```
install.packages("ggplot2")
install.packages("tidyquant")
install.packages("urca")

# Load the necessary libraries
library(ggplot2)
library(tidyquant)
library(tseries)
library(forecast)
library(rugarch)
library(dplyr)
library(urca)

# Get the S&P 500 index data from Yahoo Finance
data <- tq_get("^GSPC", start = "2013-01-01", end = "2023-01-01")

View(data)

# Extract the closing price
closing_price <- data$close

# Convert the closing price to a data frame
closing_price_df <- as.data.frame(closing_price)

# Create a timeplot using ggplot2
ggplot(data, aes(x = date, y = close)) +
  geom_line() +
  labs(title = "S&P 500 (^GSPC) Timeplot",
       x = "Date",
       y = "Close Price") +
  theme_minimal()



# Perform the ADF test
adf_test <- ur.df(data$close, type = "trend", lags = 10)

# Print the ADF test results
print(adf_test)

adf_test1 <- adf.test(closing_price)
adf_test1
```

```r
# Calculate daily returns
daily_returns <- diff(data$close) / lag(data$close, 1)

# Print the first few daily returns
head(daily_returns)

# Create a timeplot using ggplot2
ggplot(data, aes(x = date, y = daily_returns)) +
  geom_line() +
  labs(title = "S&P 500 (^GSPC) Timeplot",
       x = "Date",
       y = "Close Price") +
  theme_minimal()

adf_test1 <- adf.test(daily_returns)
adf_test1

# Calculate the first difference
first_difference <- diff(closing_price)

# Perform the ADF test on the first difference
adf_test1 <- adf.test(first_difference)
adf_test1

# Plot first_difference
plot(first_difference, main = "GSPC first_difference")
# Plot ACF and PACF of first_difference
acf(first_difference, main = "ACF of GSPC first_difference")
pacf(first_difference, main = "PACF of GSPC first_difference")

daily_returns

acf_result <- acf(daily_returns[-1], main = "ACF Plot", lag.max= 20)
pacf_result <- pacf(daily_returns[-1], main = "PACF Plot", lag.max = 20)


# Fit ARIMA models
model_1 <- arima(daily_returns, order = c(1, 1, 1))
model_2 <- arima(daily_returns, order = c(1, 1, 2))
model_3 <- arima(daily_returns, order = c(1, 1, 4))
model_4 <- arima(daily_returns, order = c(2, 1, 1))
model_5 <- arima(daily_returns, order = c(2, 1, 2))
model_6 <- arima(daily_returns, order = c(2, 1, 4))
model_7 <- arima(daily_returns, order = c(4, 1, 1))
```

```
summary(model_3)
summary(model_1)
summary(model_2)
summary(model_4)
summary(model_5)
summary(model_6)
summary(model_7)


# Calculate AIC, BIC, and HQIC for each model
aic_values <- c(AIC(model_1), AIC(model_2), AIC(model_3), AIC(model_4), AIC(model_5),
AIC(model_6), AIC(model_7))
bic_values <- c(BIC(model_1), BIC(model_2), BIC(model_3), BIC(model_4), BIC(model_5),
BIC(model_6), BIC(model_7))
hqic_values <- -2 * aic_values + 2 * 3 * log(log(length(returns1)))

# Find the index of the model with the lowest AIC, BIC, and HQIC
best_model_aic <- which.min(aic_values)
best_model_bic <- which.min(bic_values)
best_model_hqic <- which.min(hqic_values)

# Print AIC, BIC, and HQIC values
cat("AIC Values:\n", aic_values, "\n")
cat("BIC Values:\n", bic_values, "\n")
cat("HQIC Values:\n", hqic_values, "\n")

# Print the best models based on each criterion
cat("Best Model (AIC): ", best_model_aic, "\n")
cat("Best Model (BIC): ", best_model_bic, "\n")
cat("Best Model (HQIC): ", best_model_hqic, "\n")


# Load required libraries

library(rugarch)


# Create a GARCH model specification
spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0)),
  distribution.model = "norm"
)
```

```
# Fit the model to the data
model <- ugarchfit(spec = spec, data = daily_returns[-1])

# Print the model summary
summary(model)
print(model)

daily_returns

spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0)),
  distribution.model = "norm"
)

model <- ugarchfit(spec = spec, data = daily_returns[-1])

forecast <- ugarchforecast(model, n.ahead = 20)

print(forecast)
plot(forecast)
1
3


#Taking the ideal ARIMA model of order (2,1,4)
spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(2, 1, 4)),
  distribution.model = "norm"
)

modelg2 <- ugarchfit(spec = spec, data = daily_returns[-1])

# Print the model summary
summary(modelg2)
print(modelg2)

forecast1 <- ugarchforecast(modelg2, n.ahead = 20)
(forecast1)

print(forecast1)
plot(forecast1)
```

1
3