Project Work

Kimberley Maldonado, Brendan Kenny, Emily Su, Brianna Cirillo

Libraries

```
# Load necessary libraries
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tseries)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.1     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(ggplot2)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(TSA)
```

```
## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
```

```
##   plot.Arima   forecast
##
## Attaching package: 'TSA'
##
## The following object is masked from 'package:readr':
##
##     spec
##
## The following objects are masked from 'package:stats':
##
##     acf, arima
##
## The following object is masked from 'package:utils':
##
##     tar
```

```
library(rugarch)
```

```
## Loading required package: parallel
##
## Attaching package: 'rugarch'
##
## The following object is masked from 'package:purrr':
##
##     reduce
##
## The following object is masked from 'package:stats':
##
##     sigma
```

```
library(forecast)
library(PerformanceAnalytics)
```

```
## Loading required package: xts
##
## ########################## Warning from 'xts' package ##########################
## #                                                                              #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                              #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                              #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
## ################################################################################
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
```

```
##
##     first, last
##
##
## Attaching package: 'PerformanceAnalytics'
##
## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness
##
## The following object is masked from 'package:graphics':
##
##     legend
```

```
library(xts)
library(quantmod)
```

```
## Loading required package: TTR
```

FRED Monthly Retail Sales

Load & Inspect the Data

```
file_path <- "~/Documents/GitHub/MA-641-Course-Project/Scratch Work/RSXFSN.csv"

retail_sales_data <- read_csv(file_path)
```

```
## Rows: 126 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): RSXFSN
## date (1): DATE
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Convert the DATE column to Date type
retail_sales_data$DATE <- as.Date(retail_sales_data$DATE, format="%Y-%m-%d")
```

The following dataset is part of the Advance Monthly Retail Trade Survery conducted by the U.S Census Bureau, retrieved from FRED and measures the retail trade activity in the United States. It provides an early estimate of monthly sales for retail and food service companies capturing consumer demand for goods and services.

The RSXFSN data series measures the dollar value of sales and is reported on millions of dollars. This data is not seasonally adjusted and is released monthly. The period of observation for this project spans a 10 year period from January 2014 to May 2024, totaling 126 observations.

```
head(retail_sales_data)
```

```
## # A tibble: 6 x 2
##   DATE       RSXFSN
##   <date>      <dbl>
## 1 2014-01-01 340439
## 2 2014-02-01 337453
## 3 2014-03-01 383540
## 4 2014-04-01 383796
## 5 2014-05-01 408258
## 6 2014-06-01 385463
```

```
summary(retail_sales_data)
```

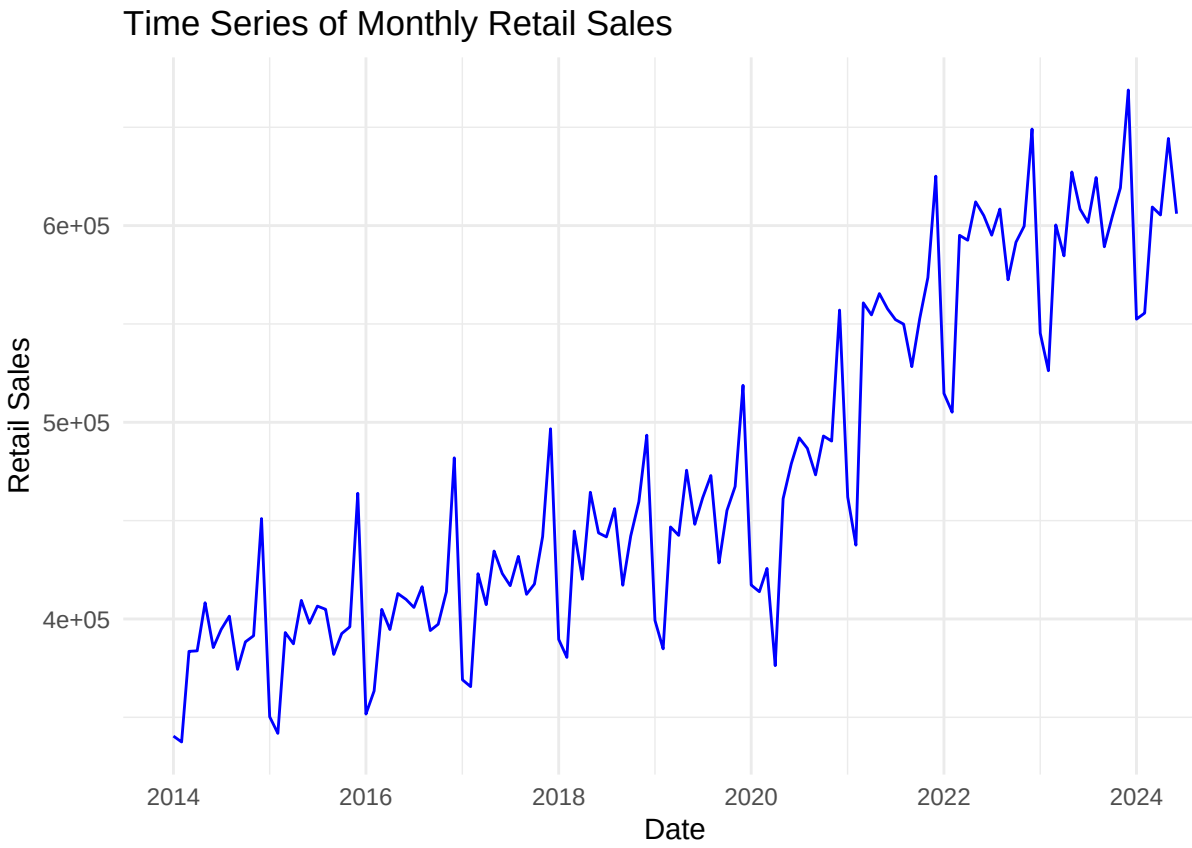
```
##       DATE                RSXFSN
##  Min.   :2014-01-01   Min.   :337453
##  1st Qu.:2016-08-08   1st Qu.:405145
##  Median :2019-03-16   Median :449590
##  Mean   :2019-03-17   Mean   :474666
##  3rd Qu.:2021-10-24   3rd Qu.:554140
##  Max.   :2024-06-01   Max.   :668957
```

```
# Check the number of rows in the original dataset
num_data_points <- nrow(retail_sales_data)
cat("Number of data points in the original dataset:", num_data_points, "\n")
```

```
## Number of data points in the original dataset: 126
```

Descriptive Analysis

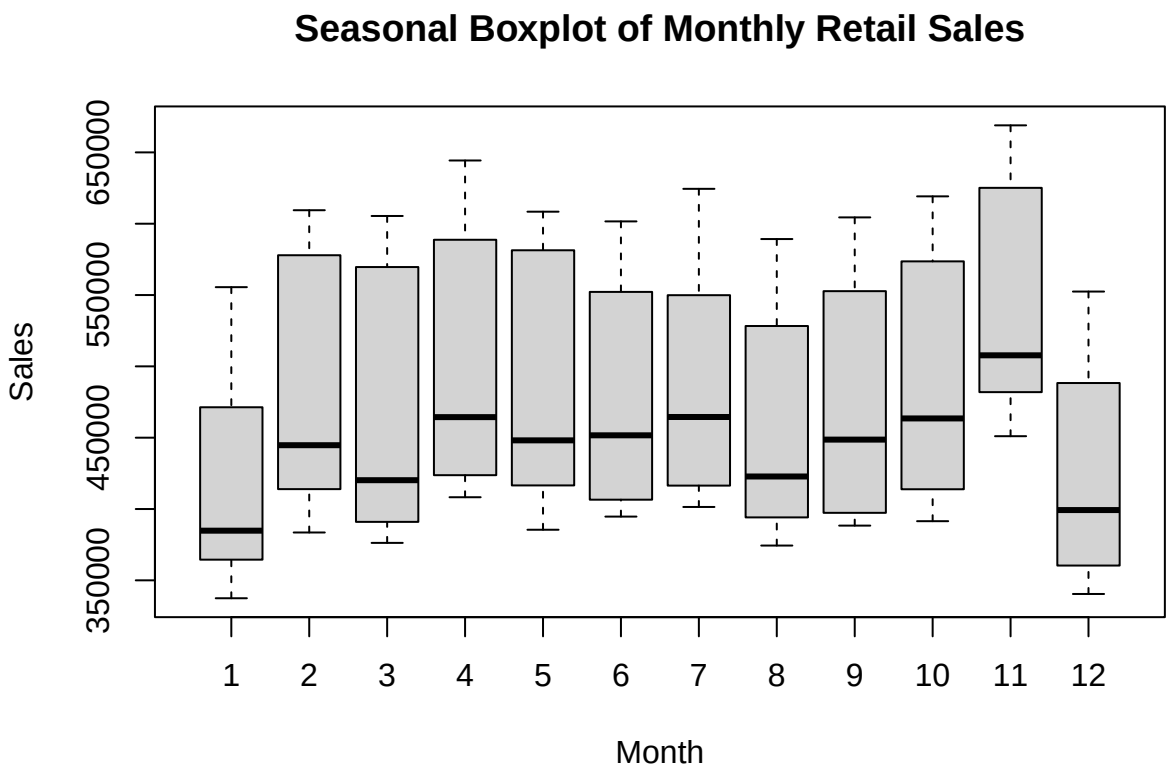
```
# Plot the original data
ggplot(data = retail_sales_data, aes(x = DATE, y = RSXFSN)) +
  geom_line(color = "blue") +
  labs(title = "Time Series of Monthly Retail Sales",
       x = "Date",
       y = "Retail Sales") +
  theme_minimal()
```



Time Series Plot

The time series displays a noticeable upward trend in retails sales with periodic spikes in the final months of the year, indicating the presence of seasonality.

```
# Create a time series object
ts_data <- ts(retail_sales_data$RSXFSN, start = c(2014, 12), frequency = 12)

boxplot(ts_data ~ cycle(ts_data), main="Seasonal Boxplot of Monthly Retail Sales", ylab = "Sales", xlab
```

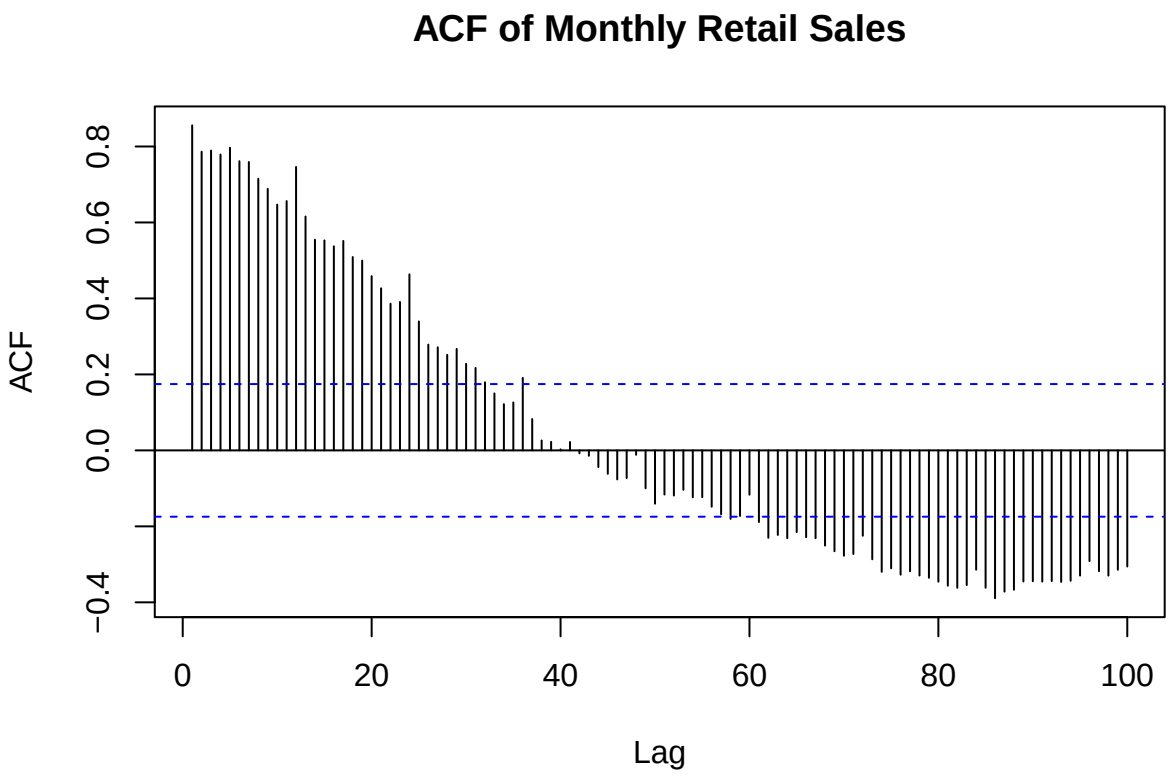


Box Plot

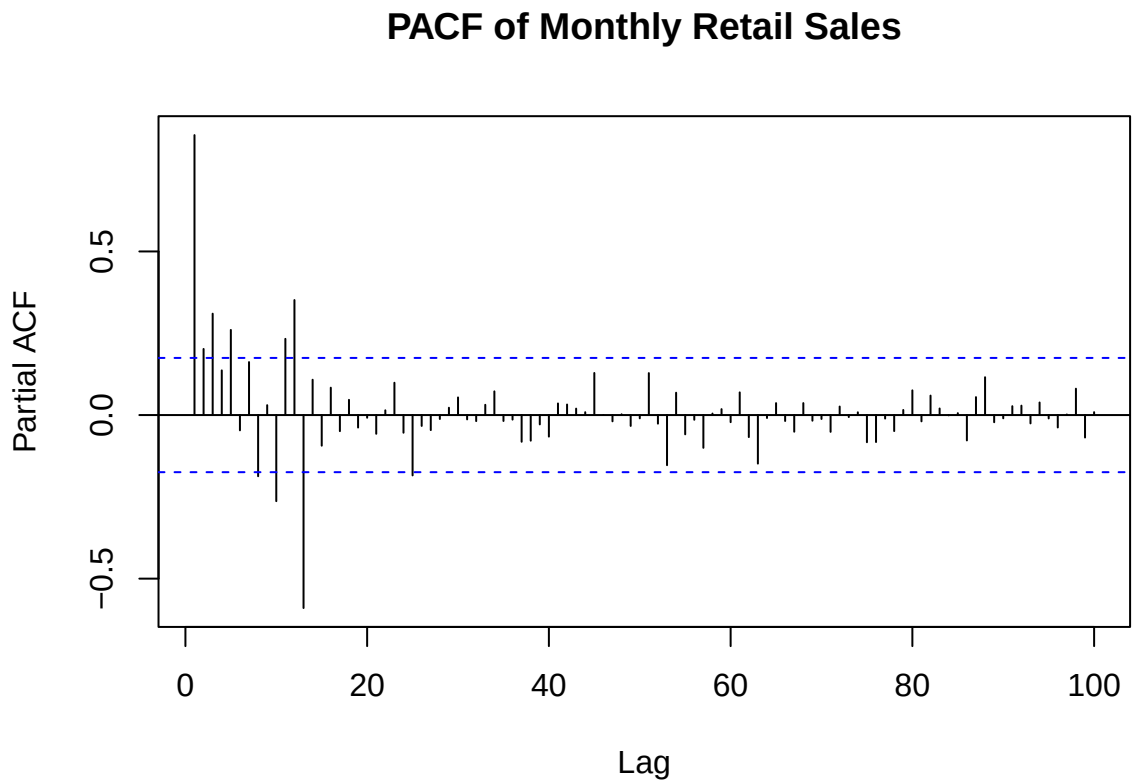
Although it makes sense that the month of November would see peak sales due to Black Friday and Cyber Monday events, it seems counter-intuitive that December would have such a stark decrease in sales, considering the holiday season.

ACF and PACF of Original Data

```
# Create ACF plot
acf(retail_sales_data$RSXFSN, main = "ACF of Monthly Retail Sales", lag.max = 100)
```




```
# Create PACF plot
par(mar=c(5, 5, 4, 2) + 0.1)
pacf(retail_sales_data$RSXFSN, main = "PACF of Monthly Retail Sales", lag.max = 100)
```



###

ADF Test

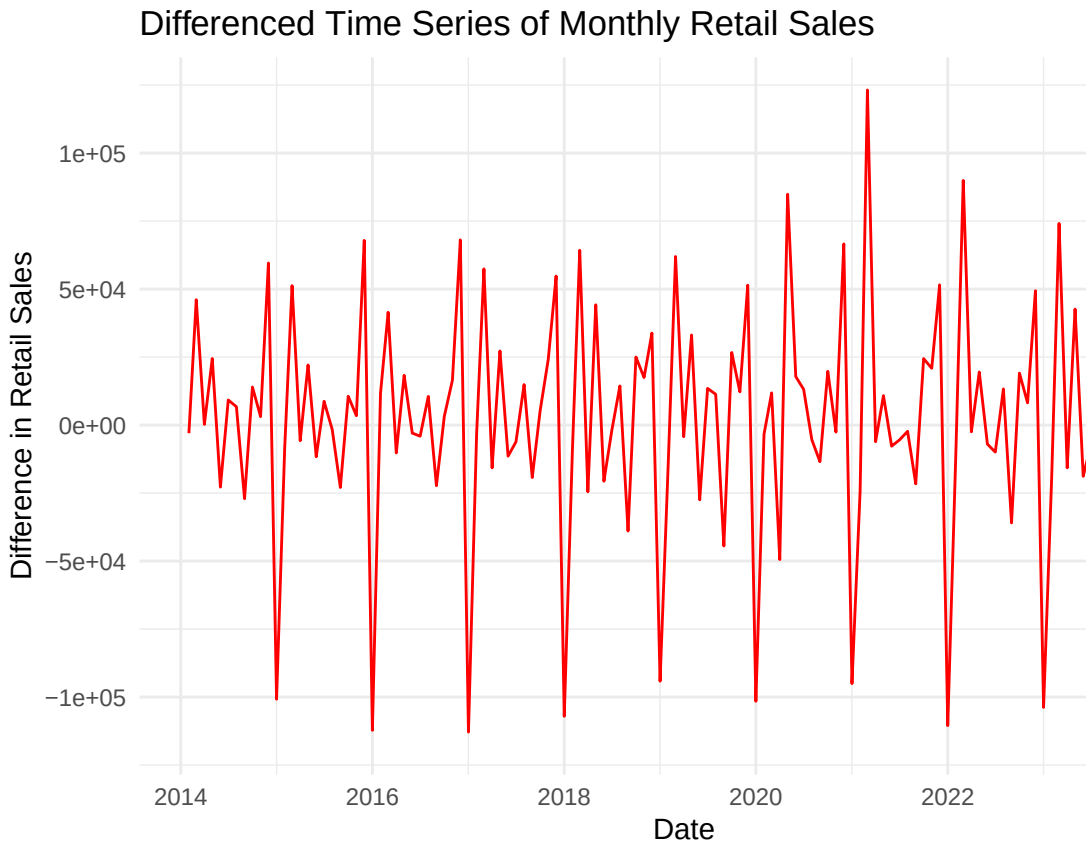
```
adf_test_result <- adf.test(retail_sales_data$RSXFSN)
print(adf_test_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  retail_sales_data$RSXFSN
## Dickey-Fuller = -2.4875, Lag order = 4, p-value = 0.3739
## alternative hypothesis: stationary
```

The data shows clear non-stationarity. Therefore, we proceed with differencing measures to achieve stationarity.

Obtaining Stationary Time Series

```
# Calculate the differences of the RSXFSN series
diff_series <- diff(retail_sales_data$RSXFSN)

# Plot the differenced data
diff_data <- data.frame(DATE = retail_sales_data$DATE[-1], Difference = diff_series)

ggplot(data = diff_data, aes(x = DATE, y = Difference)) +
  geom_line(color = "red") +
  labs(title = "Differenced Time Series of Monthly Retail Sales",
       x = "Date",
       y = "Difference in Retail Sales") +
  theme_minimal()
```



One order of differencing

```
adf_test_result <- adf.test(diff_series)
```

```
## Warning in adf.test(diff_series): p-value smaller than printed p-value
```

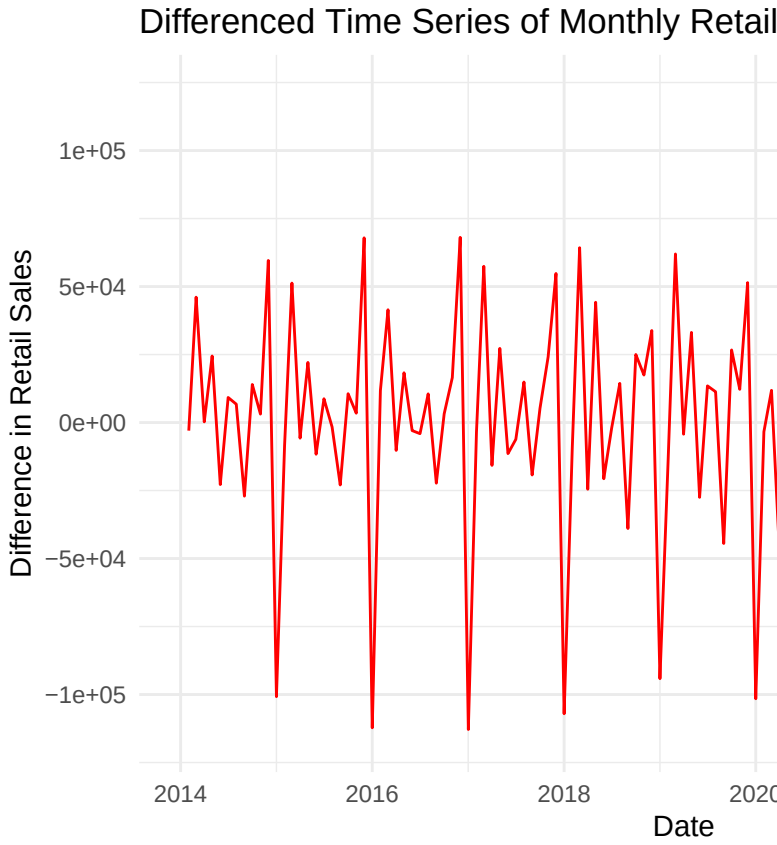
```
print(adf_test_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_series
## Dickey-Fuller = -8.0818, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

p-value from ADF $= 0.01 < 0.05$ -> stationary

```
# Take log transformation of data
log_series <- log(retail_sales_data$RSXFSN)

# Apply differencing
diff_log_series <- diff(log_series)

# Plot the differenced data
```

```
diff_log_data <- data.frame(DATE = retail_sales_data$DATE[-1], Difference = diff_log_series)

ggplot(data = diff_data, aes(x = DATE, y = Difference)) +
  geom_line(color = "red") +
  labs(title = "Differenced Time Series of Monthly Retail Sales",
       x = "Date",
       y = "Difference in Retail Sales") +
  theme_minimal()
```



Log transformation and one order of differencing

```
adf_test_result <- adf.test(diff_log_series)
```

```
## Warning in adf.test(diff_log_series): p-value smaller than printed p-value
```

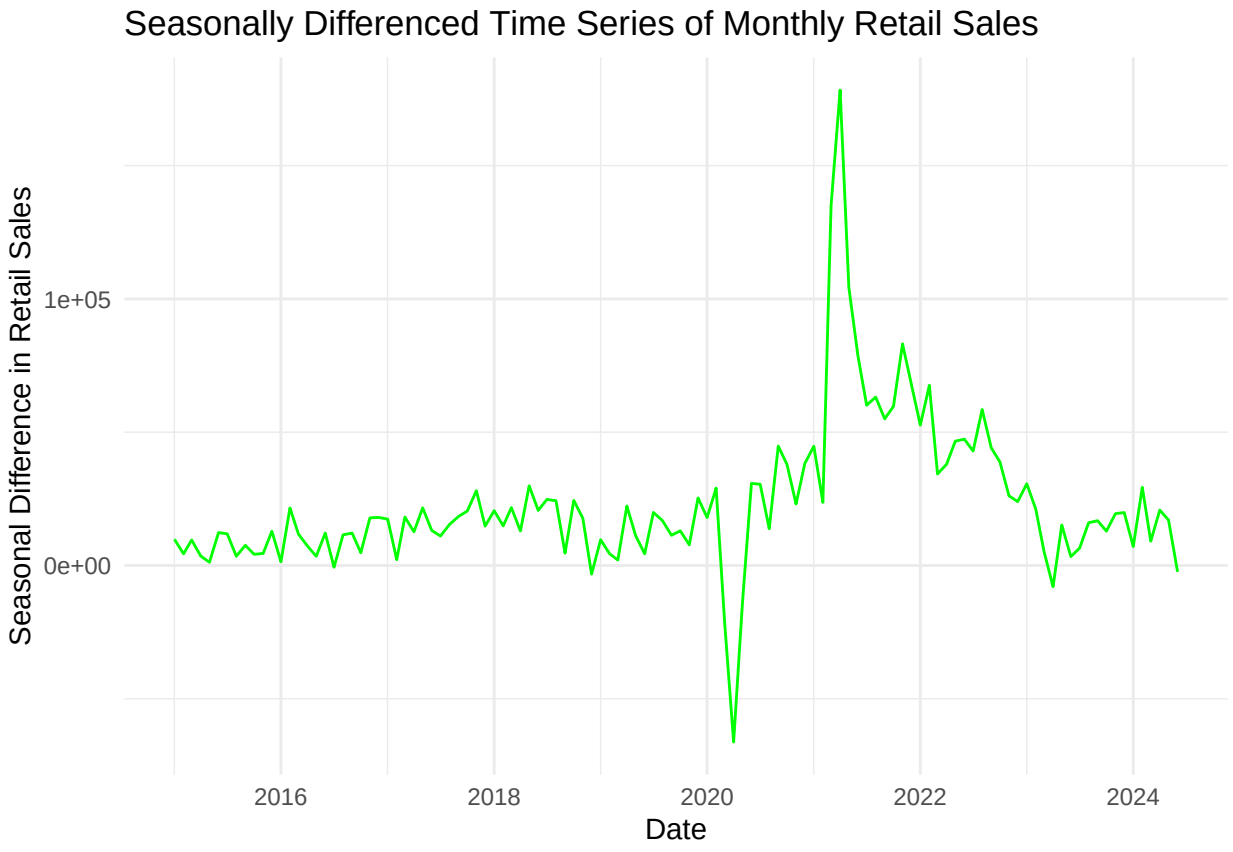
```
print(adf_test_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_log_series
## Dickey-Fuller = -7.9621, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

One order of differencing, both with and without taking the logarithmic transformation, achieves stationarity.

One order seasonal differencing

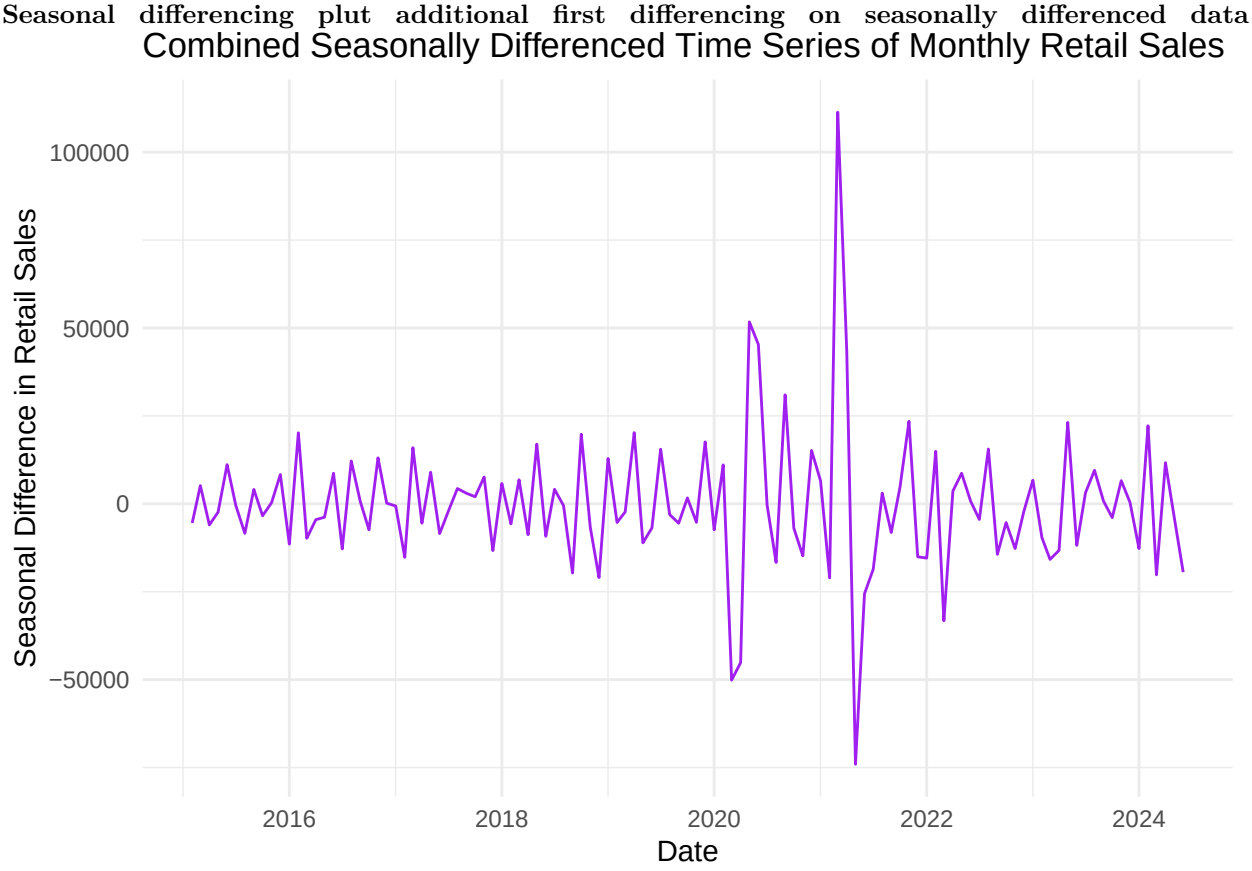
```
# Apply seasonal differencing to the RSXFSN series
seasonal_diff_series <- diff(retail_sales_data$RSXFSN, lag = 12)

# Create a data frame for the seasonally differenced data
seasonal_diff_data <- data.frame(DATE = retail_sales_data$DATE[-(1:12)], Difference = seasonal_diff_ser

# Plot the seasonally differenced data
ggplot(data = seasonal_diff_data, aes(x = DATE, y = Difference)) +
  geom_line(color = "green") +
  labs(title = "Seasonally Differenced Time Series of Monthly Retail Sales",
       x = "Date",
       y = "Seasonal Difference in Retail Sales") +
  theme_minimal()
```




```
# Apply seasonal differencing to the RSXFSN series
seasonal_diff_series <- diff(retail_sales_data$RSXFSN, lag = 12)

# Perform additional first differencing on seasonally differenced data to ensure stationarity
combined_diff_data <- diff(seasonal_diff_series)

# Create a data frame for the seasonally differenced data
```

```
doubly_diff_data <- data.frame(DATE = retail_sales_data$DATE[-(1:13)], Difference = combined_diff_data)

# Plot the seasonally differenced data
ggplot(data = doubly_diff_data, aes(x = DATE, y = Difference)) +
  geom_line(color = "purple") +
  labs(title = "Combined Seasonally Differenced Time Series of Monthly Retail Sales",
       x = "Date",
       y = "Seasonal Difference in Retail Sales") +
  theme_minimal()
```

Seasonal differencing plut additional first differencing on seasonally differenced data




```
adf_test_result <- adf.test(combined_diff_data)
```

```
## Warning in adf.test(combined_diff_data): p-value smaller than printed p-value
```

