statistics$Close\_price\_max, summary\_statistics$Close\_price\_mean, summary\_statistics$Close\_price\_na\_count)  
)

### Summary Statistics - S&P 500 Data(past 1 year)

# Display the table  
kable(summary\_df, col.names = c("\*\*Statistic\*\*", "\*\*Open Price\*\*", "\*\*High Price\*\*", "\*\*Close Price\*\*"))

| **Statistic** | **Open Price** | **High Price** | **Close Price** |
| --- | --- | --- | --- |
| **Minimum** | 413.5600 | 414.6000 | 410.6800 |
| **Maximum** | 532.9600 | 533.0700 | 531.3600 |
| **Mean** | 469.3543 | 471.3924 | 469.4955 |
| **Missing Values** | 0.0000 | 0.0000 | 0.0000 |