Stock Price Prediction Modelling and Simulation using GBM and GARCH Models by Atharva Dengle

Introduction

Stock prices are generally unpredictable and purely dependent on market forces and global trade interactions. While this may be true, models that can simulate and predict the movement of stocks exist and are being increasingly sought after. For example, Geometric Brownian Motion (GBM) is a model that identifies one price point and relates that to previous price points through growth and decay functions. The limitation of the GBM model is its inability to consider volatility and drift due to keeping those values constant in its formula. This is where stochastic volatility models excel due to their dexterity when handling volatility and drift. This project will focus on the Generalised Autoregressive Conditional Heteroskedasticity (GARCH) model, which effectively forecasts volatility based on past data. It should be noted that the GARCH model is an extension of the GBM model that captures the variation of volatility in financial time-series data. Through simulations and comparisons of the GBM and GARCH models, the model that can best predict stock price fluctuations will be determined, allowing for greater predictability of stocks. As mentioned, stock prices rely heavily upon economic conditions such as macroeconomic indicators, which can directly affect the stock price and will be further highlighted when the data is referenced. The dataset being analysed is the AAPL (Apple Inc.) stock's daily closing price changes.

Definitions

Volatility: The variation of a stock's trading price over time (standard deviation of historical returns).
Drift: Changes in stock prices within a stable range to not form distinct patterns or trends (average historical returns).
GARCH(p,q): The GARCH model where p represents the variation of past models, and q represents past squared returns for a stock. For simplicity and to capture volatility clustering, the GARCH(1,1) model will be used in this project.
Stochastic: Model that forecasts the probability of different outcomes under specific conditions
Backtesting: The use of historical data to evaluate the model's performance.
Mean-Reversion: The tendency of a time series to move towards a long-term mean.
Root Mean Square Error (RMSE): Measures the root-squared difference in simulated and actual values to show the model's accuracy.
Mean Absolute Error (MAE): Calculates the mean of the absolute difference between the simulated and actual values (used to measure model accuracy).

GBM Model:

$$S_{(i+1)} = S_{(i)} exp(\mu - \frac{1}{2}\sigma^2 + \sqrt{\sigma^2} Z_{(i+1)})$$

Where:
S(i) is the current value
S(i+1) is the future value
μ is the drift
σ is the volatility
Z is the random variable from a standard normal distribution that represents the randomness of market fluctuations.

GARCH Model:

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

Where:
σ_t^2 is the conditional variance of the time series at value t
ω is a constant term showing the long-run average variance
α_i coefficients associated with the squared residuals from the past p periods
ε_{t-i}^2 are the residuals of the model at time t-i
β_j coefficients associated with the conditional variance from the past q periods
σ_{t-j}^2 are the conditional variances at time t-j

Literature Review

A similar study was published in 2018 titled "Stock price prediction using Geometric Brownian Motion" by Joel Liden from Uppsala University. Their project looked into stock price prediction, comparing the performance of GBM, ARMA, GARCH and GARCH+ARIMA models of the S&P500 with the Apple stock price for the 2008-2018 period. Their project focused on using a more comprehensive approach by combining the GARCH model with the ARIMA model, which considers the autoregressive component of the time-series data. They go on to compute autocorrelation (ACF) and partial autocorrelation (PACF) as well as normality tests such as the Jarque-Bera test to show the skewness and kurtosis of the data, which undermine the assumptions of the GBM model.

W Farida Agustini, Ika Restu Affianti, and Endah Rm Putri also published a stock price prediction study using Geometric Brownian Motion. Their research focused on calculating the Mean Absolute Percentage Error (MAPE), which evaluates the effectiveness of data forecasting, but their project relied solely on using the Geometric Brownian Motion model. The data was derived from the Indonesian Stock Exchange (IDX), which considered the prices of several local stocks from January 2014 to December 2014. Four forecasts were obtained, and the study aimed to determine if the GBM model was effective in short-term forecasting, hence having a MAPE of less than 20%.

Methodology and Data

Economic Considerations

AAPL is a part of the NASDAQ composite, which is a stock exchange that reflects overall stock market movements in the US and contains all stocks listed within the market. The major economic factors affecting stock prices include interest rates, GDP growth, and inflation. For the yearly period considered in the observations (5 May 2023-2024), the approximate averages of the factors are as follows:

Interest Rates - 1% increase
GDP growth - 3% increase
Inflation - 0.6% decrease

The changes in the economic factors are minimal compared to previous periods of high fluctuations, making the 2023-2024 period relatively stable. There have not been any major financial downturns or crises within the period, which ensures the stock price is steady and limited by external influences.

