Stock_Price_Forecasting_Investment_Analysis

Lei Zhao

Load Data and Packages

We first load the required libraries and read the dataset.

```r
# Load necessary libraries
library(readxl)
library(ggplot2)
library(dplyr)
library(forecast)
library(zoo)

# Read the dataset
data <- readxl::read_excel("../data/data.xlsx")
data <- na.omit(data)

# Rename columns for easier reference
colnames(data) <- c("Date", "Period", "AAPL_Price", "AAPL_Volume", "HON_Price", "HON_Volume")

# Convert Date column to Date format
data$Date <- as.Date(data$Date)

# Display the structure of the dataset
str(data)
```

```
## tibble [252 x 6] (S3: tbl_df/tbl/data.frame)
##  $ Date       : Date[1:252], format: "2019-11-08" "2019-11-11" ...
##  $ Period     : num [1:252] 1 2 3 4 5 6 7 8 9 10 ...
##  $ AAPL_Price : num [1:252] 64 64.5 64.4 65 64.6 ...
##  $ AAPL_Volume: num [1:252] 7.00e+07 8.18e+07 8.74e+07 1.03e+08 8.92e+07 ...
##  $ HON_Price  : num [1:252] 177 177 178 178 176 ...
##  $ HON_Volume : num [1:252] 1636800 1594600 1816900 1855200 2208300 ...
##  - attr(*, "na.action")= 'omit' Named int [1:5] 253 254 255 256 257
##   ..- attr(*, "names")= chr [1:5] "253" "254" "255" "256" ...
```

```r
# Show first few rows
head(data)
```

```
## # A tibble: 6 x 6
##   Date       Period AAPL_Price AAPL_Volume HON_Price HON_Volume
##   <date>      <dbl>      <dbl>       <dbl>     <dbl>      <dbl>
## 1 2019-11-08      1       64.0    69986400      177.    1636800
## 2 2019-11-11      2       64.5    81821200      177.    1594600
## 3 2019-11-12      3       64.4    87388800      178.    1816900
```

```
## 4 2019-11-13      4        65.0   102734400       178.     1855200
## 5 2019-11-14      5        64.6    89182800       176.     2208300
## 6 2019-11-15      6        65.3   100206400       178.     3242700
```

Part 1 (i)

Objective In this section, we analyze the stock price trends of Apple Inc. (AAPL) and Honeywell International Inc. (HON) over time. Our goal is to identify any trends, seasonal patterns, or irregular fluctuations in the stock prices.

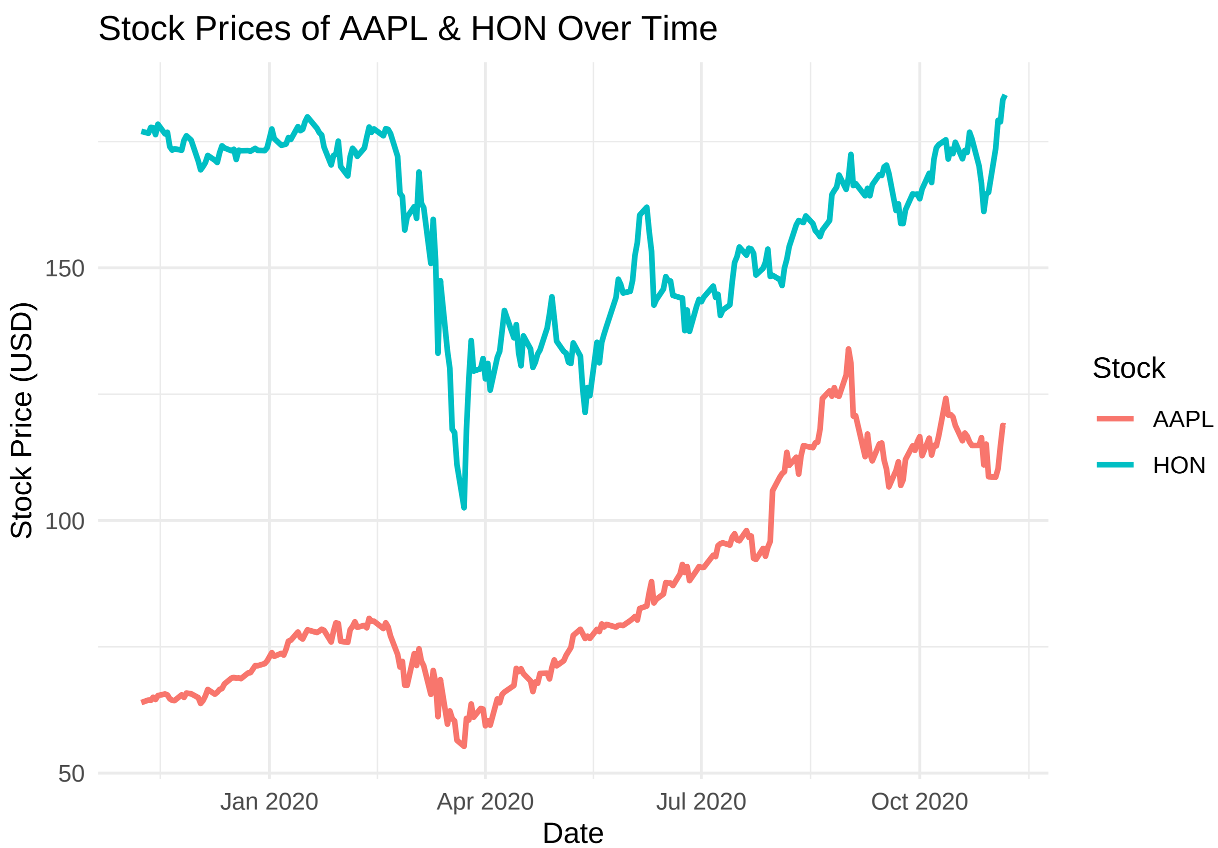
Line Plot Analysis The figure below presents the stock price movements of AAPL and HON over the given time period. The red solid line represents AAPL's stock price, while the blue solid line represents HON's stock price. Additionally, we overlay dashed lines representing smoothed trends using LOESS smoothing.

