Projet d'Analyse des Données Économiques avec R

Haithem Farjallah et Omar Majri

Table of contents

1 Introduction

1.1 Description du Projet

Notre projet tourne autour de la collecte, du nettoyage, de l'analyse et de la visualisation des données économiques en utilisant R. Nous avons choisi de nous concentrer sur des ensembles de données liés à des entreprises de renom telles qu'Amazon, Microsoft et d'autres, car elles offrent des informations précieuses sur le paysage économique. Nous explorerons divers indicateurs économiques, tendances et modèles pour extraire des insights significatifs pouvant éclairer les processus de prise de décision.

2 Collecte des Données

2.1 Source des données : Yahoo Finance - Les plus actifs

Les données utilisées dans ce projet proviennent de la page "Les plus actifs" de Yahoo Finance. Cette source fournit une liste complète des actions les plus échangées, comprenant des détails tels que la date, les prix d'ouverture, de clôture, les plus hauts et les plus bas, ainsi que le volume de transactions et le nom de l'entreprise associée.

```
     Date               Open             High              Low         
 Length:158768      Min.   :  0.000   Min.   :  0.0497   Min.   :  0.0491  
 Class :character   1st Qu.:  4.981   1st Qu.:  5.1200   1st Qu.:  4.9700  
 Mode  :character   Median : 11.730   Median : 11.8750   Median : 11.5700  
                    Mean   : 24.479   Mean   : 24.8666   Mean   : 24.2217  
                    3rd Qu.: 25.978   3rd Qu.: 26.2900   3rd Qu.: 25.6500  
                    Max.   :446.010   Max.   :446.0100   Max.   :440.1900  
     Close            Adj.Close          Volume            Company         
 Min.   :  0.0491   Min.   :  0.038   Min.   :0.000e+00   Length:158768     
 1st Qu.:  5.0400   1st Qu.:  3.050   1st Qu.:6.700e+05   Class :character  
 Median : 11.7200   Median :  8.003   Median :3.321e+06   Mode  :character  
 Mean   : 24.5475   Mean   : 20.097   Mean   :3.607e+07                     
 3rd Qu.: 25.9794   3rd Qu.: 17.416   3rd Qu.:1.186e+07                     
 Max.   :440.6200   Max.   :430.300   Max.   :7.422e+09                     
```

3 Netoyage des donées :

1. **Vérification des valeurs manquantes :**

   Nous avons commencé par vérifier et supprimer les lignes contenant des valeurs manquantes à l'aide de la fonction **`complete.cases`**. Cela garantit l'intégrité de nos données en éliminant les observations incomplètes.

```r
# Check for missing values
missing_values <- sum(is.na(data))
print(missing_values)
```

```
[1] 0
```

```r
# Remove rows with missing values
data <- data[complete.cases(data), ]
```

2. **Vérification du format de la date :**

   Ensuite, nous avons utilisé la bibliothèque lubridate pour confirmer que la colonne 'Date' était au bon format de date, et nous avons converti cette colonne en format de date avec **`as.Date`**. Cela assure la cohérence dans la gestion des dates pour nos analyses ultérieures.

```
#Check Date Format: Confirm that the 'Date' column is in the correct date format.
library(lubridate)
```

```
Attaching package: 'lubridate'
```

```
The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
is.Date(data$Date)
```

```
[1] FALSE
```

```
data$Date <- as.Date(data$Date)
# Check unique values in the 'Date' column
length(unique(data$Date)) == nrow(data)
```

```
[1] FALSE
```

3. **Vérification des valeurs négatives :**

   Nous avons également vérifié qu'il n'y avait pas de valeurs négatives dans les colonnes numériques pertinentes telles que 'Open', 'High', 'Low', 'Close', 'Adj.Close', et 'Volume'.

```
#Check for Negative Values: Ensure that there are no negative values in numerical columns
any(data$Open < 0)
```

```
[1] FALSE
```

```
any(data$High < 0)
```

```
[1] FALSE
```

```r
any(data$Low < 0)
```

```
[1] FALSE
```

```r
any(data$Close < 0)
```

```
[1] FALSE
```

```r
any(data$Adj.Close < 0)
```

```
[1] FALSE
```

```r
any(data$Volume < 0)
```

```
[1] FALSE
```

4. **Vérification de la cohérence des valeurs 'High' et 'Low' :**

   Une étape cruciale a été de garantir la cohérence des données en vérifiant que les valeurs dans la colonne 'High' étaient toujours supérieures ou égales aux valeurs correspondantes dans la colonne 'Low'.

```r
#Check Consistency: Ensure that 'High' values are greater than or equal to 'Low' values
any(data$High < data$Low)
```

```
[1] FALSE
```

5. **Résumé statistique des données nettoyées :**

   Après ce nettoyage initial, nous avons généré un résumé statistique pour la colonne 'Close', ce qui nous a donné un aperçu des propriétés centrales et de la dispersion de nos données.

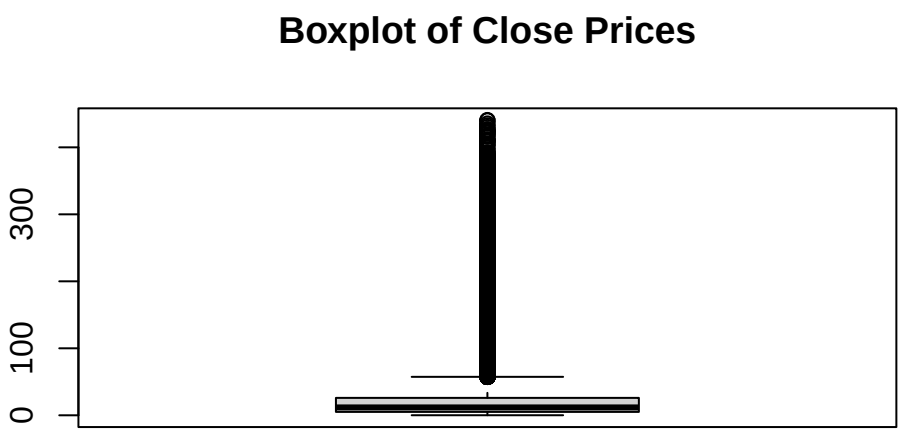
```r
# Summary statistics for the 'Close' column after cleaning
summary(data$Close)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0491  5.0400 11.7200 24.5475 25.9794 440.6200 
```

```r
# Summary statistics for the 'Close' column
summary(data)
```

```
     Date                 Open              High              Low         
Min.   :1962-01-02   Min.   :  0.000   Min.   :  0.0497   Min.   :  0.0491  
1st Qu.:1997-08-20   1st Qu.:  4.981   1st Qu.:  5.1200   1st Qu.:  4.9700  
Median :2009-04-02   Median : 11.730   Median : 11.8750   Median : 11.5700  
Mean   :2006-04-11   Mean   : 24.479   Mean   : 24.8666   Mean   : 24.2217  
3rd Qu.:2017-06-30   3rd Qu.: 25.978   3rd Qu.: 26.2900   3rd Qu.: 25.6500  
Max.   :2024-04-11   Max.   :446.010   Max.   :446.0100   Max.   :440.1900  
    Close            Adj.Close           Volume             Company         
Min.   :  0.0491   Min.   :  0.038   Min.   :0.000e+00   Length:158768     
1st Qu.:  5.0400   1st Qu.:  3.050   1st Qu.:6.700e+05   Class :character  
Median : 11.7200   Median :  8.003   Median :3.321e+06   Mode  :character  
Mean   : 24.5475   Mean   : 20.097   Mean   :3.607e+07                     
3rd Qu.: 25.9794   3rd Qu.: 17.416   3rd Qu.:1.186e+07                     
Max.   :440.6200   Max.   :430.300   Max.   :7.422e+09                     
```

```r
# Boxplot to visualize outliers
boxplot(data$Close, main = "Boxplot of Close Prices")
```



```r
# Calculate the Interquartile Range (IQR)
Q1 <- quantile(data$Close, 0.25)
Q3 <- quantile(data$Close, 0.75)
IQR <- Q3 - Q1

# Define the lower and upper bounds for outlier detection
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR

# Identify outliers
outliers <- data$Close[data$Close < lower_bound | data$Close > upper_bound]

# Remove outliers
data_clean <- data[!(data$Close %in% outliers), ]
```