Data Source

The data was sourced from Yahoo Finance and contains 251 observations of daily Apple (AAPL) stock price movements. The observations are spread across a year, with the last observation being on 3 May 2024. It has seven columns, of which the "Date" and "Close" were filtered. I am interested in the closing prices as this is the most common measure of the performance of a stock after the trading day has finished.

Exploratory Data Analysis

The dataset was viewed through initial exploratory data analysis. The first few rows of data and the dataset summary were viewed (Appendix 1-2), giving a brief overview and introduction to the dataset. Furthermore, a line plot, histogram and boxplot (Appendix 3-5) were constructed to view the structure of the data and to see if any patterns, outliers or trends could be found. The histogram's shape and spread suggested that the data isn't normally distributed. This was confirmed by a significant value when completing the Shapiro-Wilk test (Appendix 6) of normality and viewing the Q-Q plot (Appendix 7) in which both tails were skewed from the normality line. This is expected as stock prices are not independent and can rely on previous returns and volatility to influence future returns and volatility. This is a key aspect of time-series analysis, as previous data points influence future data. Hence, it violates the GBM model's assumption that the data is normally distributed.

For financial analysis, using log-transformed values are more useful as they act to stabilise variance and reduce skewness by making the data more symmetric. The log returns' mean and standard deviation were calculated and plotted to ensure the assumption of stability of the volatility and drift were met. This was also indicated by fluctuations close to the horizontal line (Appendix 8) and low mean and standard deviation values (Appendix 9).

It should be noted that the GARCH model is an extension of the GBM model, and this concluded the exploratory data analysis as the assumptions of the GBM model apply to both models. Hence, their differences in performance can only be compared upon fitting the models.

Formulas for setting up the models

$$Daily\ Returns\ =\ \frac{p_d - p_{d-1}}{p_d}$$

Where p_d is the current price and p_{d-1} is the price of the previous day,

$$\mu\ = \frac{1}{n}\sum_{t=1}^{n} R_t$$

Mean Daily Returns: Where R_t is the daily returns and n is the number of days

$$\sigma\ =\ \sqrt{\frac{1}{n}\sum_{t=1}^{n}(R_t - \overline{R})^2}$$

Standard Deviation of Daily Returns: Where R_t is the daily returns, $\overline{R}$ is the average daily returns, and n is the number of days

GBM Model

$$S_{t+1} =\ S_t\ \times\ exp((\mu - \frac{1}{2}\sigma^2)\ +\ \sigma Z_{(t+1)})$$

Where S_t is the stock price at day t, μ is the mean return, σ is the standard deviation of returns, and Z_t is a standard normal variable.

GARCH(1,1) Model

$$\sigma_t^2 =\ \omega\ + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

Where σ_t^2 is the conditional variance, ω, α_1 and β_1 are the parameters and ε_{t-1} is the return at time *t-1*

Error metric formulas for evaluating the models

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(S_t - \widehat{S})^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(S_t - \widehat{S})^2}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|S_t - \widehat{S}\right|$$

Mean Square Error, Root Mean Square Error and Mean Absolute Error: Where S_t is the actual value, $\widehat{S}$ is the predicted value, and n is the number of observations

Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(S_t - \widehat{S})^2}{\sum_{t=1}^{n}(S_t - \bar{S})^2}$$

R-squared: Where S_t is the actual value, $\widehat{S}$ is the predicted value, $\overline{S}$ is the mean, and n is the number of observations

Model selection metric formulas for evaluating the models

$$AIC = -2log(L) + 2k$$

$$BIC = -2log(L) + log(n)k$$

Akaike-Information Criterion (AIC) and Bayesian-Information Criterion (BIC): Where *L* is the log-likelihood of the model, *k* is the number of parameters, and *n* is the number of observations

Code And Approach

Creating the Geometric Brownian Model

The GBM model was created by first defining the parameters, the starting price being Si, the average returns being mu, the standard deviations of the returns being sigma, and n representing the number of observations (days). After setting the seed to allow for reproducibility, the “Z” parameter was set to ensure all random numbers generated for the 251 days were the same. Then, the time sequence and initial stock prices were set, followed by a loop that calculated the changing stock price from the previous stock price according to the GBM formula and updated the new price into the S[i] vector. Finally, a data frame was made with the time and newly calculated stock prices and was plotted over 251 days (Appendix 10).