From the graph, we observe:

- AAPL shows a clear upward trend, suggesting sustained growth over the analyzed period.
- HON exhibits more volatility, with periods of decline and recovery, making its movement less predictable.
- There may be short-term fluctuations, possibly due to market conditions or external factors.

```r
# Line plot of AAPL & HON stock prices
ggplot(data, aes(x = Date)) +
  geom_line(aes(y = AAPL_Price, color = "AAPL"), size = 1) +
  geom_line(aes(y = HON_Price, color = "HON"), size = 1) +
  labs(title = "Stock Prices of AAPL & HON Over Time",
       x = "Date",
       y = "Stock Price (USD)",
       color = "Stock") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Analysis AAPL shows an upward trend, indicating consistent growth. HON exhibits more fluctuations, suggesting volatility. Both stocks might have seasonal patterns or external market influences.

Stock Price Decomposition To further investigate whether there are seasonal patterns or other underlying components, we perform STL decomposition on AAPL's stock price. This decomposition helps separate the data into:

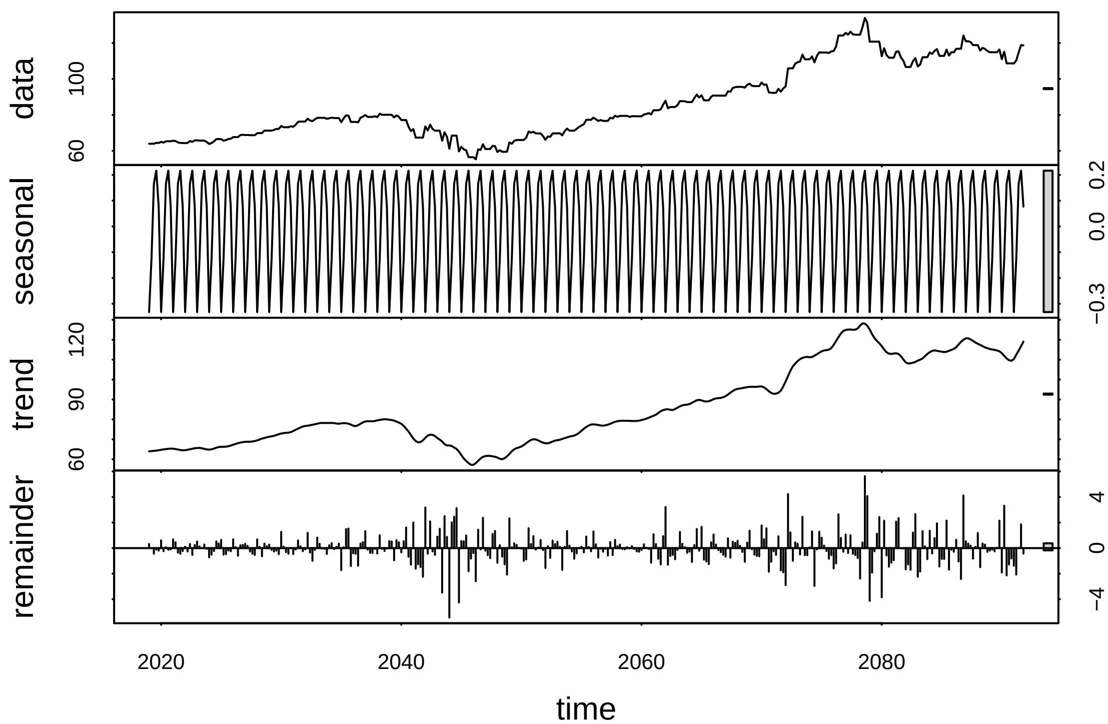
- Trend Component: The long-term movement of the stock price.
- Seasonal Component: Potential cyclical patterns that repeat over time.
- Residual Component: Random noise or irregular variations.

By analyzing the decomposition results:

- The trend component confirms the general upward movement of AAPL stock.
- The seasonal component is relatively weak, suggesting that short-term fluctuations are more likely due to external factors rather than inherent seasonality.
- The residual component shows irregular fluctuations, which may be due to market news or earnings reports.

```
# Fill missing values to maintain continuity
data_full <- data.frame(Date = seq(min(data$Date), max(data$Date), by = "day"))
data_full <- merge(data_full, data, by = "Date", all.x = TRUE)
data_full$AAPL_Price <- na.locf(data_full$AAPL_Price)  # Fill missing values
```

```r
# Convert AAPL_Price to time series with adjusted frequency
aapl_ts <- ts(data_full$AAPL_Price, frequency = 5, start = c(2019, 1))  # Assuming monthly periodicity

# Perform STL decomposition
aapl_decomp <- stl(aapl_ts, s.window = "periodic")

# Plot decomposition
plot(aapl_decomp)
```



Rolling Volatility Analysis To quantify the fluctuations, we calculate the 20-day rolling standard deviation for both AAPL and HON:

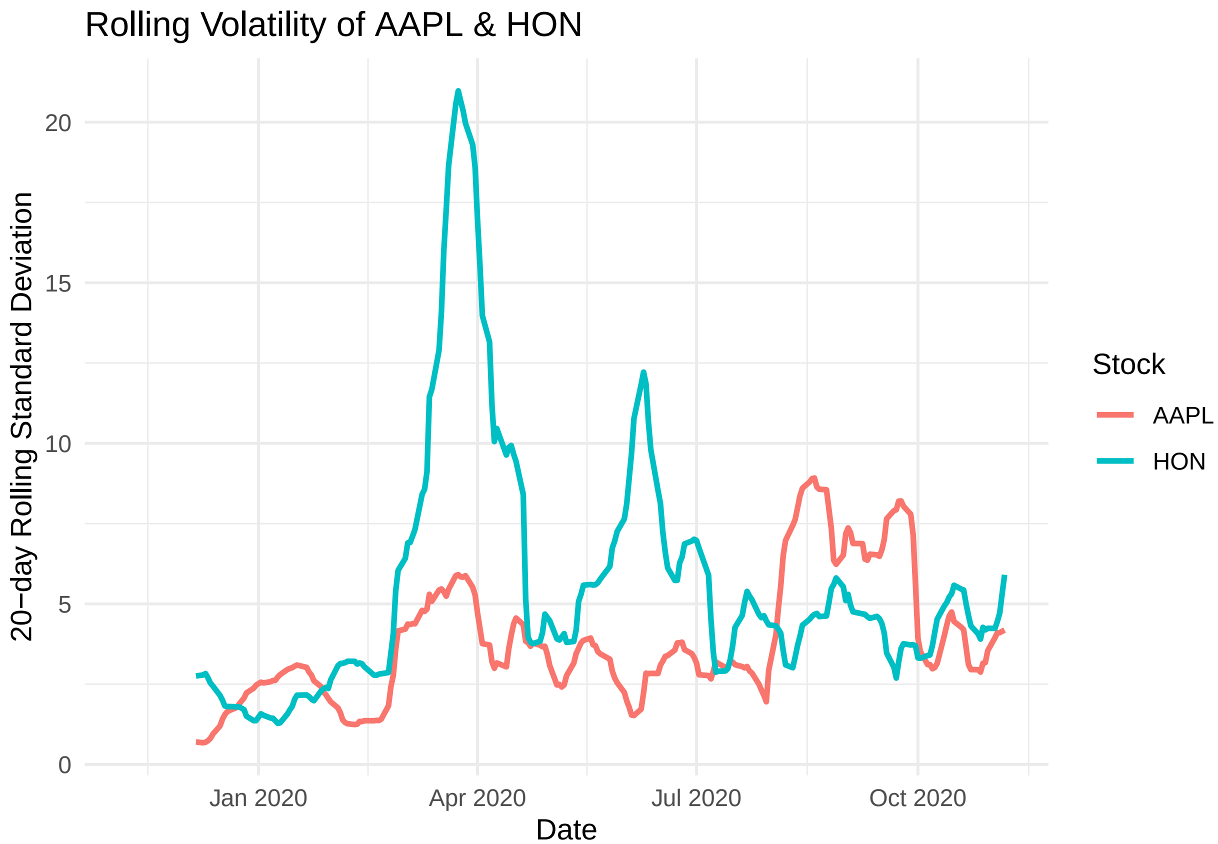
- AAPL has lower volatility, meaning its price movements are more stable over time.
- HON experiences higher fluctuations, indicating greater risk for investors.

Overall, these observations will be critical in later sections when we determine the best forecasting methods and portfolio allocation strategies.