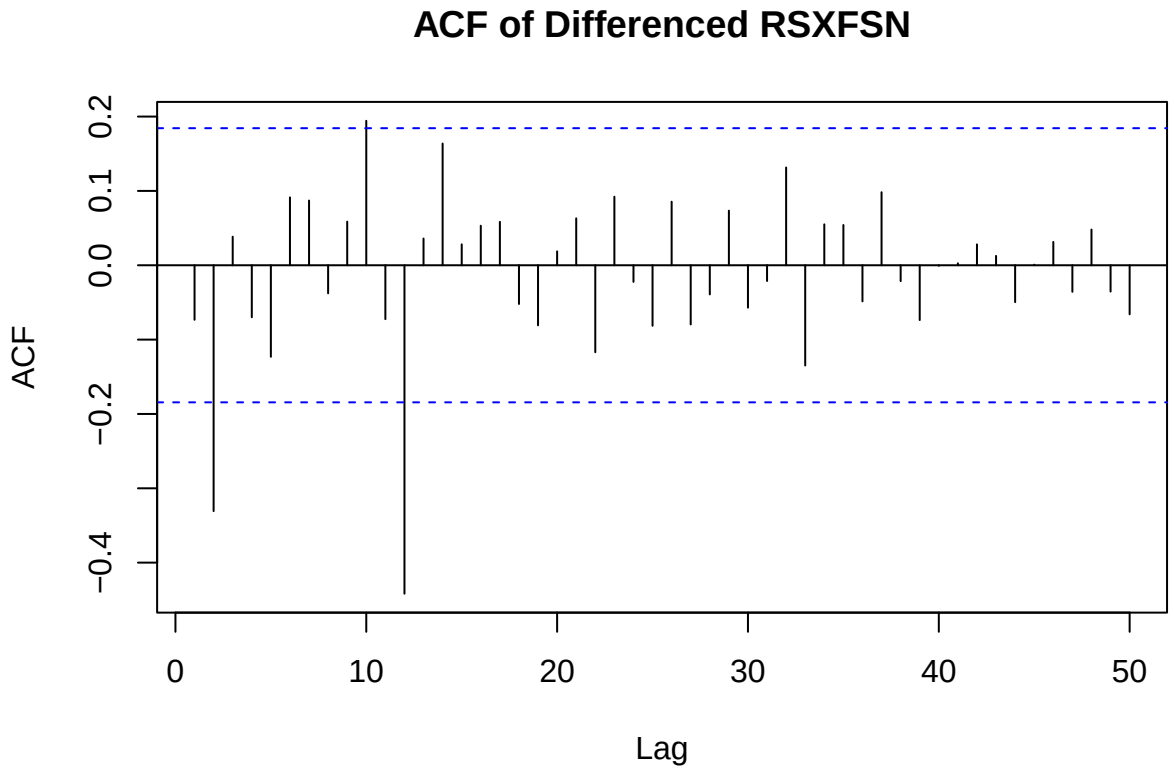
```
print(adf_test_result)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  combined_diff_data
## Dickey-Fuller = -7.1014, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Taking a seasonal difference of the data, and then an additional order of differencing produces the time series most resembling white noise, albeit with clear volatility between 2020 and 2022. This can perhaps be

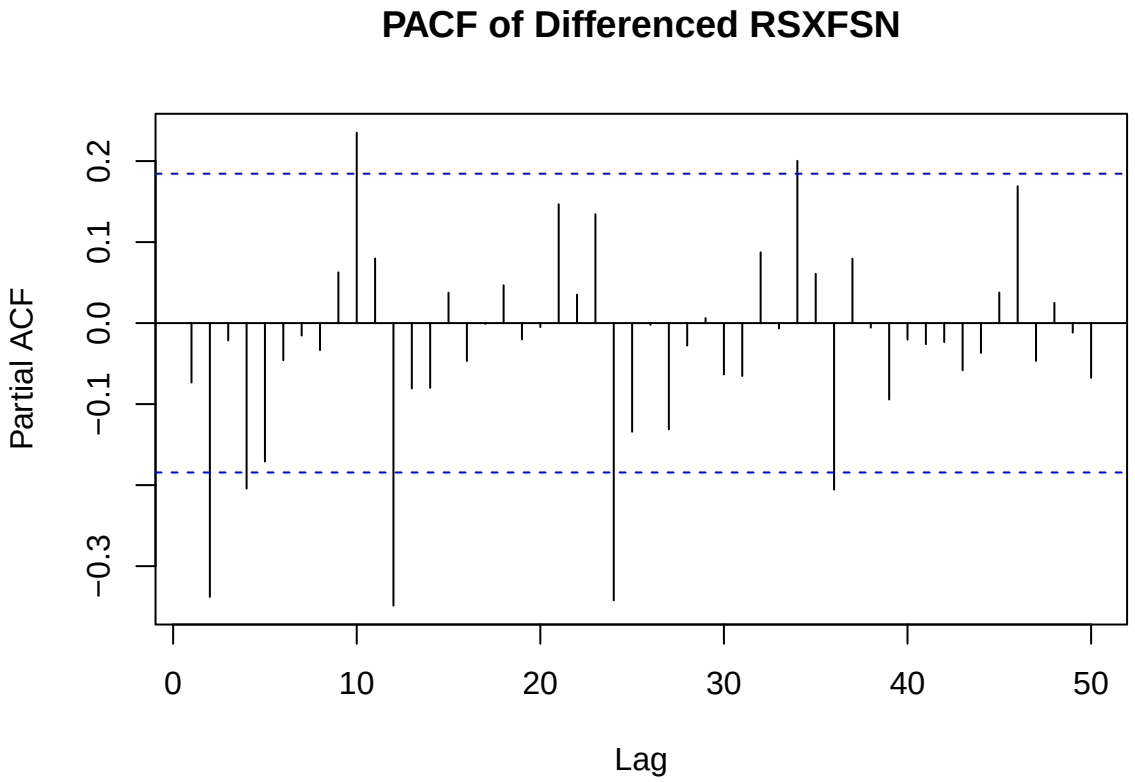
explained by the COVID-19 pandemic, which saw many store closures and and ceasing of retail activities. We now investigate the ACF and PACF plots with this data.

```
# Plot ACF for the differenced data
acf(combined_diff_data, main="ACF of Differenced RSXFSN", lag.max=50)
```



ACF and PACF

```
# PACF plot for differenced data
par(mar=c(5, 5, 4, 2) + 0.1)
pacf(combined_diff_data, main="PACF of Differenced RSXFSN", lag.max=50)
```



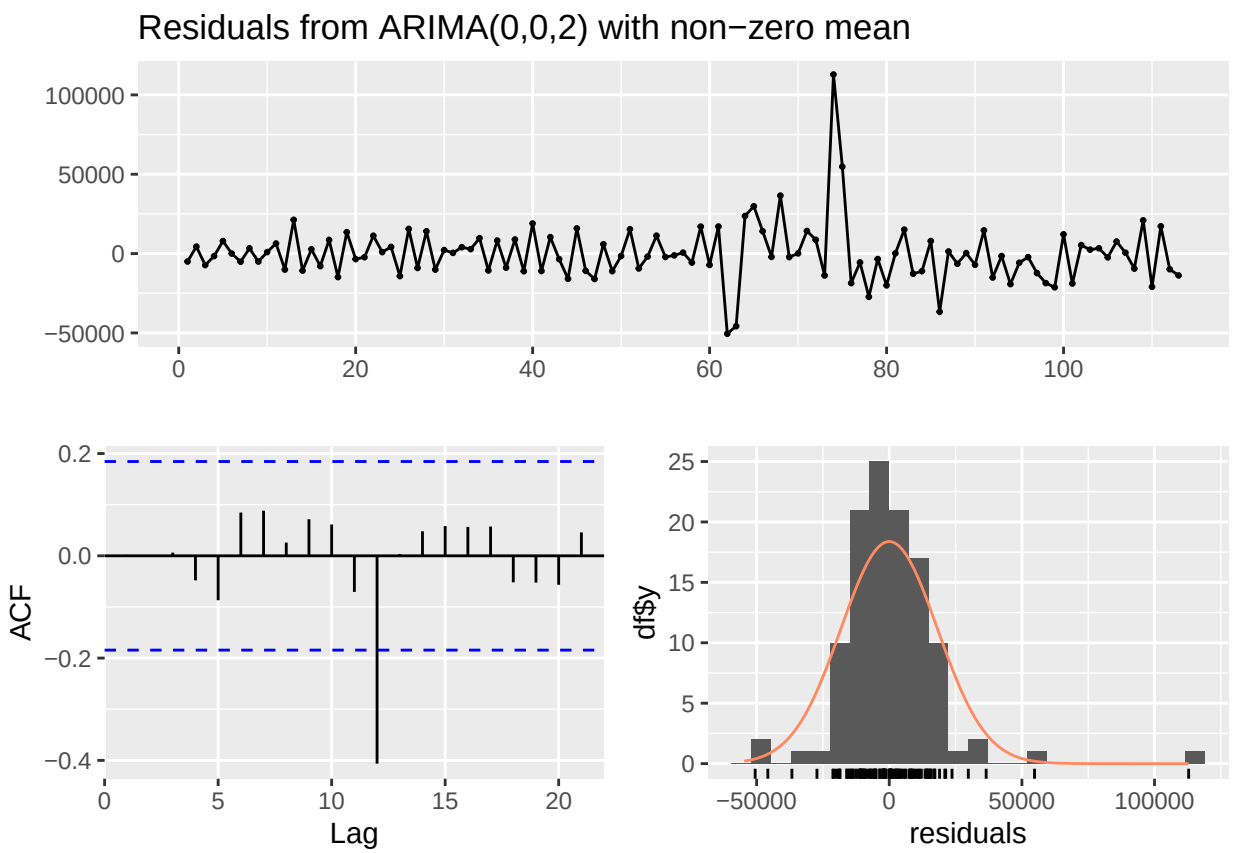
The ACF displays an abrupt cut-off after lag 12, and the PACF gradually decays, which is indicative of a Moving Average model.

Fitting a SARIMA Model

```
# Fit a seasonal ARIMA model manually
# Non-seasonal part: ARIMA(0,0,2), Seasonal part: ARIMA(0,2,12)[12]
adjusted_arima_model <- Arima(combined_diff_data, order = c(0, 0, 2), seasonal = c(0, 0, 12))

# Summary of the adjusted ARIMA model
summary(adjusted_arima_model)
```

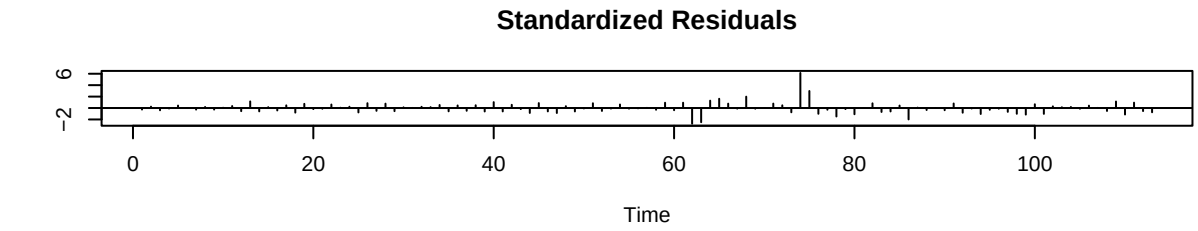
```
## Series: combined_diff_data
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##           ma1      ma2      mean
##       -0.1526  -0.4173   12.7487
## s.e.   0.0854   0.0849  749.6410
##
## sigma^2 = 3.38e+08:  log likelihood = -1268.63
## AIC=2545.26   AICc=2545.63   BIC=2556.17
##
## Training set error measures:
##                      ME     RMSE      MAE     MPE     MAPE      MASE         ACF1
## Training set -22.10046 18139.64 11832.41 56.0421 156.7261 0.5800389 0.001788122
```

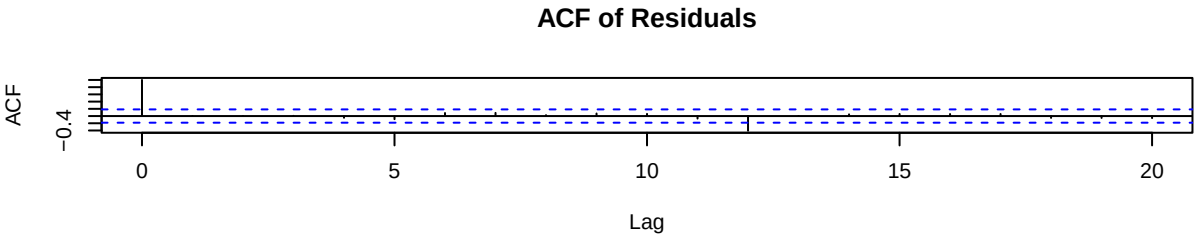
```
# Diagnostic checking of the adjusted model
checkresiduals(adjusted_arima_model)
```

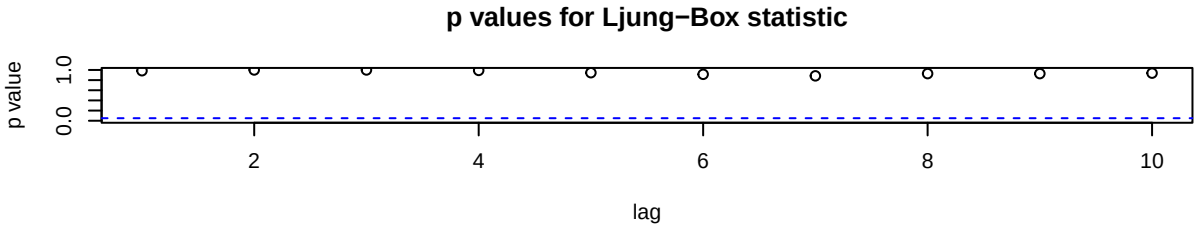



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,2) with non-zero mean
## Q* = 4.2086, df = 8, p-value = 0.8378
##
## Model df: 2.   Total lags used: 10
```

```
# Use tsdiag to generate diagnostic plots
tsdiag(adjusted_arima_model)
```



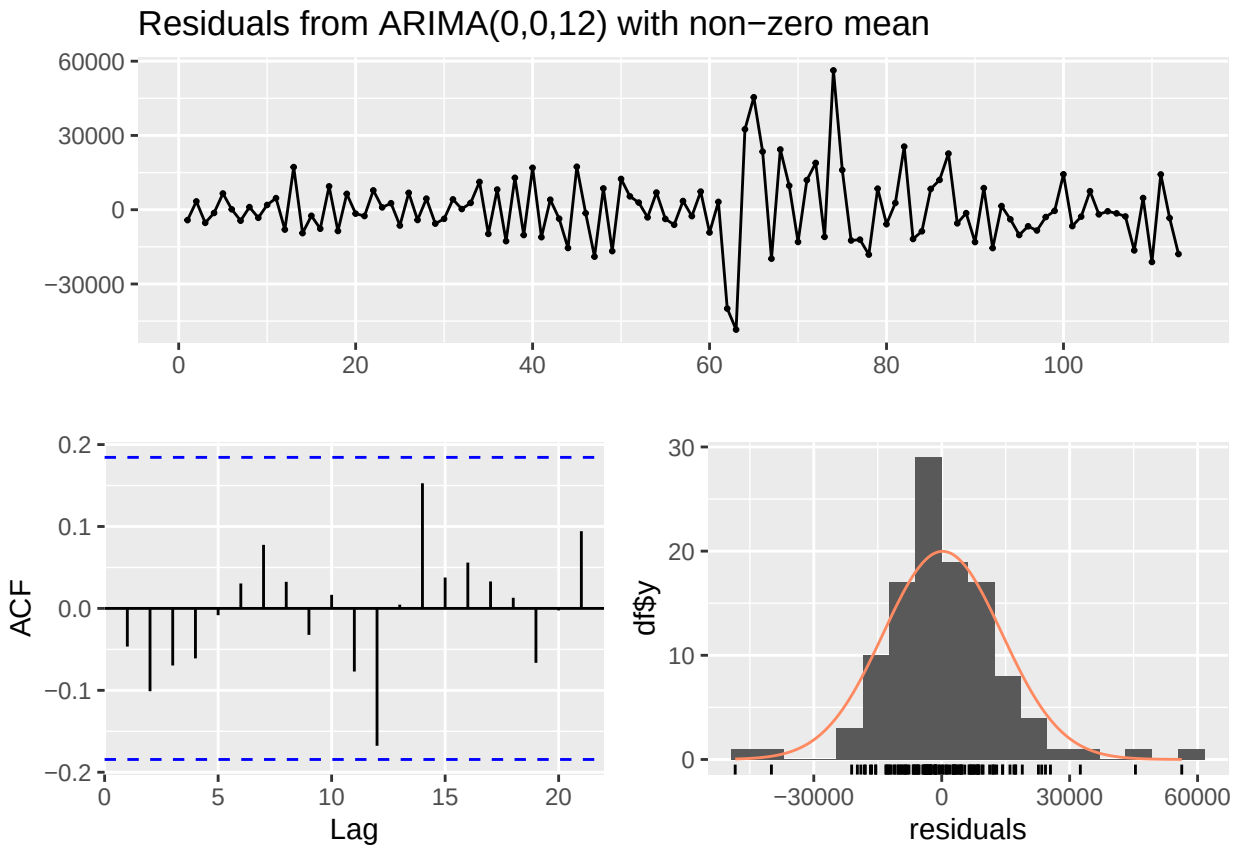





```
# Fit a seasonal ARIMA model
# Non-seasonal part: ARIMA(2,1,0), Seasonal part: ARIMA(1,1,1)[12]
adjusted_arima_model_2 <- Arima(combined_diff_data, order = c(0, 0, 12), seasonal = c(0, 0, 10))

# Summary of the adjusted ARIMA model
summary(adjusted_arima_model_2)
```

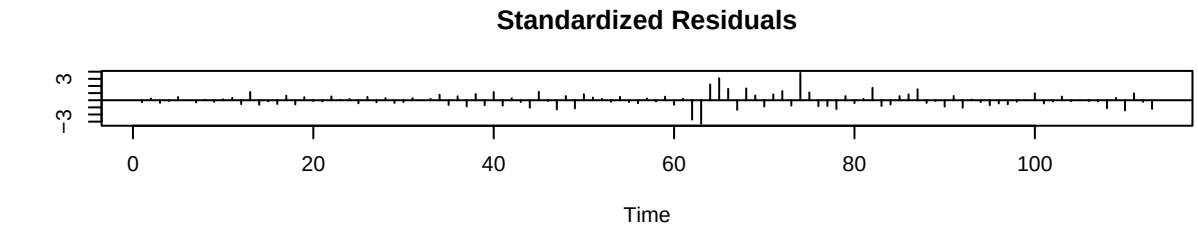
```
## Series: combined_diff_data
## ARIMA(0,0,12) with non-zero mean
##
## Coefficients:
##           ma1      ma2     ma3      ma4      ma5     ma6      ma7      ma8
##       -0.2489  -0.1880  0.1805  -0.1458  -0.0933  0.2216  -0.1103  -0.1227
## s.e.   0.0937   0.0997  0.1094   0.0942   0.0998  0.1117   0.0939   0.0954
##          ma9    ma10     ma11     ma12      mean
##       0.3371  0.0174  -0.2301  -0.6175  156.1416
## s.e.  0.1138  0.0935   0.1065   0.1106  285.6890
##
## sigma^2 = 216418099:  log likelihood = -1246.91
## AIC=2521.81   AICc=2526.1   BIC=2559.99
##
## Training set error measures:
##                     ME     RMSE     MAE      MPE     MAPE      MASE        ACF1
## Training set 203.7765 13839.09 9898.97 12.37666 183.9934 0.4852594 -0.04652619
```

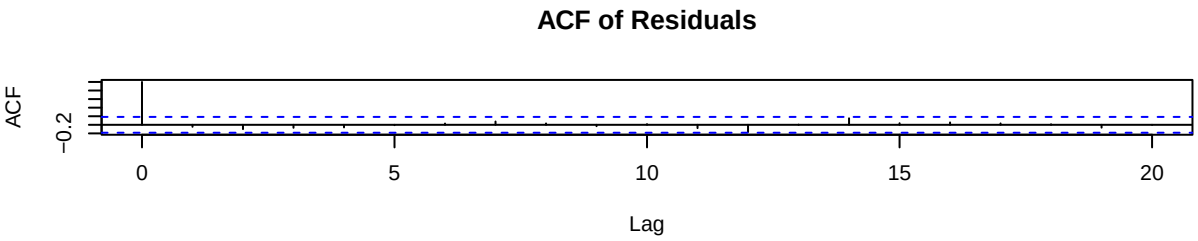
```
# Diagnostic checking of the adjusted model
checkresiduals(adjusted_arima_model_2)
```

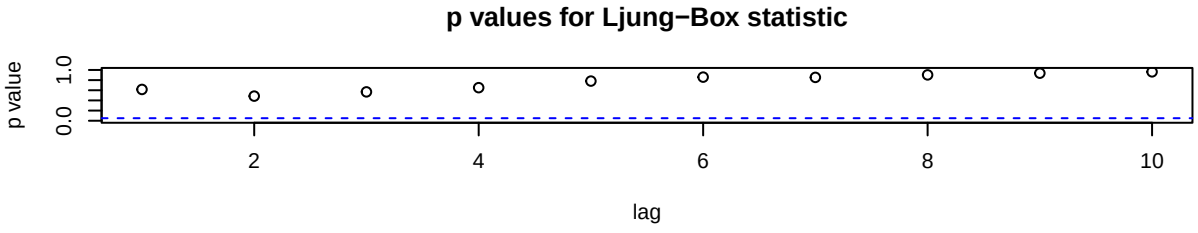



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,12) with non-zero mean
## Q* = 11.244, df = 3, p-value = 0.01048
##
## Model df: 12.   Total lags used: 15
```

```
# Use tsdiag to generate diagnostic plots
tsdiag(adjusted_arima_model_2)
```



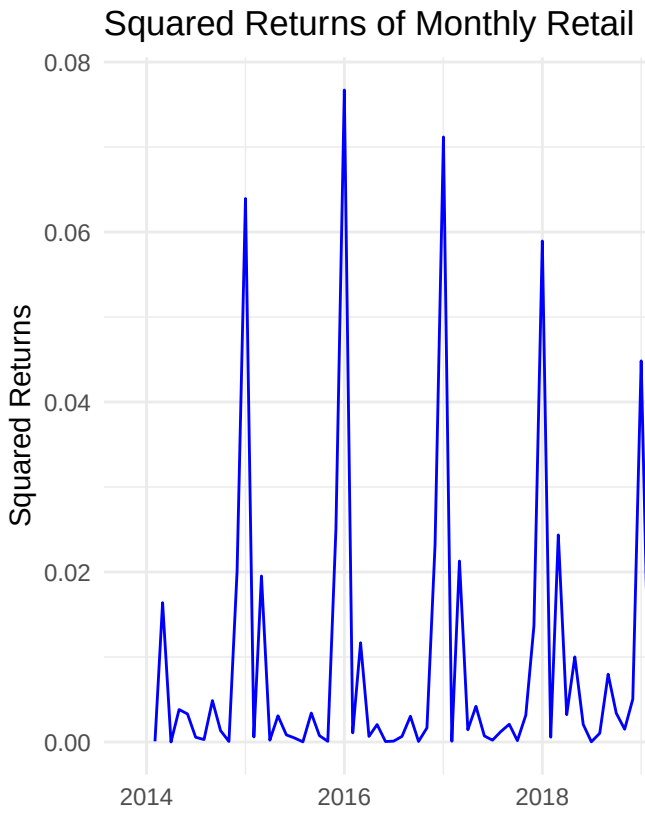




The residuals' ACF and PACF plots indicate that the model's residuals are mostly uncorrelated, suggesting a good fit, though the SARIMA(0,0,12)(0,0,10) is slightly better with an AIC of 2521.81 versus 2545.26. While the SARIMA model captures the seasonality and general trend, it may not fully account for the extreme fluctuations seen in the historical data. For better predictions, it might be necessary to explore other models or include additional explanatory variables.

Investigating GARCH Model

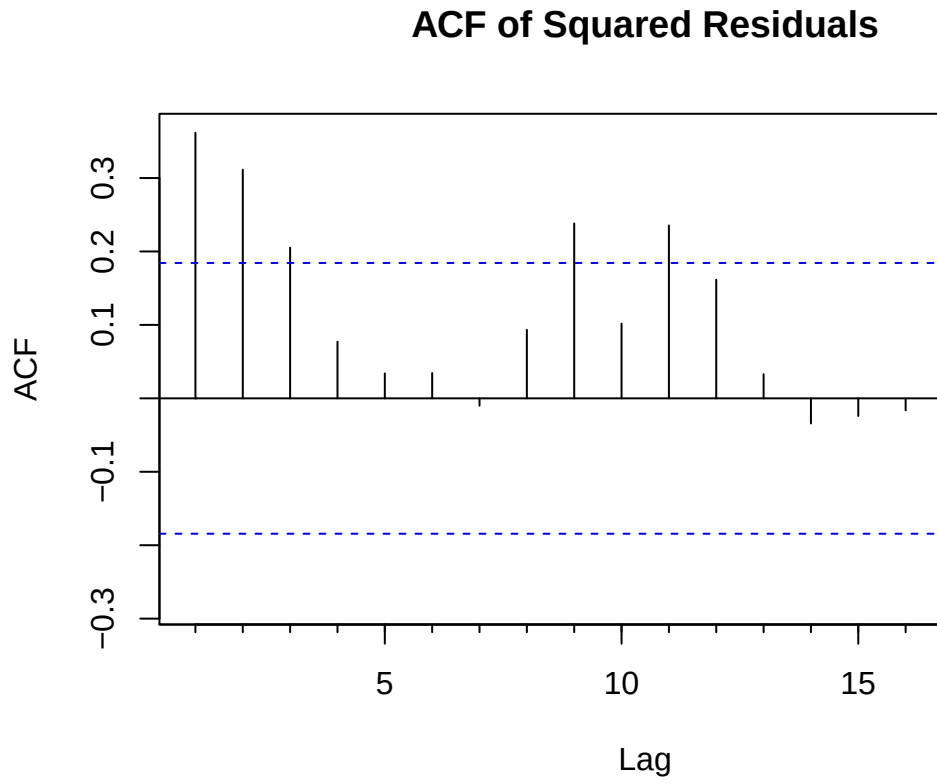
```
# Calculate returns
returns <- diff(log(retail_sales_data$RSXFSN))

# Square the returns
squared_returns <- returns^2

# Create a data frame for plotting
squared_returns_data <- data.frame(DATE = retail_sales_data$DATE[-1], Squared_Returns = squared_returns)

# Plot the squared returns
library(ggplot2)
ggplot(data = squared_returns_data, aes(x = DATE, y = Squared_Returns)) +
  geom_line(color = "blue") +
  labs(title = "Squared Returns of Monthly Retail Sales",
       x = "Date",
       y = "Squared Returns") +
  theme_minimal()
```



Investigating Squared Residuals to Justify GARCH Model

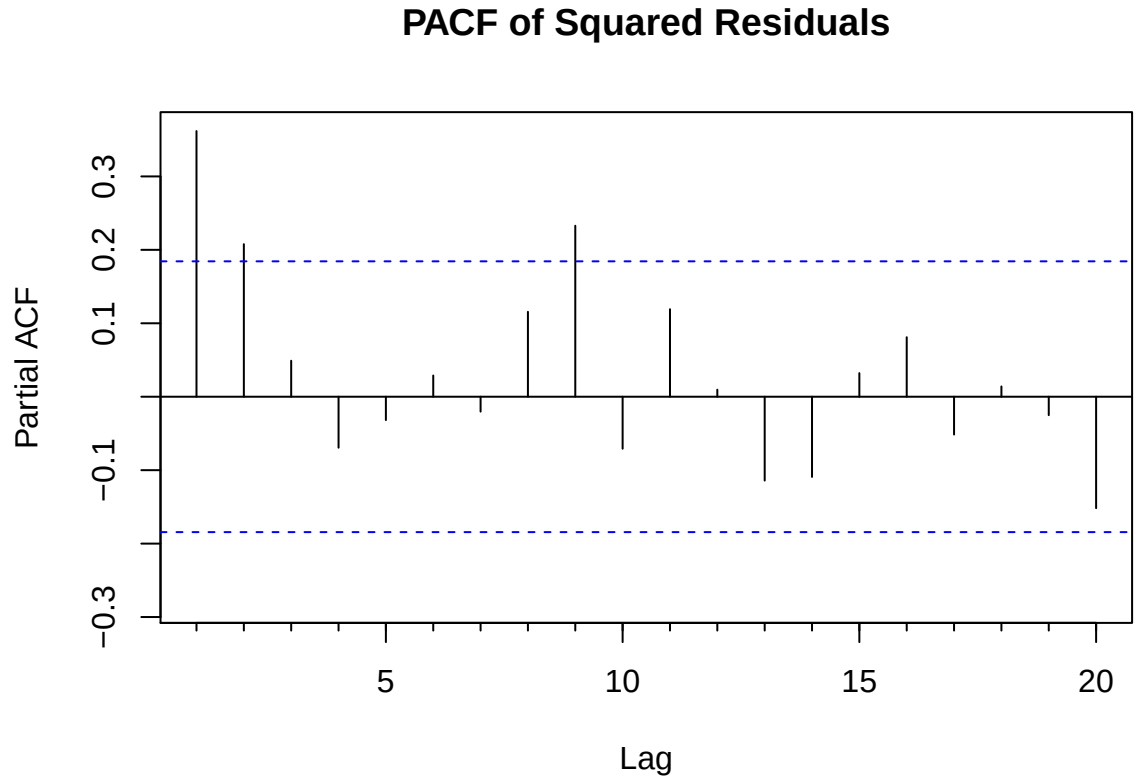
The squared returns of the monthly sales data shows much periodic volatility, which indicates that the data is a strong candidate for the GARCH model.

```
# Calculate residuals
arima_residuals <- residuals(adjusted_arima_model_2)

# Square the residuals to focus on volatility
squared_arima_residuals <- arima_residuals^2

par(mar=c(5, 5, 4, 2) + 0.1)
# Plot ACF of squared residuals
Acf(squared_arima_residuals, main="ACF of Squared Residuals")
```



ACF and PACF of Squared Residuals

```
# Plot PACF of squared residuals
Pacf(squared_arima_residuals, main="PACF of Squared Residuals")
```



The ACF of the squared residuals shows significant autocorrelation at various lags, suggesting periods of high volatility

followed by high volatility and periods of low volatility follow periods of low volatility. This pattern is indicative of volatility clustering, a characteristic of financial time series data. The presence of significant autocorrelation in squared residuals also suggests nonlinearity in variance, highlighting a need for models like GARCH.

