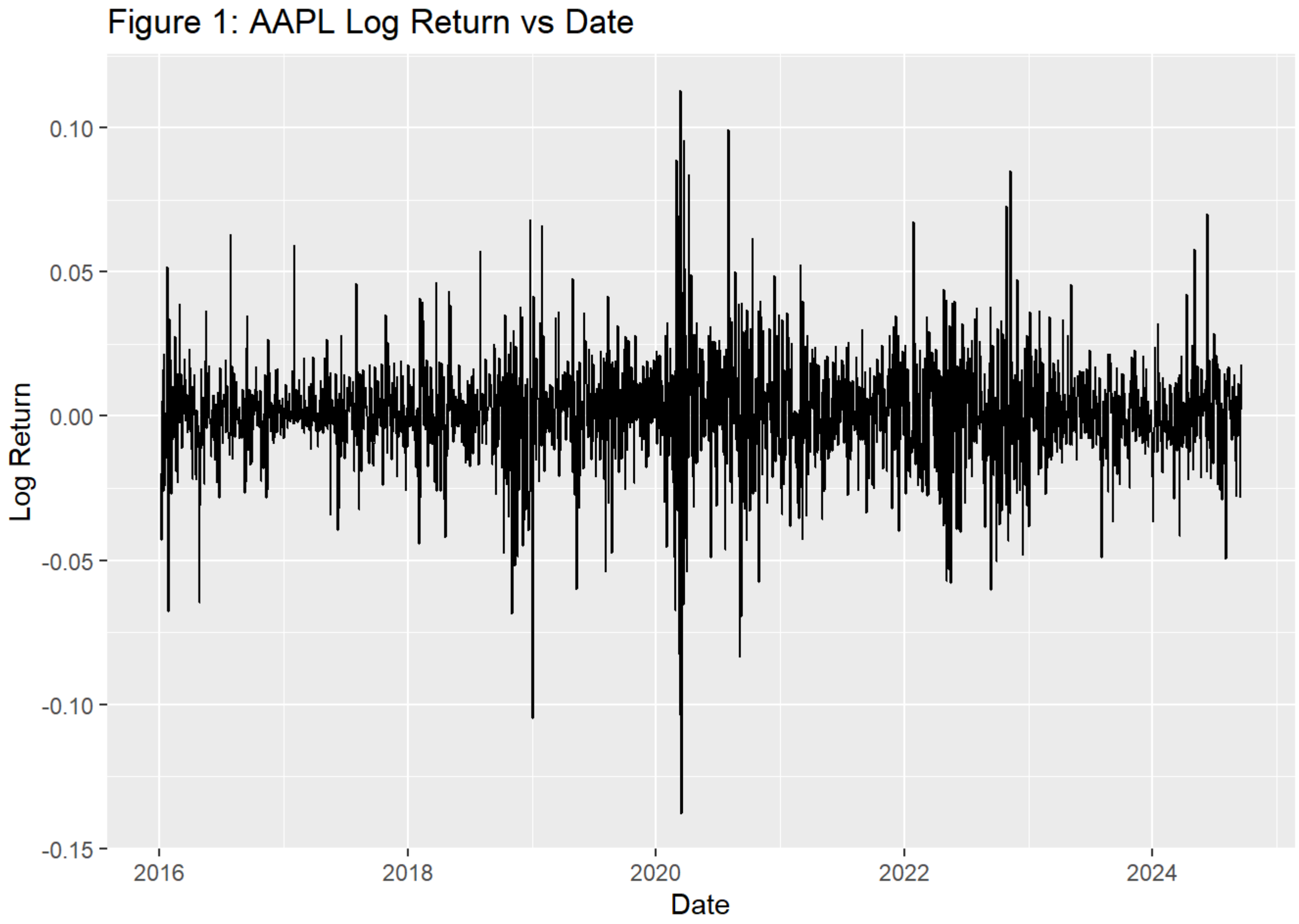
Question 5

```
# Load the data
aapl_data = read.csv("aapl_log_returns.csv")

# Load the required packages
suppressWarnings(suppressMessages(library(rugarch)))
suppressWarnings(suppressMessages(library(forecast)))
suppressWarnings(suppressMessages(library(ggplot2)))
suppressWarnings(suppressMessages(library(nortsTest)))
suppressWarnings(suppressMessages(library(tseries)))
```

Explore the Data

Let’s first visualize the data to observe any useful traits

```
# reformat the Date column to Date Format
aapl_data$Date <- as.Date(aapl_data$Date, format = "%Y/%m/%d")

# Plot the log return
ggplot(aapl_data, aes(x = Date, y = Log.Return)) + geom_line() + ggtitle("Figure 1: AAPL Log
Return vs Date") + xlab("Date") + ylab("Log Return")
```



Observations of the Data

We see that the log return of AAPL is quite volatile, with some days having a high return and others having a low return. Furthermore, the log return seems to fluctuate randomly around zero, this could suggest little autocorrelation. We can confirm this using an ACF plot. But we should check for stationarity first.