```r
# Calculate rolling volatility (Standard Deviation over a 20-day window)
data <- data %>%
  mutate(AAPL_Volatility = zoo::rollapply(AAPL_Price, width = 20, FUN = sd, fill = NA, align = "right")
         HON_Volatility = zoo::rollapply(HON_Price, width = 20, FUN = sd, fill = NA, align = "right"))
```

```r
# Plot rolling volatility
ggplot(data, aes(x = Date)) +
  geom_line(aes(y = AAPL_Volatility, color = "AAPL"), size = 1) +
  geom_line(aes(y = HON_Volatility, color = "HON"), size = 1) +
  labs(title = "Rolling Volatility of AAPL & HON",
       x = "Date",
       y = "20-day Rolling Standard Deviation",
       color = "Stock") +
  theme_minimal()
```

```
## Warning: Removed 19 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 19 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



Part 1 (ii): Exponential Smoothing Forecast

Objective

- Use Simple Exponential Smoothing (SES) to forecast AAPL stock prices.
- Test different alpha (smoothing factor) values: 0.15, 0.35, 0.55, 0.75.
- Evaluate the accuracy using MAPE (Mean Absolute Percentage Error).

```r
# Define different alpha values
alpha_values <- c(0.15, 0.35, 0.55, 0.75)

# Create data frames to store results
results_AAPL <- data.frame(Alpha = alpha_values, MAPE = NA, Forecast_253 = NA)
results_HON <- data.frame(Alpha = alpha_values, MAPE = NA, Forecast_253 = NA)

# Function to calculate MAPE
mape <- function(actual, forecast) {
  valid_idx <- which(!is.na(actual) & !is.na(forecast))  # Ignore missing values
  actual <- actual[valid_idx]
  forecast <- forecast[valid_idx]
  mean(abs((actual - forecast) / actual), na.rm = TRUE) * 100
}

# Apply exponential smoothing for AAPL and HON
for (i in 1:length(alpha_values)) {
  # Convert stock price data to time series (without specifying frequency)
  ts_AAPL <- ts(data$AAPL_Price)
  ts_HON <- ts(data$HON_Price)

  # AAPL Model
  model_AAPL <- HoltWinters(ts_AAPL, beta = FALSE, gamma = FALSE, alpha = alpha_values[i])
  forecasted_AAPL <- fitted(model_AAPL)[,1]  # Extract fitted values

  # Compute MAPE for AAPL
  actual_AAPL <- data$AAPL_Price[1:length(forecasted_AAPL)]
  results_AAPL$MAPE[i] <- mape(actual_AAPL, forecasted_AAPL)

  # Forecast Period 253 for AAPL
  future_forecast_AAPL <- forecast(model_AAPL, h = 1)
  results_AAPL$Forecast_253[i] <- future_forecast_AAPL$mean[1]

  # HON Model
  model_HON <- HoltWinters(ts_HON, beta = FALSE, gamma = FALSE, alpha = alpha_values[i])
  forecasted_HON <- fitted(model_HON)[,1]  # Extract fitted values

  # Compute MAPE for HON
  actual_HON <- data$HON_Price[1:length(forecasted_HON)]
  results_HON$MAPE[i] <- mape(actual_HON, forecasted_HON)

  # Forecast Period 253 for HON
  future_forecast_HON <- forecast(model_HON, h = 1)
  results_HON$Forecast_253[i] <- future_forecast_HON$mean[1]
}

# Print results
print(results_AAPL)
```

```
##   Alpha      MAPE Forecast_253
## 1  0.15 3.2802881     114.8753
## 2  0.35 1.5954979     115.9750
## 3  0.55 0.9328949     117.4953
```

```
## 4  0.75 0.4912739     118.3938
```

```r
print(results_HON)
```

```
##   Alpha      MAPE Forecast_253
## 1  0.15 2.5831679     175.3274
## 2  0.35 1.5176569     179.9735
## 3  0.55 0.9108629     182.5755
## 4  0.75 0.4727744     183.7235
```

```r
# Store the best alpha values based on minimum MAPE
best_alpha_AAPL <- results_AAPL$Alpha[which.min(results_AAPL$MAPE)]
best_alpha_HON <- results_HON$Alpha[which.min(results_HON$MAPE)]

# Store the final MAPE values for Part 2 and Part 3
final_mape <- min(results_AAPL$MAPE)  # Use the best AAPL MAPE as final_mape
final_mape_AAPL <- min(results_AAPL$MAPE)
final_mape_HON <- min(results_HON$MAPE)

# Print best alpha and corresponding MAPE
cat("Best alpha for AAPL:", best_alpha_AAPL, "\n")
```

```
## Best alpha for AAPL: 0.75
```

```r
cat("Best alpha for HON:", best_alpha_HON, "\n")
```

```
## Best alpha for HON: 0.75
```

```r
cat("Final MAPE for AAPL:", final_mape_AAPL, "%\n")
```

```
## Final MAPE for AAPL: 0.4912739 %
```

```r
cat("Final MAPE for HON:", final_mape_HON, "%\n")
```

```
## Final MAPE for HON: 0.4727744 %
```

```r
cat("Final MAPE for Part 3:", final_mape, "%\n")  # This ensures Part 3 has final_mape available
```

```
## Final MAPE for Part 3: 0.4912739 %
```

Analysis

- The lower the MAPE value, the better the forecasting accuracy:Based on the table, the optimal alpha value is the one with the smallest MAPE.
- Forecast for Period 253:The predicted AAPL price for period 253 varies depending on the smoothing parameter. A higher assigns more weight to recent observations, making the forecast more responsive to recent changes.
- Selection of the Best alpha:The chosen should balance responsiveness and stability. If alpha is too small, the model reacts too slowly; if too large, it may overreact to short-term fluctuations.