```
combined_xts <- xts(combined_diff_data, order.by = as.Date(time(combined_diff_data)))

# Fit the SARIMA model
# Extract residuals from the SARIMA model
sarima_residuals <- residuals(adjusted_arima_model_2)

# Extract dates from the combined_xts
dates <- index(combined_xts)

# Convert SARIMA residuals to xts format
sarima_residuals_xts <- xts(sarima_residuals, order.by = dates)

# Fit a GARCH model on the SARIMA residuals
# Define a simple GARCH(1,1) model
garch_spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "norm"
)

# Fit the GARCH model on SARIMA residuals
garch_fit <- ugarchfit(spec = garch_spec, data = sarima_residuals_xts)
```

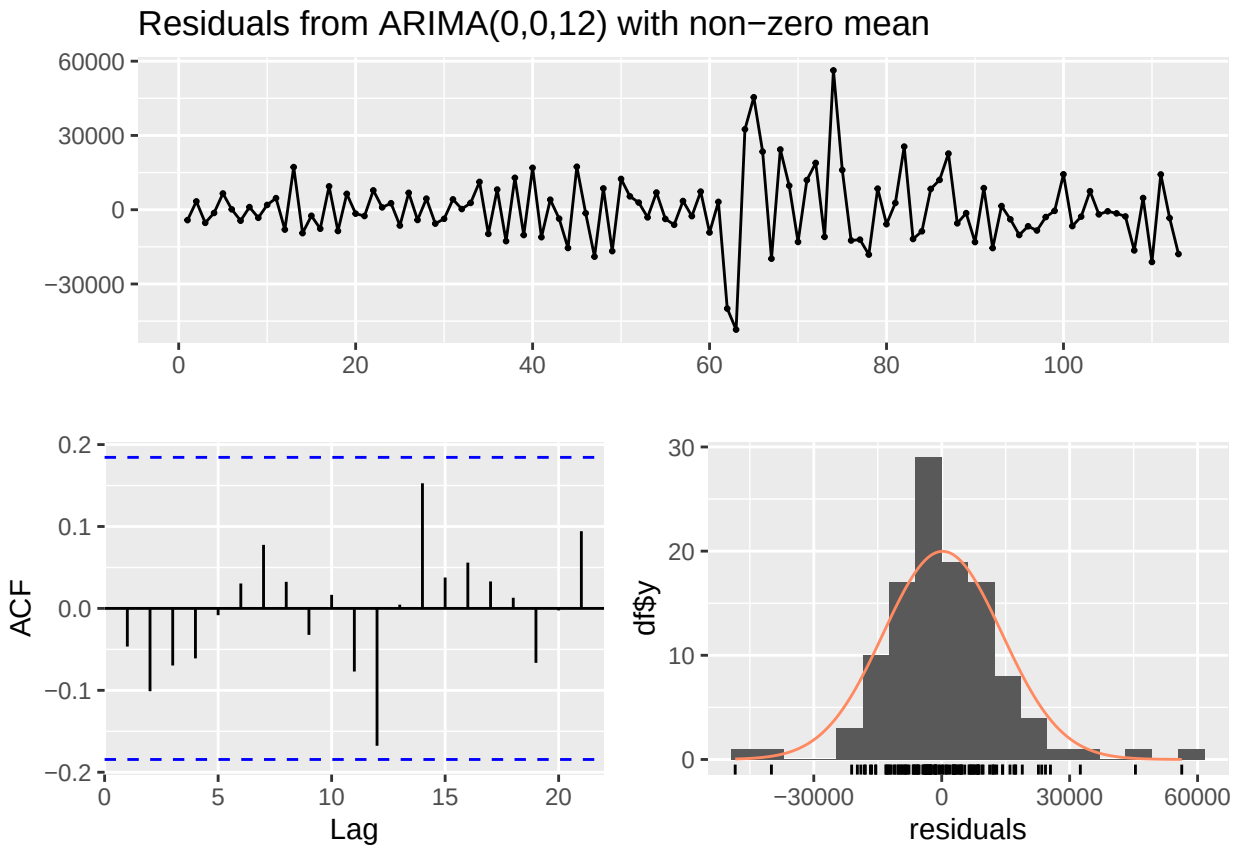
Fitting the GARCH Model to seasonally differenced data plus an additional order of differencing

```
## Warning in .makefitmodel(garchmodel = "sGARCH", f = .sgarchLLH, T = T, m = m, :
## rugarch-->warning: failed to invert hessian
```

```
# Summarize GARCH model fit
summary(garch_fit)
```

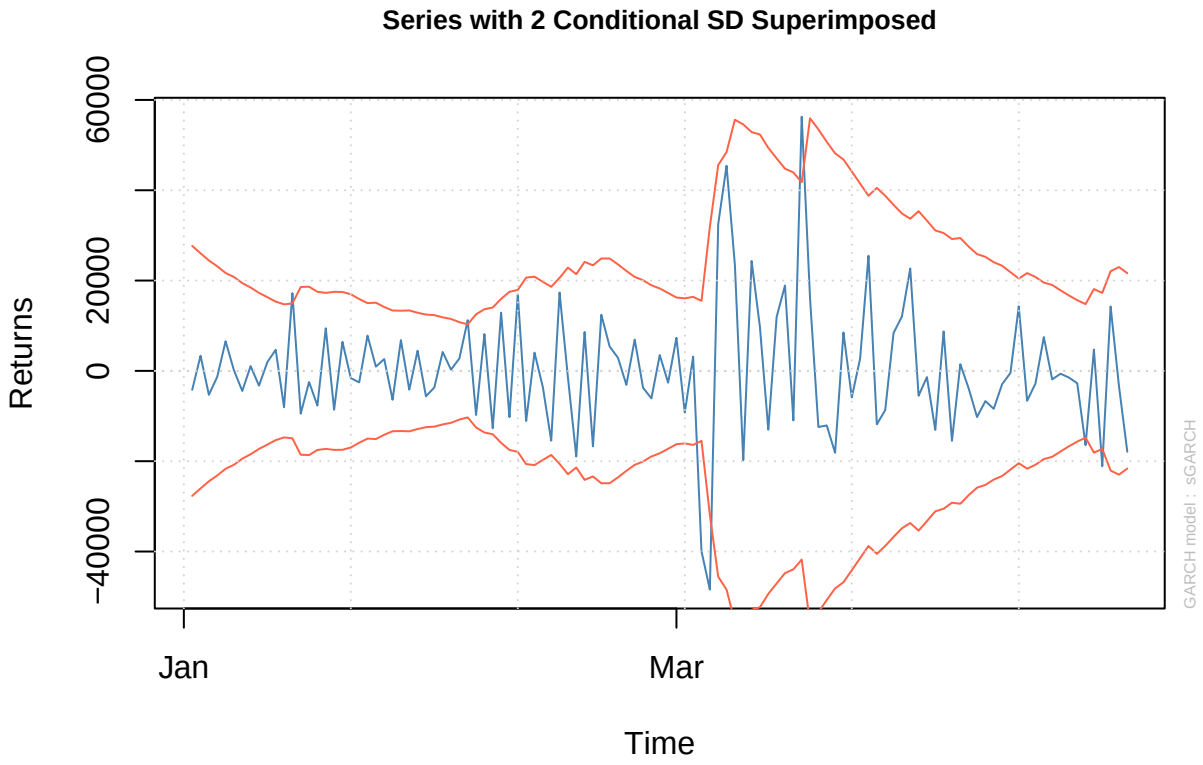
```
##    Length     Class      Mode
##         1 uGARCHfit        S4
```

```
# Evaluate the Combined Model
# Check residuals of the SARIMA model
checkresiduals(adjusted_arima_model_2)
```



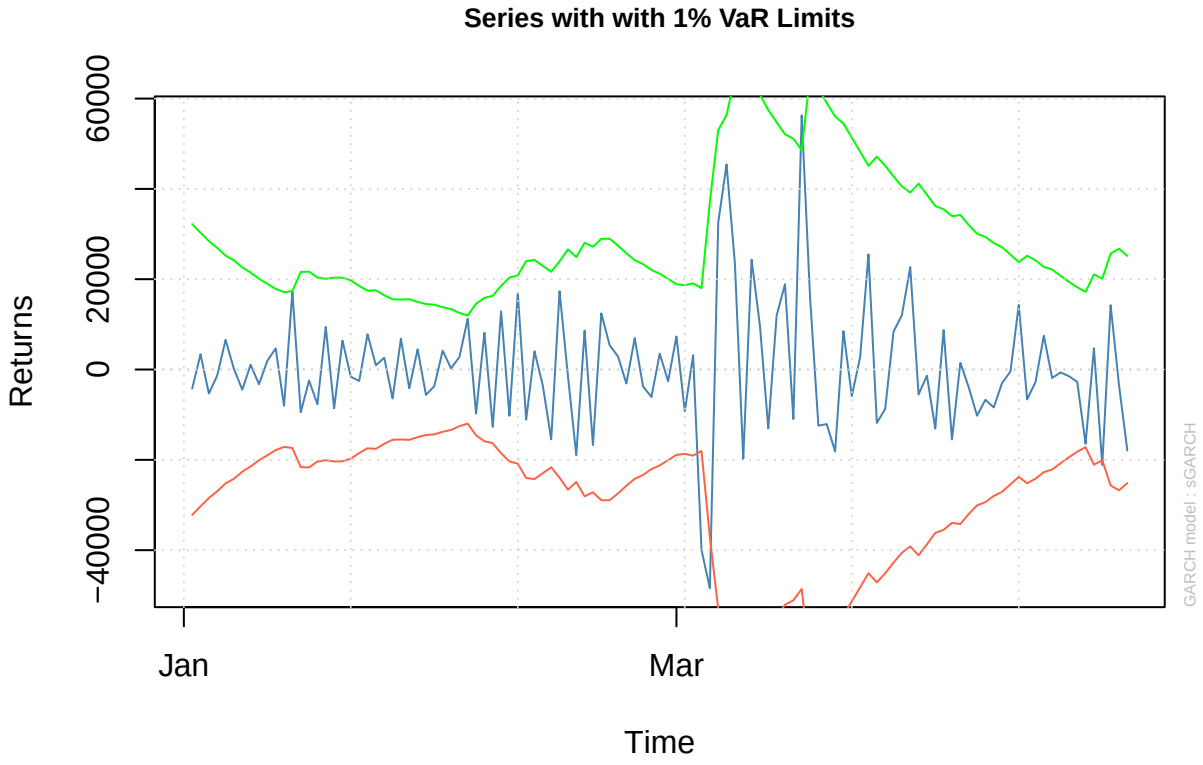

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,12) with non-zero mean
## Q* = 11.244, df = 3, p-value = 0.01048
##
## Model df: 12.   Total lags used: 15
```

```
# Plot diagnostic plots for GARCH model
# Plot diagnostics
plot(garch_fit, which = 1)
```

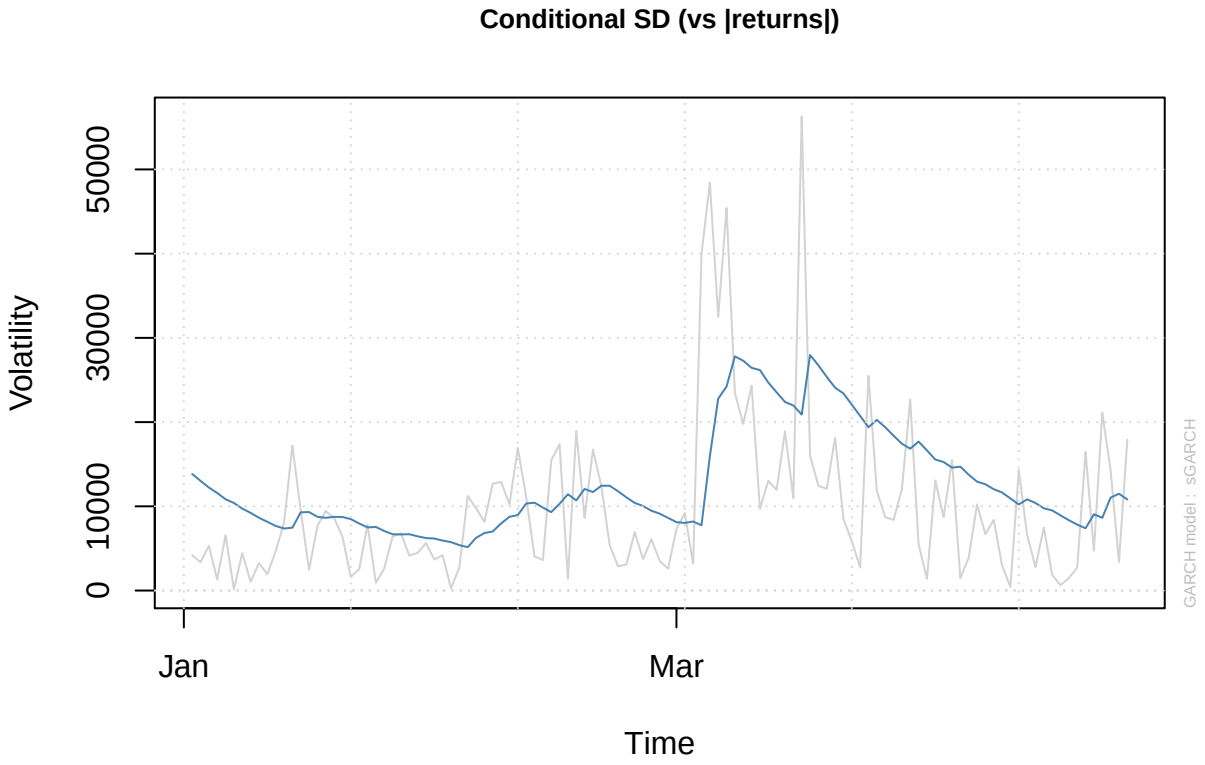



```
plot(garch_fit, which = 2)
```

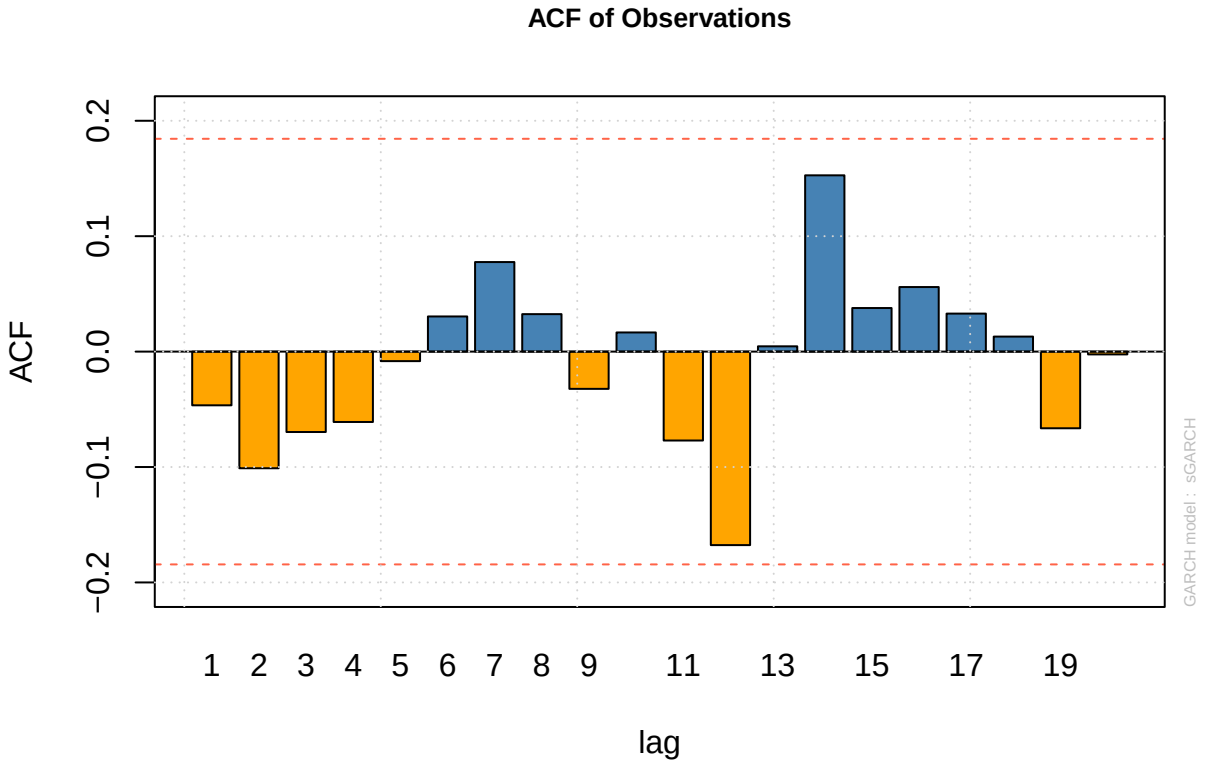
```
##
## please wait...calculating quantiles...
```