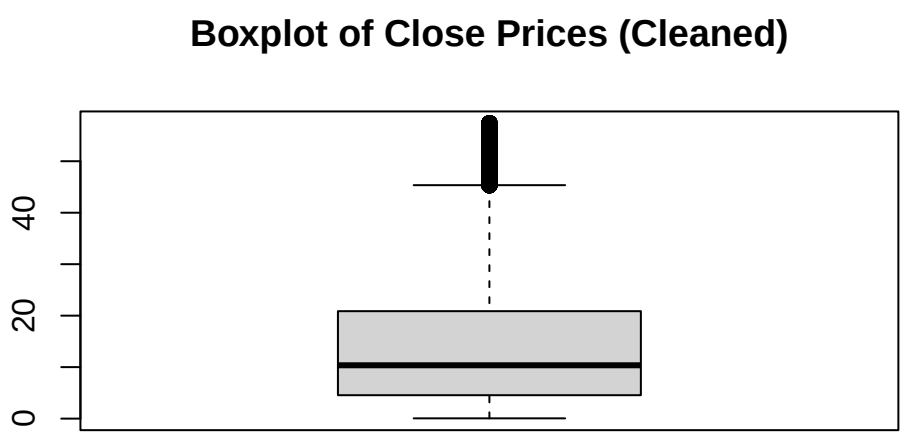
6. **Visualisation des valeurs aberrantes (outliers) :**

   Pour détecter et traiter les valeurs aberrantes, nous avons créé un diagramme en boîte pour visualiser la distribution des prix de clôture après le nettoyage. Cela nous a permis de mieux comprendre et gérer les valeurs extrêmes qui pourraient affecter nos analyses.

```r
# Summary statistics for the cleaned dataset
summary(data_clean$Close)
```

```
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.04911  4.55000 10.36000 14.39168 20.87000 57.38000
```

```r
# Boxplot of the cleaned dataset
boxplot(data_clean$Close, main = "Boxplot of Close Prices (Cleaned)")
```



4 Analyse des donnees

1. **Chargement des bibliothèques nécessaires :**

   Nous avons commencé par charger les bibliothèques essentielles telles que ggplot2, lmtest et dplyr pour effectuer nos analyses et visualisations.

```
# Load required library
library(ggplot2)

# Load necessary libraries
library(lmtest)  # For econometric modeling
```

```
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```
library(dplyr)
```

```
Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

2. **Conversion de la colonne "Date" :**

   Nous avons converti la colonne "Date" en type de données Date pour faciliter la manipulation temporelle de nos données.

```
# Convert 'Date' column to Date type
data$Date <- as.Date(data$Date)
```

3. **Filtrage des données :**

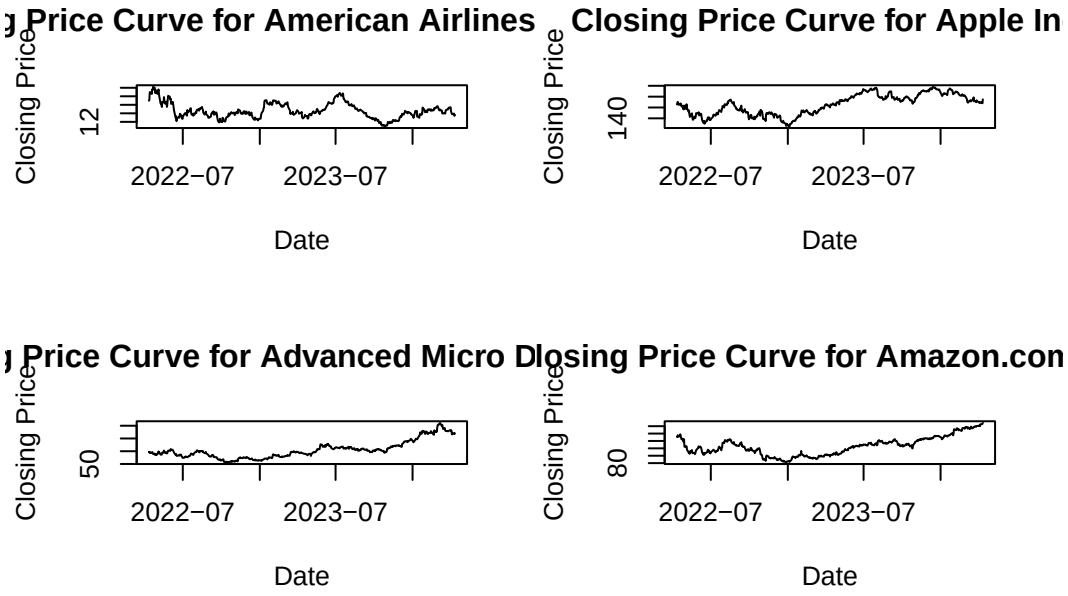
   Les données ont été filtrées pour inclure uniquement les sociétés sélectionnées et les données des six derniers mois, ce qui nous a permis de nous concentrer sur des données récentes et pertinentes.