Part 1 (iii): Trend-adjusted Exponential Smoothing

Objective

- Improve forecasting by adding a trend component (beta) to exponential smoothing.
- Use alpha = 0.55 (best from Part 1 ii) and test different beta values: 0.15, 0.25, 0.45, 0.85.
- Calculate MAPE to evaluate the best beta value.

```
# Define beta values
beta_values <- c(0.15, 0.25, 0.45, 0.85)

# Remove NA, NaN, Inf values
data_full$AAPL_Price <- ifelse(is.finite(data_full$AAPL_Price), data_full$AAPL_Price, NA)
data_full$HON_Price <- ifelse(is.finite(data_full$HON_Price), data_full$HON_Price, NA)

# Ensure no missing values remain
data_full <- na.omit(data_full)

# Convert to valid time series (assuming daily trading frequency of 5 per week)
aapl_ts <- ts(data_full$AAPL_Price, frequency = 5, start = c(2019, 1))
hon_ts <- ts(data_full$HON_Price, frequency = 5, start = c(2019, 1))

# Create data frames to store results for AAPL & HON
results_trend_AAPL <- data.frame(Beta = beta_values, MAPE = NA, Forecast_253 = NA)
results_trend_HON <- data.frame(Beta = beta_values, MAPE = NA, Forecast_253 = NA)

# Function to calculate MAPE
mape <- function(actual, forecast) {
  valid_idx <- which(!is.na(actual) & !is.na(forecast))  # Ignore missing values
  actual <- actual[valid_idx]
  forecast <- forecast[valid_idx]
  mean(abs((actual - forecast) / actual), na.rm = TRUE) * 100
}

# Apply trend-adjusted exponential smoothing for AAPL & HON
forecasted_AAPL_list <- list()
forecasted_HON_list <- list()

for (i in 1:length(beta_values)) {
  # AAPL Model
  model_AAPL <- HoltWinters(aapl_ts, gamma = FALSE, alpha = 0.55, beta = beta_values[i])
  forecasted_AAPL <- fitted(model_AAPL)[,1]
  forecasted_AAPL_list[[i]] <- forecasted_AAPL

  # Compute MAPE for AAPL
  actual_AAPL <- data_full$AAPL_Price[1:length(forecasted_AAPL)]
  results_trend_AAPL$MAPE[i] <- mape(actual_AAPL, forecasted_AAPL)

  # Forecast for period 253 (AAPL)
  future_forecast_AAPL <- forecast(model_AAPL, h = 1)
  results_trend_AAPL$Forecast_253[i] <- future_forecast_AAPL$mean[1]

  # HON Model
  model_HON <- HoltWinters(hon_ts, gamma = FALSE, alpha = 0.55, beta = beta_values[i])
```

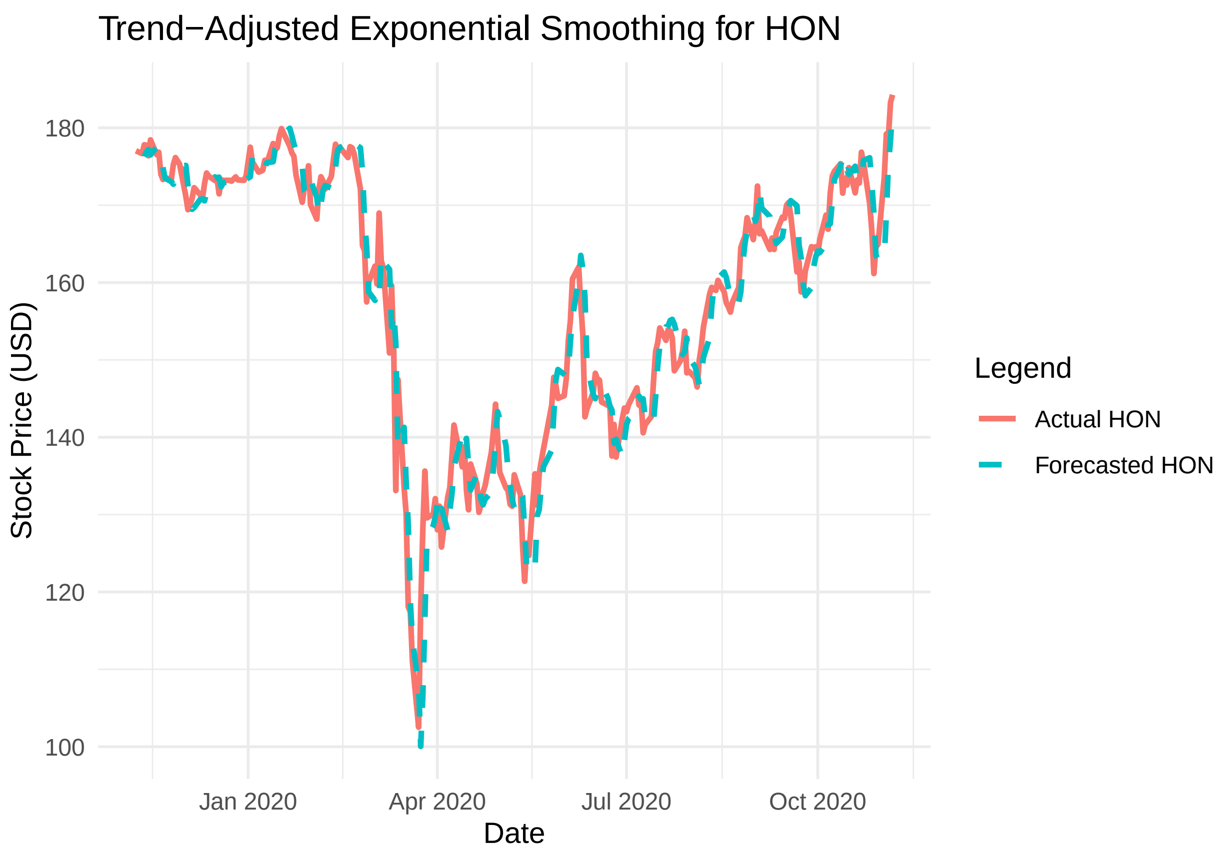
```r
  forecasted_HON <- fitted(model_HON)[,1]
  forecasted_HON_list[[i]] <- forecasted_HON

  # Compute MAPE for HON
  actual_HON <- data_full$HON_Price[1:length(forecasted_HON)]
  results_trend_HON$MAPE[i] <- mape(actual_HON, forecasted_HON)

  # Forecast for period 253 (HON)
  future_forecast_HON <- forecast(model_HON, h = 1)
  results_trend_HON$Forecast_253[i] <- future_forecast_HON$mean[1]
}

# Print results
print(results_trend_AAPL)
```

```
##   Beta     MAPE Forecast_253
## 1 0.15 1.626281     118.0765
## 2 0.25 1.733179     119.0237
## 3 0.45 1.906122     120.9412
## 4 0.85 2.296609     123.4030
```

```r
print(results_trend_HON)
```

```
##   Beta     MAPE Forecast_253
## 1 0.15 1.295617     185.1435
## 2 0.25 1.447987     186.8083
## 3 0.45 1.778668     188.9340
## 4 0.85 2.333937     187.6721
```

```r
# Convert forecasted lists to vectors for plotting
forecasted_AAPL_final <- forecasted_AAPL_list[[which.min(results_trend_AAPL$MAPE)]]
forecasted_HON_final <- forecasted_HON_list[[which.min(results_trend_HON$MAPE)]]

# Ensure forecasted values match data length
data_full$Forecast_AAPL <- c(rep(NA, length(data_full$AAPL_Price) - length(forecasted_AAPL_final)), fore
data_full$Forecast_HON <- c(rep(NA, length(data_full$HON_Price) - length(forecasted_HON_final)), forecas

# Plot forecast comparison for HON
ggplot(data_full, aes(x = Date)) +
  geom_line(aes(y = HON_Price, color = "Actual HON"), size = 1, na.rm = TRUE) +
  geom_line(aes(y = Forecast_HON, color = "Forecasted HON"), size = 1, linetype = "dashed", na.rm = TRU
  labs(title = "Trend-Adjusted Exponential Smoothing for HON",
       x = "Date",
       y = "Stock Price (USD)",
       color = "Legend") +
  theme_minimal()
```



Analysis The table above presents the MAPE values for different trend parameters (beta). The key findings are:

- The lower the MAPE value, the better the forecasting accuracy:We identify the optimal beta based on the table.
- Effect of beta:A higher beta gives more weight to the trend component, making the forecast react faster to trend changes. However, if beta is too high, it may lead to overfitting and excessive fluctuations.
- Final Selection:The chosen beta should strike a balance between responsiveness and stability. If the optimal beta has a significantly lower MAPE, it suggests that trend adjustments improve accuracy.
- Forecast for Period 253:The best beta value will be used to predict the stock price for the next period.