Check for Stationarity

```
# Check for stationarity using ADF Test
adf.test(aapl_data$Log.Return, alternative = "stationary")
```

```
## Warning in adf.test(aapl_data$Log.Return, alternative = "stationary"): p-value
## smaller than printed p-value
```

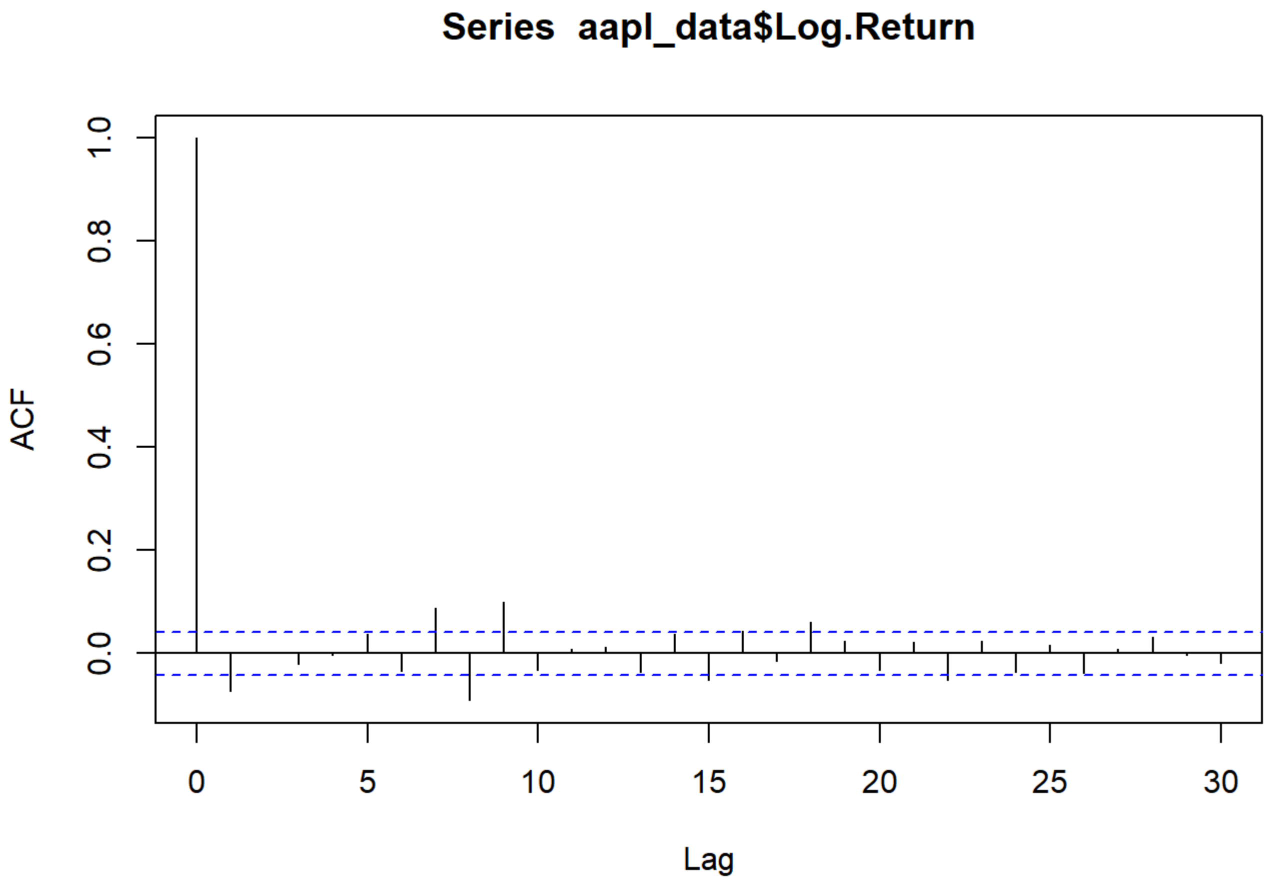
```
##
##  Augmented Dickey-Fuller Test
##
## data:  aapl_data$Log.Return
## Dickey-Fuller = -12.573, Lag order = 12, p-value = 0.01
## alternative hypothesis: stationary
```