```
# Specify the companies to include in the plots
selected_companies <- c("American Airlines Group Inc.", "Apple Inc.", "Advanced Micro

# Filter data for selected companies
selected_data <- data[data$Company %in% selected_companies, ]

# Filter data for the last 2 years
six_months_ago <- as.Date("2022-04-111")
selected_data <- selected_data[selected_data$Date >= six_months_ago, ]
```

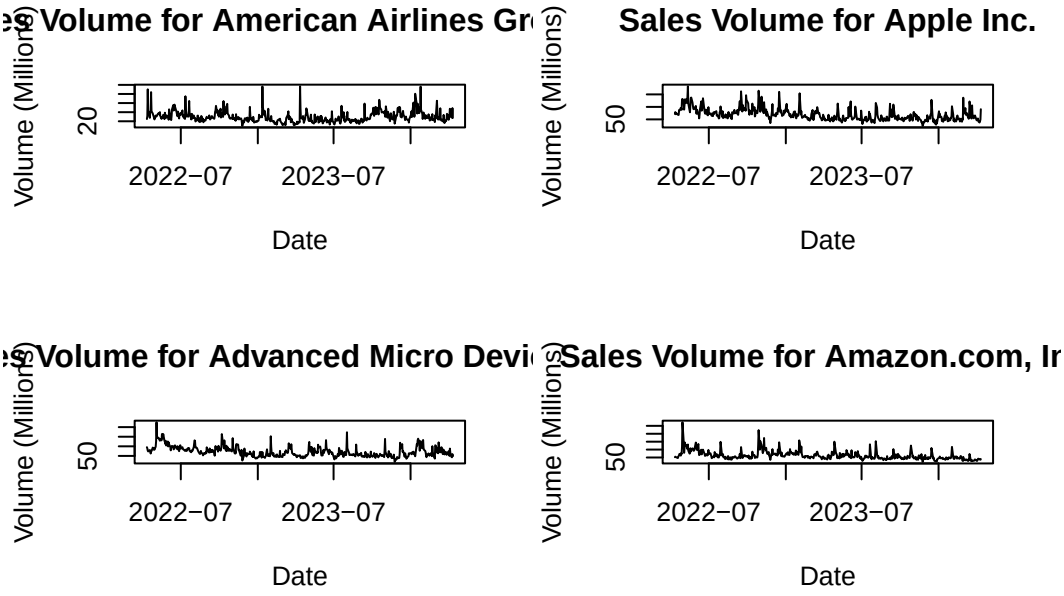
4. **Création de graphiques individuels :**

   Des graphiques ont été créés pour chaque société sélectionnée, montrant l'évolution du prix de clôture et du volume de ventes, offrant ainsi une perspective visuelle sur les performances des entreprises étudiées.

```r
# Set up layout for plots
par(mfrow = c(2, 2))

# Create individual plots for each selected company - Closing Price
for (company in selected_companies) {
  subset_data <- selected_data[selected_data$Company == company, ]
  plot(subset_data$Date, subset_data$Close, type = "l",
       main = paste("Closing Price Curve for", company),
       xlab = "Date", ylab = "Closing Price")
}
```




```r
# Create individual plots for sales volume for each selected company
for (company in selected_companies) {
  subset_data <- selected_data[selected_data$Company == company, ]
  plot(subset_data$Date, subset_data$Volume / 1e6, type = "l",
       main = paste("Sales Volume for", company),
       xlab = "Date", ylab = "Volume (Millions)")
}
```

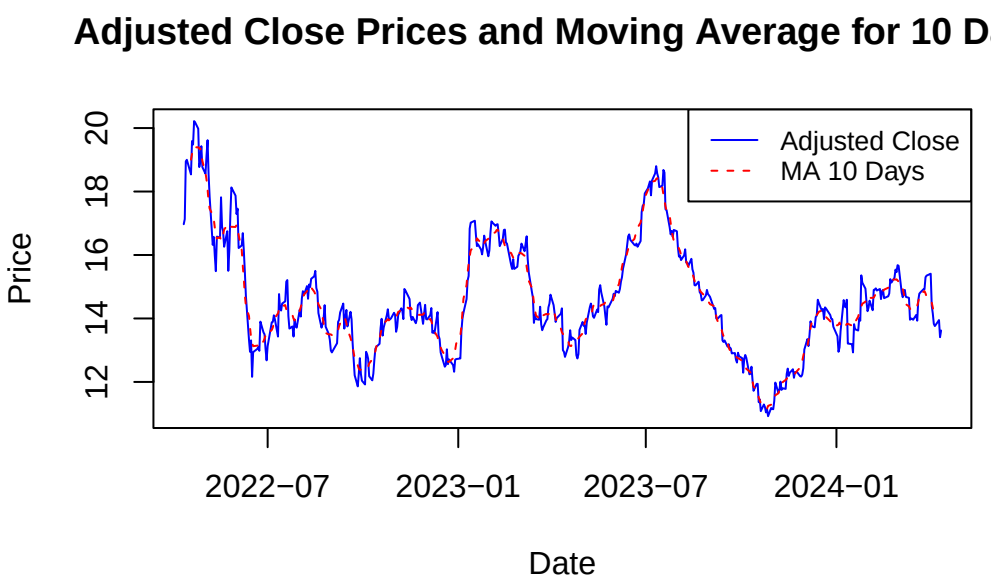


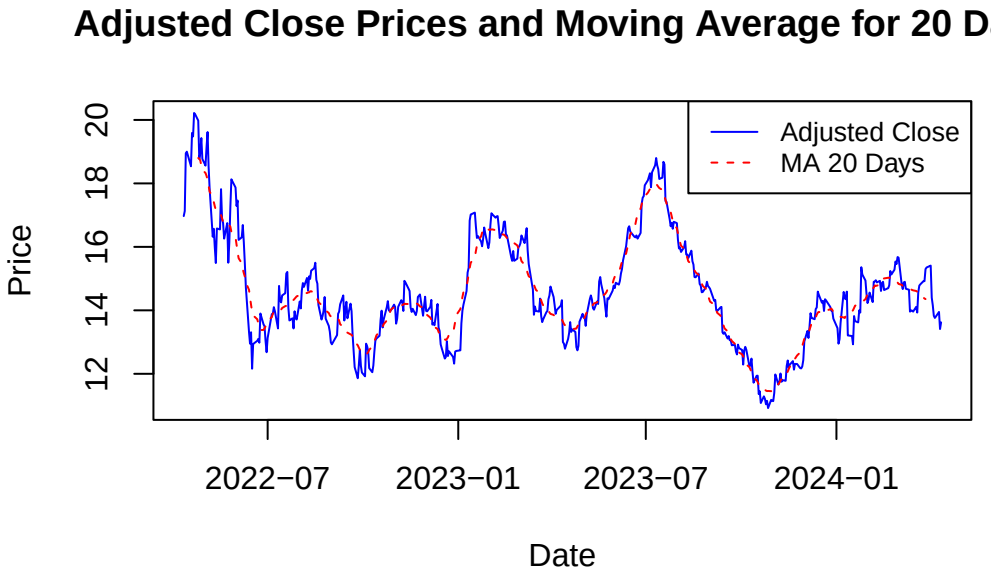
5. **Calcul des moyennes mobiles :**

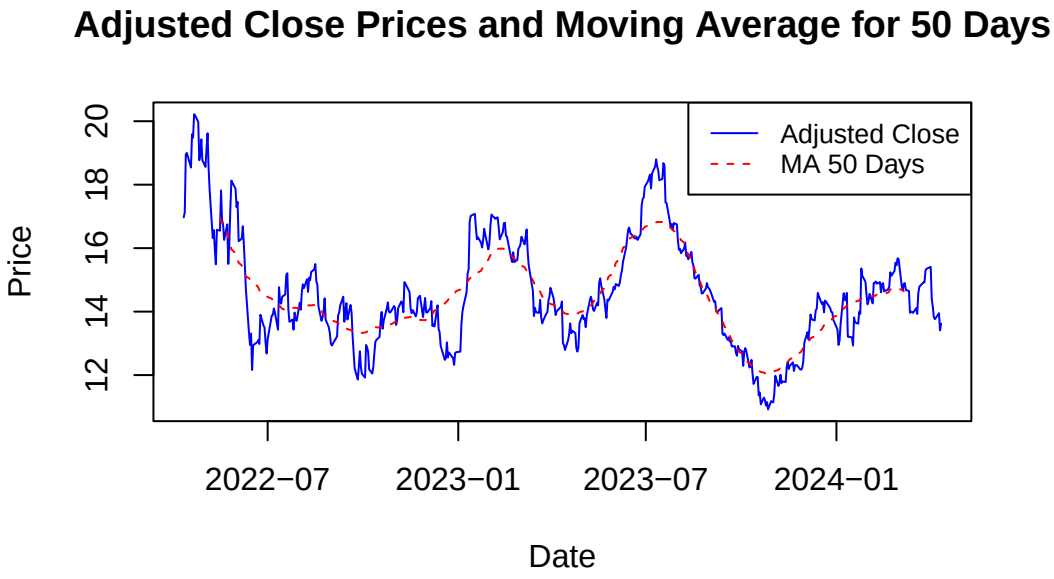
   Nous avons calculé les moyennes mobiles pour chaque société et représenté graphiquement les prix de clôture ajustés et les moyennes mobiles pour observer les tendances à long terme.

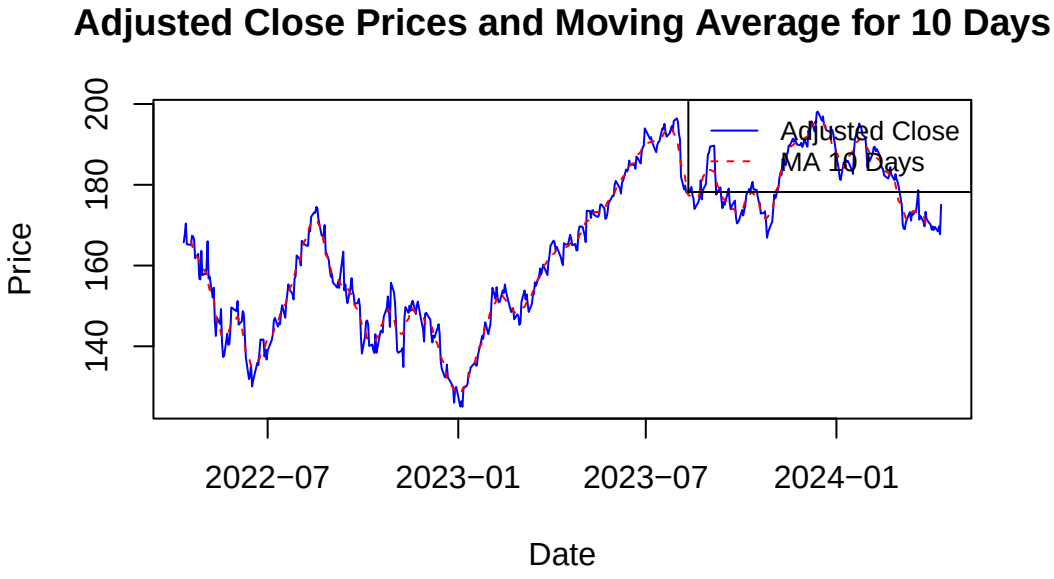
```
# Calculate moving averages for each company
ma_periods <- c(10, 20, 50)  # Moving average periods