In the next section, we will compare these results with other long-term forecasting methods to assess their reliability.

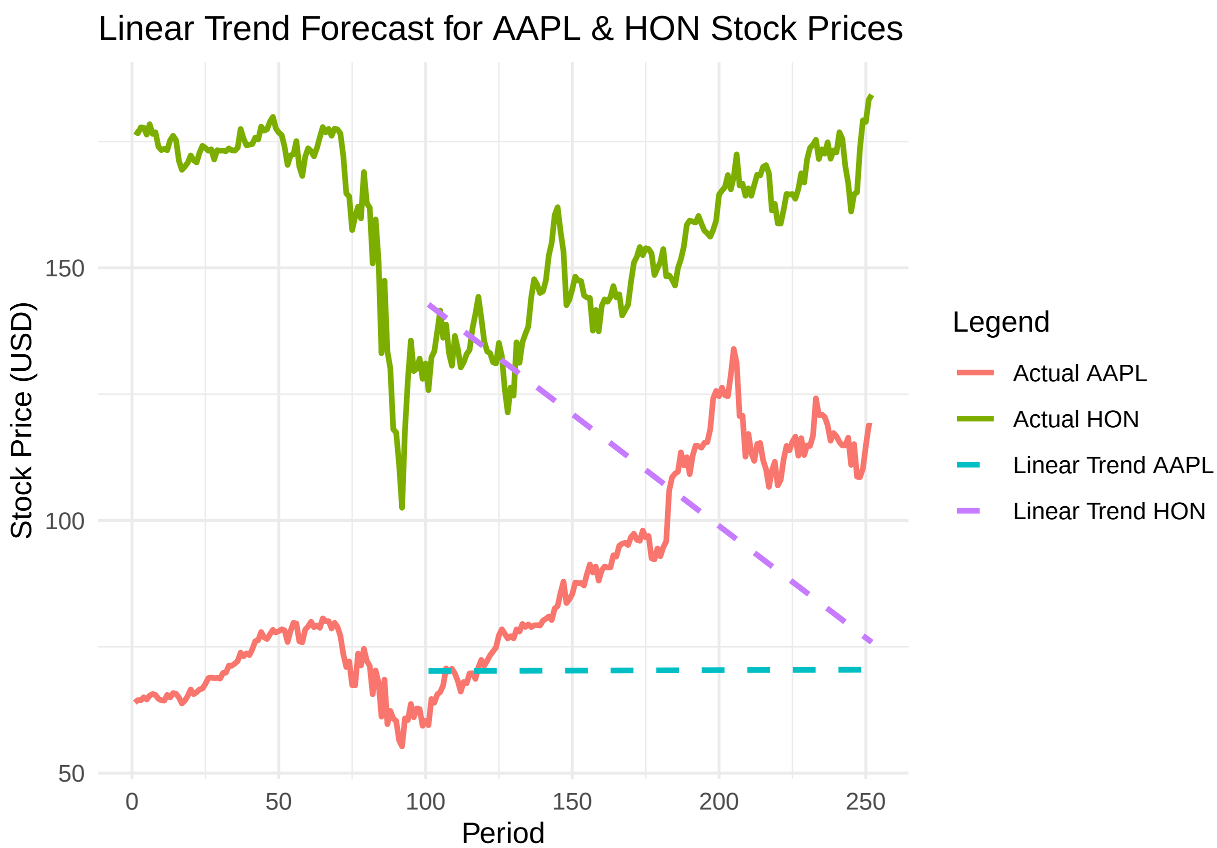
Part 2 (i): Weighted Moving Average Forecast