Observations of the ADF Test

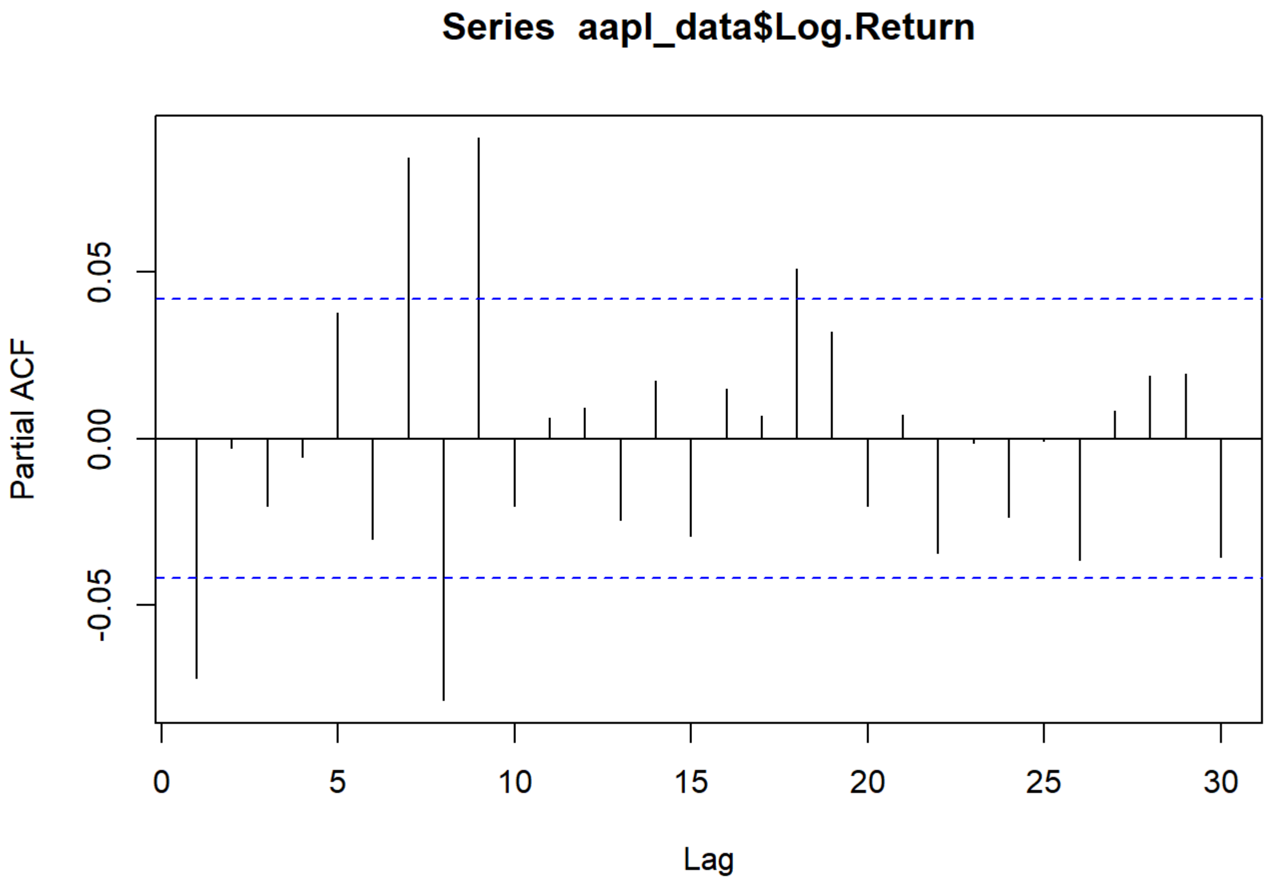
The p-value of the ADF test is less than 0.05, suggesting that we can reject the null hypothesis that the data is non-stationary. This means that the data is stationary, and we can proceed with analysis of ACF and PACF plots.