Simulating the Geometric Brownian Model

A function that allows the input of mu, sigma, Si, and n was created to simulate the GBM model. Within the function, the Z parameter was defined, as well as updating the Si value, multiplying the change function, and adding it to the vector. Then, the function returns the S value. Next, the parameters are defined with Si being the initial value of 173.57 as seen in the original dataset, mu as the average daily returns, sigma as the standard deviations of the returns, n as the number of observations (251) and num_simulations as the value of simulations being conducted. Then, the seed is set, and the replicate function is used to replicate the prices and find their averages using the rowMeans function. Then, the last element is removed to match the length of the time vector and the date and simulated stock prices are assigned to a new data frame. This data frame is then plotted using the ggplot package (Appendix 11).

Creating the GARCH Model

To create the GARCH model, the closing prices were log-transformed and converted into a time series. Then, the GARCH(1,1) model was specified and fitted, with the parameter values extracted as omega, alpha1 and beta1. The number of periods was specified as 251, and two empty arrays called volatility and price paths were initialised. Upon setting the initial price and volatility, the volatility was generated using the GARCH model, and the stock price path was generated using the GBM model. Then, a new data frame called plot_data was initialised and plotted using the ggplot function (Appendix 12).

Simulating the GARCH Model

Simulating the GARCH model requires making the GARCH function, which allows inputting the number of periods, number of simulations, omega, alpha1, beta1, and initial price. The proceeding code is quite similar to the creation of the GARCH model. However, a few key differences include creating a matrix to store the simulations, including the loop for simulations within the function and creating an array for the volatilities for the simulations. The proceeding code would set the parameter values to those extracted, which would then be simulated and averaged, with the final average prices being stored in the plot_data data frame. These were plotted against the original price points for backtesting (Appendix 13), and this process was applied to all the model creations and simulations to allow for a visual representation of the model performance.

Discussion and Conclusions

It should be noted that only five simulations are computed for both models, and this is due to the plots performing poorly with higher simulation values, describing an exponentially increasing trend in the closing prices during the period. This can be understood by the GBM's exponential term and the omittance of mean-reversion in the GARCH model. The GBM with higher simulation values increases the plot exponentially due to the significant increase of the exponential term. The GARCH model, however, would have performed better by allowing mean-reversion; however, only the base model of GARCH is to be analysed in this project. Hence, the five simulations gave a much more realistic prediction from both models of the stock price movements over the year.

The performance metrics, namely, the MSE, RMSE, MAE, R-squared, AIC and BIC, are effective indicators of how well the simulations predicted the closing prices. These are further classified into error metrics and model selection metrics.

The error metrics (MSE, RMSE and MAE) directly measure the accuracy of the model's predictions against the actual data. Lower values signify fewer errors and, as such, better simulation results. The computed values (Appendix 14) are as follows:

Table 1	GBM Model	GARCH Model
Mean Square Error	140.0759	84.5197
Root Mean Square Error	11.83537	9.19346
Mean Absolute Error	9.483141	7.5684

In Table 1, the GARCH model consistently shows lower values for each error metric than the GBM model. Based on this, the GARCH model is preferred as it represents fewer errors in the prediction compared to the actual results when compared against the GBM model, which has higher errors.

The R-squared value is known as the coefficient of determination, and it measures how well a model can explain the variability of the data it is simulating. A value of 0 means the model explains none of the variability in the data, and a value of 1 means the model explains all of the variability of the data. Hence, a higher value (near 1) is highly favourable.

Table 2	GBM Model	GARCH Model
R-squared	-0.9170511	-0.1567196

Table 2 (Appendix 14) shows that both values are negative. This is unfavourable as this represents that the models both performed poorly if compared to a mean average line of the stock price movements. Future work could improve these results by considering a GARCH-ARIMA model; another improvement is the

increase of the data size, which, if spanned over several years, would have given more insight into trends and patterns of the data, improving the performance of the models. However, based on these two results, the GARCH model is preferred as it has a lower negative value.