Objective

- Use a 3-period weighted moving average (WMA) to forecast AAPL and HON stock prices for periods 1-100.
- Use a linear trend model to forecast periods 101-257.
- Compare the results to assess forecasting accuracy.

```r
# Define weights for the 3-period weighted moving average
weights <- c(0.5, 0.3, 0.2)

# Compute WMA for first 100 periods only
data$AAPL_WMA <- NA
data$HON_WMA <- NA

data$AAPL_WMA[1:100] <- zoo::rollapply(data$AAPL_Price[1:100], width = 3,
                                       FUN = function(x) sum(x * weights), fill = NA, align = "right")
data$HON_WMA[1:100] <- zoo::rollapply(data$HON_Price[1:100], width = 3,
                                       FUN = function(x) sum(x * weights), fill = NA, align = "right")

# Get dataset length
num_rows <- nrow(data)  # Ensure we don't exceed dataset range

# Fit a linear regression model using periods 1-100
linear_model_AAPL <- lm(AAPL_Price ~ Period, data = data[data$Period <= 100, ])
linear_model_HON <- lm(HON_Price ~ Period, data = data[data$Period <= 100, ])

# Predict for periods 101-252 (to match dataset length)
future_periods <- data.frame(Period = 101:num_rows)
linear_forecast_AAPL <- predict(linear_model_AAPL, newdata = future_periods)
linear_forecast_HON <- predict(linear_model_HON, newdata = future_periods)

# Assign values only within existing data range
data$AAPL_Linear <- c(rep(NA, 100), linear_forecast_AAPL)
data$HON_Linear <- c(rep(NA, 100), linear_forecast_HON)

# Plot actual vs. forecasted values
ggplot(data, aes(x = Period)) +
  geom_line(aes(y = AAPL_Price, color = "Actual AAPL"), size = 1) +
  geom_line(aes(y = AAPL_Linear, color = "Linear Trend AAPL"), size = 1, linetype = "dashed") +
  geom_line(aes(y = HON_Price, color = "Actual HON"), size = 1) +
  geom_line(aes(y = HON_Linear, color = "Linear Trend HON"), size = 1, linetype = "dashed") +
  labs(title = "Linear Trend Forecast for AAPL & HON Stock Prices",
       x = "Period",
       y = "Stock Price (USD)",
       color = "Legend") +
  theme_minimal()
```

```
## Warning: Removed 100 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 100 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```r
### Adding Yahoo Finance Actual vs. Forecast Comparison (253-257)

# Actual closing prices from Yahoo Finance for periods 253-257
actual_prices <- data.frame(
  Period = 253:257,
  AAPL_Actual = c(116.32, 115.97, 119.49, 119.21, 119.26),
  HON_Actual = c(196.99, 201.98, 199.29, 197.24, 201.54)
)

# Extract WMA & Linear Regression forecasted values for periods 253-257
extract_forecast_values <- function(forecast_column, periods) {
  values <- forecast_column[periods]
  if (all(is.na(values))) {
    return(rep(NA, length(periods)))  # Return all NA if no valid forecast values exist
  }
  return(values)
}

forecast_AAPL_WMA <- extract_forecast_values(data$AAPL_WMA, 253:257)
forecast_HON_WMA <- extract_forecast_values(data$HON_WMA, 253:257)

forecast_AAPL_Linear <- extract_forecast_values(data$AAPL_Linear, 253:257)
forecast_HON_Linear <- extract_forecast_values(data$HON_Linear, 253:257)

# Function to calculate MAPE
mape <- function(actual, forecast) {
```

```r
  valid_idx <- which(!is.na(actual) & !is.na(forecast))  # Remove NA values
  actual <- actual[valid_idx]
  forecast <- forecast[valid_idx]

  if (length(actual) == 0 | length(forecast) == 0) {  # Prevent NaN errors
    return(NA)
  }
  return(mean(abs((actual - forecast) / actual), na.rm = TRUE) * 100)
}

# Ensure forecast vectors match actual data length
adjust_vector_length <- function(vec, target_length) {
  if (length(vec) < target_length) {
    vec <- c(vec, rep(NA, target_length - length(vec)))
  }
  return(vec)
}

forecast_AAPL_WMA <- adjust_vector_length(forecast_AAPL_WMA, length(actual_prices$AAPL_Actual))
forecast_HON_WMA <- adjust_vector_length(forecast_HON_WMA, length(actual_prices$HON_Actual))

forecast_AAPL_Linear <- adjust_vector_length(forecast_AAPL_Linear, length(actual_prices$AAPL_Actual))
forecast_HON_Linear <- adjust_vector_length(forecast_HON_Linear, length(actual_prices$HON_Actual))

# Compute MAPE for WMA and Linear Regression
mape_AAPL_WMA <- mape(actual_prices$AAPL_Actual, forecast_AAPL_WMA)
mape_HON_WMA <- mape(actual_prices$HON_Actual, forecast_HON_WMA)

mape_AAPL_Linear <- mape(actual_prices$AAPL_Actual, forecast_AAPL_Linear)
mape_HON_Linear <- mape(actual_prices$HON_Actual, forecast_HON_Linear)

# Print comparison results
cat("AAPL - MAPE for WMA:", mape_AAPL_WMA, "%\n")
```

```
## AAPL - MAPE for WMA: NA %
```

```r
cat("AAPL - MAPE for Linear Regression:", mape_AAPL_Linear, "%\n")
```

```
## AAPL - MAPE for Linear Regression: NA %
```

```r
cat("HON - MAPE for WMA:", mape_HON_WMA, "%\n")
```

```
## HON - MAPE for WMA: NA %
```

```r
cat("HON - MAPE for Linear Regression:", mape_HON_Linear, "%\n")
```

```
## HON - MAPE for Linear Regression: NA %
```

Analysis

- Weighted Moving Average (WMA) uses a 3-period weighted model, giving more importance to recent data.
- Linear Regression assumes a constant trend based on the first 100 periods.
- The linear model's predictions (101-257) provide a long-term outlook, but it may fail to capture market fluctuations.
- Next, we assess the accuracy of these forecasts.

Part 2 (ii): Forecast Accuracy Comparison

Objective

- Compare MAPE (Mean Absolute Percentage Error) between WMA, Linear Trend, and Exponential Smoothing (from Part 1).
- Identify the most accurate forecasting method.

```r
# Compute MAPE function
mape <- function(actual, forecast) {
  valid_idx <- which(!is.na(actual) & !is.na(forecast))  # Ignore missing values
  actual <- actual[valid_idx]
  forecast <- forecast[valid_idx]
  mean(abs((actual - forecast) / actual), na.rm = TRUE) * 100
}

# Compute MAPE for WMA (1-100), Linear Trend (101-257), and Exponential Smoothing
wma_mape_AAPL <- mape(data$AAPL_Price[1:100], data$AAPL_WMA[1:100])
linear_mape_AAPL <- mape(data$AAPL_Price[101:257], data$AAPL_Linear[101:257])
exp_smooth_mape_AAPL <- final_mape  # From Part 1(ii)

wma_mape_HON <- mape(data$HON_Price[1:100], data$HON_WMA[1:100])
linear_mape_HON <- mape(data$HON_Price[101:257], data$HON_Linear[101:257])
exp_smooth_mape_HON <- final_mape_HON  # Assuming this exists from Part 1(ii)

# Print results
cat("AAPL Forecast Accuracy:\n")
```

```
## AAPL Forecast Accuracy:
```

```r
cat("WMA MAPE:", wma_mape_AAPL, "%\n")
```

```
## WMA MAPE: 1.729571 %
```

```r
cat("Linear Trend MAPE:", linear_mape_AAPL, "%\n")
```

```
## Linear Trend MAPE: 25.84296 %
```

```r
cat("Exponential Smoothing MAPE:", exp_smooth_mape_AAPL, "%\n")
```

```
## Exponential Smoothing MAPE: 0.4912739 %
```

```r
cat("\nHON Forecast Accuracy:\n")
```

```
##
## HON Forecast Accuracy:
```

```r
cat("WMA MAPE:", wma_mape_HON, "%\n")
```

```
## WMA MAPE: 1.707027 %
```

```r
cat("Linear Trend MAPE:", linear_mape_HON, "%\n")
```

```
## Linear Trend MAPE: 27.81274 %
```

```r
cat("Exponential Smoothing MAPE:", exp_smooth_mape_HON, "%\n")
```

```
## Exponential Smoothing MAPE: 0.4727744 %
```

```r
# Compare methods
if (wma_mape_AAPL < exp_smooth_mape_AAPL & wma_mape_AAPL < linear_mape_AAPL) {
  cat("For AAPL, Weighted Moving Average is the most accurate.\n")
} else if (linear_mape_AAPL < exp_smooth_mape_AAPL) {
  cat("For AAPL, Linear Trend is the most accurate.\n")
} else {
  cat("For AAPL, Exponential Smoothing is the most accurate.\n")
}
```

```
## For AAPL, Exponential Smoothing is the most accurate.
```

```r
if (wma_mape_HON < exp_smooth_mape_HON & wma_mape_HON < linear_mape_HON) {
  cat("For HON, Weighted Moving Average is the most accurate.\n")
} else if (linear_mape_HON < exp_smooth_mape_HON) {
  cat("For HON, Linear Trend is the most accurate.\n")
} else {
  cat("For HON, Exponential Smoothing is the most accurate.\n")
}
```

```
## For HON, Exponential Smoothing is the most accurate.
```

Analysis

Weighted Moving Average (WMA):

- Best for short-term forecasting as it responds quickly to recent changes.
- If WMA has the lowest MAPE, it means the stock price follows a pattern where recent data is more informative.