ACF and PACF Plots

```
# Define lag max param
n = length(aapl_data$Log.Return)
lag_max = round(10 * log10(n/2)) # Rule of thumb for lag.max, N/m where N: num of datapoints,
m: dimension of data

# ACF plot against lag
acf(aapl_data$Log.Return, lag.max = lag_max)
```



```
# PACF plot against lag
pacf(aapl_data$Log.Return, lag.max = lag_max)
```



Observations of the ACF and PACF Plots

The ACF plot shows a sharp drop-off after the first lag, while the PACF plot shows exponential decay. Our observations suggest that the ACF plot exhibits MA signatures and this indicates that an MA model might be suitable for the data.

Fit an ARIMA Model

```
# Fit an ARIMA model
fit <- auto.arima(aapl_data$Log.Return)
fit
```

```
## Series: aapl_data$Log.Return
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##           ar1   mean
##       -0.0721  1e-03
## s.e.   0.0213  4e-04
##
## sigma^2 = 0.0003304:  log likelihood = 5672.62
## AIC=-11339.24   AICc=-11339.23   BIC=-11322.16
```

Observations of the ARIMA Model

The output suggests that an ARIMA(1,0,0) is the best fit for the data. This means that a first-order autoregressive (AR(1)) model is suitable. This contradicts our initial analysis from the ACF and PACF plots.

Now, to further decide on which model is most suitable, we compare the BIC values of the ARIMA(1,0,0) and ARIMA(0,0,1) models.

I am choosing to use BIC values based on my qualitative understanding of AAPL stock. I believe that despite AAPL being affected by the global market environment, its strong product power and franchise loyalty will help it maintain a long term growth trend. Thus, I am prioritizing simplicity and generalizability over complexity in the model selection.

Compare ARIMA Models using BIC

Note: GARCH model parameters are kept constant for the comparison.

```
# AR(1) model
spec_ar <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(1, 0))
)

# MA(1) model
spec_ma <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 1))
)

# Fit the models
fit_ar <- ugarchfit(spec_ar, data = aapl_data$Log.Return)
fit_ma <- ugarchfit(spec_ma, data = aapl_data$Log.Return)

# Extract BIC values
ar_info <- infocriteria(fit_ar)
bic_ar <- ar_info[2]
ma_info <- infocriteria(fit_ma)
bic_ma <- ma_info[2]

# Output and compare BIC values
cat("BIC for AR(1) model: ", bic_ar, "\n")
```

```
## BIC for AR(1) model:  -5.366199
```

```
cat("BIC for MA(1) model: ", bic_ma, "\n")
```

```
## BIC for MA(1) model:  -5.366199
```

```
if (bic_ar == bic_ma) {
  print("Both models have the same BIC value")
  print("Hence, both models are equally suitable")
} else if (bic_ar < bic_ma) {
  print("AR(1) model has lower BIC")
  print("Hence, ARIMA(1,0,0) model is more suitable")
} else if (bic_ar > bic_ma){
  print("MA(1) model has lower BIC")
  print("Hence, ARIMA(0,0,1) model is more suitable")
}
```

```
## [1] "MA(1) model has lower BIC"
## [1] "Hence, ARIMA(0,0,1) model is more suitable"
```

```
print(bic_ar - bic_ma)
```

```
## [1] 1.949975e-07
```

Observations of the BIC Comparison

The BIC values of the ARIMA(1,0,0) and ARIMA(0,0,1) models are very close, with the ARIMA(0,0,1) model having a slightly lower BIC value. This suggests that the ARIMA(0,0,1) model is slightly more suitable for the data. However, since this difference is minimal, this indicates that either models might be suitable for the data.

To conclude, we will use MA(1) or ARIMA(0,0,1) model for forecasting the log returns of AAPL stock.