# Create plots
par(mfrow = c(1, 1))  # Set up plotting layout
for (company in selected_companies) {
  for (period in ma_periods) {
    subset_data <- selected_data[selected_data$Company == company, ]
    ma_column_name <- paste("MA", period, "Days")
    subset_data[[ma_column_name]] <- stats::filter(subset_data$Close, rep(1/period, period

    # Plot adjusted close prices and moving averages
    plot(subset_data$Date, subset_data$Close, type = "l", col = "blue", xlab = "Date", yla
         main = paste("Adjusted Close Prices and Moving Average for", period, "Days"))
    lines(subset_data$Date, subset_data[[ma_column_name]], col = "red", lty = "dashed")
    legend("topright", legend = c("Adjusted Close", paste("MA", period, "Days")),
           col = c("blue", "red"), lty = c("solid", "dashed"), cex = 0.8)
```

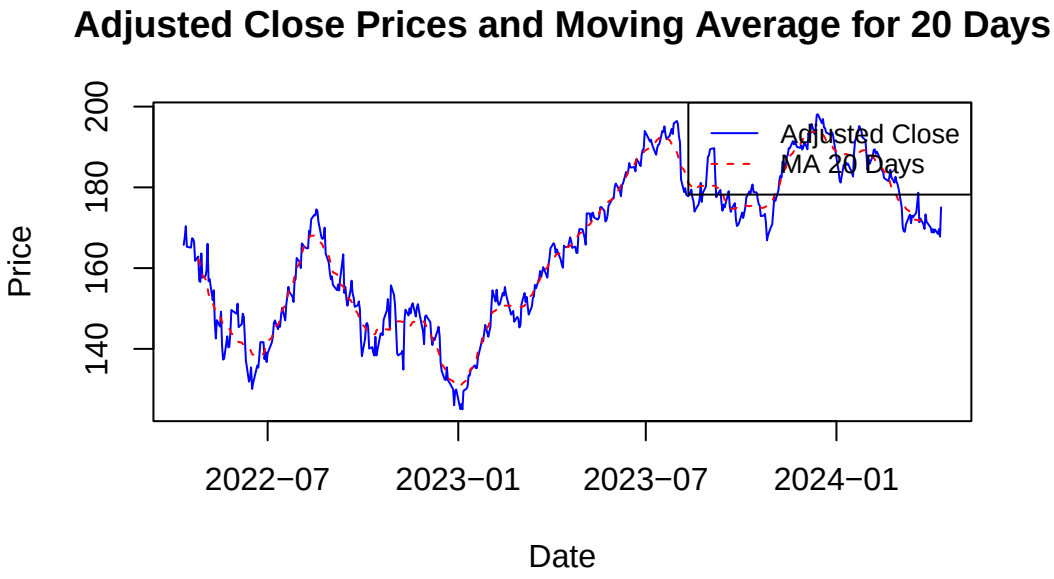
```
  }
}
```

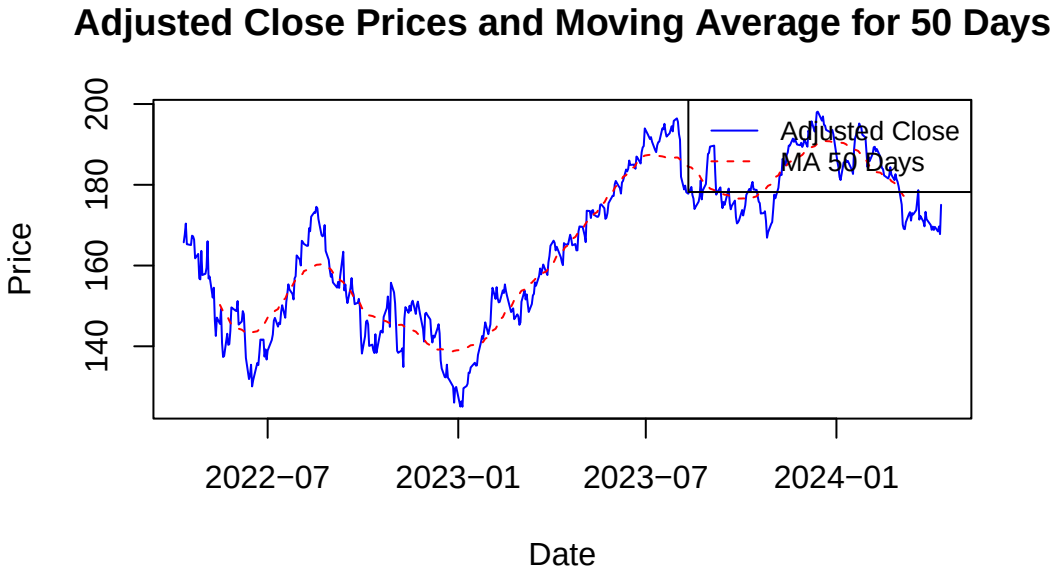

Adjusted Close Prices and Moving Average for 10 Days
Price