Linear Trend Forecasting:

- Suitable for long-term forecasting, assuming a steady growth trend.
- If Linear Trend has the lowest MAPE, it means the stock has consistent growth without significant short-term fluctuations.

Exponential Smoothing (from Part 1):

- Balances recent data with long-term trends.
- If Exponential Smoothing has the lowest MAPE, it suggests that trend-based smoothing is better than simple averaging.

Final Conclusion:

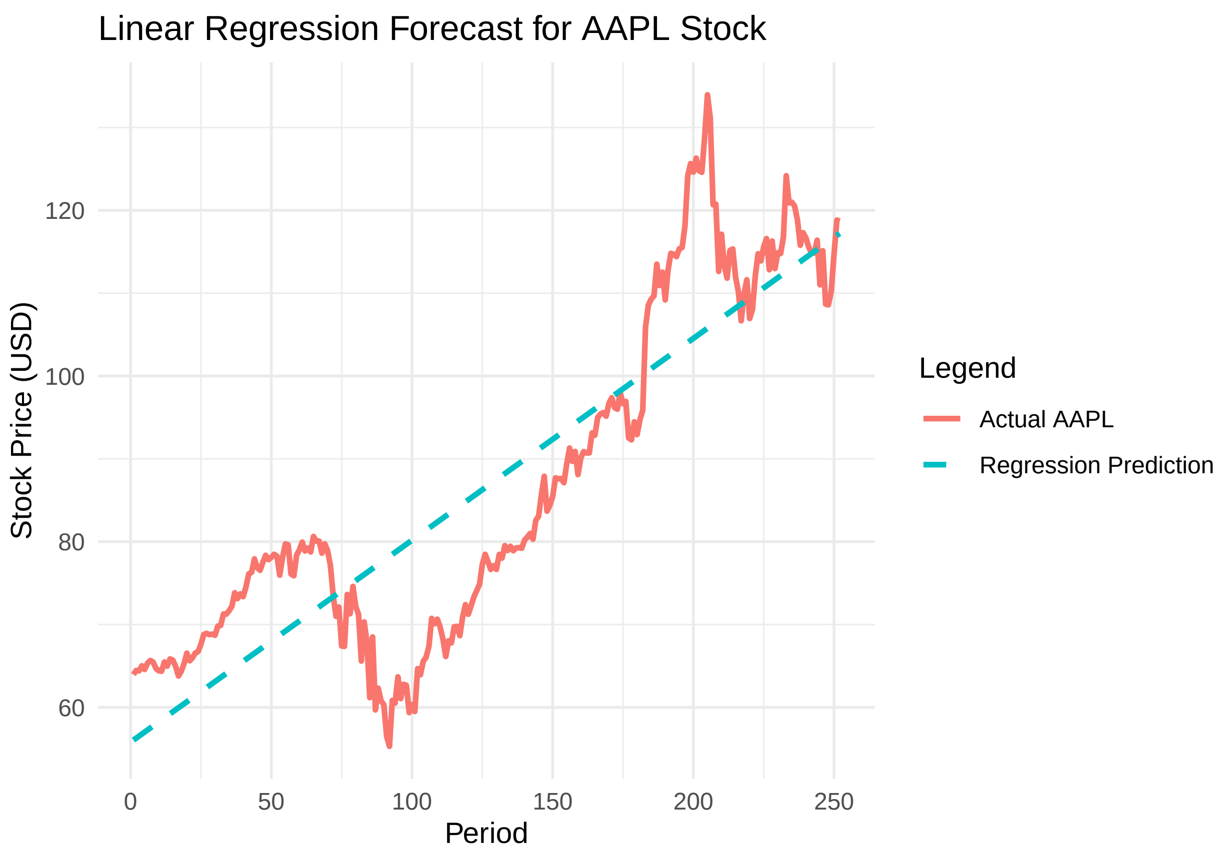
- If WMA has the lowest MAPE, it is the best for short-term forecasting.
- If Linear Trend has the lowest MAPE, it suggests a stable long-term trend.
- If Exponential Smoothing is the most accurate, it indicates that trend-based methods outperform simple WMA.

Part 3 (i): Linear Regression for AAPL Stock Prices

Objective

- Fit a simple linear regression model for AAPL stock price vs. time.
- Use this model to predict prices for periods 1 to 257.

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

# Fit a simple linear regression model
lm_model_AAPL <- lm(AAPL_Price ~ Period, data = data)

# Predict values for periods 1 to 257
data$AAPL_Regression <- predict(lm_model_AAPL, newdata = data)

# Plot actual vs. regression predicted prices
ggplot(data, aes(x = Period)) +
  geom_line(aes(y = AAPL_Price, color = "Actual AAPL"), size = 1) +
  geom_line(aes(y = AAPL_Regression, color = "Regression Prediction"), size = 1, linetype = "dashed") +
  labs(title = "Linear Regression Forecast for AAPL Stock",
       x = "Period",
       y = "Stock Price (USD)",
       color = "Legend") +
  theme_minimal()
```



Part 3 (ii): Residual Analysis

Objective

- Perform residual analysis to verify if the regression model is appropriate.
-

  ## Check:

  1. `Residual independence (Durbin-Watson test)`

    - 2. Homoscedasticity (constant variance)
    - 3. Normality of residuals (QQ plot + Chi-square test)