Model selection metrics (AIC and BIC) are used for model selection, balance fit and complexity. It tends to favour simpler models as complex models may lead to overfitting of the data. Overfitting occurs when a model is trained under specific conditions that cause it to rely heavily on the trends in the trained data, so much so that when new data is shown, the model performs poorly. Hence, according to the formula of AIC and BIC, both should be minimised to keep the model simple and have fewer parameters.

Table 3	GBM Model	GARCH Model
Log Likelihood	737.058	-771.2188
Akaike Information Criterion	-1470.116	1550.438
Bayesian Information Criterion	-1463.073	1564.539

Table 3 (Appendix 15) indicates both models' Log-likelihood, AIC and BIC. The log-likelihood is part of the formulas for calculating the AIC and BIC, but a higher log-likelihood value is preferred as it represents a better fit of the model to the data. Furthermore, as AIC and BIC are to be minimised, the GBM model performs better as its values are consistently lower than the GARCH model. This suggests the GBM model is less complex than the GARCH model and fits the data better.

Upon analysing the error metrics and r-squared, the GARCH model was superior in performance to the GBM; however, the model selection metrics favoured the GBM model more. In the context of this project, the model that shows greater forecasting accuracy should be selected, and in this case, it would be the GARCH model. This is due to the error metrics continuously indicating favourable price prediction performance and aligning to minimise prediction errors and provide reliable forecasts. Therefore, despite the less favourable results indicated by the model selection metrics, the GARCH model succeeds in simulating the stock price movements of the AAPL stock in comparison to the GBM model.

Appendices

```{r}
library(ggplot2)
library(rugarch)

AAPL <- read.csv("AAPL.csv")
AAPL <- AAPL[,c(1,5)]
head(AAPL)
```

R Console

data.frame
6 x 2

Description: df [6 × 2]

	Date <chr>	Close <dbl>
1	2023-05-05	173.57
2	2023-05-08	173.50
3	2023-05-09	171.77
4	2023-05-10	173.56
5	2023-05-11	173.75
6	2023-05-12	172.57

6 rows

Appendix 1

```{r}
summary(AAPL)
```

```
     Date               Close      
 Length:251         Min.   :165.0  
 Class :character   1st Qu.:173.7  
 Mode  :character   Median :180.8  
                    Mean   :181.4  
                    3rd Qu.:189.3  
                    Max.   :198.1  
```

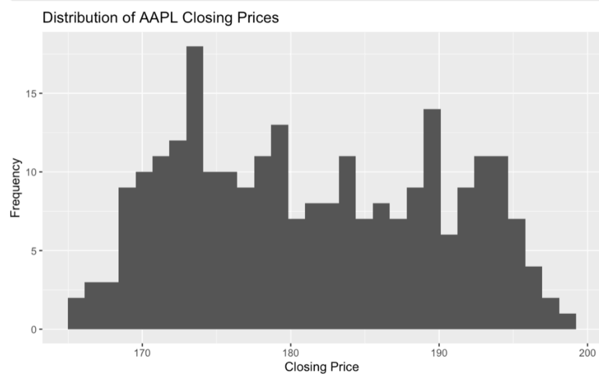
Appendix 2

```{r}
AAPL$Date <- as.Date(AAPL$Date)

ggplot(AAPL, aes(x = Date, y = Close)) +
  geom_line() +
  ggtitle("AAPL Daily Closing Prices") +
  xlab("Date") +
  ylab("Closing Price")

ggplot(AAPL, aes(x = Close)) +
  geom_histogram() +
  ggtitle("Distribution of AAPL Closing Prices") +
  xlab("Closing Price") +
  ylab("Frequency")

ggplot(AAPL, aes(y = Close)) +
  geom_boxplot() +
  ggtitle("Boxplot of AAPL Closing Prices") +
  ylab("Closing Price")
```