12
14
16
18
20
2022−07
2023−01
2023−07
2024−01
Date
Adjusted Close
MA 10 Days

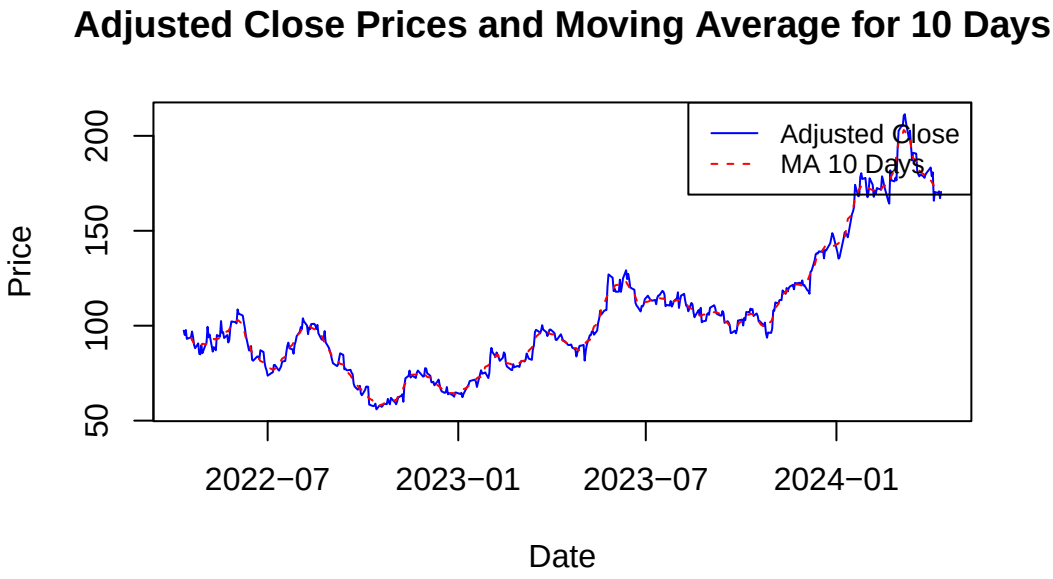

Adjusted Close Prices and Moving Average for 20 Days
Price
12
14
16
18
20
2022−07
2023−01
2023−07
2024−01
Date
Adjusted Close
MA 20 Days

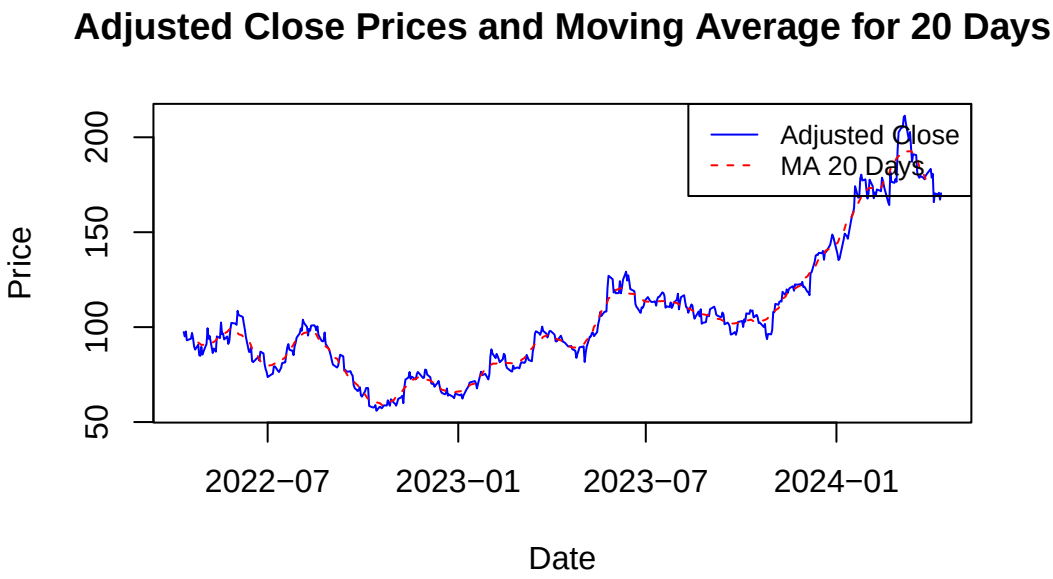

Adjusted Close Prices and Moving Average for 50 Days
Price
12
14
16
18
20
2022−07
2023−01
2023−07
2024−01
Date
Adjusted Close
MA 50 Days

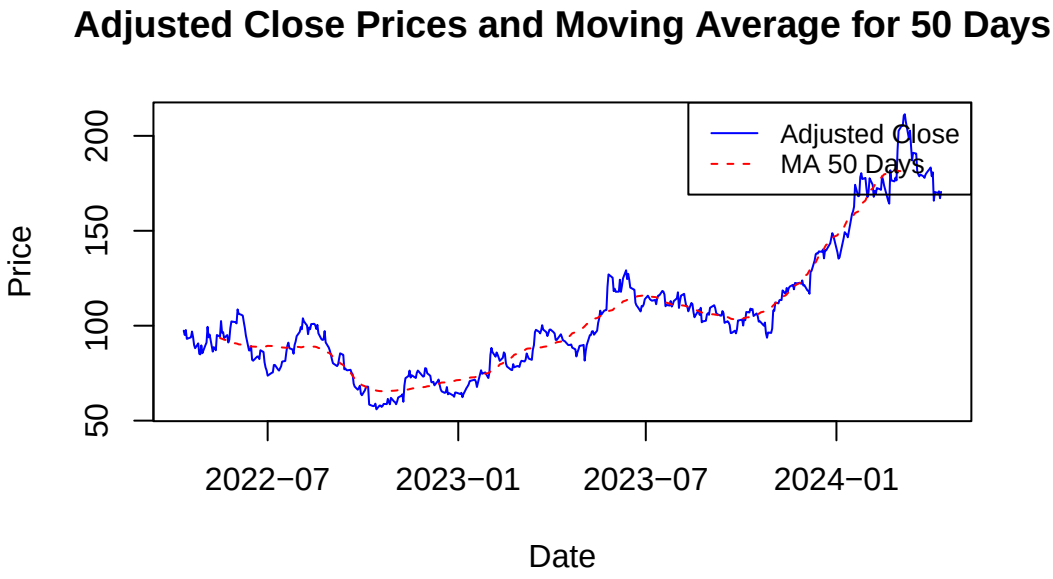

Adjusted Close Prices and Moving Average for 10 Days
Price
140
160
180
200
2022−07
2023−01
2023−07
2024−01
Date
Adjusted Close
MA 10 Days

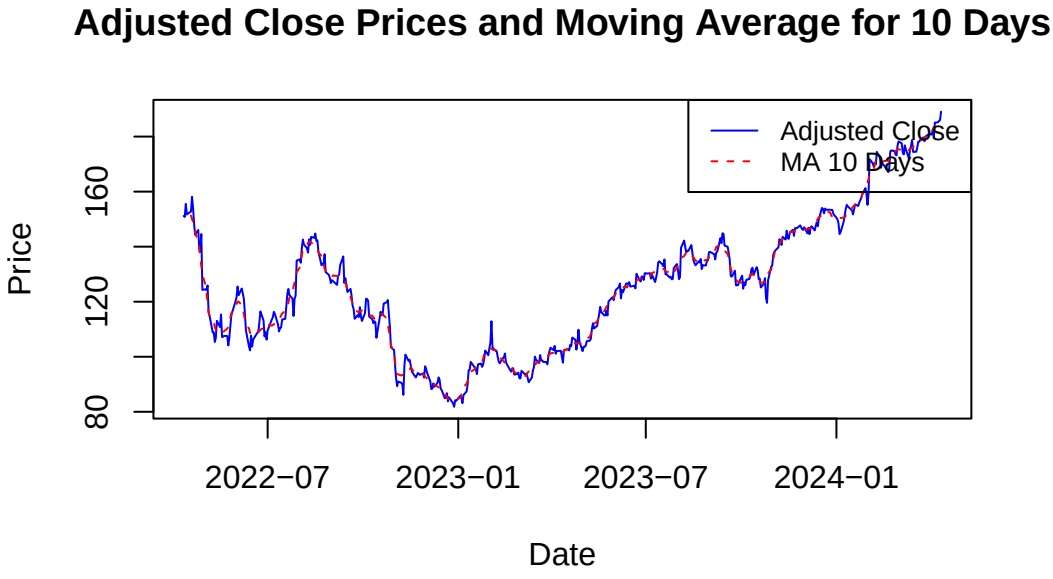

Adjusted Close Prices and Moving Average for 20 Days
Price
140
160
180
200
2022−07
2023−01
2023−07
2024−01
Date
Adjusted Close
MA 20 Days

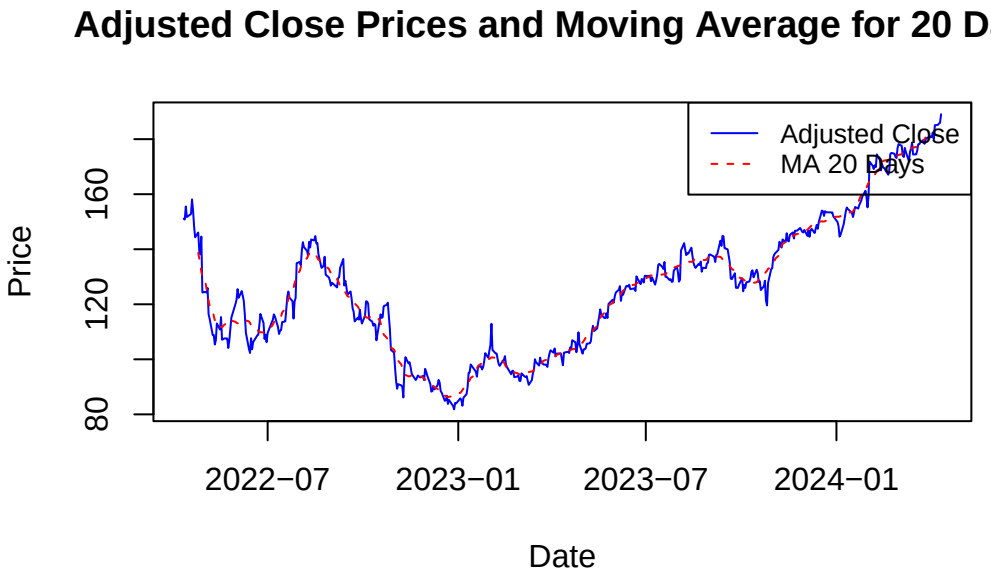

Adjusted Close Prices and Moving Average for 50 Days
Price
140
160
180
200
2022−07
2023−01
2023−07
2024−01
Date
Adjusted Close