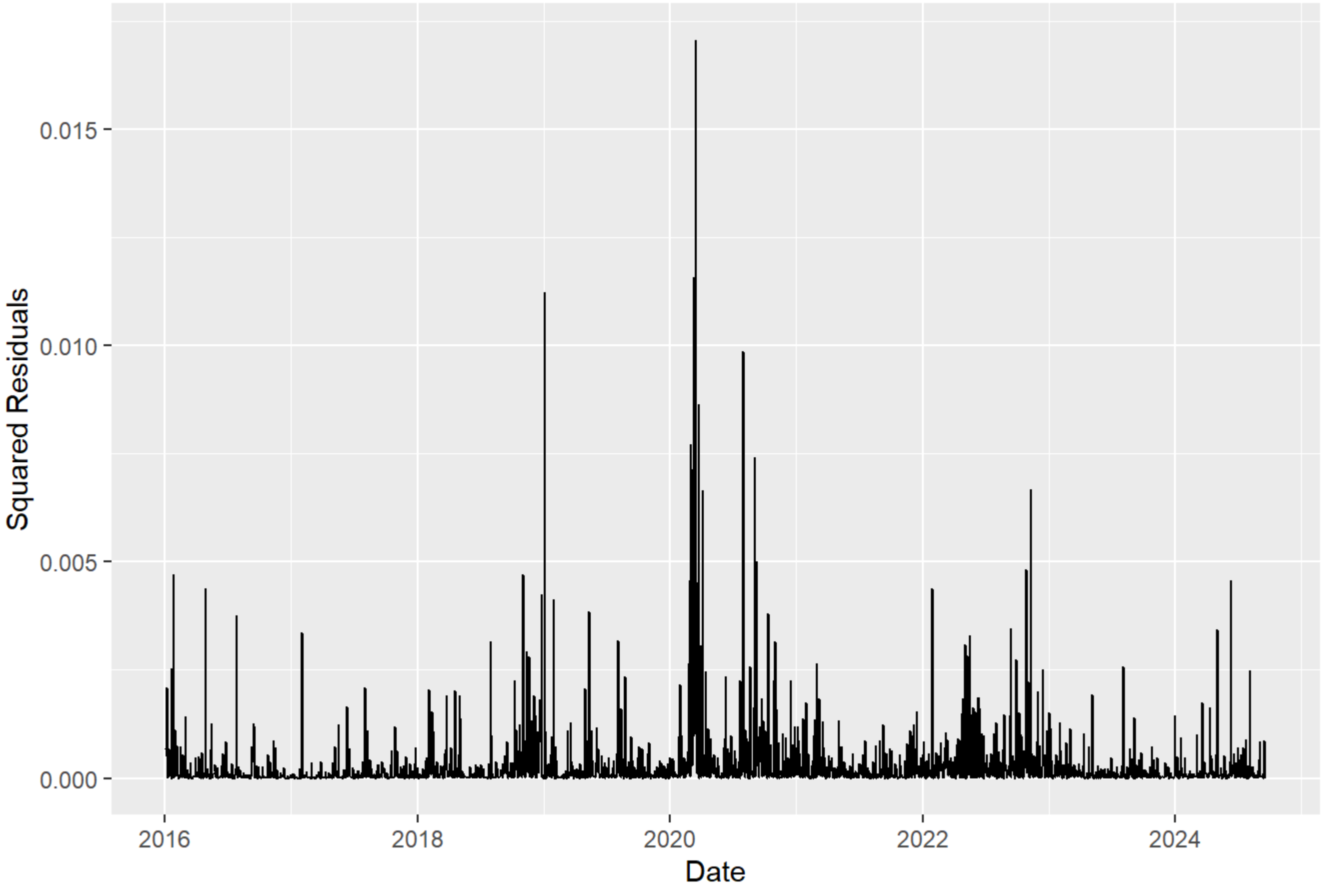
Handling Conditional Heteroskedasticity

From Figure 1, it can be observed that the log returns of AAPL stock exhibit volatility clustering. This suggests that the data is conditionally heteroskedastic. We can double confirm this by looking at the squared residuals of the ARIMA model.

```
# Extract residuals from ARIMA model
residuals <- residuals(fit)

# Plot the squared residuals
ggplot(aapl_data, aes(x = Date, y = residuals^2)) + geom_line() + ggtitle("Figure 2: Squared
Residuals of ARIMA Model vs Date") + xlab("Date") + ylab("Squared Residuals")
```

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

Figure 2: Squared Residuals of ARIMA Model vs Date



Observations of the Squared Residuals

It is clear that the squared residuals exhibit volatility clustering as seen by the large spikes in the graph, confirming that the data is conditionally heteroskedastic. To account for this, we will fit a GARCH model to the data.