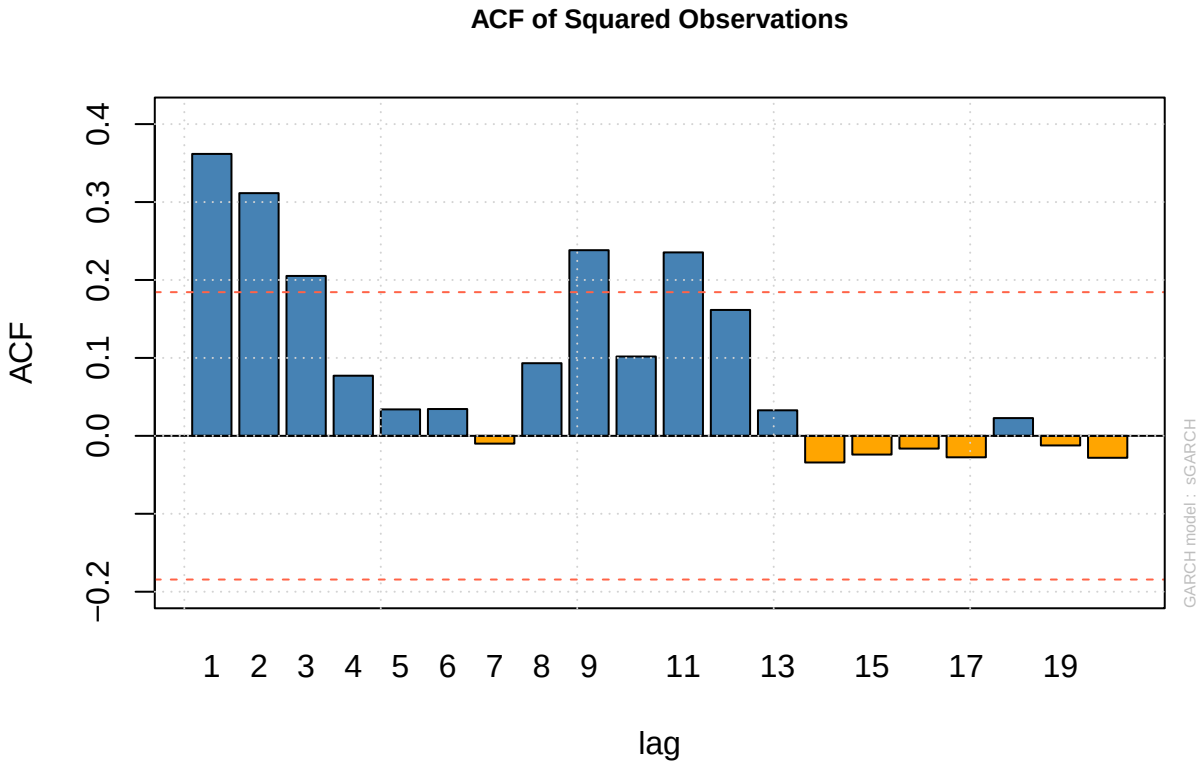
```
plot(garch_fit, which = 3)
```



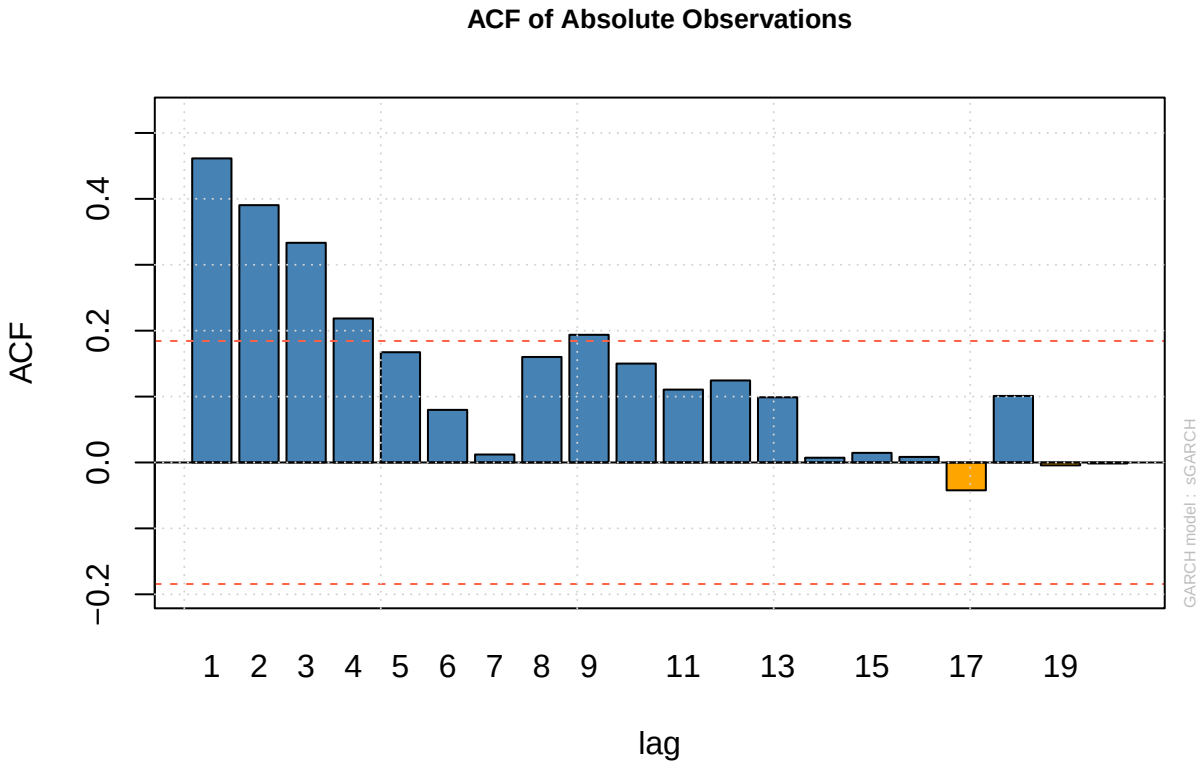

```
plot(garch_fit, which = 4)
```



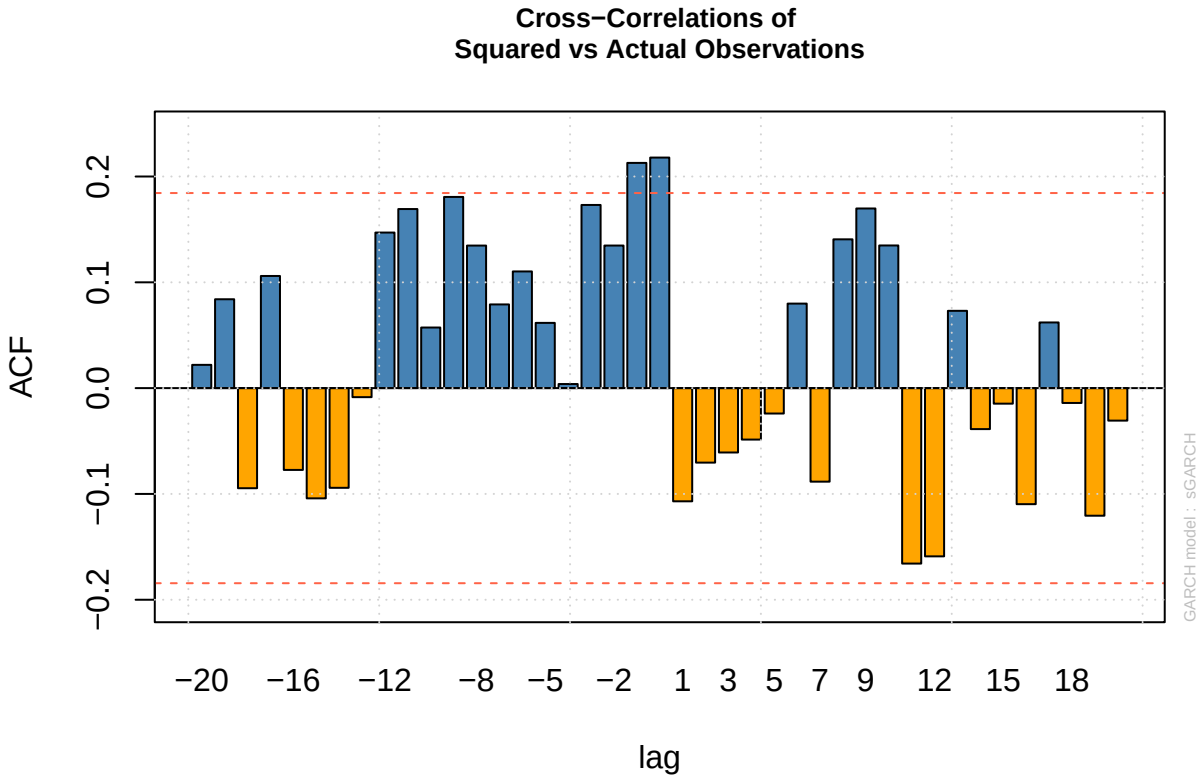
```
plot(garch_fit, which = 5)
```



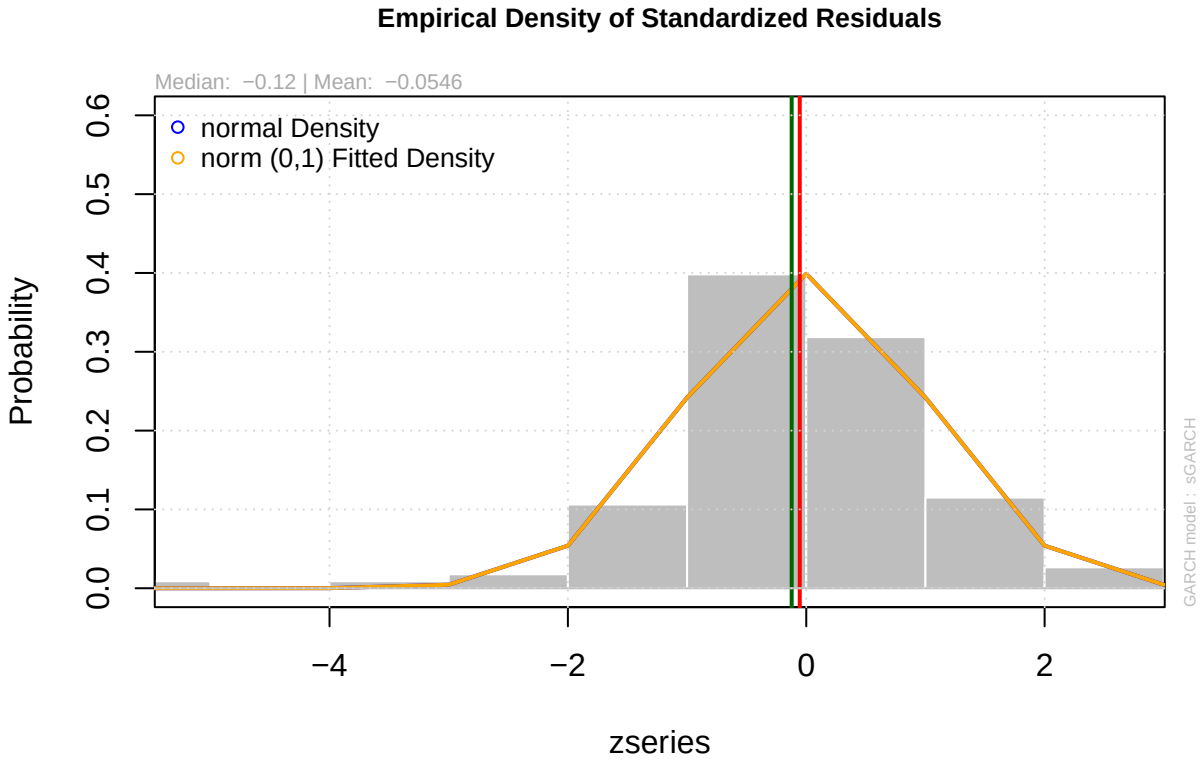

```
plot(garch_fit, which = 6)
```



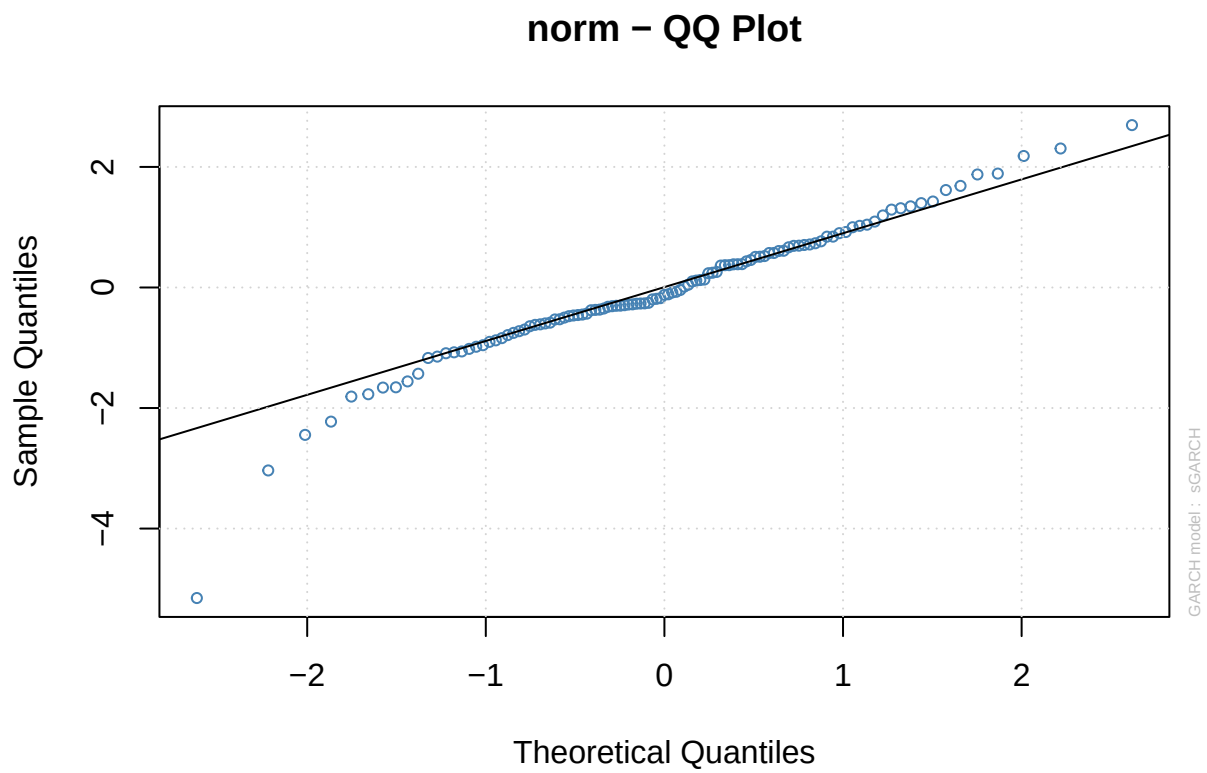
```
plot(garch_fit, which = 7)
```



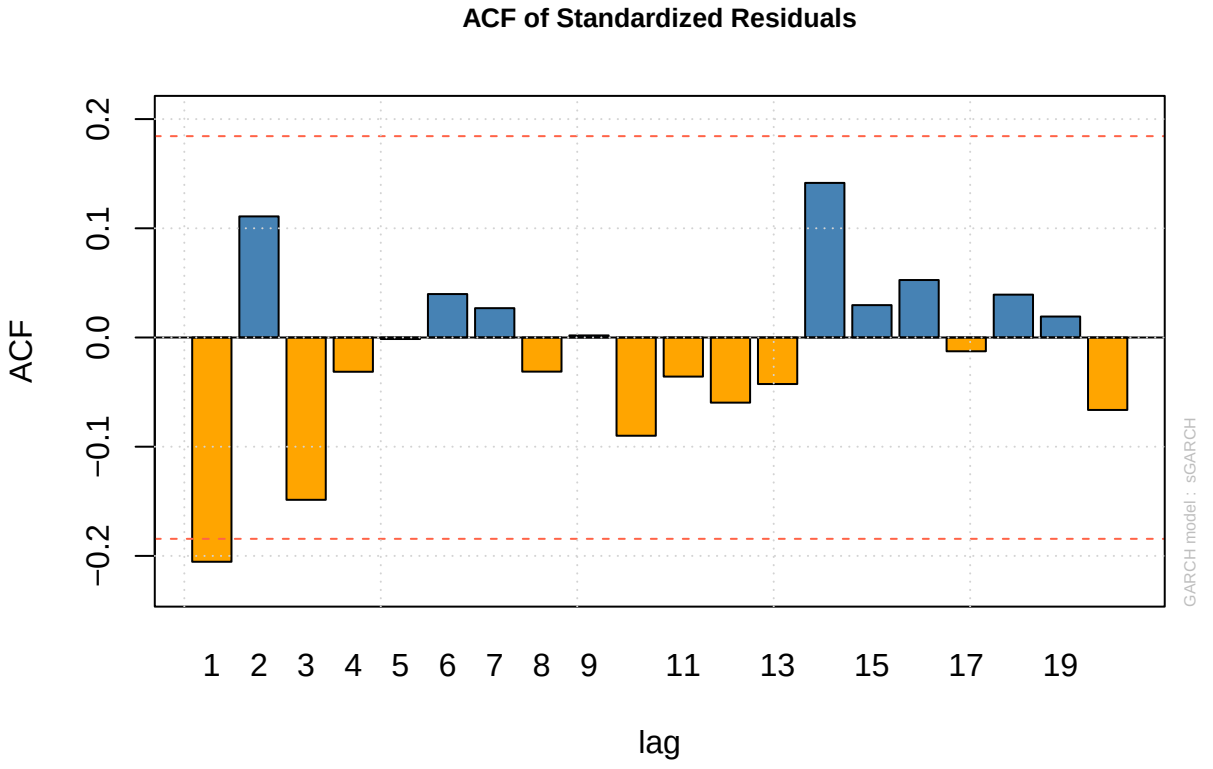

```
plot(garch_fit, which = 8)
```



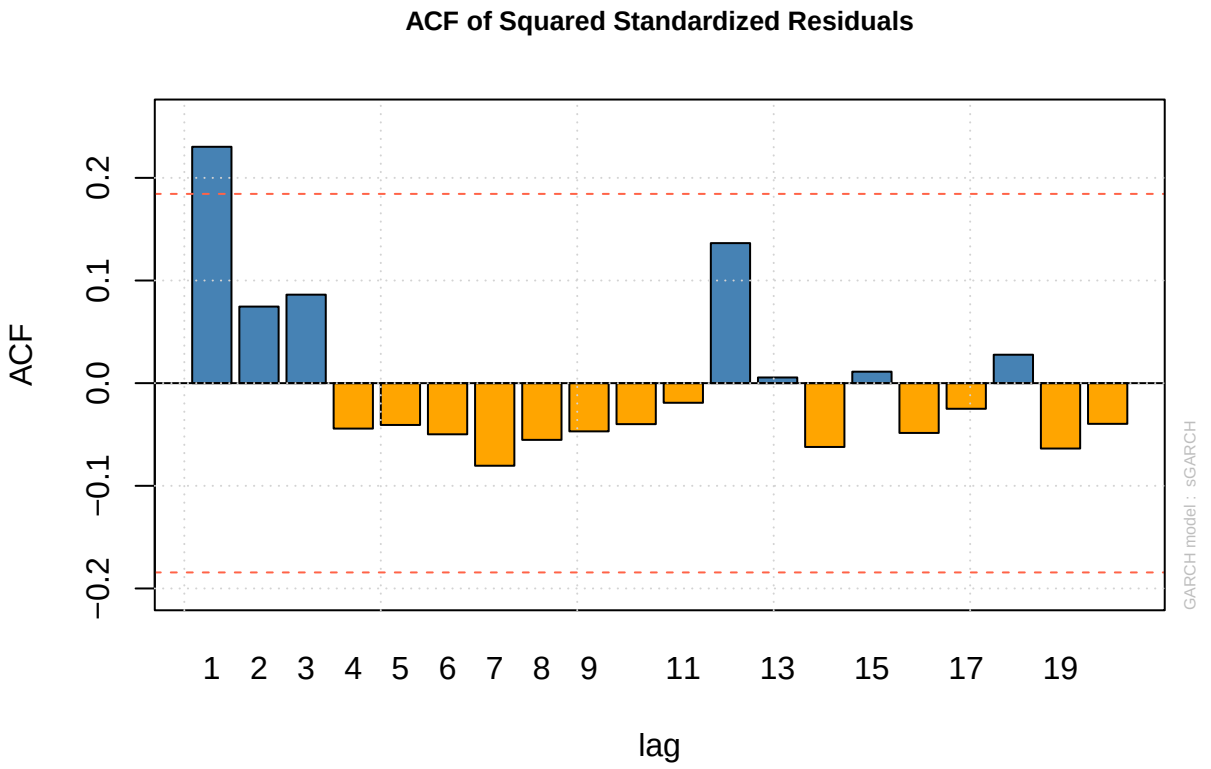
```
plot(garch_fit, which = 9)
```



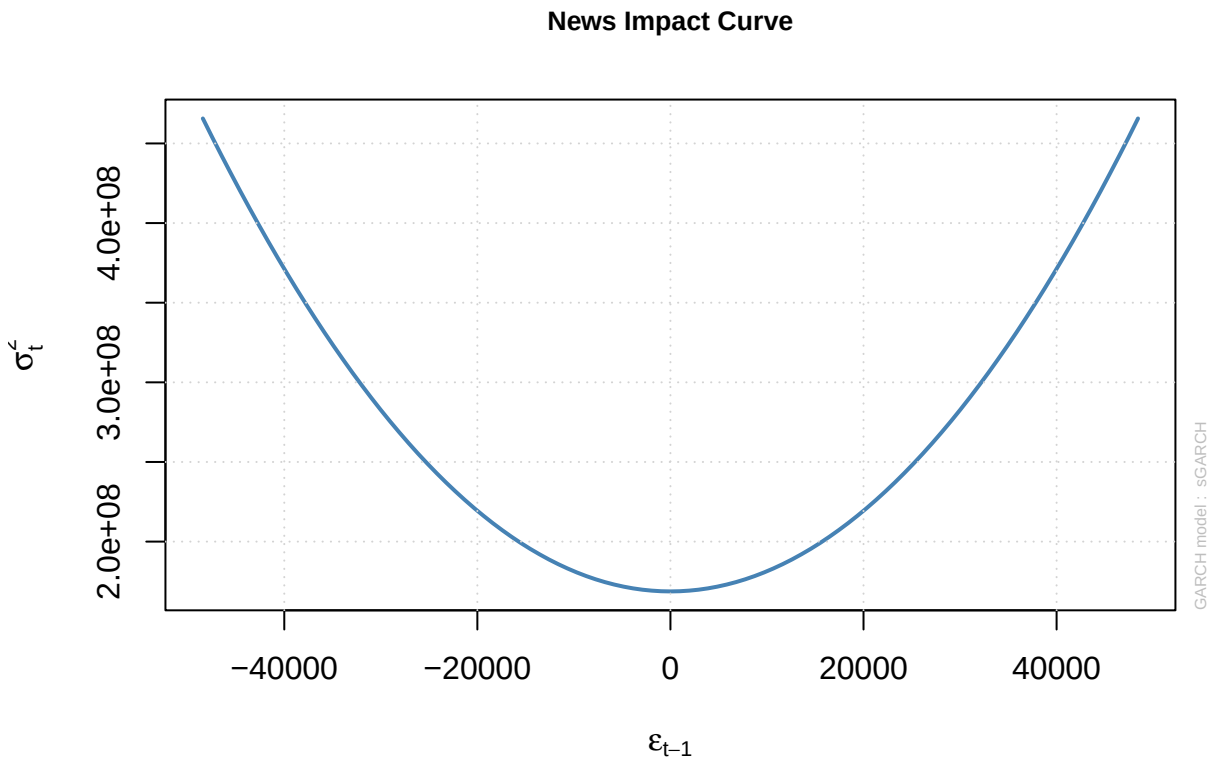

```
plot(garch_fit, which = 10)
```



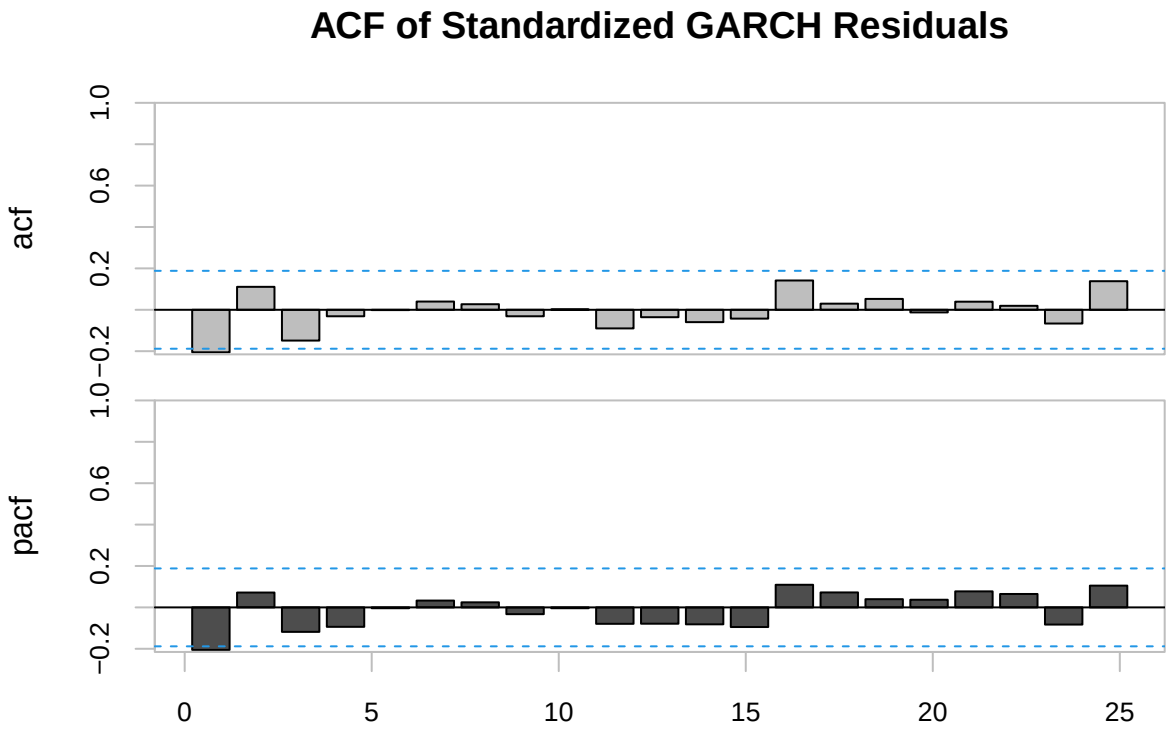
```
plot(garch_fit, which = 11)
```



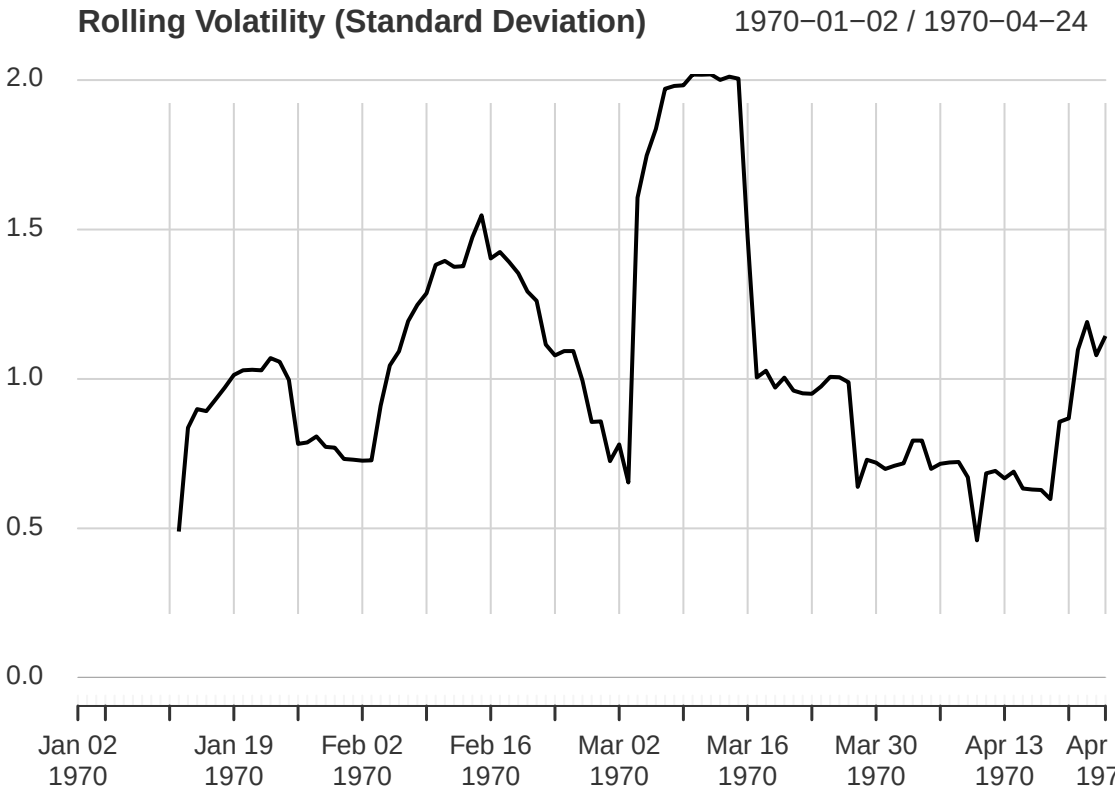

```
plot(garch_fit, which = 12)
```



```r
# Check ACF of standardized residuals from the GARCH model
garch_residuals <- residuals(garch_fit, standardize = TRUE)

# Convert GARCH residuals to xts format, adjust for any differencing
garch_residuals_xts <- xts(garch_residuals, order.by = dates[1:length(garch_residuals)])  # Adjust indes

# Plot ACF using PerformanceAnalytics
chart.ACFplus(garch_residuals_xts, main = "ACF of Standardized GARCH Residuals")
```

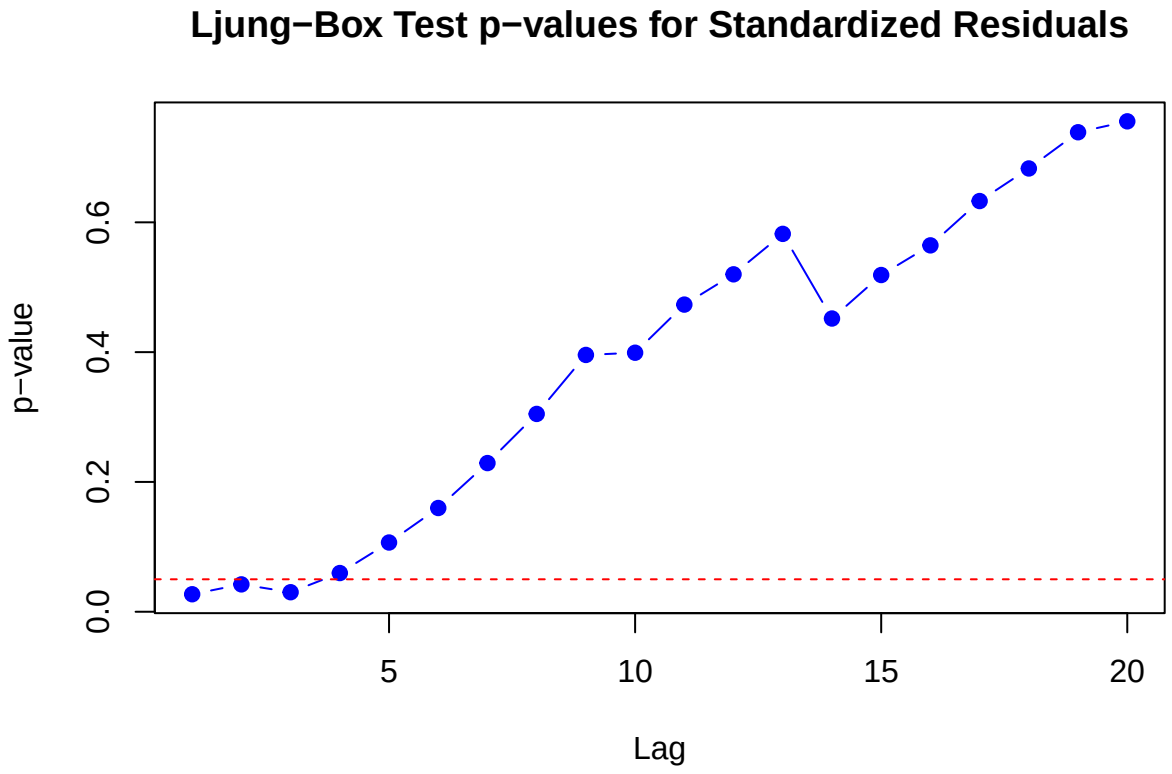



```r
# Plot rolling volatility to assess dynamic volatility over time
chart.RollingPerformance(garch_residuals_xts, width = 12, FUN = "sd", main = "Rolling Volatility (Standa
```



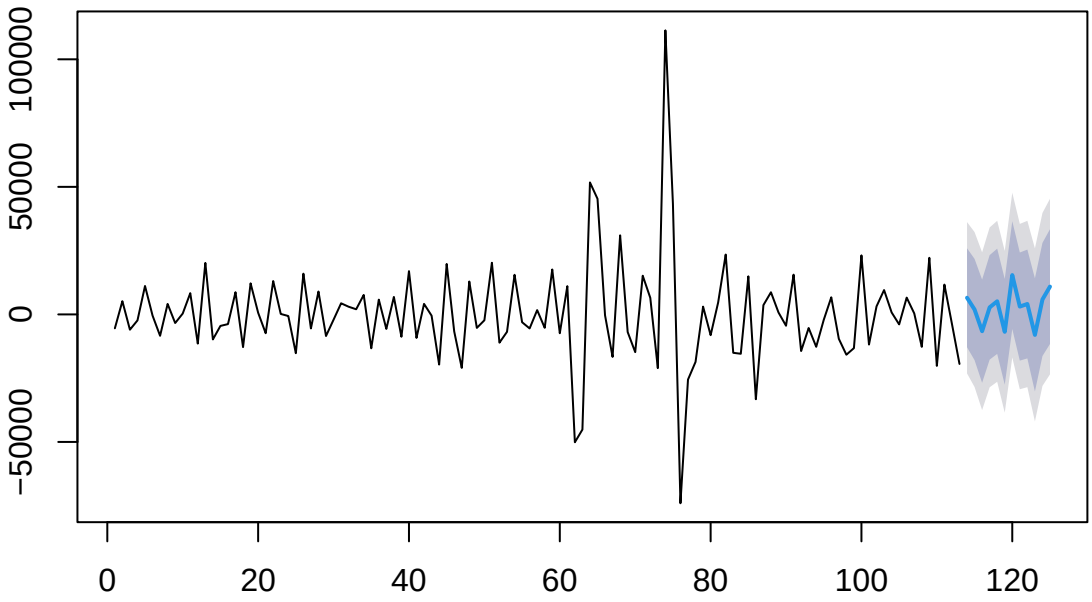
The ACF and PACF of the Standardized Residuals show no significant spikes except for the first, indicating that the residuals are uncorrelated and resemble white noise. In the QQ plot, the points fall close to the 45-degree line along the center and top-right-most portion of the graph, albeit with significant deviations at the bottom tail.

```
# Set the maximum number of lags for the Ljung-Box test
max_lags <- 20

# Initialize a vector to store p-values
p_values <- numeric(max_lags)

# Calculate the Ljung-Box test p-values for each lag
for (lag in 1:max_lags) {
  lb_test <- Box.test(garch_residuals, lag = lag, type = "Ljung-Box", fitdf = 0)
  p_values[lag] <- lb_test$p.value
}

# Create a plot of p-values
plot(1:max_lags, p_values, type = "b", pch = 19, col = "blue",
     xlab = "Lag", ylab = "p-value",
     main = "Ljung-Box Test p-values for Standardized Residuals")
abline(h = 0.05, col = "red", lty = 2)  # Add a line at the 0.05 significance level
```




```r
# Creating a dataframe for plotting
residuals_df <- data.frame(Residuals = as.numeric(garch_residuals))
```

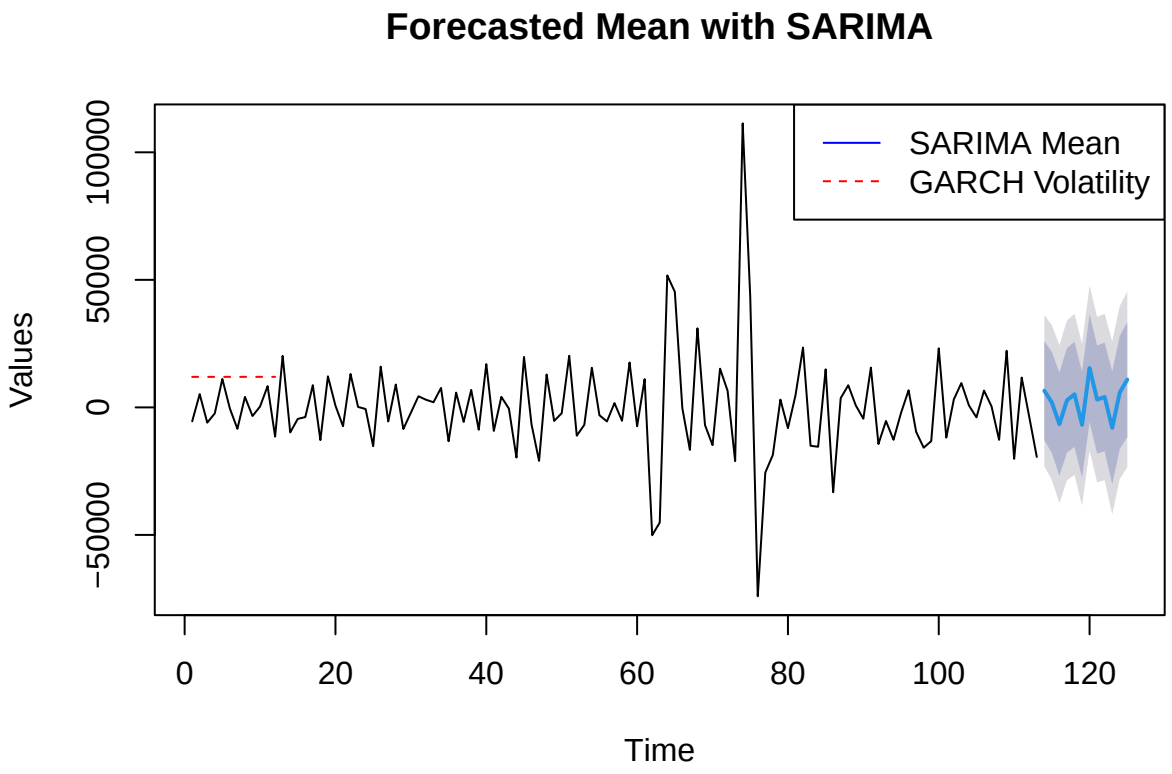
```r
# Specify the number of periods you want to forecast
forecast_horizon <- 12

# Forecast future values using the SARIMA model
sarima_forecast <- forecast(adjusted_arima_model_2, h = forecast_horizon)

# Plot SARIMA forecast
plot(sarima_forecast, main = "SARIMA Forecast")
```




```r
# Step 4: Forecast using the GARCH model
garch_forecast <- ugarchforecast(garch_fit, n.ahead = forecast_horizon)

# Extract forecasted volatility (standard deviation) from the GARCH model
volatility_forecast <- sigma(garch_forecast)

# Print volatility forecast
print(volatility_forecast)
```

```
##      1970-04-24
## T+1    11949.97
## T+2    11952.08
## T+3    11954.19
## T+4    11956.29
## T+5    11958.39
## T+6    11960.49
## T+7    11962.58
## T+8    11964.68
## T+9    11966.77
## T+10   11968.86
## T+11   11970.94
## T+12   11973.03
```

```r
# Combine SARIMA forecast with GARCH volatility forecast
plot(sarima_forecast, main = "Forecasted Mean with SARIMA", xlab = "Time", ylab = "Values")
lines(volatility_forecast, col = "red", lty = 2)

legend("topright", legend = c("SARIMA Mean", "GARCH Volatility"), col = c("blue", "red"), lty = 1:2)
```



AAPL Monthly Data 2016-2024

Load & Inspect the Data

```
# Load the data
aapl_monthly_data <- read.csv("~/Documents/GitHub/MA-641-Course-Project/AAPL_Monthly2016.csv")

# Convert the date column to Date type
aapl_monthly_data$Date <- as.Date(aapl_monthly_data$Date, format="%Y-%m-%d")

# Inspect the data
head(aapl_monthly_data)
```

```
##         Date    Open    High     Low   Close Adj.Close     Volume
## 1 2016-01-01 25.6525 26.4625 23.0975 24.3350  22.09621 5087392000
## 2 2016-02-01 24.1175 24.7225 23.1475 24.1725  21.94866 3243450400
## 3 2016-03-01 24.4125 27.6050 24.3550 27.2475  24.87501 2984198400
## 4 2016-04-01 27.1950 28.0975 23.1275 23.4350  21.39447 3489534800
## 5 2016-05-01 23.4925 25.1825 22.3675 24.9650  22.79125 3602686000
## 6 2016-06-01 24.7550 25.4725 22.8750 23.9000  21.95182 3117990800
```

```
summary(aapl_monthly_data)
```

```
##       Date                 Open             High             Low
##  Min.   :2016-01-01   Min.   : 23.49   Min.   : 24.72   Min.   : 22.37
##  1st Qu.:2018-02-01   1st Qu.: 41.74   1st Qu.: 44.30   1st Qu.: 40.16
```

```
##  Median :2020-03-01   Median : 71.56   Median : 81.06   Median : 64.09
##  Mean   :2020-03-01   Mean   : 94.47   Mean   :101.04   Mean   : 88.97
##  3rd Qu.:2022-04-01   3rd Qu.:148.99   3rd Qu.:157.50   3rd Qu.:138.27
##  Max.   :2024-05-01   Max.   :196.24   Max.   :199.62   Max.   :187.45
##      Close          Adj.Close          Volume
##  Min.   : 23.43   Min.   : 21.39   Min.   :9.697e+08
##  1st Qu.: 41.95   1st Qu.: 39.72   1st Qu.:1.676e+09
##  Median : 73.45   Median : 71.54   Median :2.240e+09
##  Mean   : 95.93   Mean   : 93.98   Mean   :2.320e+09
##  3rd Qu.:149.80   3rd Qu.:147.50   3rd Qu.:2.801e+09
##  Max.   :196.45   Max.   :195.41   Max.   :6.280e+09
```

About the Data:

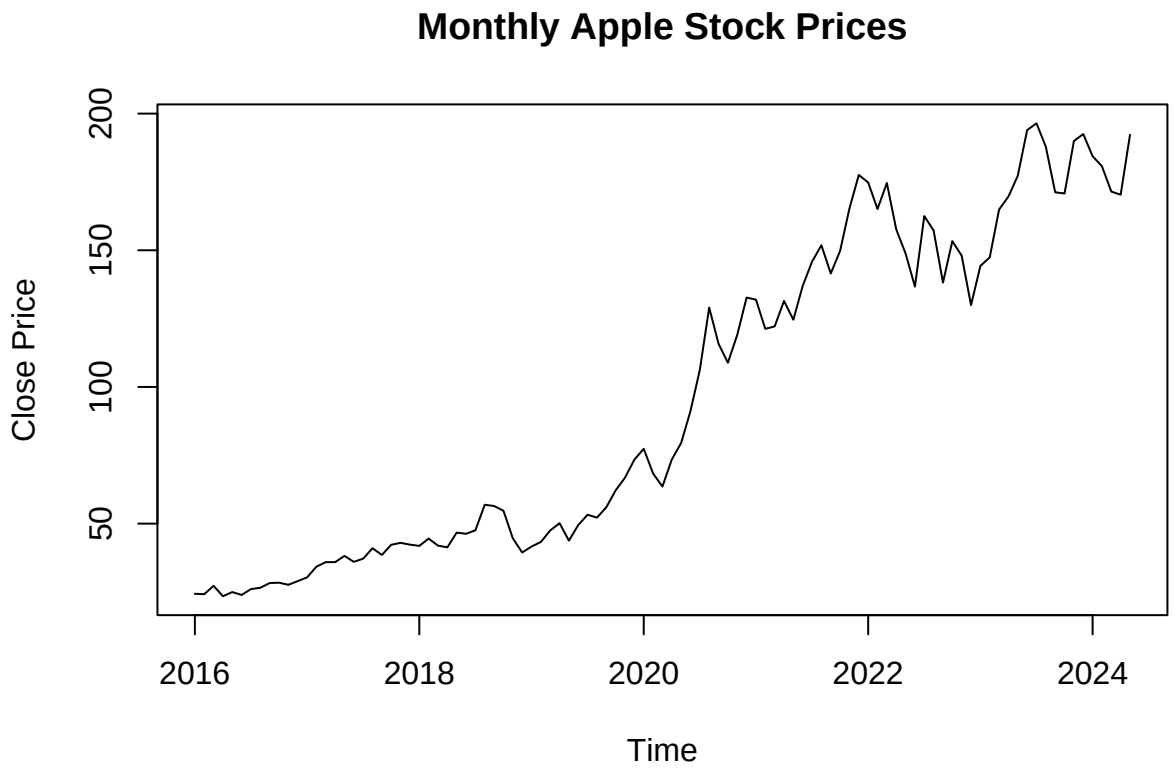
- 101 data points spanning from 01/01/16 to 05/01/24
- Data includes the monthly open, high, low, close, and adjusted close prices of the apple stock

Create a Time Series Object

```
# Create a time series object
aaplmonthly_ts <- ts(aapl_monthly_data$Close, start=c(2016, 01), end = c(2024, 05), frequency=12)
```

Descripvtive Analysis

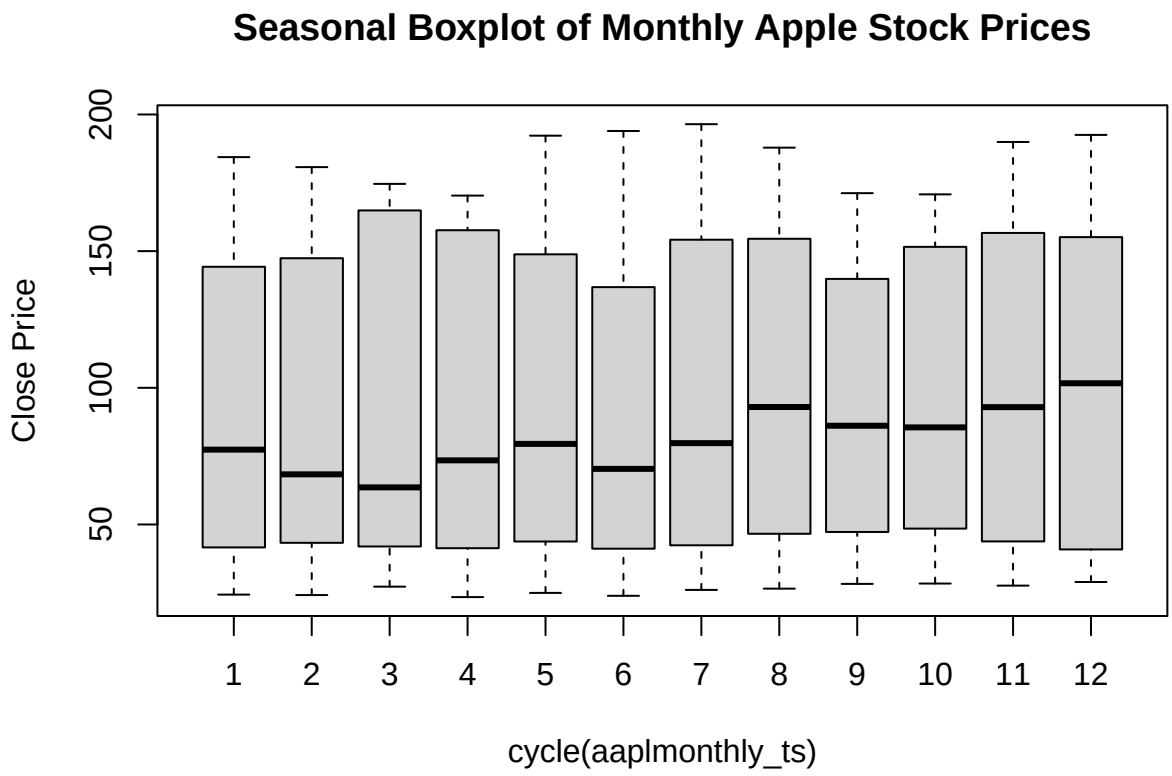
```
# Descriptive Analysis
plot(aaplmonthly_ts, main="Monthly Apple Stock Prices", ylab="Close Price", xlab="Time")
```



```
summary(aaplmonthly_ts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   23.43   41.95   73.45   95.93  149.80  196.45
```

```
boxplot(aaplmonthly_ts ~ cycle(aaplmonthly_ts), main="Seasonal Boxplot of Monthly Apple Stock Prices", y
```



Time Series Plot:

- There is a clear upward trend in Apple stock prices over the period. The prices show a substantial increase, particularly starting around 2019.
- There is visible volatility in the stock prices, with fluctuations becoming more pronounced in the later years.
- The increased volatility may imply higher risk for investors, as the stock prices have larger swings.