MA 50 Days

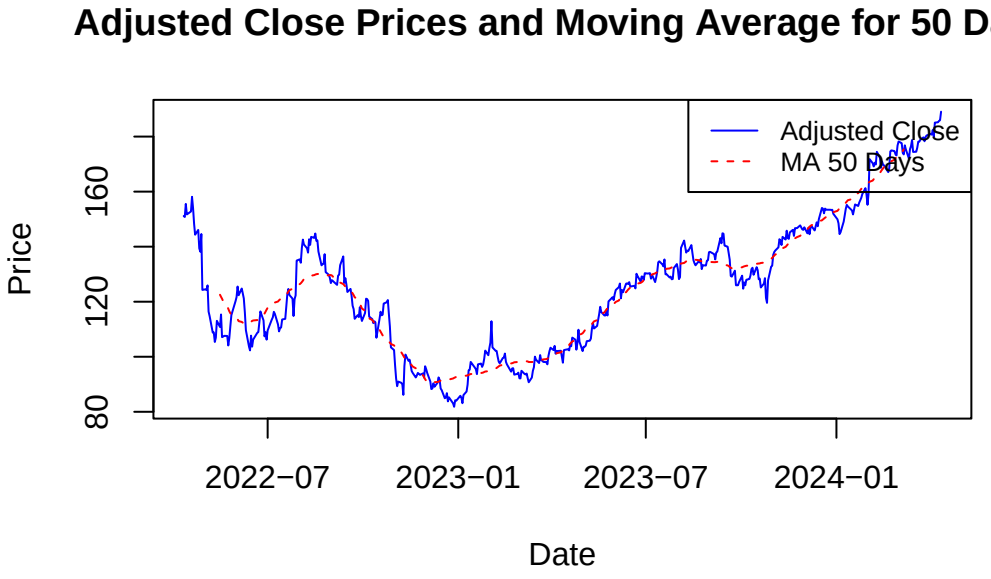

Adjusted Close Prices and Moving Average for 10 Days
Adjusted Close
MA 10 Days
Price
50
100
150
200
2022−07
2023−01
2023−07
2024−01
Date


Adjusted Close Prices and Moving Average for 20 Days
Adjusted Close
MA 20 Days
Price
50
100
150
200
2022−07
2023−01
2023−07
2024−01
Date


Adjusted Close Prices and Moving Average for 50 Days
Adjusted Close
MA 50 Days
Price
50
100
150
200
2022−07
2023−01
2023−07
2024−01
Date


Adjusted Close Prices and Moving Average for 10 Days
Adjusted Close
MA 10 Days
Price
80
120
160
2022−07
2023−01
2023−07
2024−01
Date





6. **Calcul de la corrélation :**

   En calculant la corrélation entre les prix d'ouverture et de clôture, ainsi qu'entre les

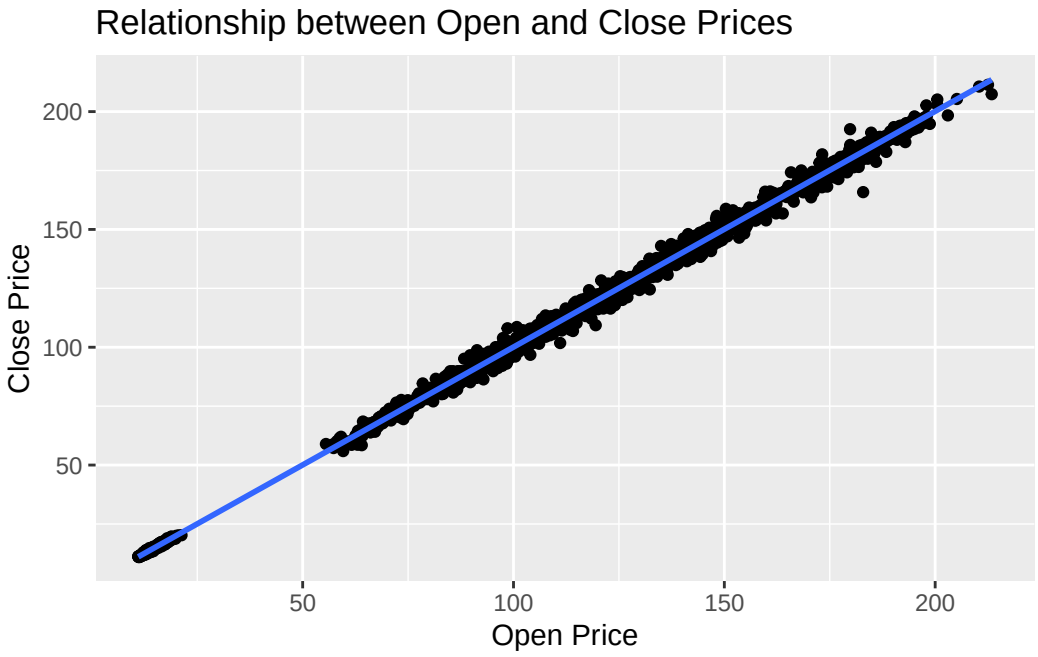
prix les plus élevés et les prix les plus bas, nous avons pu visualiser ces relations à l'aide de graphiques de dispersion, ce qui nous a donné des informations sur les liens entre différentes variables.