Fit a GARCH Model

```
# GARCH(1,1) specification
spec11 <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 1))
)

# GARCH(1,2) specification
spec12 <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 2)),
  mean.model = list(armaOrder = c(0, 1))
)

# GARCH(2,1) specification
spec21 <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(2, 1)),
  mean.model = list(armaOrder = c(0, 1))
)

# GARCH(2,2) specification
spec22 <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(2, 2)),
  mean.model = list(armaOrder = c(0, 1))
)

# Fit the GARCH model
fit11 <- ugarchfit(spec11, data = aapl_data$Log.Return)
fit12 <- ugarchfit(spec12, data = aapl_data$Log.Return)
fit21 <- ugarchfit(spec21, data = aapl_data$Log.Return)
fit22 <- ugarchfit(spec22, data = aapl_data$Log.Return)

# output results
fit_garch_11 <- infocriteria(fit11)
cat("GARCH(1,1) BIC: ", fit_garch_11[2], "\n")
```

```
## GARCH(1,1) BIC:  -5.366199
```

```
fit_garch_12 <- infocriteria(fit12)
cat("GARCH(1,2) BIC: ", fit_garch_12[2], "\n")
```

```
## GARCH(1,2) BIC:  -5.362436
```

```
fit_garch_21 <- infocriteria(fit21)
cat("GARCH(2,1) BIC: ", fit_garch_21[2], "\n")
```

```
## GARCH(2,1) BIC:  -5.362431
```

```
fit_garch_22 <- infocriteria(fit22)
cat("GARCH(2,2) BIC: ", fit_garch_22[2], "\n")
```

```
## GARCH(2,2) BIC:  -5.35958
```

Observations of the GARCH Model

From the BIC values, the GARCH(1,1) model has the lowest BIC value, suggesting that it is the most suitable model for the data.

Data & Model Analysis

```
# Peform Shapiro-Wilk test on the data to check for normality
shapiro.test(aapl_data$Log.Return)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aapl_data$Log.Return
## W = 0.93688, p-value < 2.2e-16
```

```
# Redefining the GARCH (1,1) model using Student's t distribution
spec11_t <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 1)),
  distribution.model = "std"
)

# Fit the GARCH (1,1) model with Student's t distribution
fit11_t <- ugarchfit(spec11_t, data = aapl_data$Log.Return)

# Output the results
fit_garch_11_t <- infocriteria(fit11_t)
cat("GARCH(1,1) with Student's t BIC: ", fit_garch_11_t[2], "\n")
```

```
## GARCH(1,1) with Student's t BIC:  -5.459818
```

Some Observations

Since the data does not follow a normal distribution, we can use the GARCH(1,1) model with Student’s t distribution to account for the heavy tails in the data. This model has a lower BIC value compared to the GARCH(1,1) model with normal distribution, suggesting that it is more suitable for the data.

```
# obtain the degrees of freedom of the Student's t distribution
df = coef(fit11_t)[6]
print("Degrees of Freedom of Student's t Distribution:")
```

```
## [1] "Degrees of Freedom of Student's t Distribution:"
```

```
print(df)
```

```
##    shape
## 4.464358
```

