AAPL R Code

```r
# Load necessary libraries
library(forecast)
library(tseries)
library(tidyverse)
library(readr)
library(ggplot2)
library(zoo)
library(TSA)
library(rugarch)
library(forecast)
library(PerformanceAnalytics)
library(xts)
library(quantmod)
```

AAPL Monthly Data 2016-2024

```r
# 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)
summary(aapl_monthly_data)
```

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

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

```r
summary(aaplmonthly_ts)
```

```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)
```

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

```r
# 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
}
```

```r
# 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)
```

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

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

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

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

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

# 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)")

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

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

```r
print(comparison)

# 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)")
```

```r
checkresiduals(arma_model2)
```

```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")
```

```r
# Forecasting with the ARIMA(2,1,1) model
arma2_forecast <- forecast(arma_model2, h=12)
plot(arma2_forecast, main="Forecasts from ARIMA(2,1,1)")
```

```r
# GARCH Models
# 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")
```

```r
# Specify ARMA(1,0)-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_garchAR1 <- ugarchfit(spec = spec_garch, data = returns)

# Specify ARMA(2,0)-GARCH(1,1) model
spec_garch <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
                         mean.model = list(armaOrder = c(2, 0)),
                         distribution.model = "norm")

fit_garchAR2 <- ugarchfit(spec = spec_garch, data = returns)

# Specify ARMA(0,2)-GARCH(1,1) model
spec_garch <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
                         mean.model = list(armaOrder = c(0, 2)),
                         distribution.model = "norm")

fit_garchMA2 <- ugarchfit(spec = spec_garch, data = returns)

# Specify ARMA(2,1)-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")
```

```r
fit_garchARMA21 <- ugarchfit(spec = spec_garch, data = returns)

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

fit_garchARMA22 <- ugarchfit(spec = spec_garch, data = returns)

# Comparing Models using AIC and BIC
model_names <- c("ARFIMA(1,0)-GARCH(1,1)", "ARFIMA(2,0)-GARCH(1,1)", "ARFIMA(0,2)-GARCH(1,1)", "ARFIMA(1

aic_values <- c(-2.0775, -2.0822, -2.0873, -2.0705, -2.0843)
bic_values <- c(-1.9473, -1.9259, -1.9310, -1.8882, -1.8759)

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

```r
# GARCH Model Assumptions Check
# Extract standardized residuals
std_residuals <- residuals(fit_garchMA2, standardize = TRUE)

# Remove time index, if present
std_residuals <- as.numeric(std_residuals)
checkresiduals(std_residuals)
```

```r
# Plot diagnostics
plot(fit_garchMA2, which = 1)
```

```r
plot(fit_garchMA2, which = 2)
```

```r
plot(fit_garchMA2, which = 3)
```

```r
plot(fit_garchMA2, which = 4)
```

```r
plot(fit_garchMA2, which = 5)
```

```r
plot(fit_garchMA2, which = 6)
```

```r
plot(fit_garchMA2, which = 7)
```

```r
plot(fit_garchMA2, which = 8)
```

```r
plot(fit_garchMA2, which = 9)
```

```r
plot(fit_garchMA2, which = 10)
```

```r
plot(fit_garchMA2, which = 11)
```

```r
plot(fit_garchMA2, which = 12)
```

```r
# Forecasting with the GARCH model
forecast_garch <- ugarchforecast(fit_garchMA2, n.ahead=12)
plot(forecast_garch, which=1)  # Forecast series
```