```
# Calculate correlation between Open and Close prices
correlation <- cor(selected_data$Open, selected_data$Close)
print(paste("Correlation between Open and Close prices:", correlation))
```

```
[1] "Correlation between Open and Close prices: 0.999346192428546"
```

```
# Plot the relationship between Open and Close prices
ggplot(selected_data, aes(x = Open, y = Close)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Open Price", y = "Close Price", title = "Relationship between Open and Close P
```

```
`geom_smooth()` using formula = 'y ~ x'
```



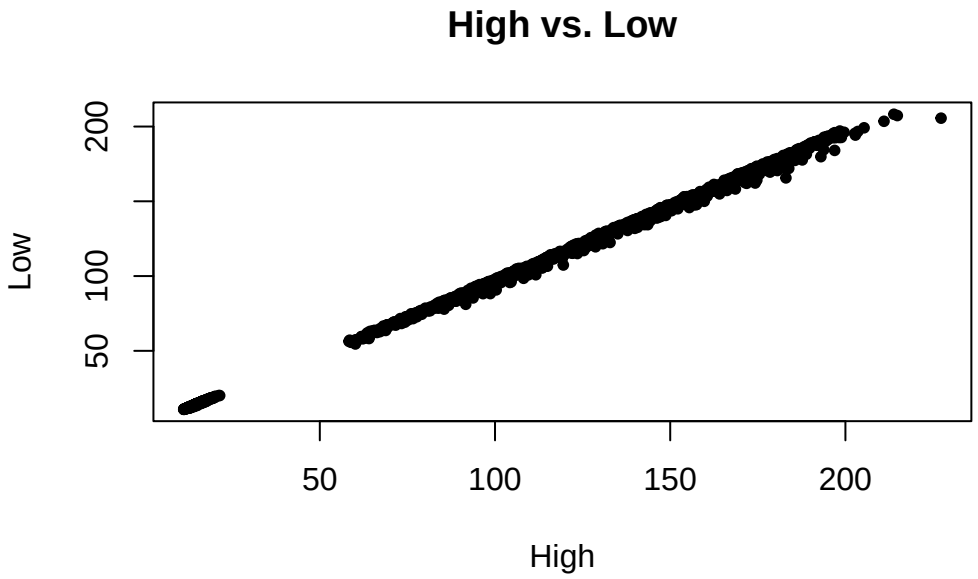
```r
# Perform econometric modeling
model <- lm(Close ~ Open, data = selected_data)
print(summary(model))
```

```
Call:
lm(formula = Close ~ Open, data = selected_data)

Residuals:
     Min       1Q   Median       3Q      Max
-17.1968  -0.9122  -0.0257   1.0144  12.6041

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 0.0519271  0.0958536    0.542    0.588
Open        1.0002998  0.0008072 1239.210   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.167 on 2010 degrees of freedom
Multiple R-squared:  0.9987,    Adjusted R-squared:  0.9987
F-statistic: 1.536e+06 on 1 and 2010 DF,  p-value: < 2.2e-16
```

```r
# Perform heteroscedasticity test
bptest(model)
```

```
    studentized Breusch-Pagan test

data:  model
BP = 84.993, df = 1, p-value < 2.2e-16
```

```r
# Calculate correlation between High and Low columns
correlation <- cor(selected_data$High, selected_data$Low)