Summary:

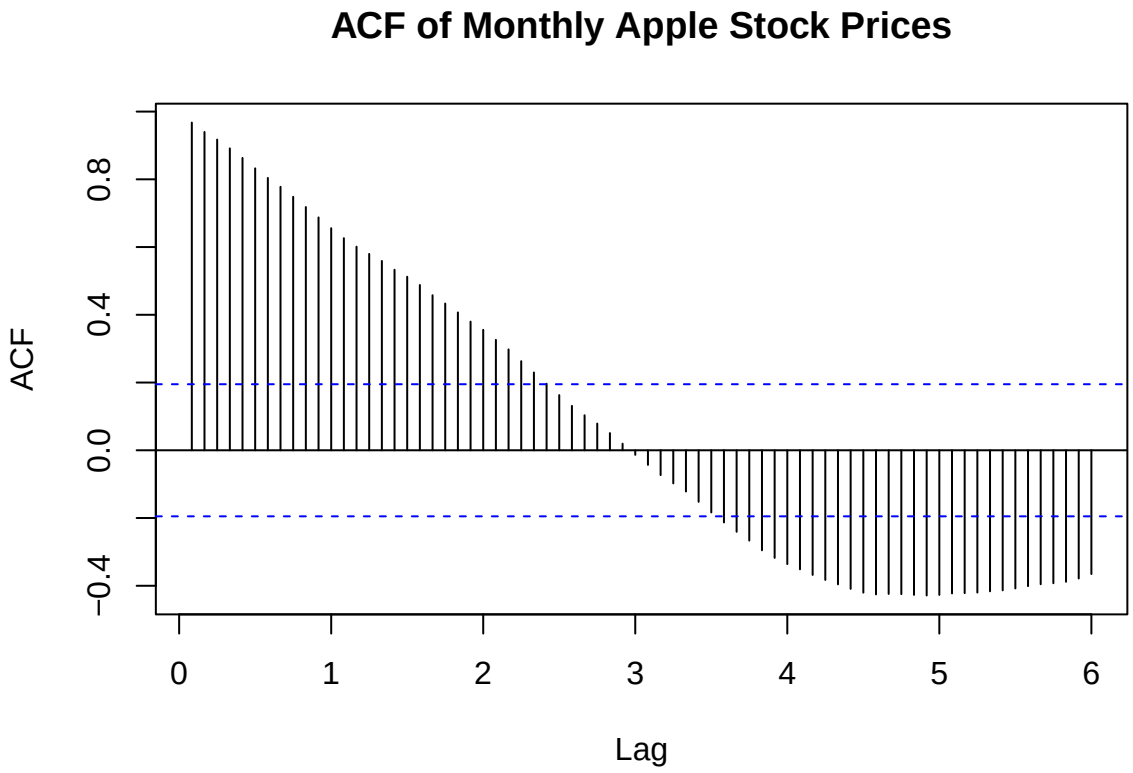
- The mean and median values suggest that the central tendency of the stock prices is around 73 to 96.
- The range indicates that the stock price has varied significantly over the period.

Seasonal Boxplot:

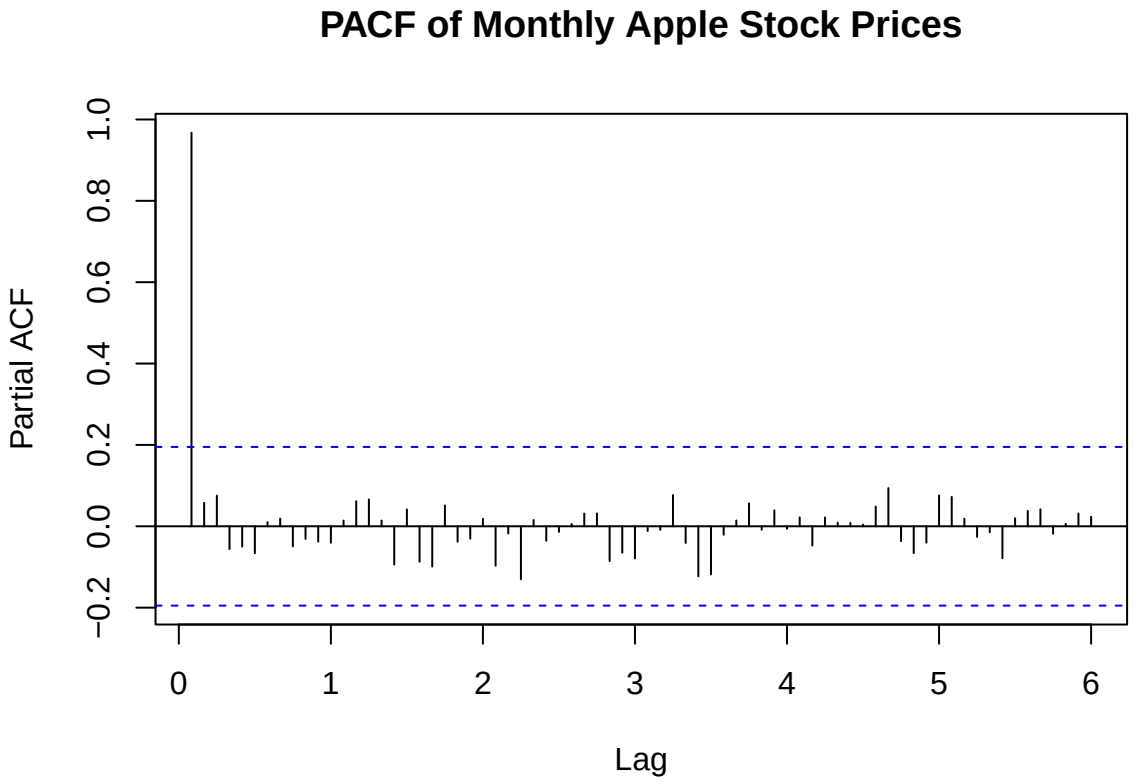
- The presence of seasonality suggests that certain months tend to have higher or lower stock prices consistently, which can be crucial for seasonal trading strategies.
- The seasonal boxplot reveals minor monthly patterns and variability, suggesting that seasonality should be considered in trading strategies and risk

ACF, PACF, & EACF Plots

```
# ACF and PACF Plots
par(mar=c(5, 5, 4, 2) + 0.1)
acf(aaplmonthly_ts, main="ACF of Monthly Apple Stock Prices", lag.max = 72)
```




```
pacf(aaplmonthly_ts, main="PACF of Monthly Apple Stock Prices", lag.max = 72)
```




```
eacf(aaplmonthly_ts)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x x x x x x x x  x  x  x
## 1 o x o o o o o o o o o  o  o  o
## 2 x x o o o o o o x o o  o  o  o
## 3 o o o o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 o o o o o o o o o o o  o  o  o
## 6 o o o o o o o o o o o  o  o  o
## 7 o o o x o o o o o o o  o  o  o
```

ACF Plot:

- The gradual decay in the ACF indicates the presence of a trend component in the time series. The series is likely non-stationary.
- Significant autocorrelations suggest that the data is not random and past values can help predict future values.
- A high degree of positive autocorrelation at the first few lags implies momentum in the stock prices, which is common in financial time series.

PACF Plot:

- The sharp drop after the first lag in the PACF suggests that an AR(1) model may be appropriate for capturing the relationship in the data.
- The significant first lag indicates that the immediate past value has a strong influence on the current value, while the influence of values further in the past diminishes quickly.

- This pattern supports the use of a simple autoregressive model, as the complexity beyond the first lag does not add much explanatory power.

ADF Test

```
# Augmented Dickey-Fuller Test
adf_test <- adf.test(aaplmonthly_ts, alternative="stationary")
print(adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  aaplmonthly_ts
## Dickey-Fuller = -2.3411, Lag order = 4, p-value = 0.4353
## alternative hypothesis: stationary
```

- Since the p-value is greater than 0.05, we fail to reject the null hypothesis that the time series has a unit root. This indicates that the series is non-stationary.
- The non-stationarity observed from the ADF test results implies that differencing the time series is necessary to achieve stationarity.

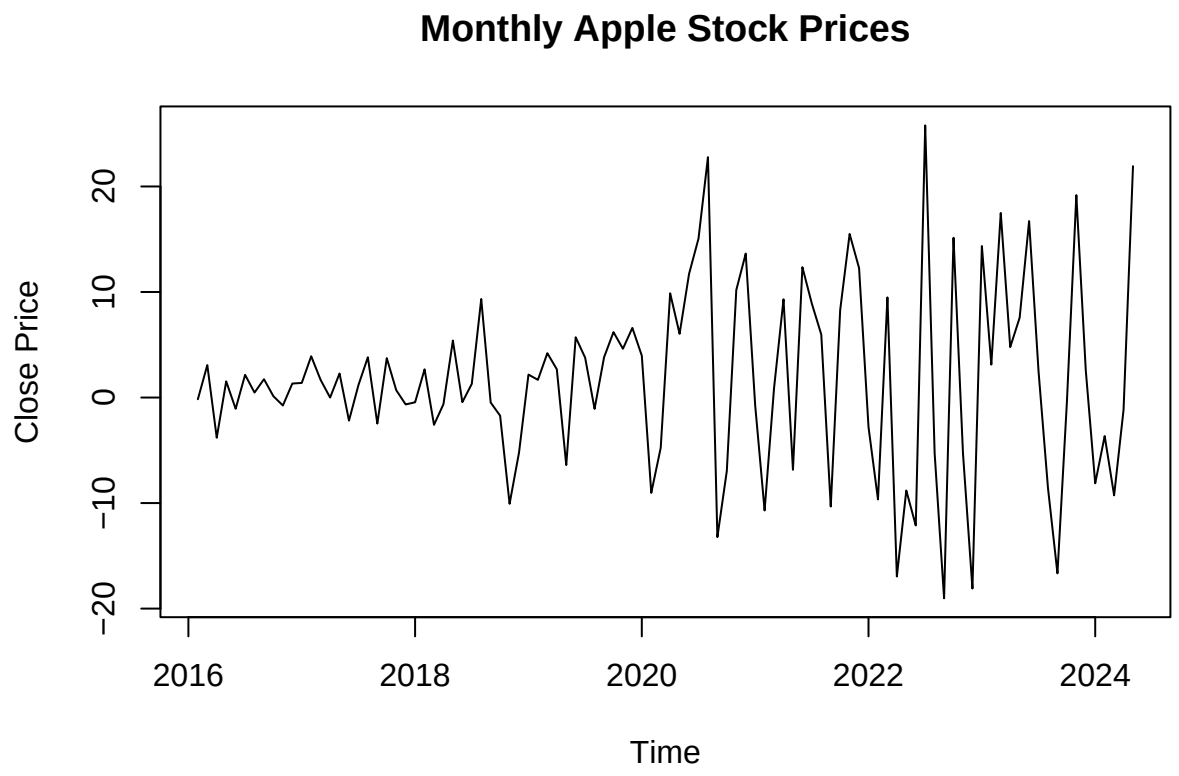
```
# Differencing the series if it is not stationary
if (adf_test$p.value > 0.05) {
  ts_data_diff <- diff(aaplmonthly_ts, differences=1)
  adf_test_diff <- adf.test(ts_data_diff, alternative="stationary")
  print(adf_test_diff)

  # Update the time series data to the differenced series
  aaplmonthly_ts <- ts_data_diff
}
```

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

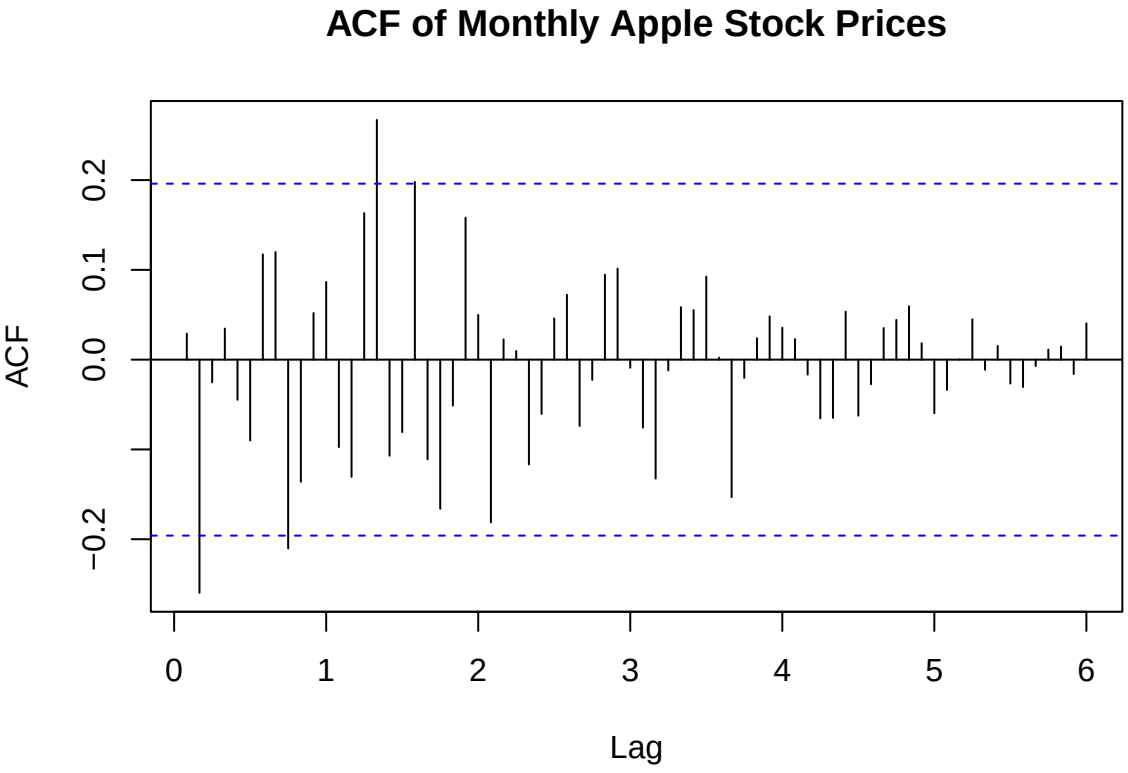
```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts_data_diff
## Dickey-Fuller = -5.0114, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

- Since the p-value is less than 0.05, we reject the null hypothesis that the time series has a unit root. This indicates that the differenced series is stationary.
- With the differenced series being stationary, it is now suitable for fitting ARIMA models.

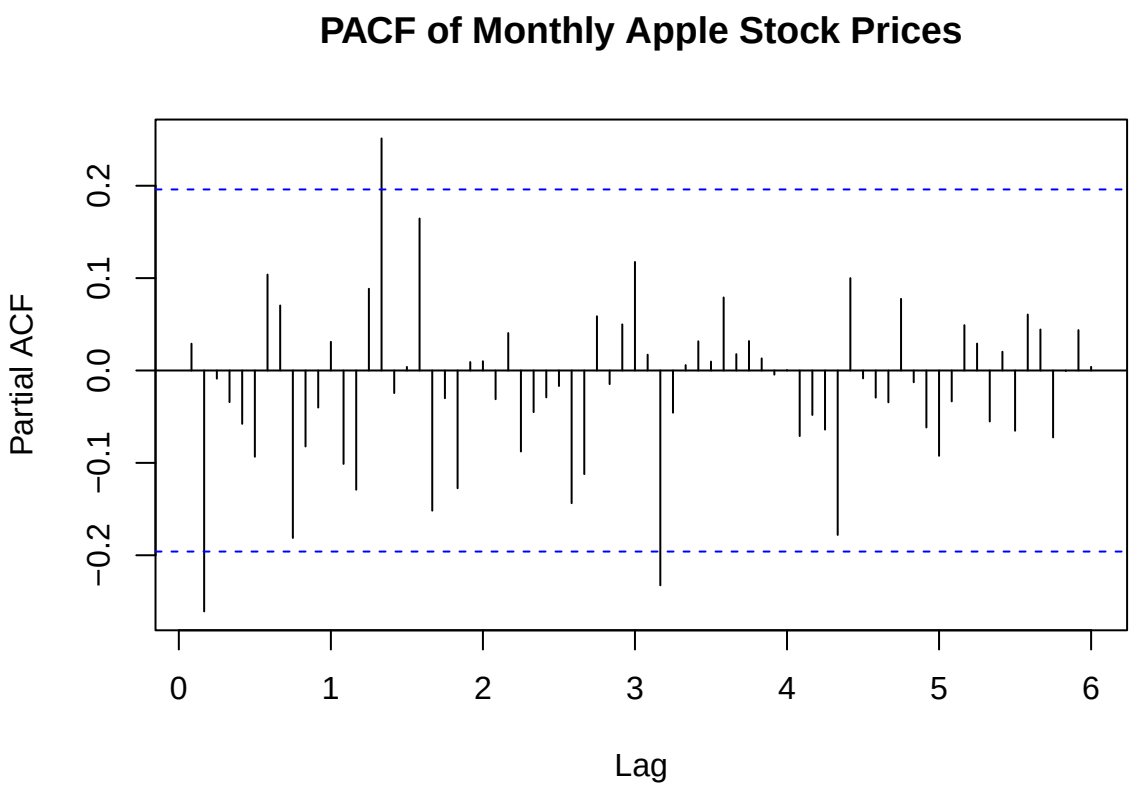
```
# Time Series Plot after Differencing
plot(aaplmonthly_ts, main="Monthly Apple Stock Prices", ylab="Close Price", xlab="Time")
```




```r
# ACF and PACF Plots
par(mar=c(5, 5, 4, 2) + 0.1)
acf(aaplmonthly_ts, main="ACF of Monthly Apple Stock Prices", lag.max = 72)
```



```r
pacf(aaplmonthly_ts, main="PACF of Monthly Apple Stock Prices", lag.max = 72)
```




```r
eacf(aaplmonthly_ts)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o x o o o o o o x o o  o  o  o
## 1 o x o o o o o o x o o  o  o  o
## 2 o o o o o o o o o o o  o  o  o
## 3 o o o o o o o o o o o  o  o  o
## 4 x o o o o o o o o o o  o  o  o
## 5 x o o o o o o o o o o  o  o  o
## 6 x x o o o o o o o o o  o  o  o
## 7 x x o o x o x o o o o  o  o  o
```

- We can see that after differencing the time-series appears to be stationary.

ACF Plot:

- Exponential decay, suggests an AR model.
- Spike at certain lags and then no significant autocorrelation indicates an MA model.
- Lag 1: Strong positive autocorrelation, possibly suggesting an MA(1) component.
- Further Lags: Gradual decay suggests potential AR component, but initial spikes may also indicate an MA component.

PACF Plot:

- Lag 1: A significant spike, suggesting an AR(1) component.
- Further Lags: The lack of significant spikes after the initial ones implies no strong additional AR terms.

EACF Plot:

- ARIMA(1,0,1): Due to the significant spike at lag 1 in both ACF and PACF plots.
- ARIMA(0,0,1): Due to the initial spikes in the ACF and a quickly decaying PACF.
- ARIMA(1,0,0): Due to the AR component suggested by PACF.
- ARIMA(2,0,1) or ARIMA(1,0,2): EACF indicates potential combined AR and MA terms.

Fit AR, MA, and ARMA Models

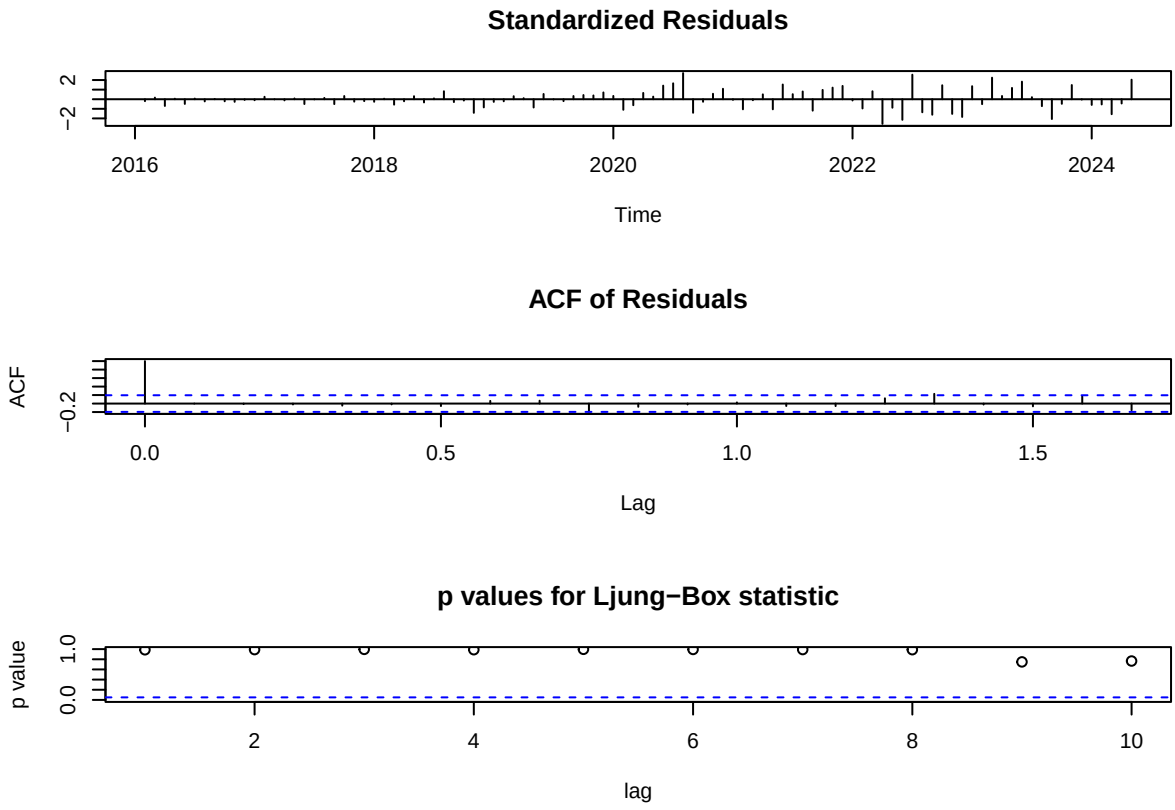
```
# Fit AR model
ar_model <- Arima(aaplmonthly_ts, order=c(2,0,0))
summary(ar_model)
```

AR Model

```
## Series: aaplmonthly_ts
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2    mean
##       0.0409  -0.2708  1.6476
## s.e.  0.0988   0.0980  0.6883
##
## sigma^2 = 73.24:  log likelihood = -355.13
## AIC=718.27   AICc=718.69   BIC=728.69
##
## Training set error measures:
##                        ME     RMSE     MAE      MPE     MAPE      MASE
## Training set -0.000308276 8.428628 6.23919 507.0341 563.7521 0.7036177
##                      ACF1
## Training set -0.001614169
```

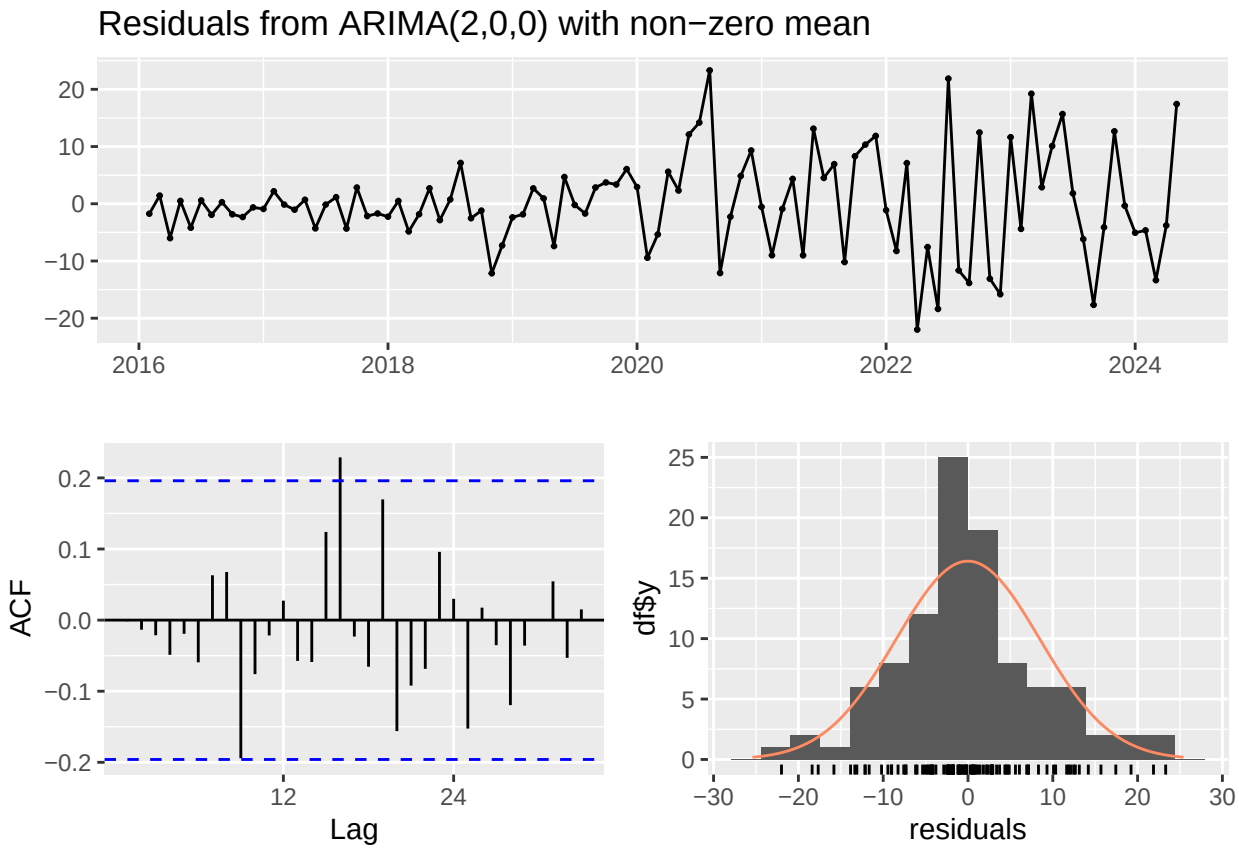
- AR(1) Coefficient=0.0409, is a small positive value close to zero which suggests a weak positive correlation with the immediate past month's value
- AR(2) Coefficient=-0.2708, is a negative value which suggests that the price two months ago has a moderate inverse relationship with the current month's price
- Mean=1.6476, the average level of the series after removing the autoregressive effects
- The AR(1) coefficient is small, while the AR(2) coefficient is moderate and negative, suggesting some complexity in how past values relate to current values
- The model accounts for the influence of two previous months' prices, capturing both immediate and delayed effects
- The variance is relatively high, which may indicate substantial unexplained variability
- RMSE and MAE are moderate, indicating that the model has reasonable accuracy but could be improved
- High MPE and MAPE suggest some forecasts might be significantly off from actual values

```r
# Perform diagnostics for AR(2)
par(mar=c(5, 5, 4, 2) + 0.1)
tsdiag(ar_model, gof.lag = 10, main = "Diagnostics for AR(2)")
```



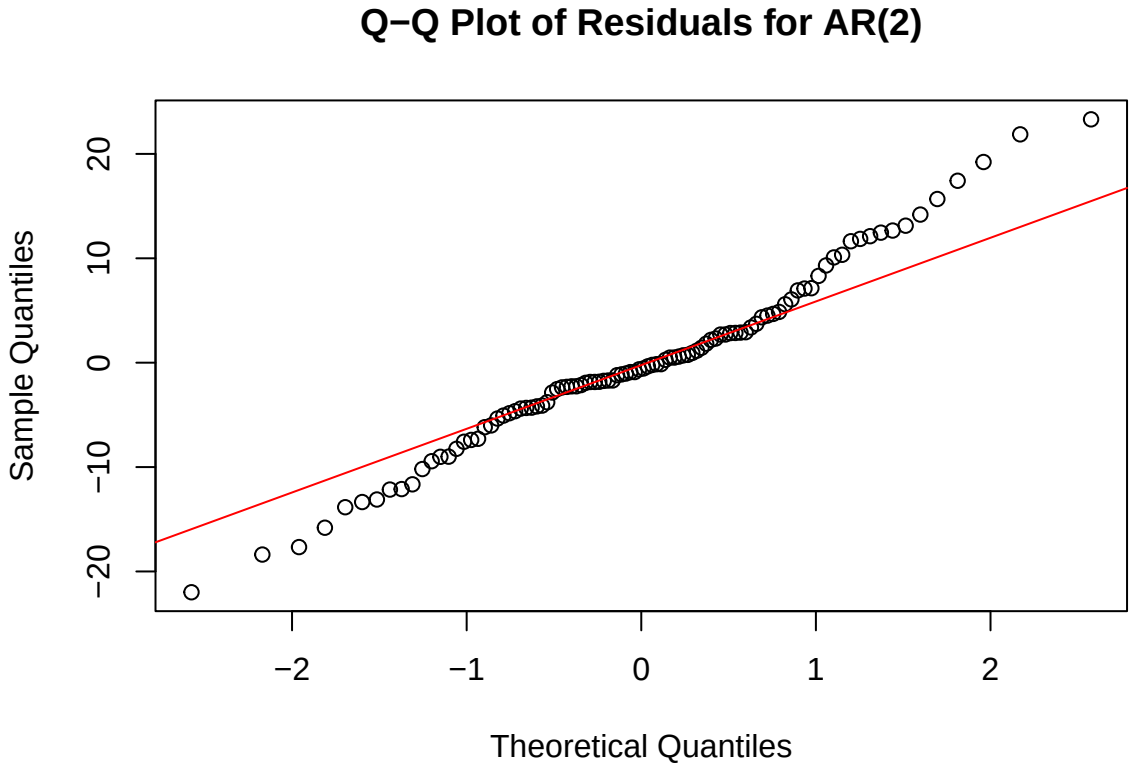
Residual Analysis

```r
checkresiduals(ar_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0) with non-zero mean
## Q* = 23.044, df = 18, p-value = 0.1889
##
## Model df: 2.   Total lags used: 20
```

```r
# Q-Q plot for AR(2)
residuals_ar2 <- residuals(ar_model)
qqnorm(residuals_ar2, main = "Q-Q Plot of Residuals for AR(2)")
qqline(residuals_ar2, col = "red")
```



Q-Q Plot:

- The points in the middle of the plot are closely aligned with the red line, indicating that the central residuals are approximately normally distributed.
- There is some deviation at the tails of the distribution, with points falling off the line, suggesting the presence of some outliers or non-normality in the extreme values.
- The Q-Q plot indicates that while most residuals follow a normal distribution, the extreme values do not. This is common in financial data where extreme values (volatility clusters) are not uncommon.

Residuals vs. Time Plot:

- The residuals appear to be randomly scattered around zero with no discernible pattern over time.
- This randomness suggests that the AR(2) model has effectively captured the linear patterns in the data, leaving white noise residuals.

ACF of Residuals:

- The autocorrelation function (ACF) of the residuals shows that most spikes are within the confidence bands, indicating no significant autocorrelations.
- Lack of significant autocorrelation implies that there are no patterns left unexplained by the AR(2) model.

Histogram of Residuals:

- The histogram with the superimposed normal curve suggests a somewhat normal distribution of residuals, although there might be slight skewness.
- This further supports the finding from the Q-Q plot that the residuals are approximately normal, but with some deviations at the tails.

Ljung-Box Test:

- With a p-value of 0.1889, we fail to reject the null hypothesis. This indicates that the residuals do not exhibit significant autocorrelation, suggesting that the AR(2) model fits the data well in terms of capturing the temporal dependencies.

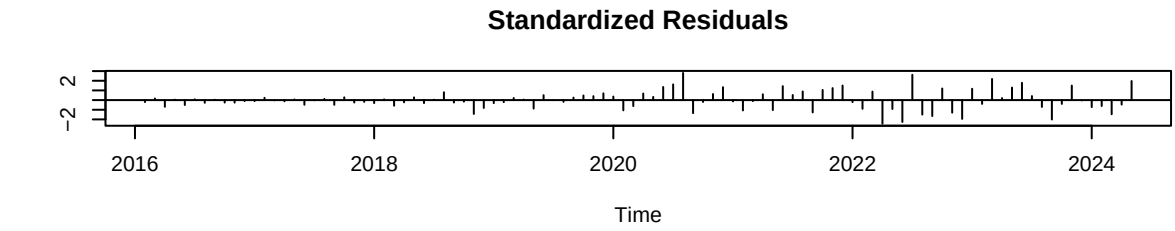
```
# Fit MA model
ma_model <- Arima(aaplmonthly_ts, order=c(0,0,2))
summary(ma_model)
```

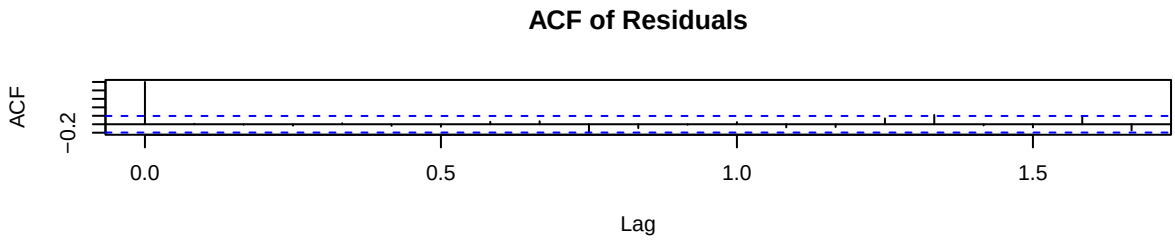
MA Model

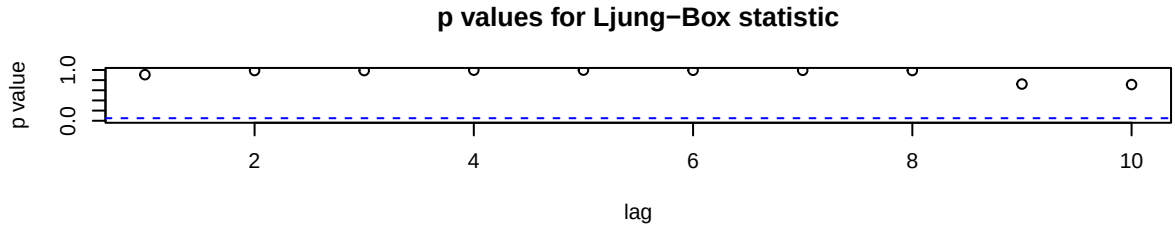
```
## Series: aaplmonthly_ts
## ARIMA(0,0,2) with non-zero mean
##
## Coefficients:
##          ma1      ma2    mean
##       0.0231  -0.2775  1.6525
## s.e.  0.0979   0.0971  0.6327
##
## sigma^2 = 73.16:  log likelihood = -355.09
## AIC=718.17   AICc=718.59   BIC=728.59
##
## Training set error measures:
##                        ME     RMSE     MAE      MPE     MAPE      MASE
## Training set -0.006781185 8.424279 6.24754 520.9368 585.7699 0.7045594
##                    ACF1
## Training set 0.0114845
```

- The small MA(1) coefficient suggests a weak impact of the error from one period ago, while the moderate negative MA(2) coefficient indicates a more noticeable inverse relationship with errors from two periods ago.
- The variance (sigma^2) is moderate, consistent with the AR(2) model, suggesting a similar level of unexplained volatility.
- RMSE and MAE are slightly lower than those of the AR(2) model, indicating that the MA(2) model may fit the data slightly better in terms of absolute error metrics.
- The AIC and BIC values are very similar to those of the AR(2) model, suggesting that both models are comparable in fit.
- The slightly lower AIC suggests the MA(2) model might be marginally better in terms of balancing model complexity and goodness of fit.