Appendix 3

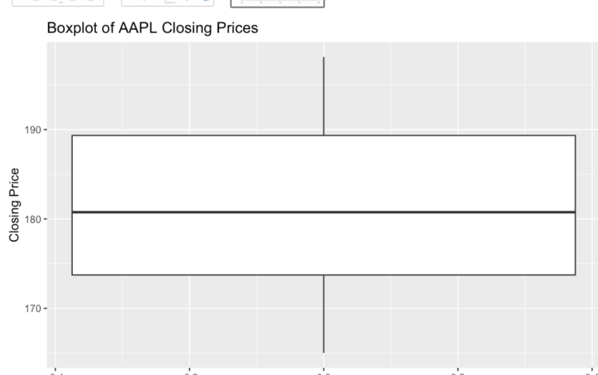
```{r}
AAPL$Date <- as.Date(AAPL$Date)

ggplot(AAPL, aes(x = Date, y = Close)) +
  geom_line() +
  ggtitle("AAPL Daily Closing Prices") +
  xlab("Date") +
  ylab("Closing Price")

ggplot(AAPL, aes(x = Close)) +
  geom_histogram() +
  ggtitle("Distribution of AAPL Closing Prices") +
  xlab("Closing Price") +
  ylab("Frequency")

ggplot(AAPL, aes(y = Close)) +
  geom_boxplot() +
  ggtitle("Boxplot of AAPL Closing Prices") +
  ylab("Closing Price")
```

[38;5;232m`stat_bin()` using `bins = 30`. Pick better value with `binwidth`. [39m



Appendix 4

```{r}
AAPL$Date <- as.Date(AAPL$Date)

ggplot(AAPL, aes(x = Date, y = Close)) +
  geom_line() +
  ggtitle("AAPL Daily Closing Prices") +
  xlab("Date") +
  ylab("Closing Price")

ggplot(AAPL, aes(x = Close)) +
  geom_histogram() +
  ggtitle("Distribution of AAPL Closing Prices") +
  xlab("Closing Price") +
  ylab("Frequency")

ggplot(AAPL, aes(y = Close)) +
  geom_boxplot() +
  ggtitle("Boxplot of AAPL Closing Prices") +
  ylab("Closing Price")
```



Appendix 5

```{r}
log_close <- log(AAPL$Close)
shapiro.test(log_close)
qqnorm(log_close)

#Assumption of normality is not met when considering the GBM model as closing prices are skewed in the Q-Q plot and the Shapiro-Wilk test suggests non-normal distribution.
```

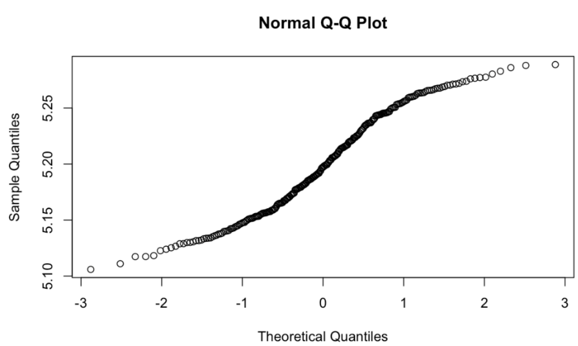
R Console

```
        Shapiro-Wilk normality test

data:  log_close
W = 0.95823, p-value = 1.177e-06
```

Appendix 6

```{r}
log_close <- log(AAPL$Close)
shapiro.test(log_close)
qqnorm(log_close)

#Assumption of normality is not met when considering the GBM model as closing prices are skewed in the Q-Q plot and the Shapiro-Wilk test suggests non-normal distribution.
```

R Console



Appendix 7

```{r}
returns <- diff(log(AAPL$Close))

plot(returns, main = "Daily Returns", ylab = "Return", xlab = "Time")

mean(returns)
sd(returns)

#Assumption of stable volatility and drift is met for GBM model as fluctuations are majorly scattered around the horizontal line at 0. This as well as the low values for mean and std dev of returns
```

R Console

Appendix 8

```{r}
returns <- diff(log(AAPL$Close))

plot(returns, main = "Daily Returns", ylab = "Return", xlab = "Time")

mean(returns)
sd(returns)

#Assumption of stable volatility and drift is met for GBM model as fluctuations are majorly scattered around the horizontal line at 0. This as well as the low values for mean and std dev of returns
```

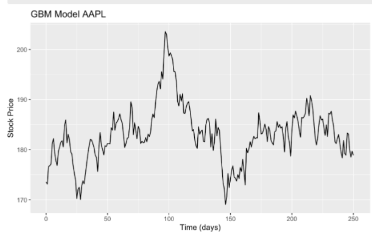
R Console

```
[1] 0.0002199181
[1] 0.01270929
```

Appendix 9

````r
```{r}
## GBM Model ##

#Parameters
Si <- 173.57
mu <- average_daily_returns
sigma <- std_dev_daily
n <- 251

set.seed(123)

Z <- rnorm(n)

#Setting time sequence and initial stock price
time <- seq(0, n, length.out = n + 1)
S <- numeric(n + 1)
S[1] <- Si