Observations of the Student’s t Distribution

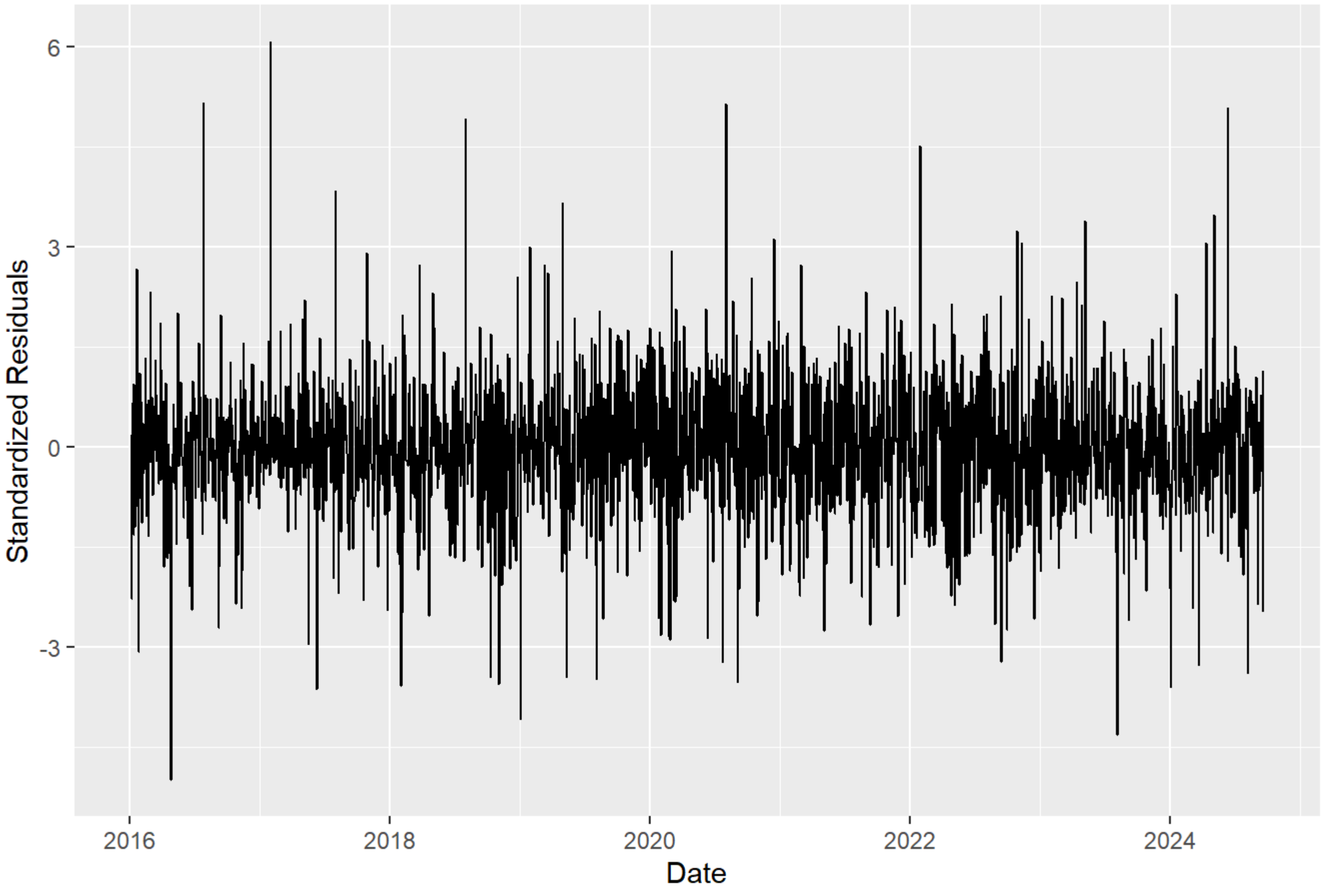
The degrees of freedom of the Student's t distribution is approximately 4.46. This suggests that the data has heavier tails compared to a normal distribution.

Standardized Residuals

```
# Extract residual from mean model MA(1)
residuals_garch <- residuals(fit11_t)

# Extract estimated conditional volatility from the GARCH model
volatility <- sigma(fit11_t)

# Calculate the standardized residuals
std_resid <- residuals_garch / volatility

# Plot the standardized residuals
ggplot(aapl_data, aes(x = Date, y = std_resid)) + geom_line() + ggtitle("Figure 3: Standardized Residuals of GARCH(1,1) Model vs Date") + xlab("Date") + ylab("Standardized Residuals")
```

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

Figure 3: Standardized Residuals of GARCH(1,1) Model vs Date



Observations of the Standardized Residuals

The standardized residuals of the GARCH(1,1) model show that the residuals are centered around zero, indicating that the model is capturing the mean of the data. However, there are still some spikes in the graph, suggesting that the model might not be capturing all the volatility in the data.

Analysis of Standardized Residuals

1. Ljung-Box Test

```
# Perform Ljung-Box test on the standardized residuals
Box.test(std_resid, lag = lag_max, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  std_resid
## X-squared = 37.4, df = 30, p-value = 0.1658
```

Observations of the Ljung-Box Test

The p-value of the Ljung-Box Test is greater than 0.05, this suggests there is no strong autocorrelation in the standardized residuals. This indicates that the GARCH(1,1) model is capturing the autocorrelation in the data effectively.

2. Ljung Box Test on Squared Residuals

```
# Perform Ljung-Box test on the squared standardized residuals
Box.test(std_resid^2, lag = lag_max, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  std_resid^2
## X-squared = 17.325, df = 30, p-value = 0.9685
```

Observations of the Ljung-Box Test on Squared Residuals

Similarly, the p-value of the Ljung-Box Test on the squared standardized residuals is greater than 0.05, suggesting that there is no strong autocorrelation in the squared standardized residuals. This indicates that the GARCH(1,1) model is capturing the volatility clustering/time-varying variance in the data effectively.