```
# Perform diagnostics for MA(2)
par(mar=c(5, 5, 4, 2) + 0.1)
tsdiag(ma_model, gof.lag = 10, main = "Diagnostics for MA(2)")
```

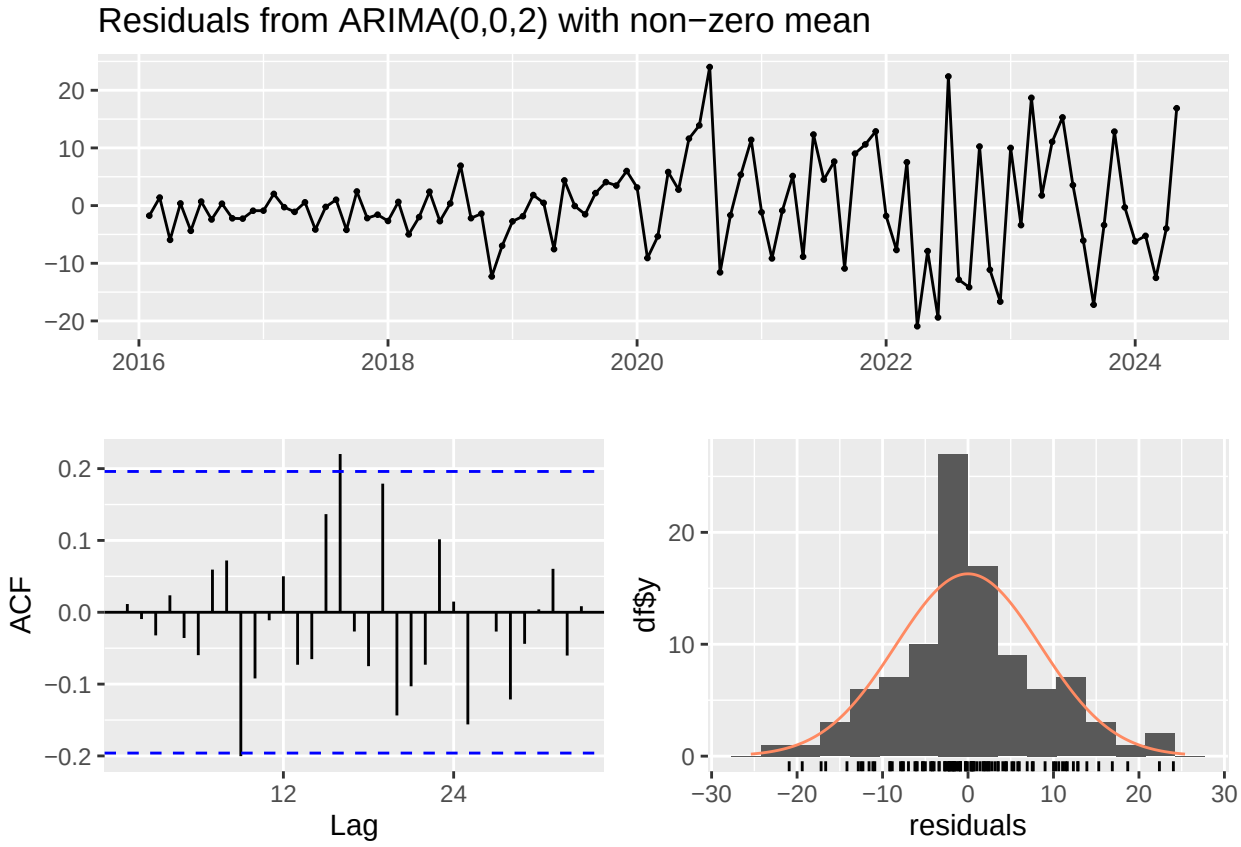






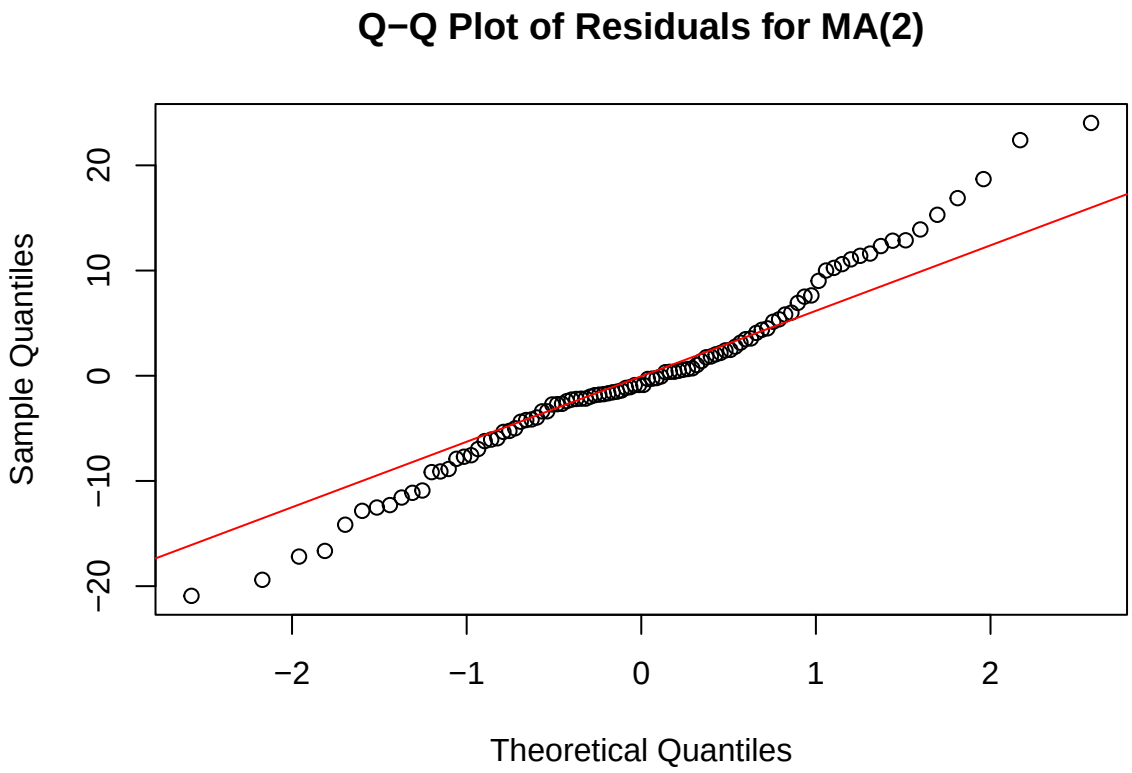
Residual Analysis

```
checkresiduals(ma_model)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,2) with non-zero mean
## Q* = 24.159, df = 18, p-value = 0.1499
##
## Model df: 2.   Total lags used: 20
```

```r
# Q-Q plot for MA(2)
residuals_ma2 <- residuals(ma_model)
qqnorm(residuals_ma2, main = "Q-Q Plot of Residuals for MA(2)")
qqline(residuals_ma2, col = "red")
```



Q-Q Plot:

- The plot shows that the residuals closely follow the red line in the middle section, suggesting that the residuals are approximately normally distributed.
- However, there are deviations at the tails, indicating potential issues with normality at the extremes (outliers).

Residuals vs. Time Plot:

- The residuals appear to be scattered randomly around zero, indicating no obvious patterns or trends.
- This suggests that the model has captured most of the structure in the data.

ACF of Residuals:

- Most of the spikes are within the blue dashed lines, indicating that the residuals do not exhibit significant autocorrelation.

- A few spikes slightly exceed the bounds, which may indicate some remaining autocorrelation, but it's not substantial.

Histogram of Residuals:

- The residuals appear to follow a bell-shaped curve, which suggests approximate normality.
- There is a slight skewness, as seen by the mismatch between the histogram and the normal curve.

Ljung-Box Test:

- The p-value is 0.1499, which is greater than 0.05. This indicates that there is no significant autocorrelation in the residuals. The model's residuals can be considered independently distributed.

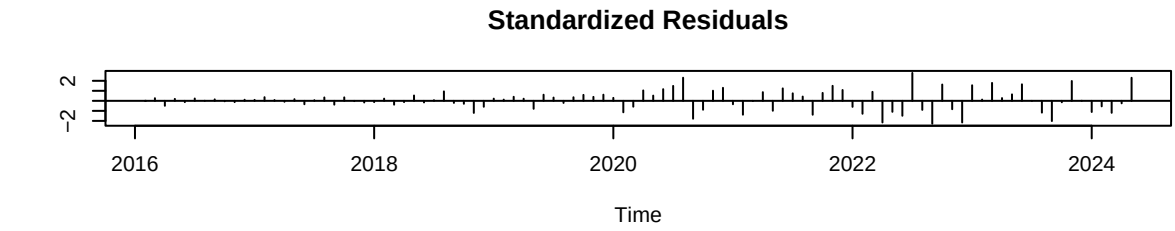
```
# Fit ARMA(1,1,1) model
arma_model1 <- Arima(aaplmonthly_ts, order=c(1,1,1))
summary(arma_model1)
```

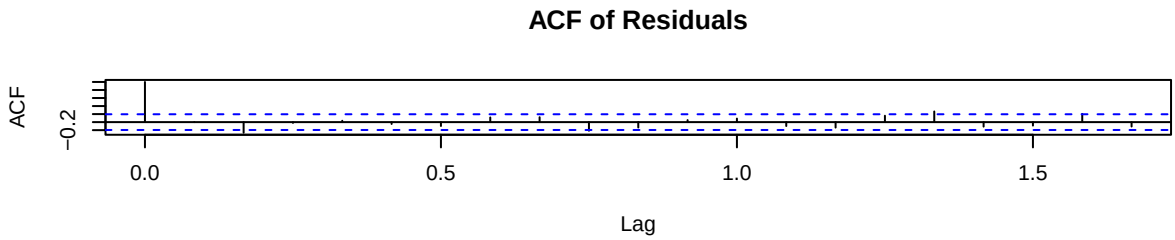
ARIMA(1,1,1) Model

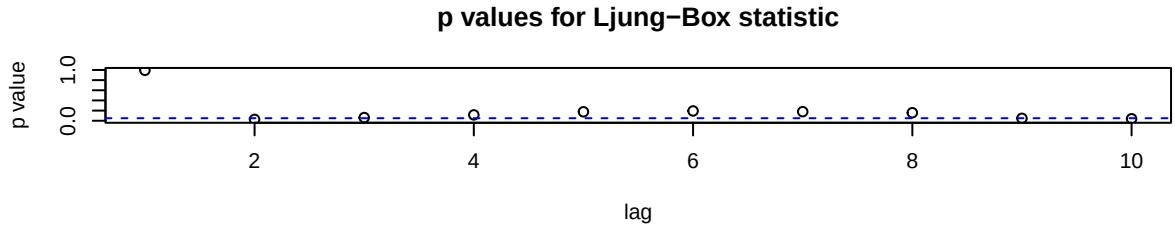
```
## Series: aaplmonthly_ts
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.0413  -1.0000
## s.e.  0.1034   0.0308
##
## sigma^2 = 78.94:  log likelihood = -357.98
## AIC=721.96   AICc=722.21   BIC=729.74
##
## Training set error measures:
##                      ME     RMSE      MAE     MPE     MAPE      MASE
## Training set 0.5179315 8.750795 6.505048 433.929 441.6441 0.7335995
##                       ACF1
## Training set -0.0008596899
```

- The AR coefficient is weak, indicating a limited effect of past values on current values. The MA coefficient is very strong, indicating significant correction based on past errors.
- The differencing part of the ARIMA model suggests that the data has been transformed to achieve stationarity, which might explain why the AR component is weak.
- The variance of the residuals is moderate, suggesting some unexplained variability in the data.
- RMSE and MAE are higher than in AR(2) and MA(2), indicating less precise predictions.
- The AIC and BIC values suggest this model might not be the best fit compared to simpler models like AR(2) and MA(2), which had lower AIC/BIC values.

```
# Perform diagnostics for ARIMA(1,1,1)
par(mar=c(5, 5, 4, 2) + 0.1)
tsdiag(arma_model1, gof.lag = 10, main = "Diagnostics for ARIMA(1,1,1)")
```

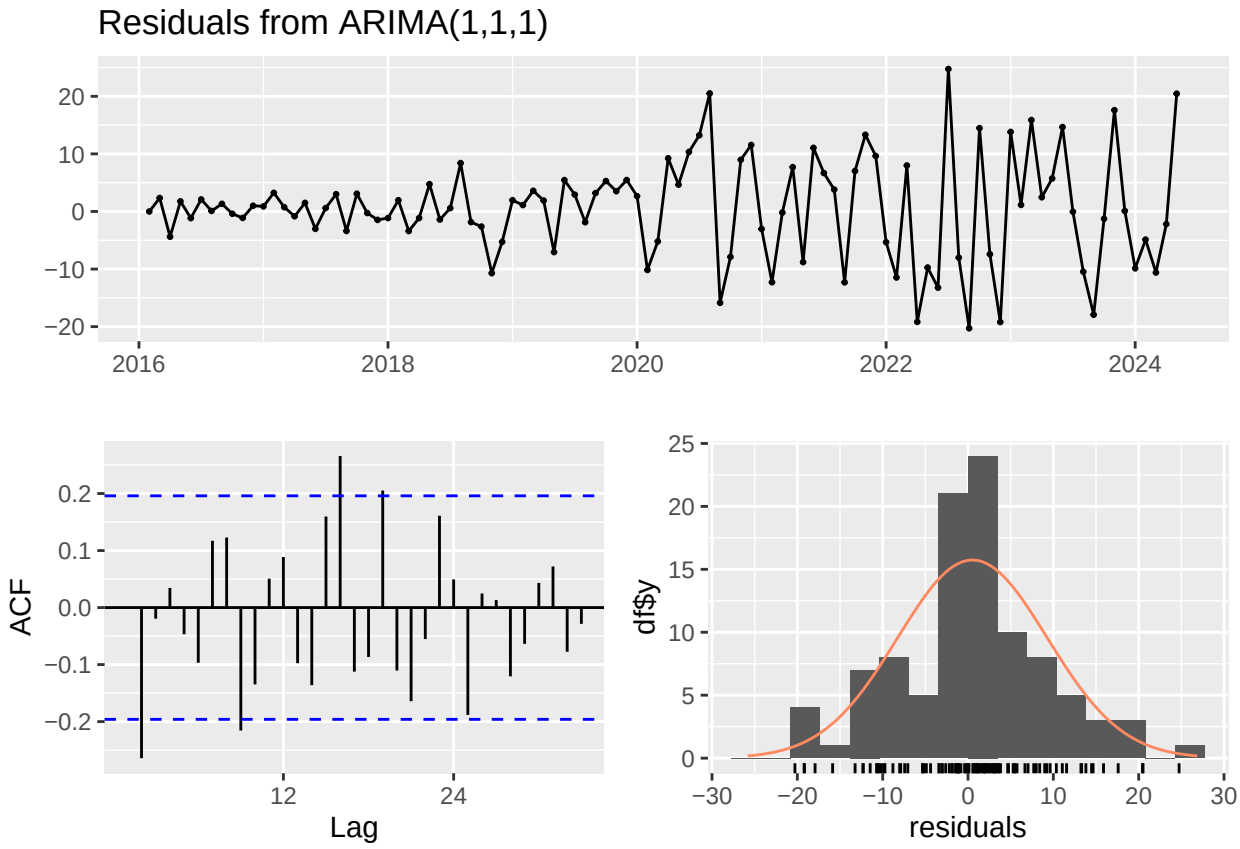






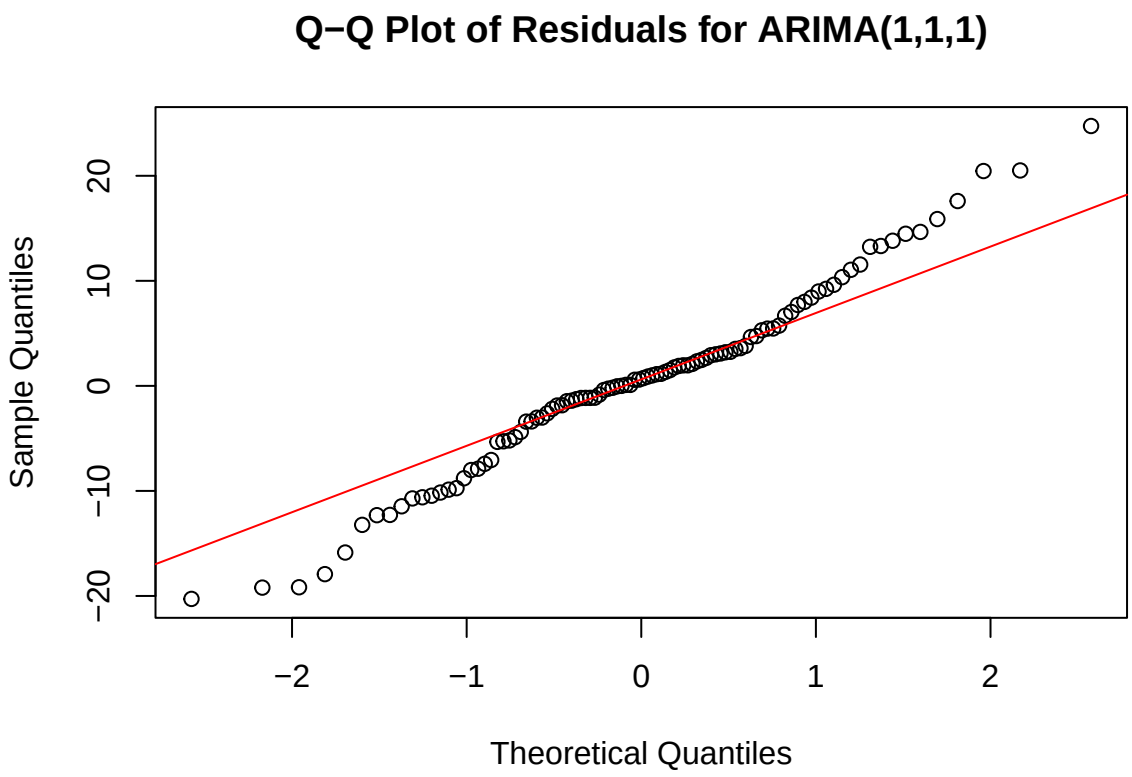
Residual Analysis

```
checkresiduals(arma_model1)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 44.611, df = 18, p-value = 0.0004715
##
## Model df: 2.   Total lags used: 20
```

```r
# Q-Q plot for ARIMA(1,1,1)
residuals_arma111 <- residuals(arma_model1)
qqnorm(residuals_arma111, main = "Q-Q Plot of Residuals for ARIMA(1,1,1)")
qqline(residuals_arma111, col = "red")
```



Q-Q Plot:

- The residuals are mostly aligned with the straight line, suggesting that they are approximately normally distributed.
- However, there are some deviations at both tails, indicating potential issues with extreme values or heavy tails.
- The plot indicates that while most of the residuals follow a normal distribution, there may be some extreme values (outliers) that deviate from this assumption.

Residuals vs. Time Plot:

- The residuals appear to be centered around zero, which is a good sign for unbiased predictions.
- There are some visible spikes and variations over time, which could indicate non-constant variance or potential patterns not captured by the model.

ACF of Residuals:

- The ACF values are mostly within the blue dashed lines (confidence intervals), suggesting that the residuals do not exhibit significant autocorrelation.
- This indicates that the ARIMA(1,1,1) model has successfully captured most of the autocorrelation structure in the data.

Histogram of Residuals:

- The distribution appears to be approximately normal, but with slight skewness and kurtosis, as indicated by the histogram.
- There are some residuals on the tails that are further away from the normal curve, which could affect model accuracy for extreme values.

Ljung-Box Test:

- The low p-value suggests that there is significant autocorrelation present in the residuals. This indicates that the ARIMA(1,1,1) model may not have captured all the autocorrelation in the data, leading to potential improvement in model fit.

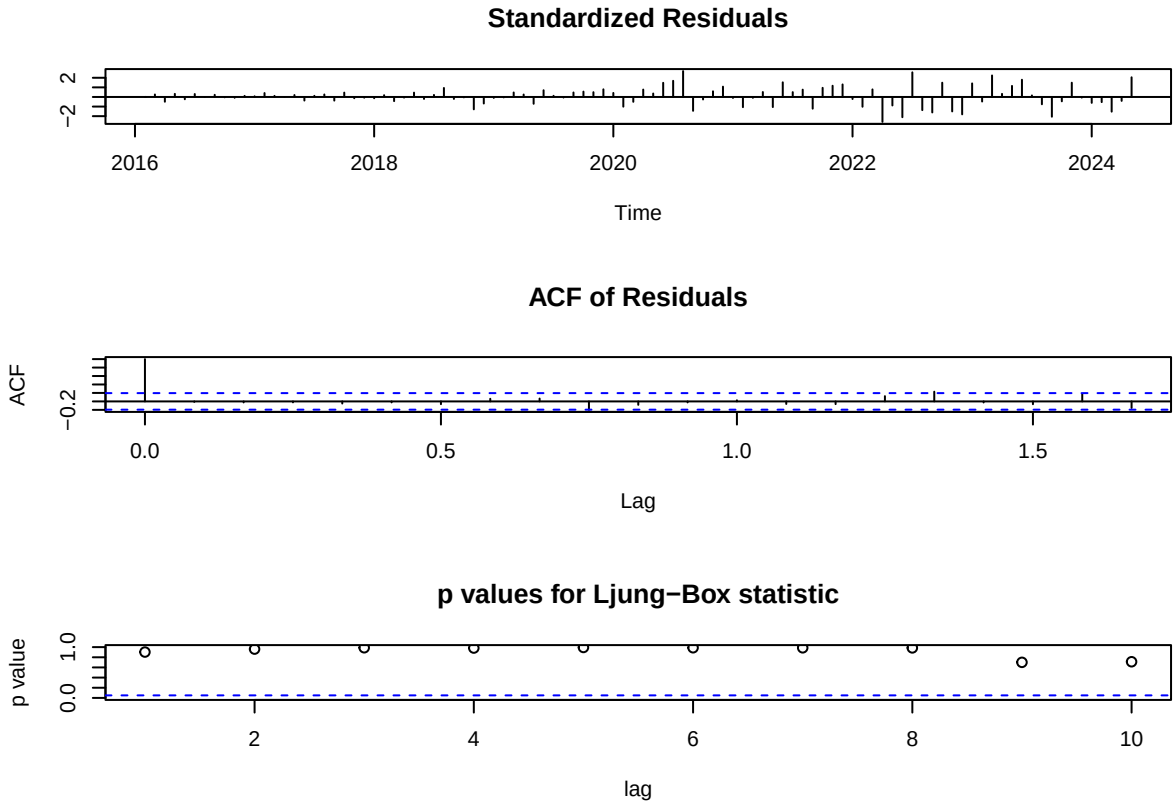
```
# ARMA(2,1,1) Model
arma_model2 <- Arima(aaplmonthly_ts, order=c(2,1,1))
summary(arma_model2)
```

ARIMA(2,1,1) Model

```
## Series: aaplmonthly_ts
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2      ma1
##       0.0486  -0.2632  -1.0000
## s.e.  0.0996   0.0988   0.0377
##
## sigma^2 = 74.02:  log likelihood = -354.58
## AIC=717.16   AICc=717.59   BIC=727.54
##
## Training set error measures:
##                      ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.6253754 8.429532 6.225648 78.44247 127.6236 0.7020905
##                      ACF1
## Training set -0.01202426
```

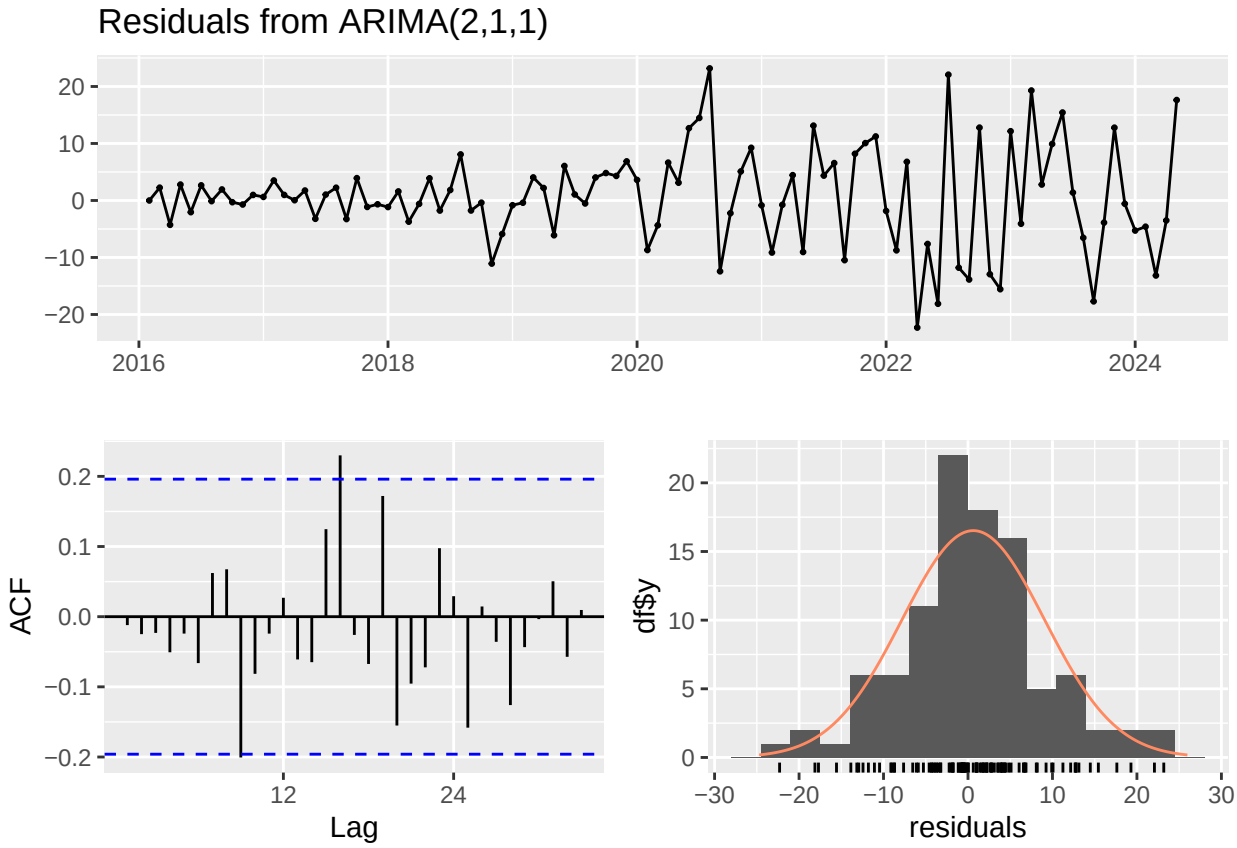
- The AR coefficients show a mixed impact, with a weak positive effect from the last period and a moderate negative effect from two periods ago. The strong MA(1) term suggests that recent errors are heavily corrected, which may smooth the series too aggressively.
- The variance of the residuals is moderate, indicating a fair amount of explained variability. The error measures (RMSE and MAE) suggest that the model provides reasonably accurate predictions, although there are still areas for improvement.
- The AIC and BIC values are relatively low, indicating a good fit compared to other models. The log likelihood is also higher, supporting this conclusion.
- The ARIMA(2,1,1) model captures the data dynamics well, with strong correction for past errors and a reasonable account of past values. The model is effective in handling the data's structure and trends, as indicated by the low error measures and ACF1.