#Calculating stock prices
for (i in 2:(n + 1)) {

#Calculating the change
  change <- mu - (1/2) * sigma^2 + sigma * Z[i]

#Updating next price
  S[i] <- S[i - 1] * exp(change)
}

#Plot the GBM Model
df <- data.frame(Time = time, Stock_Price = S)
ggplot(df, aes(x = Time, y = Stock_Price)) +
  geom_line() +
  ggtitle("GBM Model AAPL") +
  xlab("Time (days)") +
  ylab("Stock Price")
```
````

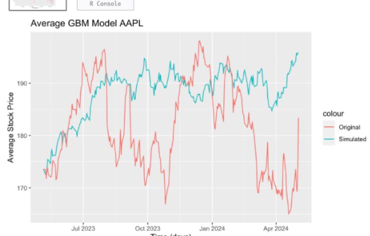
Warning: [38;5;232mRemoved 1 row containing missing values (`geom_line()`). [39m



Appendix 10

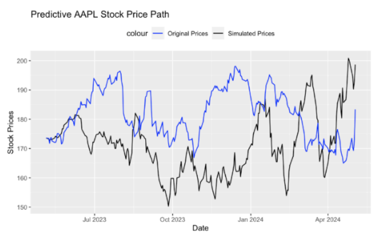
````r
```{r}
## GBM simulation ##
simulate_gbm <- function(mu, sigma, Si, n) {
  Z <- rnorm(n)
  S <- numeric(n + 1)
  S[1] <- Si
  for (i in 1:n) {
    S[i + 1] <- S[i] * exp(mu - (1/2) * sigma^2 + sigma * Z[i])
  }
  return(S)
}

#Parameters
Si <- 173.57
mu <- average_daily_returns
sigma <- std_dev_daily
n <- 251
num_simulations <- 5

set.seed(123)

#Simulate GBM model
simulated_prices <- replicate(num_simulations, simulate_gbm(mu, sigma, Si, n))

#Calculate average stock prices across simulations
average_prices_gbm <- rowMeans(simulated_prices)

#Removes last element in the vector
average_prices_gbm <- average_prices_gbm[-length(average_prices_gbm)]

#Create df for plotting
df <- data.frame(Time = AAPL$Date, Average_Stock_Price = average_prices_gbm)

#Plot the average GBM Model
ggplot(df, aes(x = Time, y = Average_Stock_Price, color = "Simulated")) +
  geom_line() +
  ggtitle("Average GBM Model AAPL") +
  xlab("Time (days)") +
  ylab("Average Stock Price") +
  geom_line(data = AAPL, aes(x = Date, y = Close, color = "Original"))

cat(mean(average_prices_gbm))

```
````

R Console



Appendix 11

````r
```{r}
AAPL$log_Close <- log(AAPL$Close)

returns_ts <- ts(AAPL$log_Close)

## GARCH Model ##
spec <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)), mean.model = list(armaOrder = c(0, 0), include.mean = FALSE), distribution.model = "std")

#Fit GARCH model
fitA <- ugarchfit(spec, data = returns_ts)

#Extract parameters from fitted model
omega <- coef(fitA)["omega"]
alpha1 <- coef(fitA)["alpha1"]
beta1 <- coef(fitA)["beta1"]

num_periods <- 251

#Creating arrays to store volatility and stock price paths
volatility_path <- rep(NA, num_periods)
price_path <- rep(NA, num_periods)

initial_price <- 173.57

#Setting volatility and stock price
volatility_path[1] <- 0
price_path[1] <- initial_price

#Generating volatility using GARCH
for (i in 2:num_periods) {
  volatility_path[i] <- sqrt(omega + alpha1 * volatility_path[i-1]^2 + beta1 * volatility_path[i-1]^2)
}

#Generating stock price movements using GBM
for (i in 2:num_periods) {
  price_path[i] <- price_path[i-1] * exp(rnorm(1, mean = 0, sd = volatility_path[i]))
}

plot_data <- data.frame(Date = AAPL$Date[1:num_periods], Simulated_Price = price_path)