# Plot High vs. Low
plot(selected_data$High, selected_data$Low, xlab = "High", ylab = "Low", main = "High vs.
```




```
# Print the correlation coefficient
print(correlation)
```

```
[1] 0.9995816
```

```
# Fit an econometric model
model <- lm(Close ~ Open + High + Low + Volume, data = selected_data)

# Summary of the model
summary(model)
```

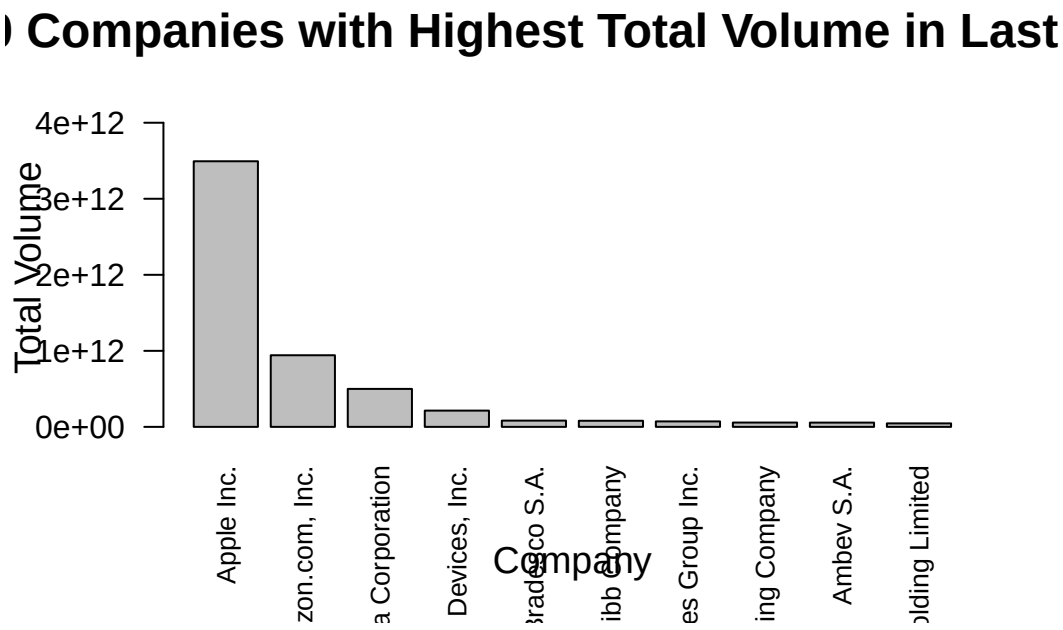
```
Call:
lm(formula = Close ~ Open + High + Low + Volume, data = selected_data)

Residuals:
    Min      1Q  Median      3Q     Max
-9.8163 -0.4028  0.0095  0.4464  5.6636

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) -6.630e-02  5.309e-02  -1.249 0.211894
Open        -6.125e-01  1.724e-02 -35.526  < 2e-16 ***
High         7.273e-01  1.828e-02  39.779  < 2e-16 ***
Low          8.864e-01  1.810e-02  48.985  < 2e-16 ***
Volume       3.464e-09  1.041e-09   3.326 0.000896 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9343 on 2007 degrees of freedom
Multiple R-squared:  0.9998,    Adjusted R-squared:  0.9998
F-statistic: 2.067e+06 on 4 and 2007 DF,  p-value: < 2.2e-16
```

7. **Groupement et analyse du volume des ventes :** Nous avons groupé les données par société pour calculer le volume total des ventes et identifier les dix sociétés ayant le volume le plus élevé au cours des six derniers mois, ce qui est essentiel pour comprendre l'activité économique de ces entreprises.

```r
# Group by 'Company' and calculate total volume
total_volume <- data %>%
  group_by(Company) %>%
  summarise(Total_Volume = sum(Volume)) %>%
  arrange(desc(Total_Volume))  # Sort by total volume in descending order

# Select the topdata# Select the top 10 companies
top_10_companies <- total_volume %>%
  top_n(10, Total_Volume)

# Increase plot size
options(repr.plot.width = 10, repr.plot.height = 6)  # Adjust width and height as needed

# Plot bar chart with custom design
barplot(top_10_companies$Total_Volume,
        names.arg = top_10_companies$Company,
        xlab = "Company",
        ylab = "Total Volume",
        main = "Top 10 Companies with Highest Total Volume in Last 6 Months",
        border = "black",
        ylim = c(0, max(total_volume$Total_Volume) * 1.2),
        las = 2,  # Rotate company names if needed for better visualization
        cex.names = 0.8,  # Adjust font size for company names
        cex.lab = 1.2,  # Adjust font size for axis labels
        cex.main = 1.5,  # Adjust font size for main title
```

```
        width = 0.5)  # Adjust bar width as needed
```



5 Prediction :

1. **Filtrage des données pour AMD :**

   Les données ont été filtrées pour inclurex uniquement les entrées liées à AMD, une étape préalable à la prédiction spécifique pour cette entreprise.

```
# Filter data for AMD
amd_data <- data[data$Company == "Advanced Micro Devices, Inc.", ]
```

2. **Imputation des valeurs manquantes :**

   Nous avons remplacé les valeurs manquantes dans la colonne “Close” par la moyenne des valeurs disponibles pour AMD, assurant ainsi la continuité des données nécessaires à la prédiction.

```
# Impute missing values with mean for AMD
amd_data$Close <- ifelse(is.na(amd_data$Close), mean(amd_data$Close, na.rm = TRUE), amd_da
```

```
# Ensure 'amd_data' is sorted by date
amd_data <- amd_data[order(amd_data$Date), ]
```

3. **Ingénierie des fonctionnalités :**

   En créant des variables retardées basées sur les prix de clôture antérieurs, nous avons préparé nos données pour l'entraînement du modèle de prédiction.

```
# Feature engineering: Create lagged variables for previous closing prices
lagged_prices <- c(1, 2, 3, 4, 5)  # Lagged periods
for (lag in lagged_prices) {
  amd_data[[paste0("Close_lag", lag)]] <- lag(amd_data$Close, lag)
}
# Remove rows with NA values in lagged variables
amd_data <- na.omit(amd_data)
```

4. **Division des données et définition des caractéristiques :**

   Les données ont été divisées en ensembles d'entraînement et de test, avec la définition des caractéristiques à utiliser pour la prédiction et la variable cible, à savoir les prix de clôture.

```
# Split the data into training and testing sets
train_size <- 0.8  # 80% of data for training
train_index <- 1:round(nrow(amd_data) * train_size)
train_data <- amd_data[train_index, ]
test_data <- amd_data[-train_index, ]

# Define features and target variable
features <- c("Close_lag1", "Close_lag2", "Close_lag3", "Close_lag4", "Close_lag5")
target <- "Close"
```

5. **Entraînement du modèle et évaluation :**

   Un modèle de forêt aléatoire a été entraîné en utilisant les données d'entraînement, puis utilisé pour prédire les prix de clôture sur les données de test. Enfin, les métriques de précision telles que MAE, MSE et RMSE ont été calculées pour évaluer la performance du modèle.

‘‘‘