```r
# Perform diagnostics for ARIMA(2,1,1)
par(mar=c(5, 5, 4, 2) + 0.1)
tsdiag(arma_model2, gof.lag = 10, main = "Diagnostics for ARIMA(2,1,1)")
```



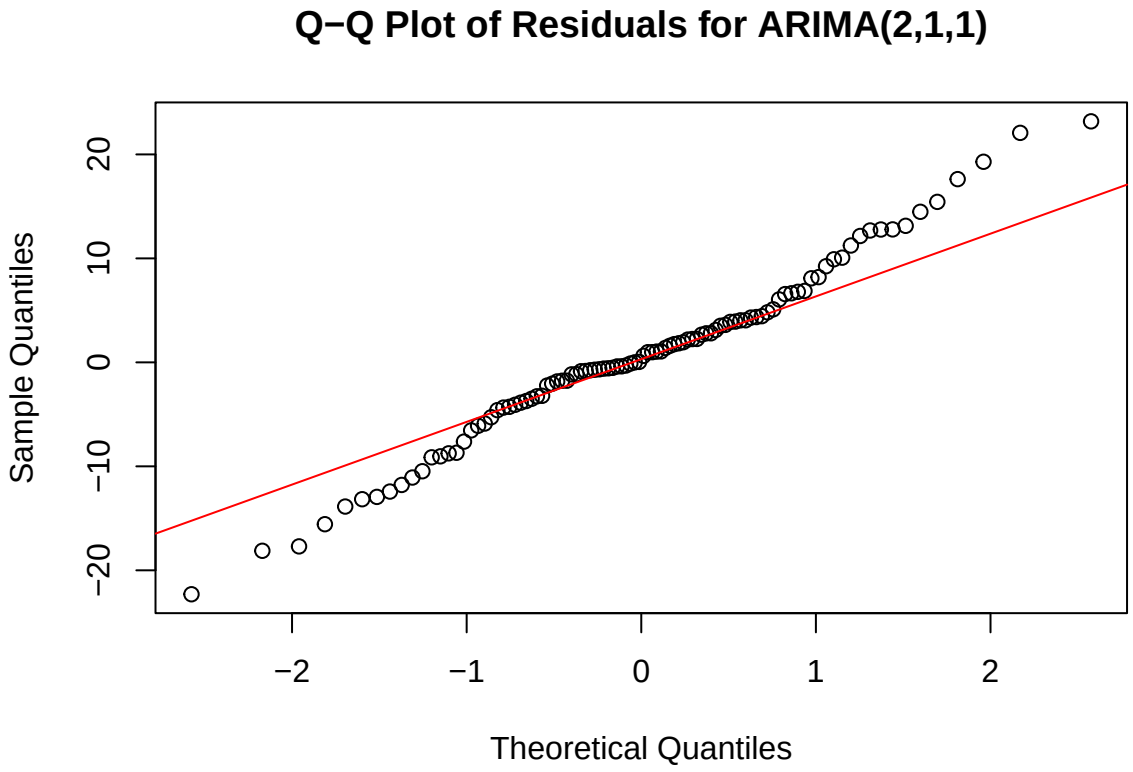
Residual Analysis

```r
checkresiduals(arma_model2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)
## Q* = 23.957, df = 17, p-value = 0.1206
##
## Model df: 3.   Total lags used: 20
```

```r
# Q-Q plot for ARIMA(2,1,1)
residuals_arma211 <- residuals(arma_model2)
qqnorm(residuals_arma211, main = "Q-Q Plot of Residuals for ARIMA(2,1,1)")
qqline(residuals_arma211, col = "red")
```



Q-Q Plot:

- The residuals are close to the line, especially in the middle range, indicating that they are approximately normally distributed.
- The tails deviate slightly from the line, suggesting potential outliers or non-normality in the extremes. This is common in financial data.

Residuals vs. Time Plot:

- The residuals fluctuate around zero, indicating that the model has captured the trend well.
- The variance of residuals appears constant over time, which satisfies one of the assumptions of ARIMA modeling.
- There are no obvious patterns, suggesting the model is capturing most of the signal.

ACF of Residuals:

- The majority of the autocorrelation values fall within the blue confidence bands, indicating no significant autocorrelation.
- This suggests that the model has effectively captured the autocorrelations in the data.

Histogram of Residuals:

- The histogram shows a bell-shaped curve, with some deviation at the tails, which is typical for time series data.
- The overlaying normal curve fits reasonably well, suggesting residuals are approximately normally distributed.

Ljung-Box Test:

- The p-value is greater than 0.05, indicating that we fail to reject the null hypothesis of no autocorrelation in the residuals. This means the residuals are likely random, supporting the model's adequacy.

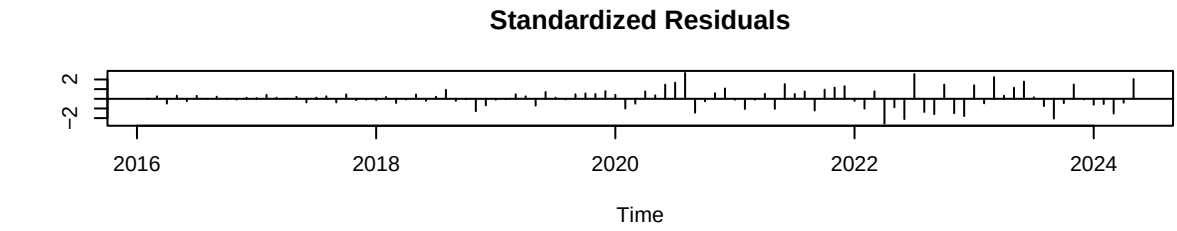
```r
# ARMA(2,1,2) Model
arma_model3 <- Arima(aaplmonthly_ts, order=c(2,1,2))
summary(arma_model3)
```

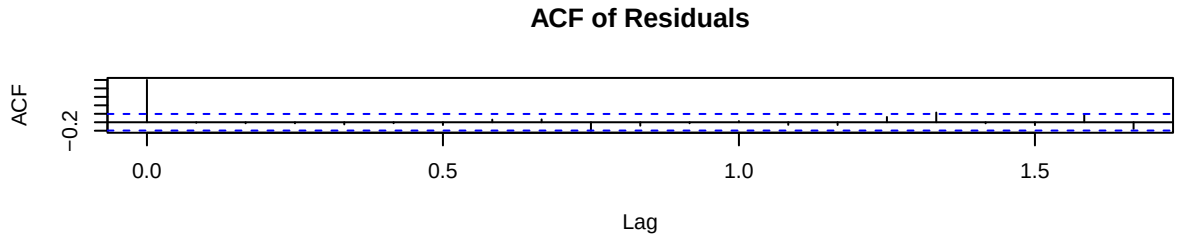
ARIMA(2,1,2) Model

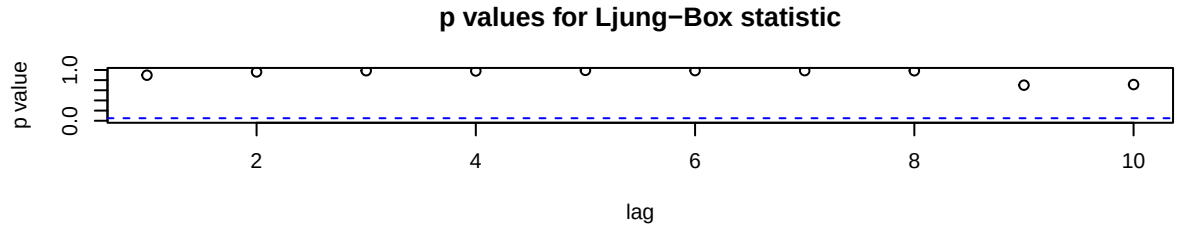
```
## Series: aaplmonthly_ts
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2      ma1      ma2
##       0.0375  -0.2628  -0.9880  -0.0120
## s.e.  0.4098   0.1004   0.4284   0.4267
##
## sigma^2 = 74.8:  log likelihood = -354.58
## AIC=719.16   AICc=719.8   BIC=732.13
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.6240689 8.429788 6.223491 79.15774 126.6565 0.7018472
##                     ACF1
## Training set -0.01277316
```

- The AR(2) term seems significant, while the AR(1) and MA(2) terms are less impactful. The MA(1) term is strong, suggesting reliance on correcting the previous period's error.
- The residual variance (sigma^2) is slightly higher than desired, indicating some unexplained variability. The ACF1 is close to zero, which is positive, indicating little remaining autocorrelation.
- Compared to the ARIMA(2,1,1) model, this model has slightly higher AIC and BIC, suggesting it might not fit as well. However, the performance metrics (RMSE, MAE) are competitive, indicating this model is also a valid candidate.
- While the error measures (MAPE, MPE) show some prediction errors, the model captures the general trend well. Improvements could be made by refining the model or testing alternative specifications.

```r
# Perform diagnostics for ARIMA(2,1,2)
par(mar=c(5, 5, 4, 2) + 0.1)
tsdiag(arma_model3, gof.lag = 10, main = "Diagnostics for ARIMA(2,1,2)")
```

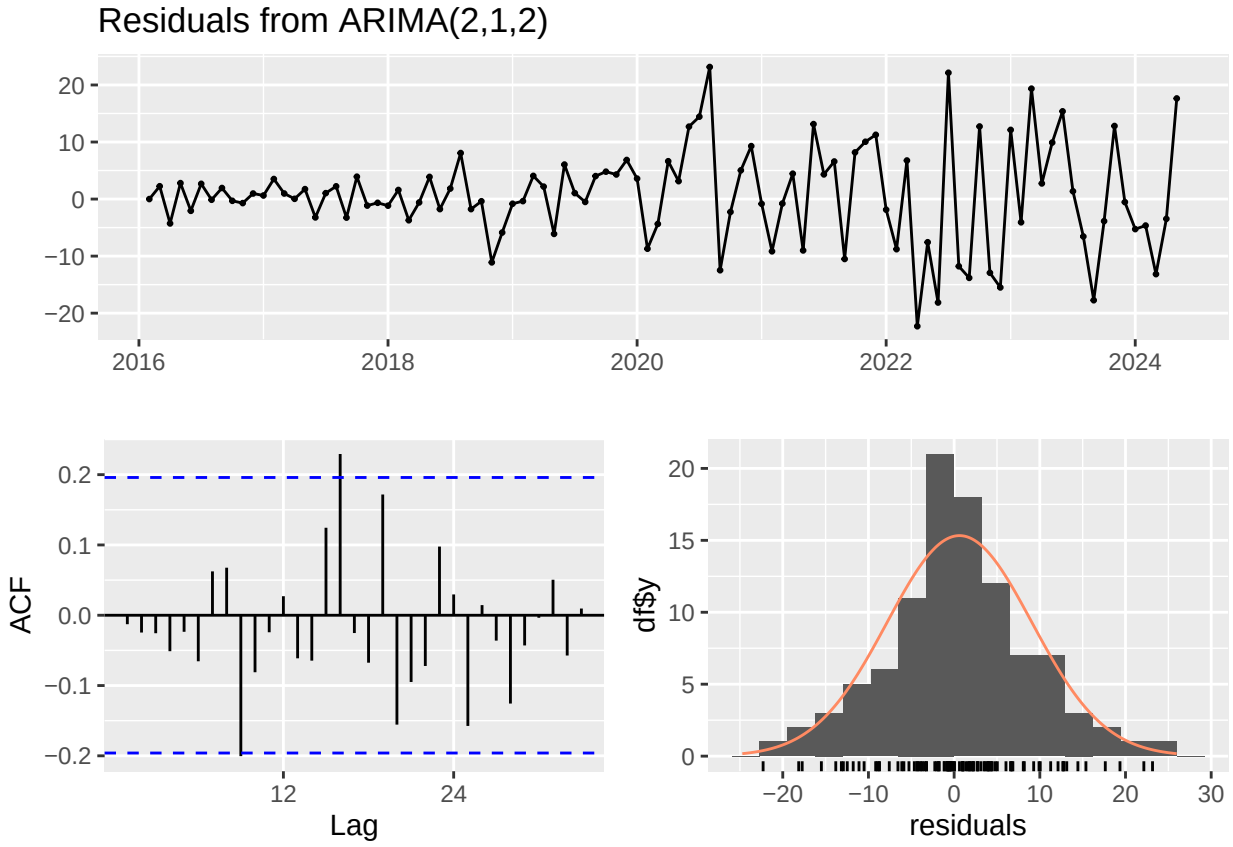






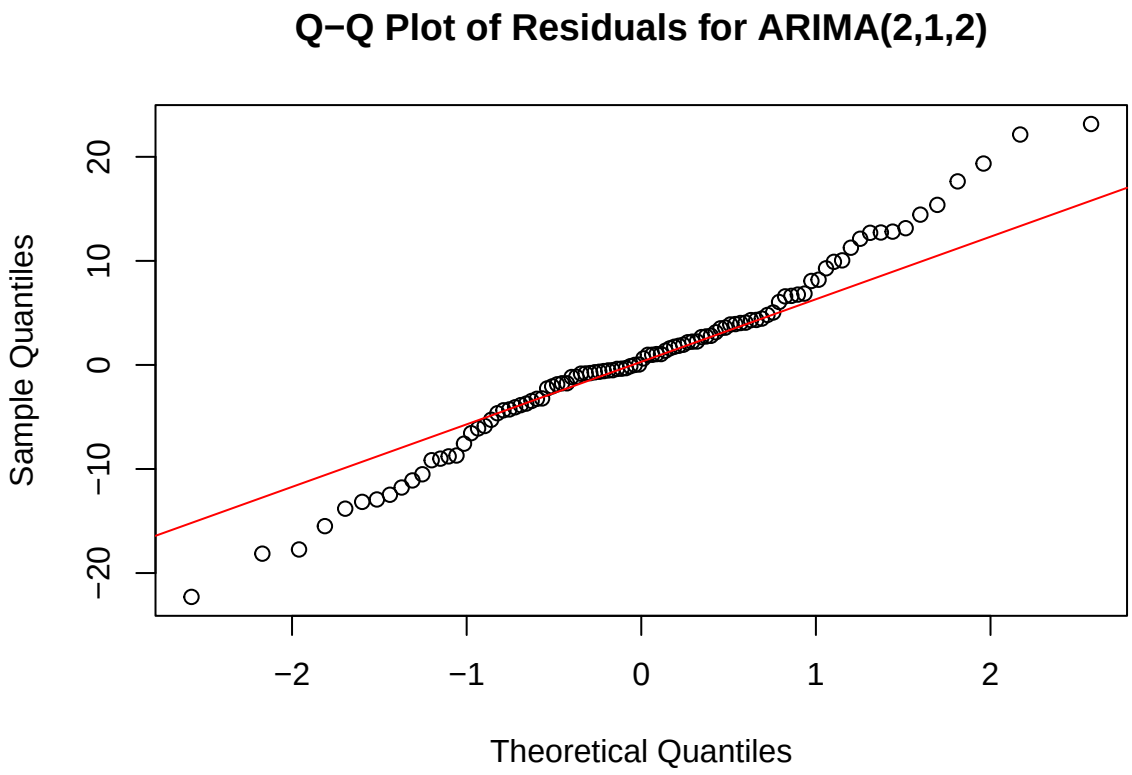
Residual Analysis

```
checkresiduals(arma_model3)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2)
## Q* = 23.916, df = 16, p-value = 0.09136
##
## Model df: 4.   Total lags used: 20
```

```r
# Q-Q plot for ARIMA(2,1,2)
residuals_arma212 <- residuals(arma_model3)
qqnorm(residuals_arma212, main = "Q-Q Plot of Residuals for ARIMA(2,1,2)")
qqline(residuals_arma212, col = "red")
```



Q-Q Plot:

- The residuals are close to the line, especially in the middle range, indicating that they are approximately normally distributed.
- The tails deviate slightly from the line, suggesting potential outliers or non-normality in the extremes. This is common in financial data.

Residuals vs. Time Plot:

- The residuals fluctuate around zero, indicating that the ARIMA(2,1,2) model has successfully captured the main structure of the data.
- There are no obvious patterns or trends visible, suggesting that the model accounts well for the systematic components of the time series.
- The model appears to capture the dynamics of the data well, with residuals that resemble white noise.

ACF of Residuals:

- Most autocorrelation values lie within the blue dashed confidence bounds, which suggests that the residuals are uncorrelated.
- The lack of significant autocorrelation indicates that the model has effectively captured the autocorrelation structure of the series, resulting in residuals that are close to white noise.

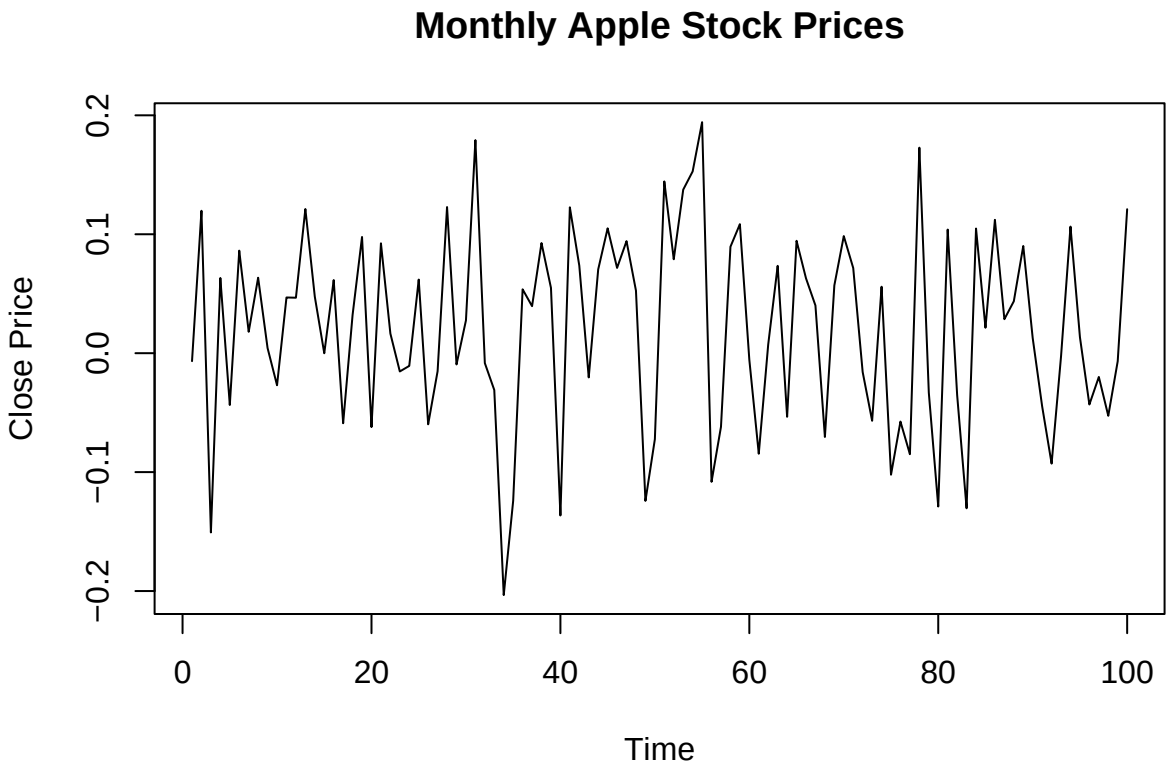
Histogram of Residuals:

- The residuals show a distribution that is fairly symmetric, though there might be some skewness or kurtosis.
- The residuals' distribution is fairly normal but with slight deviations, likely due to outliers or model imperfections.

Ljung-Box Test:

- Since the p-value is greater than 0.05, we fail to reject the null hypothesis of no autocorrelation, suggesting that the residuals are indeed white noise.

```
# Create a time series object
aaplmonthly_ts2 <- ts(aapl_monthly_data$Close, start=c(2016, 01), end = c(2024, 05), frequency=12)
# Calculate returns for modeling
returns <- diff(log(aaplmonthly_ts2))
returns <- returns[!is.na(returns)]
plot(returns, main="Monthly Apple Stock Prices", ylab="Close Price", xlab="Time", type="l")
```



GARCH Models

```r
# Specify GARCH(1,1) model
spec_garch <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
                         mean.model = list(armaOrder = c(1, 0)),
                         distribution.model = "norm")

fit_garch <- ugarchfit(spec = spec_garch, data = returns)

# Display the fit summary
fit_garch
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(1,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.020696    0.008198  2.52440  0.01159
## ar1     0.009283    0.100840  0.09206  0.92665
## omega   0.000212    0.000204  1.03801  0.29927
## alpha1  0.000000    0.014482  0.00000  1.00000
## beta1   0.969597    0.051349 18.88262  0.00000
##
## Robust Standard Errors:
##         Estimate  Std. Error   t value Pr(>|t|)
## mu      0.020696    0.006649  3.112732 0.001854
## ar1     0.009283    0.103105  0.090038 0.928257
## omega   0.000212    0.000379  0.558168 0.576730
## alpha1  0.000000    0.005313  0.000000 1.000000
## beta1   0.969597    0.059723 16.234816 0.000000
##
## LogLikelihood : 108.877
##
## Information Criteria
## ------------------------------------
##
## Akaike       -2.0775
## Bayes        -1.9473
## Shibata      -2.0822
## Hannan-Quinn -2.0248
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                  0.0001857  0.9891
## Lag[2*(p+q)+(p+q)-1][2] 1.2346095  0.5948
```

```
## Lag[4*(p+q)+(p+q)-1][5] 2.4164966  0.5903
## d.o.f=1
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.449  0.2287
## Lag[2*(p+q)+(p+q)-1][5]     3.745  0.2876
## Lag[4*(p+q)+(p+q)-1][9]     6.376  0.2572
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     3.543 0.500 2.000 0.05978
## ARCH Lag[5]     3.661 1.440 1.667 0.20717
## ARCH Lag[7]     5.696 2.315 1.543 0.16300
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.4525
## Individual Statistics:
## mu     0.07109
## ar1    0.14752
## omega  0.14307
## alpha1 0.12991
## beta1  0.14386
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias          0.04934 0.9608
## Negative Sign Bias 0.84586 0.3998
## Positive Sign Bias 0.20106 0.8411
## Joint Effect       0.99590 0.8022
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20      20.4       0.3709
## 2    30      21.8       0.8284
## 3    40      34.4       0.6796
## 4    50      42.0       0.7504
##
##
## Elapsed time : 0.037956
```

```r
# Specify GARCH(1,1) model
spec_garch <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
                         mean.model = list(armaOrder = c(2, 1)),
                         distribution.model = "norm")

fit_garch <- ugarchfit(spec = spec_garch, data = returns)

# Display the fit summary
fit_garch
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(2,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.020739    0.005744  3.61058 0.000306
## ar1     0.604229    0.341075  1.77154 0.076471
## ar2    -0.151955    0.108878 -1.39564 0.162824
## ma1    -0.610786    0.339011 -1.80167 0.071598
## omega   0.000208    0.000238  0.87143 0.383518
## alpha1  0.000000    0.013219  0.00000 1.000000
## beta1   0.968842    0.047477 20.40660 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.020739    0.005612  3.69541 0.000220
## ar1     0.604229    0.323349  1.86866 0.061671
## ar2    -0.151955    0.095674 -1.58826 0.112228
## ma1    -0.610786    0.293617 -2.08021 0.037506
## omega   0.000208    0.000316  0.65791 0.510598
## alpha1  0.000000    0.003947  0.00000 1.000000
## beta1   0.968842    0.051773 18.71330 0.000000
##
## LogLikelihood : 110.5264
##
## Information Criteria
## ------------------------------------
##
## Akaike       -2.0705
## Bayes        -1.8882
## Shibata      -2.0795
## Hannan-Quinn -1.9967
##
## Weighted Ljung-Box Test on Standardized Residuals
```

```
## -----------------------------------
##                           statistic p-value
## Lag[1]                    0.0007236  0.9785
## Lag[2*(p+q)+(p+q)-1][8]   0.7800872  1.0000
## Lag[4*(p+q)+(p+q)-1][14] 1.9624525  0.9999
## d.o.f=3
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## -----------------------------------
##                          statistic p-value
## Lag[1]                       2.977 0.08444
## Lag[2*(p+q)+(p+q)-1][5]      4.546 0.19337
## Lag[4*(p+q)+(p+q)-1][9]      6.847 0.21233
## d.o.f=2
##
## Weighted ARCH LM Tests
## -----------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]     2.326 0.500 2.000  0.1273
## ARCH Lag[5]     2.329 1.440 1.667  0.4030
## ARCH Lag[7]     4.181 2.315 1.543  0.3212
##
## Nyblom stability test
## -----------------------------------
## Joint Statistic:  2.2869
## Individual Statistics:
## mu     0.14030
## ar1    0.06721
## ar2    0.04291
## ma1    0.07467
## omega  0.13026
## alpha1 0.10397
## beta1  0.13095
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.69 1.9 2.35
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## -----------------------------------
##                    t-value   prob sig
## Sign Bias           0.1523 0.8793
## Negative Sign Bias  1.1242 0.2638
## Positive Sign Bias  0.8288 0.4093
## Joint Effect        1.9837 0.5758
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## -----------------------------------
##   group statistic p-value(g-1)
## 1    20      27.6     0.091435
## 2    30      40.4     0.077606
## 3    40      47.2     0.172404
```

```
## 4    50      77.0     0.006497
##
##
## Elapsed time : 0.06286407
```

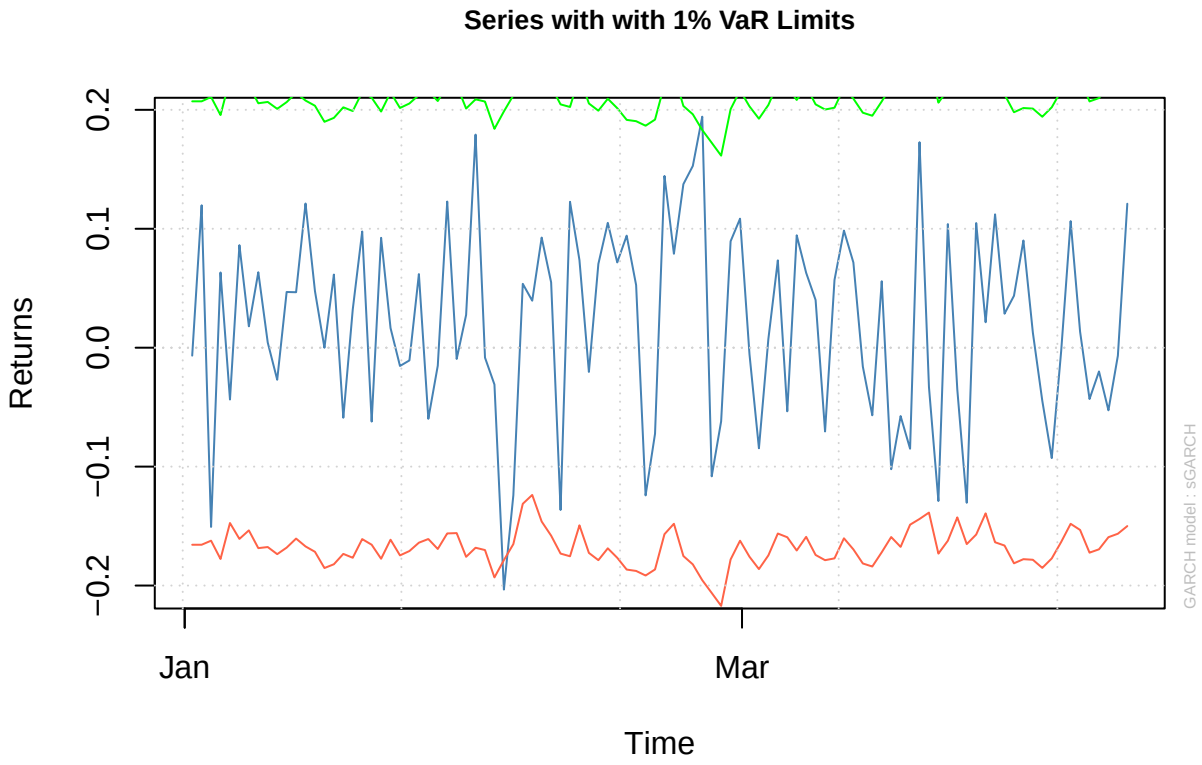
Model Fit:

- The GARCH(1,1) with ARFIMA(1,0,0) mean equation provides a reasonable fit to the data. The high significance of the β_1 parameter highlights the importance of past volatility in explaining current volatility.
- The absence of significant serial correlation in both raw and squared residuals indicates that the model captures the autocorrelation and volatility clustering effectively.
- The p-values from the diagnostic tests suggest that the residuals approximate a normal distribution, making the model suitable for the data.
- The stability of individual parameters suggests the model's robustness over time, though there is a slight indication of overall instability.
- The model is useful for forecasting future volatility and capturing the volatility clustering typical in financial time series like stock prices.

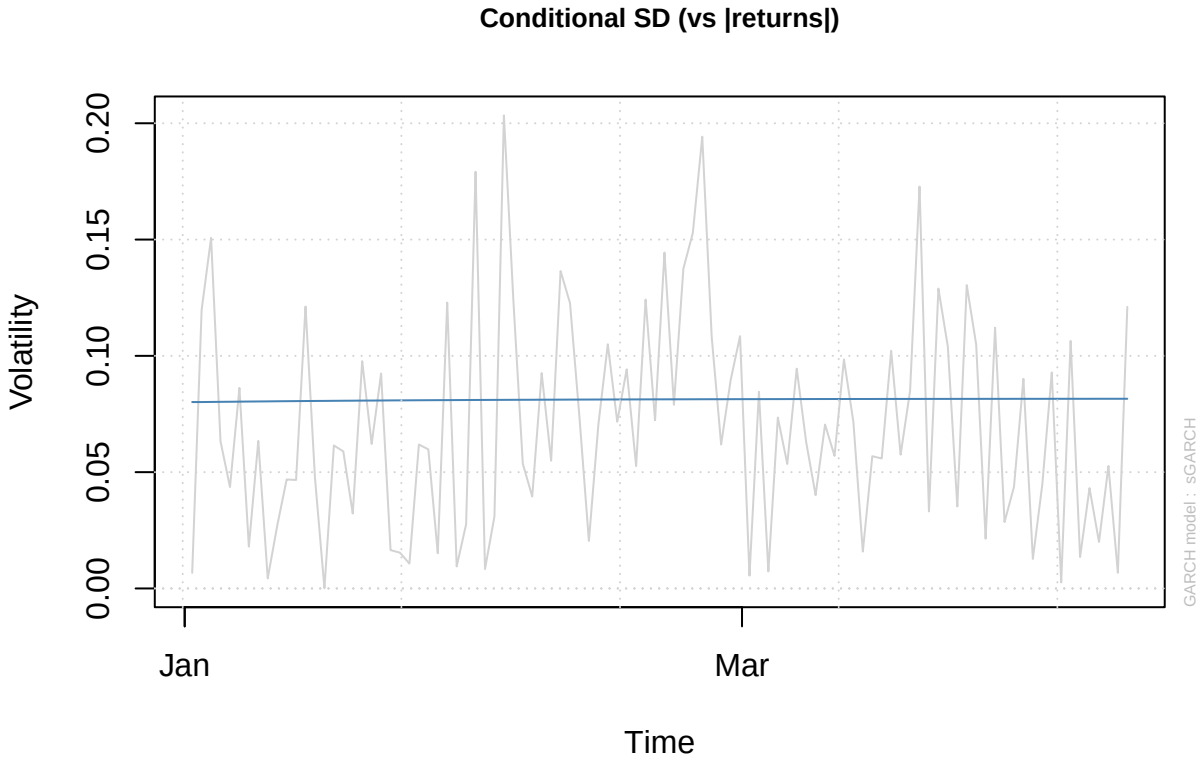
```
# Plot diagnostics
plot(fit_garch, which = 1)
```