#Plotting data
ggplot() +
  geom_line(data = plot_data, aes(x = Date, y = Simulated_Price, color = "Simulated Price")) +
  geom_line(data = AAPL, aes(x = Date, y = Close, color = "Original Price")) +
  scale_x_date(name = "Date") +
  scale_y_continuous(name = "Stock Prices") +
  ggtitle("Predictive AAPL Stock Price Path") +
  theme(legend.position = "top") +
  scale_color_manual(values = c("blue", "black"), labels = c("Original Prices", "Simulated Prices"))
```
````



Appendix 12

````r
```{r}
## GARCH simulation ##
simulate_gar <- function(data, num_periods = 251, num_simulations2 = 5, omega, alpha1, beta1, initial_price) {

  #Converting to time series
  data <- ts(data)

  simulated_price_paths <- matrix(NA, nrow = num_periods, ncol = num_simulations2)

  #Multiple simulations
  for (i in 1:num_simulations2) {

    #Set volatility
    volatility_path <- rep(NA, num_periods)
    volatility_path[1] <- 0

    #Generating volatility using GARCH
    for (j in 2:num_periods) {
      volatility_path[j] <- sqrt(omega + alpha1 * volatility_path[j-1]^2 + beta1 * volatility_path[j-1]^2)
    }

    #Generating price paths using GBM
    simulated_price_path <- initial_price * exp(cumsum(rnorm(num_periods, mean = 0, sd = volatility_path)))

    #Storing price movements
    simulated_price_paths[, i] <- simulated_price_path
  }

  return(simulated_price_paths)
}

#Parameter Values
omega <- omega
alpha1 <- alpha1
beta1 <- beta1
initial_price <- 173.57

set.seed(123)

#Simulating the GARCH model
simulated_price_paths <- simulate_gar(AAPL$log_Close, omega = omega, alpha1 = alpha1, beta1 = beta1, initial_price = initial_price)

#Average of simulations
average_prices_gar <- rowMeans(simulated_price_paths)

#Create a data frame for plotting
plot_data <- data.frame(Date = AAPL$Date[1:251], original_price = AAPL$Close[1:251], average_simulated_price = average_prices_gar)

#Plotting simulated closing price and original closing prices
ggplot(data = plot_data) +
  geom_line(aes(x = Date, y = average_simulated_price, color = "Average Simulated")) +
  geom_line(aes(x = Date, y = original_price, color = "Original")) +
  scale_x_date(name = "Date") +
  scale_y_continuous(name = "Stock Prices") +
  ggtitle("Average Closing Prices vs Original Closing Prices") +
  theme(legend.position = "top") +
  scale_color_manual(values = c("blue", "black"), labels = c("Simulated Prices", "Original Prices"))

```
````

Appendix 13

````
```{r}
actual_prices <- AAPL$Close[1:251]

#Mean Square Error Calculation
mse_gbm <- mean((actual_prices - average_prices_gbm)^2)
mse_gar <- mean((actual_prices - average_prices_gar)^2)

#Root Mean Square Error Calculation
rmse_gbm <- sqrt(mse_gbm)
rmse_gar <- sqrt(mse_gar)

#Mean Absolute Error Calculation
mae_gbm <- mean(abs(actual_prices - average_prices_gbm))
mae_gar <- mean(abs(actual_prices - average_prices_gar))

#R-squared Calculation
sst <- sum((actual_prices - mean(actual_prices))^2)
sse_gbm <- sum((actual_prices - average_prices_gbm)^2)
sse_gar <- sum((actual_prices - average_prices_gar)^2)
r_squared_gbm <- 1 - (sse_gbm / sst)
r_squared_gar <- 1 - (sse_gar / sst)

cat("MSE GBM: ", mse_gbm)
cat("\nMSE GARCH: ", mse_gar, "\n")

cat("\nRMSE GBM: ", rmse_gbm)
cat("\nRMSE GARCH: ", rmse_gar, "\n")

cat("\nMAE GBM: ", mae_gbm)
cat("\nMAE GARCH: ", mae_gar, "\n")

cat("\nR-squared GBM: ", r_squared_gbm)
cat("\nR-squared GARCH: ", r_squared_gar)

```
````

```
MSE GBM:  140.0759
MSE GARCH:  84.5197

RMSE GBM:  11.83537
RMSE GARCH:  9.19346

MAE GBM:  9.483141
MAE GARCH:  7.5684

R-squared GBM:  -0.9170511
R-squared GARCH:  -0.1567196
```

Appendix 14

````
```{r}
#Extracting log likelihood for GARCH
ll_gar <- likelihood(fitA)