3. LM Test for Conditional Hetroskedasticity

```
# Perform the LM test for conditional heteroskedasticity
arch.test(std_resid, lag.max = lag_max)
```

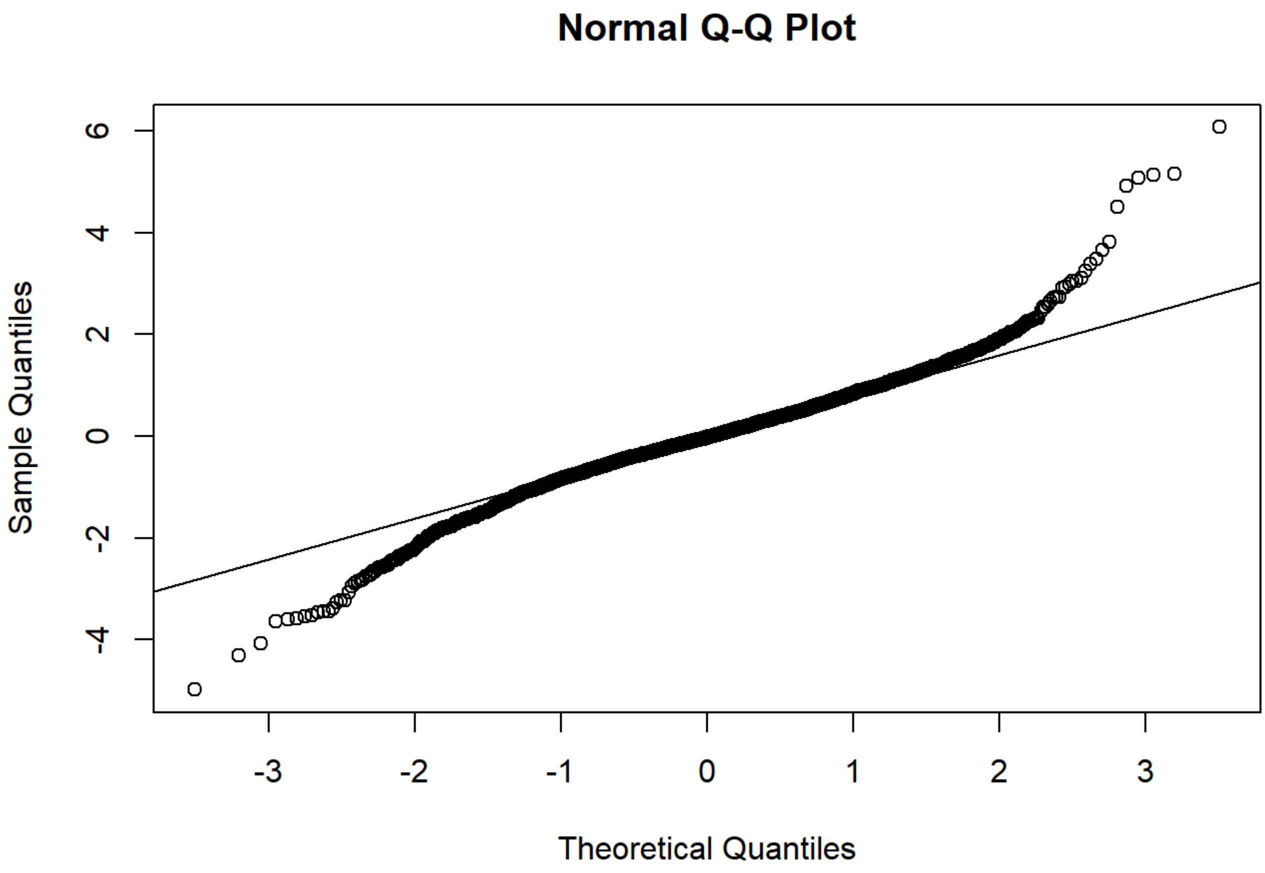
```
##
##  Box-Ljung test
##
## data:  y^2
## X-squared = 17.325, df = 30, p-value = 0.9685
## alternative hypothesis: y is heteroscedastic
```

Observations of the LM Test

The p-value of the LM Test is greater than 0.05, suggesting that there is no strong evidence of conditional heteroskedasticity in the standardized residuals. This indicates that the GARCH(1,1) model has properly captured the volatility clustering/time-varying variance in the data.

4. Distributional Assumption Test

```
# Using QQplot to check if the standardized residuals follow a normal distribution
qqnorm(std_resid)
qqline(std_resid)
```



QQ Plot Observations

From the plot, we can see that the standardized residuals do not follow a normal distribution as seen from the large tails at each end. This is consistent with the Shapiro-Wilk test results. This suggests that the Student’s t distribution is more suitable for the data compared to the normal distribution.