```
plot(fit_garch, which = 2)
```

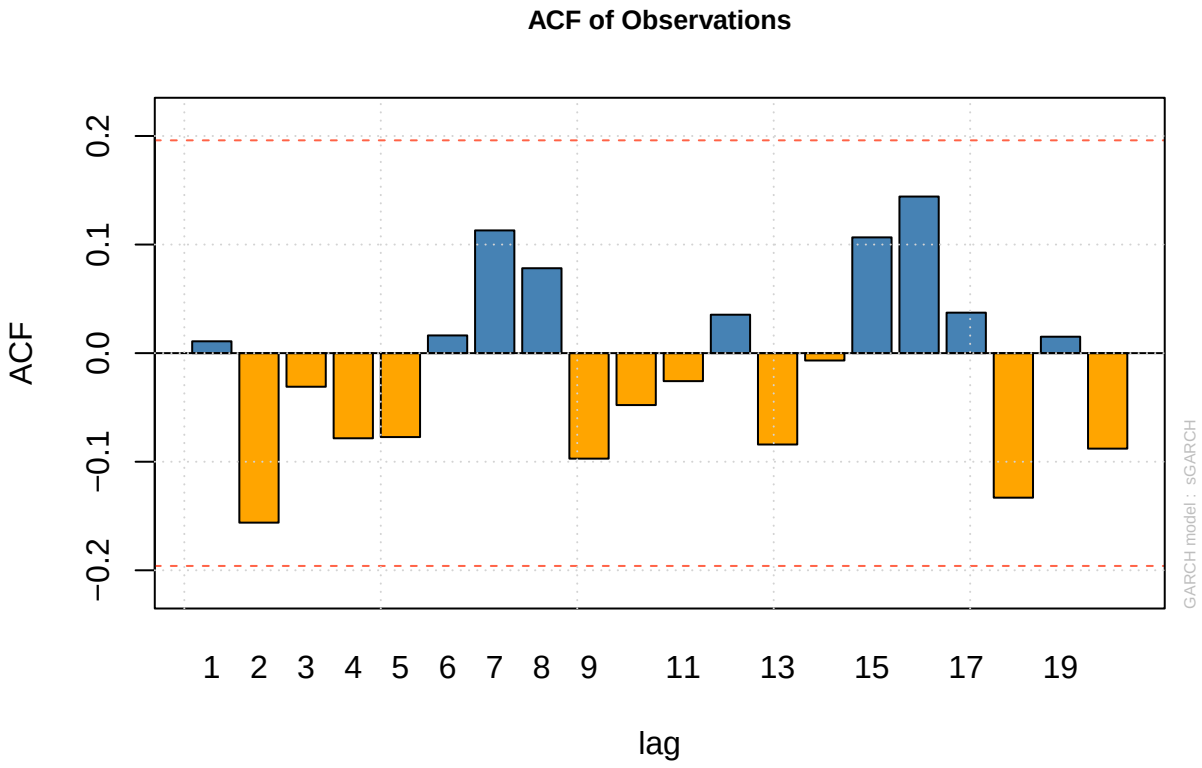
```
##
## please wait...calculating quantiles...
```



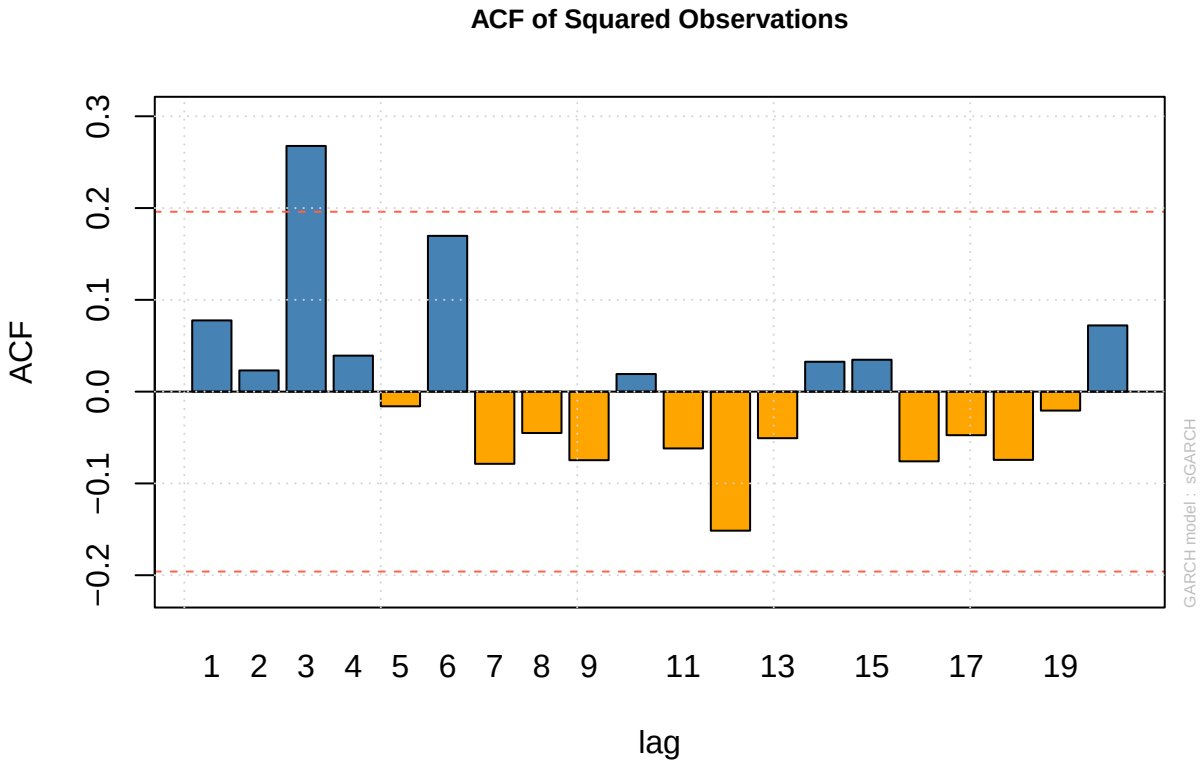

```
plot(fit_garch, which = 3)
```



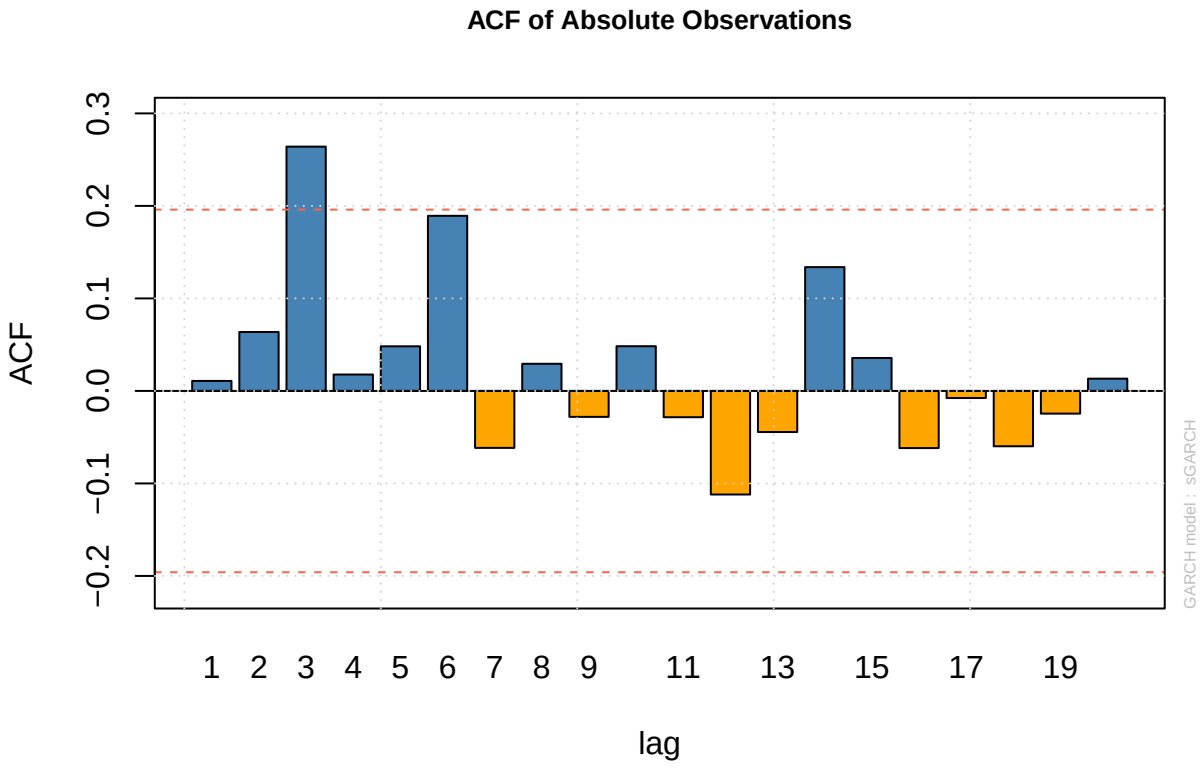

```
plot(fit_garch, which = 4)
```



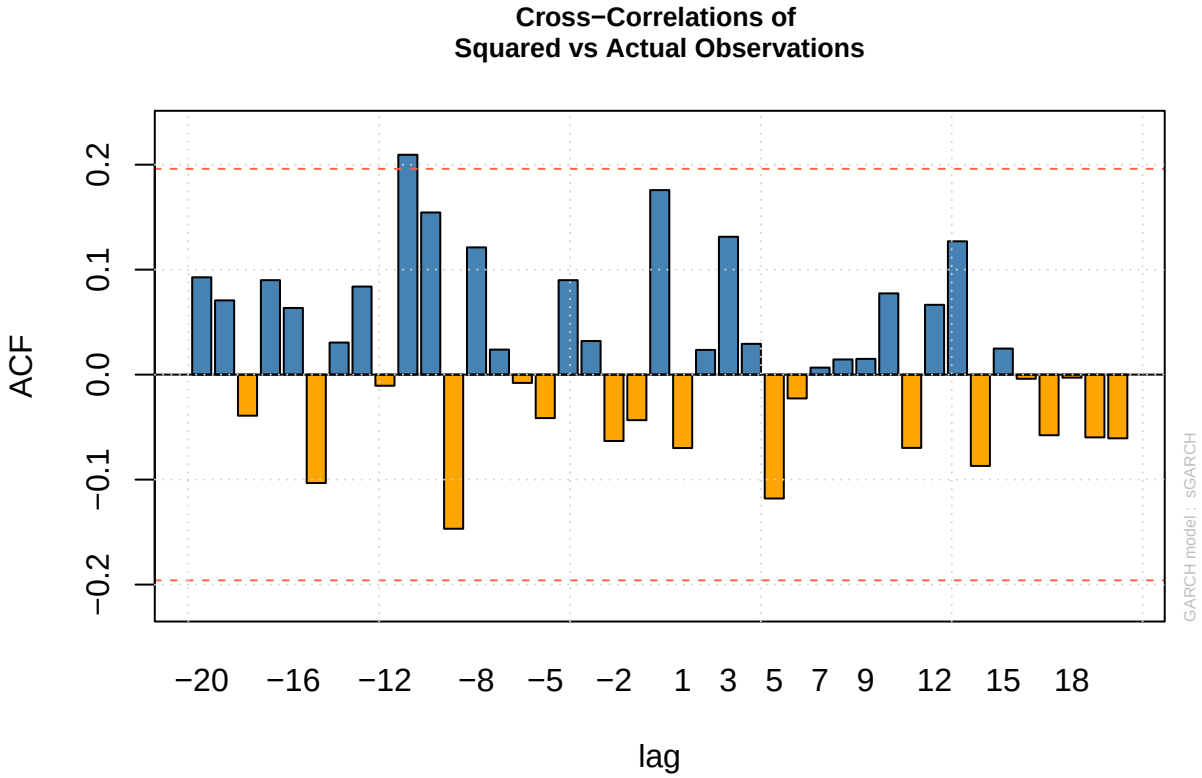

```
plot(fit_garch, which = 5)
```



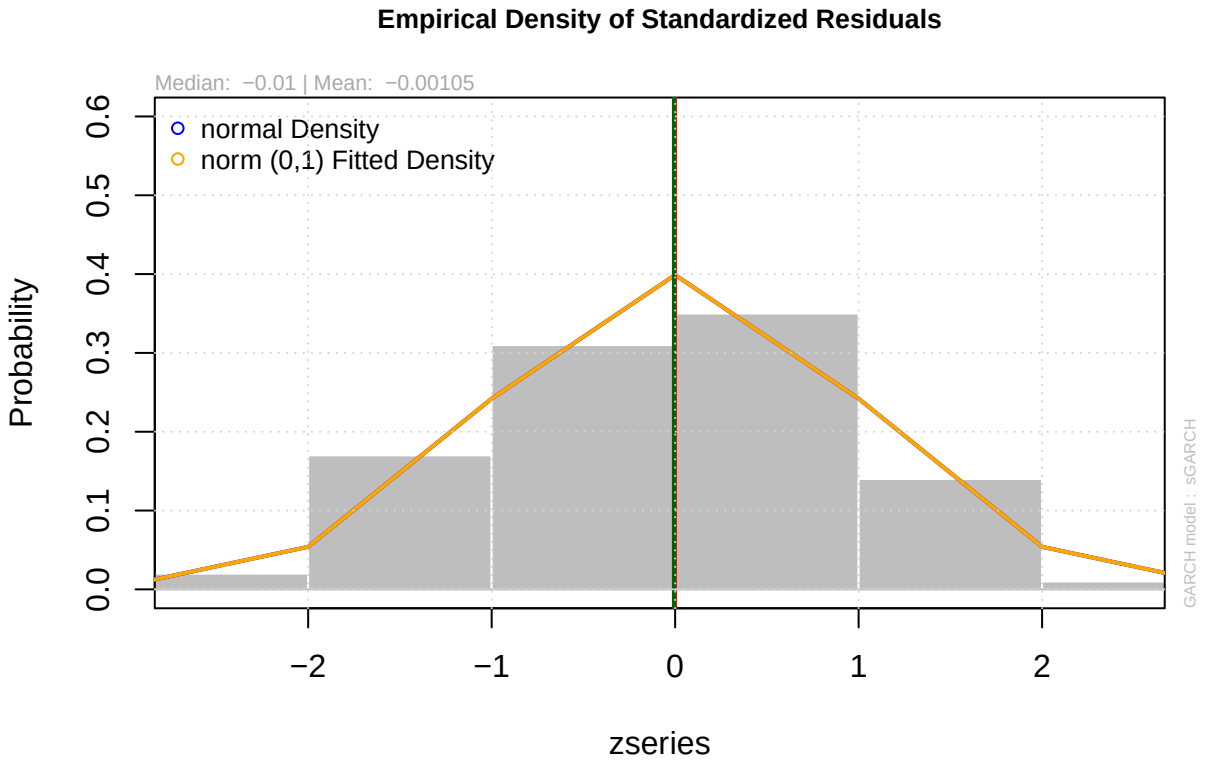

```
plot(fit_garch, which = 6)
```



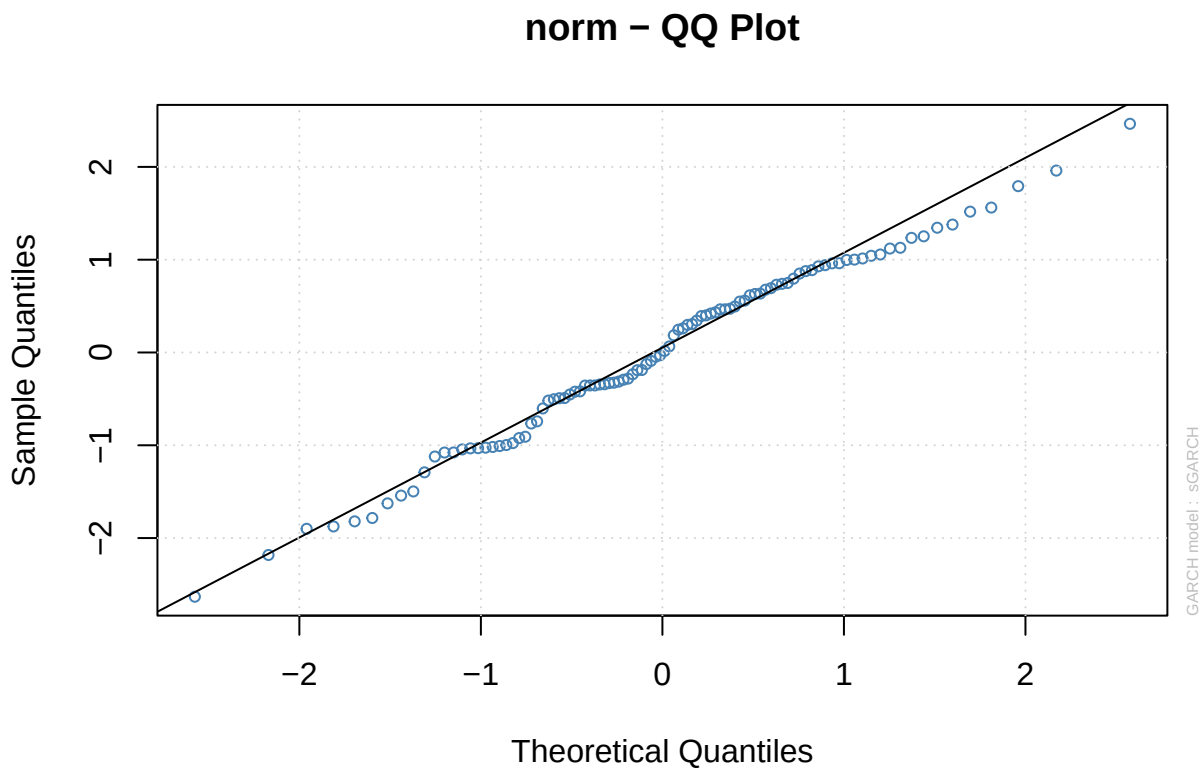

```
plot(fit_garch, which = 7)
```



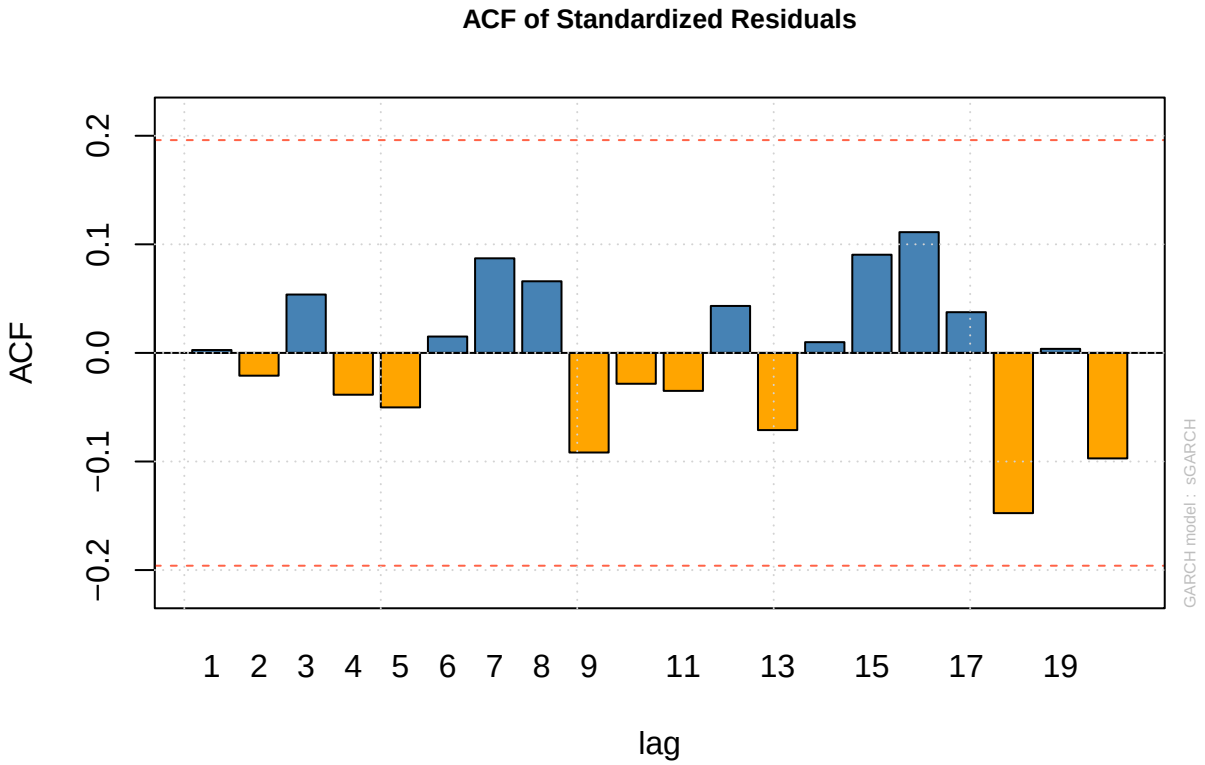
```
plot(fit_garch, which = 8)
```



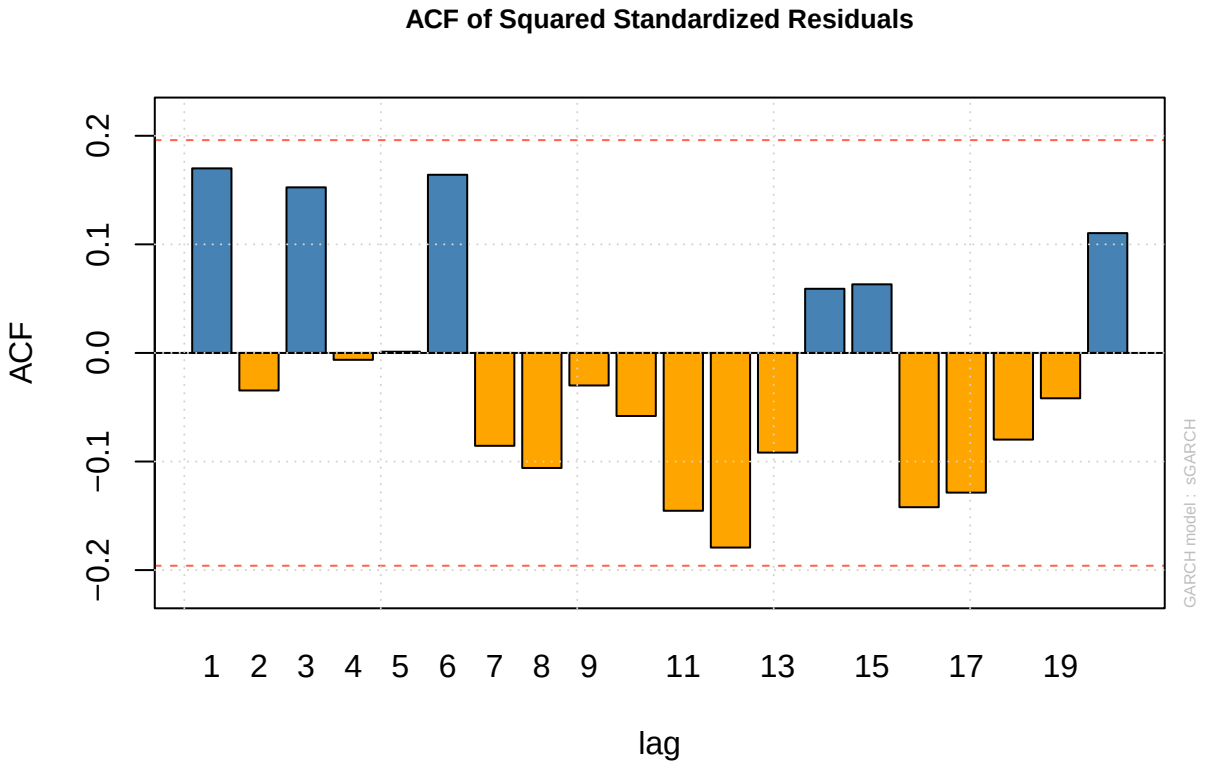

```
plot(fit_garch, which = 9)
```



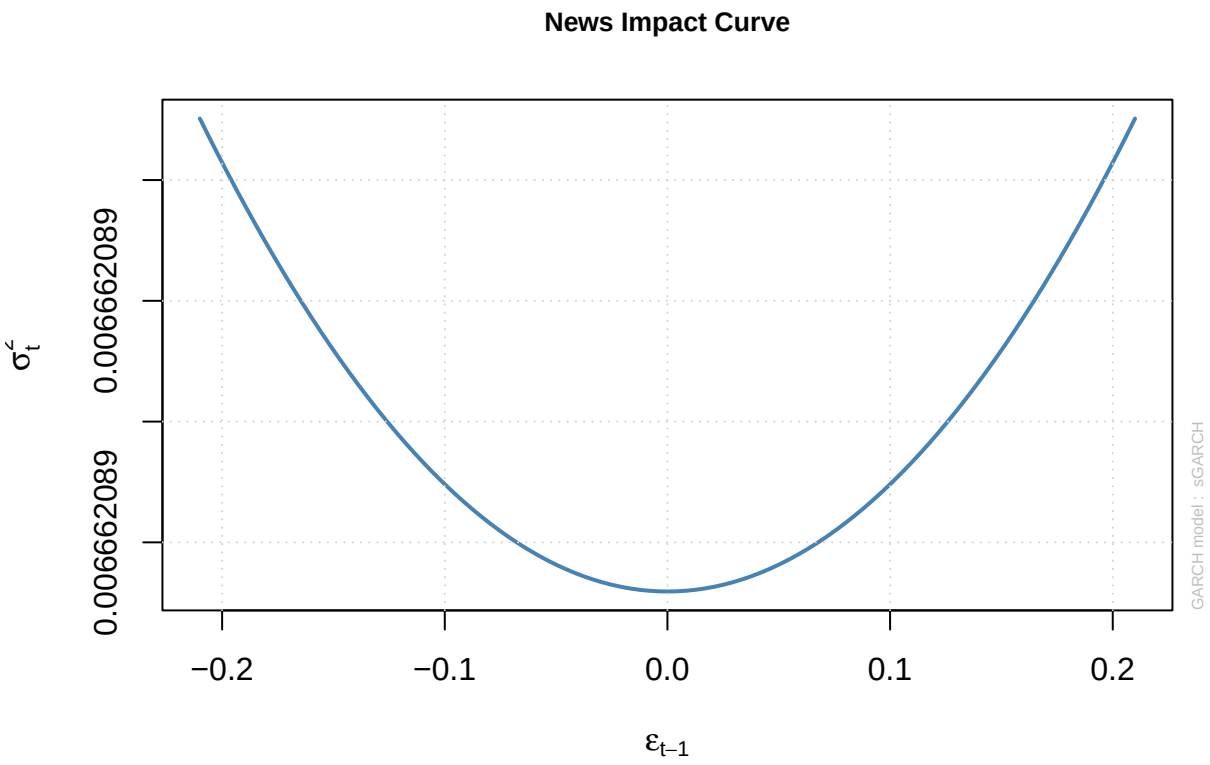
```
plot(fit_garch, which = 10)
```




```
plot(fit_garch, which = 11)
```



```
plot(fit_garch, which = 12)
```



News Impact Curve:

- The curve is symmetrical, indicating that both positive and negative shocks of equal magnitude have a similar impact on volatility.
- The U-shape shows that both positive and negative returns increase volatility equally, which is typical for symmetric GARCH models.

ACF of Squared Standardized Residuals:

- Most of the autocorrelation values fall within the confidence bands, indicating no significant autocorrelation in the squared residuals.
- This suggests that the GARCH model has successfully captured the volatility clustering in the data, meaning the model's volatility structure is adequate.

ACF of Standardized Residuals:

- The residuals appear to be white noise since most autocorrelations lie within the confidence bands.
- This suggests the mean equation part of the ARFIMA-GARCH model has been well specified and captures the data's dynamics effectively.

Q-Q Plot:

- The points largely follow the line, indicating the residuals are approximately normally distributed. Some deviations at the tails might suggest potential fat tails.

Empirical Density of Standardized Residuals:

- The residuals closely follow a normal distribution, implying that the model does not have significant misspecification in terms of distributional assumptions.

Cross-Correlations of Squared vs. Actual Observations:

- Ideally, there should be no significant correlation, implying that the squared returns (proxy for volatility) do not show further predictable patterns.
- The one significant bar indicates a potential additional pattern the model hasn't captured.

ACF of Absolute Observations:

- Absence of significant autocorrelations indicates that the GARCH model adequately captures volatility clustering.

ACF of Squared Observations:

- The lack of significant autocorrelation implies that the model has captured the serial dependence in the volatility of the series.

ACF of Observations:

- Lack of significant autocorrelation indicates that the mean model is capturing the dynamics effectively, suggesting an adequate ARFIMA fit.

Conditional Standard Deviation vs. Returns:

- The plot reveals the model's ability to track periods of high and low volatility. If the GARCH model is capturing the time-varying volatility adequately, the conditional SD should match periods of high returns variance.

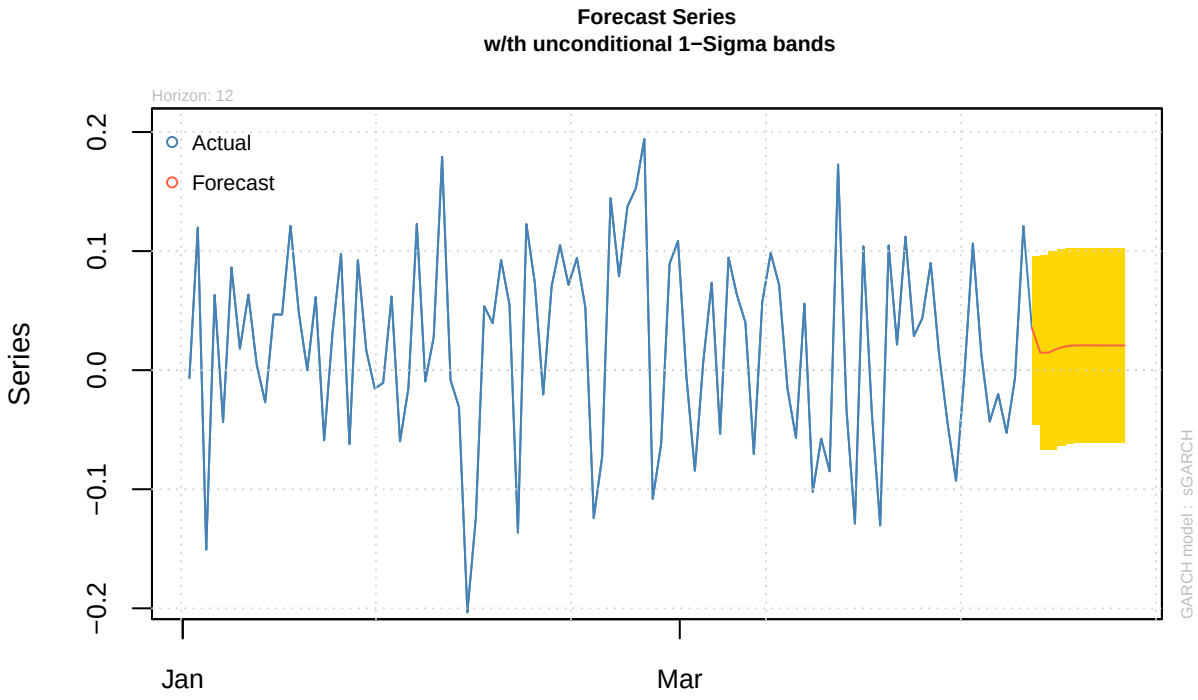
Series with 2 Conditional SD Superimposed:

- The red lines represent the 2 conditional standard deviations above and below the mean.
- The model fits well, most of the data points should fall within these bounds, indicating that the model accurately captures the volatility in the data.

Series with 1% VaR Limits:

- The green and red lines represent the upper and lower VaR limits, respectively.
- Overall the model fits well, the returns rarely exceed these limits, indicating that the model provides a good estimate of the risk.

```
# Forecasting with the GARCH model
forecast_garch <- ugarchforecast(fit_garch, n.ahead=12)
plot(forecast_garch, which=1)  # Forecast series
```



Forecasting

- The forecasted values are pretty close to the actual historical values towards the end of the time series
- The the red line appears relatively stable compared to the historical series, indicating that the model expects the future returns to stabilize somewhat, with less extreme fluctuations than observed in the past.
- The yellow bands widen as the forecast horizon extends, which is typical in time series forecasting. This widening indicates that uncertainty increases the further into the future we predict. The actual returns may deviate significantly from the forecasted line, especially as we move further into the forecast horizon.
- Since the historical series ends within the red line and yellow bands, it suggests that the model is capturing the central tendency of the time series well. However, the increasing width of the bands highlights that while the model expects less volatility, there is still considerable uncertainty about future values.
-

```
# Comparing Models using AIC and BIC
models <- list(ar_model, ma_model, arma_model1, arma_model2, arma_model3)
model_names <- c("AR", "MA", "ARIMA(1,1,1)", "ARIMA(2,1,1)", "ARIMA(2,1,2", "GARCH")

aic_values <- c(sapply(models, AIC), -2.0775)
bic_values <- c(sapply(models, BIC), -1.9473)

comparison <- data.frame(Model=model_names, AIC=aic_values, BIC=bic_values)
print(comparison)
```

Model Comparison

```
##          Model      AIC      BIC
## 1           AR 718.2679 728.6886
## 2           MA 718.1726 728.5933
## 3 ARIMA(1,1,1) 721.9588 729.7442
## 4 ARIMA(2,1,1) 717.1604 727.5408
## 5  ARIMA(2,1,2 719.1593 732.1349
## 6        GARCH  -2.0775  -1.9473
```

- Based on both AIC and BIC criteria, the GARCH model is the best-suited model for your data.
- GARCH is often preferred for financial time series data where volatility clustering is present, as it models changing variance over time, which ARIMA models do not handle well.
- The significantly lower AIC and BIC values for the GARCH model suggest it fits the data better, likely capturing patterns of volatility more effectively than the AR or ARIMA models.