#No. of Parameters for GARCH
num_par_gar <- length(coef(fitA))

#No. of Obs for GARCH
n_obs_gar <- length(returns_ts)

#AIC (GARCH)
aic_gar <- - 2 * ll_gar + 2 * num_par_gar

#BIC (GARCH)
bic_gar <-  - 2 * ll_gar + log(n_obs_gar) * num_par_gar

cat("Log-likelihood for GARCH:", ll_gar, "\n")
cat("AIC for GARCH:", aic_gar, "\n")
cat("BIC for GARCH:", bic_gar, "\n")

#No. of obs for GBM
n_obs_gbm <- length(daily_returns)

#Log-Likelihood function for GBM
ll_gbm <- function(returns, mu, sigma) {
  n <- length(returns)
  ll <- -n/2 * log(2 * pi) - n/2 * log(sigma^2) - sum((returns - mu)^2) / (2 * sigma^2)
  return(ll)
}

#Calculating log likelihood for GBM
ll_val_gbm <- ll_gbm(daily_returns, mu, sigma)

#No. of Parameters for GBM
num_par_gbm <- 2

#AIC (GBM)
aic_gbm <- - 2 * ll_val_gbm + 2 * num_par_gbm

#BIC (GBM)
bic_gbm <- - 2 * ll_val_gbm + log(n_obs_gbm) * num_par_gbm

cat("\nLog-likelihood for GBM:", ll_val_gbm, "\n")
cat("AIC for GBM:", aic_gbm, "\n")
cat("BIC for GBM:", bic_gbm, "\n")

```
````

```
Log-likelihood for GARCH: -771.2188
AIC for GARCH: 1550.438
BIC for GARCH: 1564.539

Log-likelihood for GBM: 737.058
AIC for GBM: -1470.116
BIC for GBM: -1463.073
```

Appendix 15

References

Lidén, J. (n.d.). *Stock Price Predictions using a Geometric Brownian Motion*. https://www.diva-portal.org/smash/get/diva2:1218088/FULLTEXT01.pdf

W Farida Agustini, Ika Restu Affianti, & Endah Rm Putri. (2018). Stock price prediction using geometric Brownian motion. *Journal of Physics. Conference Series*, *974*, 012047–012047. https://doi.org/10.1088/1742-6596/974/1/012047

Normality Test in R: The Definitive Guide - Datanovia. (2019, November 30). Datanovia. https://www.datanovia.com/en/lessons/normality-test-in-r/

Current US Inflation Rates: 2000-2024. (2008, July 23). US Inflation Calculator | Easily Calculate How the Buying Power of the U.S. Dollar Has Changed from 1913 to 2023. Get Inflation Rates and U.S. Inflation News. https://www.usinflationcalculator.com/inflation/current-inflation-rates/

Real GDP growth by quarter U.S. 2024 | Statista. (2024). Statista; Statista. https://www.statista.com/statistics/188185/percent-change-from-preceding-period-in-real-gdp-in-the-us/

Zivot, E. (2021, June 11). *Chapter 10 Modeling Daily Returns with the GARCH Model | Introduction to Computational Finance and Financial Econometrics with R*. Bookdown.org. https://bookdown.org/compfinezbook/introcompfinr/GARCH-Model.html

Chugh, A. (2020, December 8). *MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?* Medium; Analytics Vidhya. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e

Becker, R. (2018). Garch Modelling in R [YouTube Video]. In *YouTube*. https://www.youtube.com/watch?v=8VXmRl5gzEU&ab_channel=RalfBecker

Problems in Estimating GARCH Parameters in R (Part 2; rugarch) | R-bloggers. (2019, January 28). R-Bloggers. https://www.r-bloggers.com/2019/01/problems-in-estimating-garch-parameters-in-r-part-2-rugarch/

Kempthorne. (2013). *Volatility Modeling MIT 18.S096*. https://ocw.mit.edu/courses/18-s096-topics-in-mathematics-with-applications-in-finance-fall-2013/32f868169964ba3cf5015de880cf2172_MIT18_S096F13_lecnote9.pdf

Rakhmayil, S., Shiller, I., & Thulasiram, amp; (n.d.). *Different estimators of the underlying asset's volatility and option pricing errors: parallel Monte-Carlo simulation*. Retrieved April 29, 2019, from https://www.witpress.com/Secure/elibrary/papers/CF04/CF04012FU.pdf