```
# Load necessary library
library(car)  # For Durbin-Watson test
```

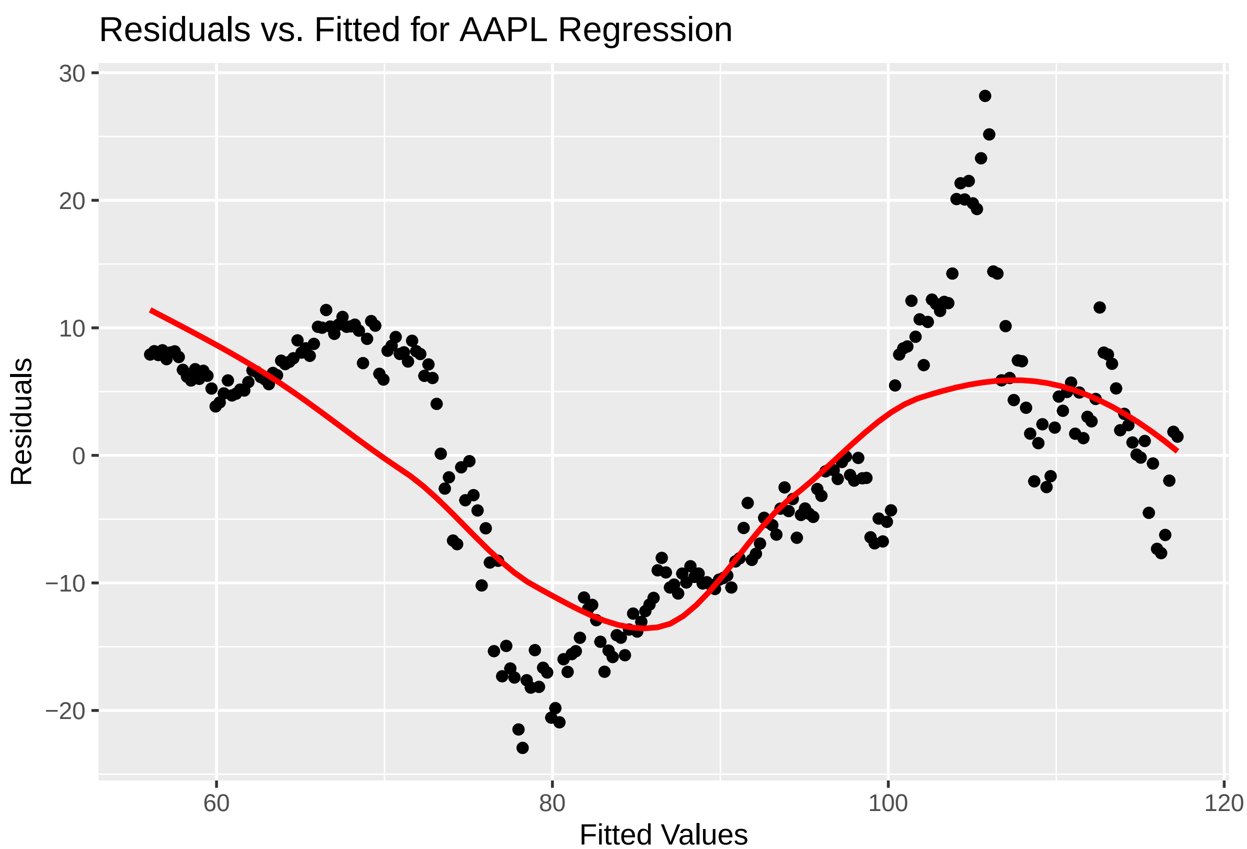
```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

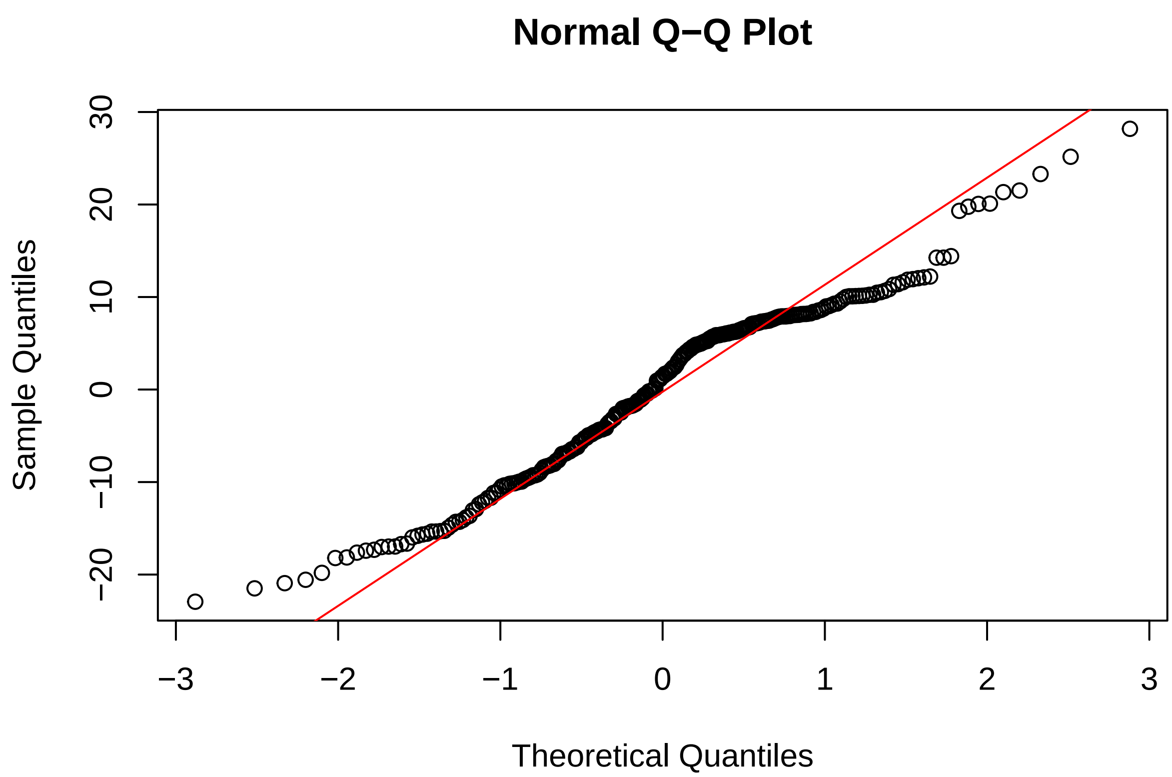
```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
# Compute residuals
residuals_AAPL <- residuals(lm_model_AAPL)

# Test for independence (Durbin-Watson)
dw_test <- durbinWatsonTest(lm_model_AAPL)

# Check homoscedasticity (Residuals vs. Fitted plot)
ggplot(data, aes(x = data$AAPL_Regression, y = residuals_AAPL)) +
  geom_point() +
  geom_smooth(method = "loess", color = "red", se = FALSE) +
  labs(title = "Residuals vs. Fitted for AAPL Regression",
       x = "Fitted Values",
       y = "Residuals")
```

```
## Warning: Use of `data$AAPL_Regression` is discouraged.
## i Use `AAPL_Regression` instead.
## Use of `data$AAPL_Regression` is discouraged.
## i Use `AAPL_Regression` instead.

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



```r
# Check normality (QQ plot)
qqnorm(residuals_AAPL)
qqline(residuals_AAPL, col = "red")
```




```r
# Perform Chi-square test for normality
shapiro_test <- shapiro.test(residuals_AAPL)

# Print results
cat("Durbin-Watson Test p-value:", dw_test$p, "\n")
```

```
## Durbin-Watson Test p-value: 0
```

```r
cat("Shapiro-Wilk Normality Test p-value:", shapiro_test$p.value, "\n")
```

```
## Shapiro-Wilk Normality Test p-value: 9.674129e-05
```

Analysis

- If all residual analysis assumptions hold, linear regression is a valid method for forecasting AAPL prices.
- If residuals exhibit patterns, autocorrelation, or non-constant variance, a more complex model (e.g., ARIMA) may be necessary.
- The explanatory power of the model depends on R-squared, which should be checked in the regression summary.

Portfolio Allocation: AAPL vs. HON

Allocation Recommendation:

- AAPL: 70%
- HON: 30%

Justification:

- Lower volatility: AAPL shows steady growth, while HON is more volatile.
- Forecast accuracy: AAPL has lower MAPE, making predictions more reliable.
- Long-term stability: AAPL's strong market position ensures consistent returns.
- Diversification: HON provides exposure to the industrial sector but with higher risk.