ECON 144: Project 1

Shannan Liu, Zach Wrubel, Austin Pham

Contents

I) Introduction

Apple is a multinational technology company, listed on the NASDAQ stock exchange, that specializes in consumer electronics, computer software and online services. We pulled our data from Yahoo Finance consisting of the weekly stock prices of APPL over the past 10 years, starting from January 1, 2010 until present day.

As you will notice later, the APPL time series object itself has very insignificant partial autocorrelations making an autoregressive model hard to fit. This is inline with the classical challenge of predicting stock prices. To help solve this, we decided to take other relevant semiconductor companies to supplement our model with distributed lags.

The three main companies of focus were ON Semiconductor, VanEck Semiconductor ETF, and Taiwan Semiconductor. Semiconductors are used within many electronic appliances and are a vital eletronic component in the modern technologies that we use today. The number of transistors on an integrated circuit has roughly doubled every two years for the past fifty years, in line with Moore's Law.

According to the International Roadmap for Devices and Systems released in 2020, new semiconductor applications are now driving technological development and creating new innovation.We wanted to track the relationship between parallel industries of the growing semiconductor industry and its affect on a multinational technology company like Apple.

```
appl <-  read.csv("AAPL.csv", sep = ",")
on <-  read.csv("ON.csv", sep = ",")
smh <-  read.csv("SMH.csv", sep = ",")
tsm <-  read.csv("TSM.csv", sep = ",")

appl_ts <- ts(appl[, 5], start = 2010, freq = 52)
on_ts <- ts(on[, 5], start = 2010, freq = 52)
smh_ts <- ts(smh[, 5], start = 2010, freq = 52)
tsm_ts <- ts(tsm[, 5], start = 2010, freq = 52)
```

II) Results

1) Modeling and Forecasting Trend

A) Time-Series Plot

```r
par(mfrow = c(2, 2))
plot(appl_ts,
     xlab = "Time (Year)",
     ylab = "Price USD",
     main = "Apple Stock")
plot(on_ts,
     xlab = "Time (Year)",
     ylab = "Price USD",
     main = "ON Semiconductor Stock")
plot(smh_ts,
     xlab = "Time (Year)",
     ylab = "Price USD",
     main = "VanEck Semiconductor ETF")
plot(tsm_ts,
     xlab = "Time (Year)",
     ylab = "Price USD",
     main = "Taiwan Semiconductor")
```

Apple Stock

ON Semiconductor Stock

VanEck Semiconductor ETF

Taiwan Semiconductor
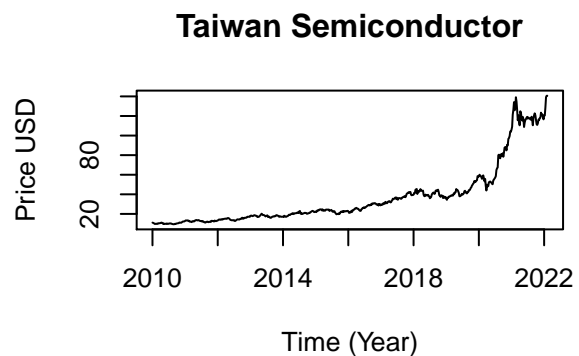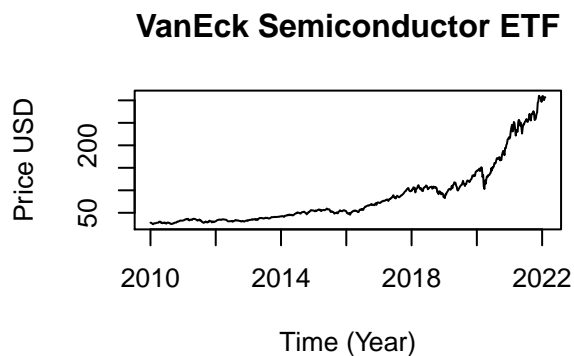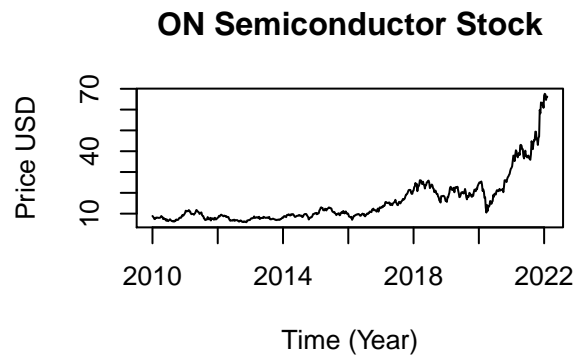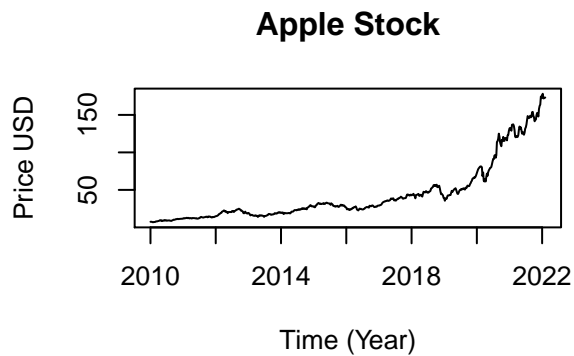
B) Covariance Stationary

The Apple time series does not appear to be covariance stationary. The object is not mean reverting as each random variable appears to have non-constant means. This can be seen in the positive linear trend in the time series. Additionally, it appears that the variance structure is not constant over time as well. Starting at around year 2018-2019, we can see a drastic increase in fluctuation overall. This is contrasted to the slow, steady, lienar growth of Apple in the years prior.

C) Autocorrelation & Partial ACF

As expected in a time series involving a stock price, the Autocorrelation graph shows highly correlated lags. This is of no surprise, especially since we are observing at a weekly periodicity. The stock price a week ago should be highly correlated to the stock price now. That affect is only compounded across several months and years.

Regarding the Partial Autocorrelation graph, we can see that there are no individual lags that are statistically significant. This signifies that though previous stock prices are overall correlated to stock prices now, we cannot firmly attribute any one lag to the overall stock price now. This is the nature of stock prices and is the reason as to why they are so hard to model.

```
par(mfrow = c(1, 2))
acf(appl_ts, main = "Autocorrelation")
pacf(appl_ts, main = "Partial Autocorrelation")
```
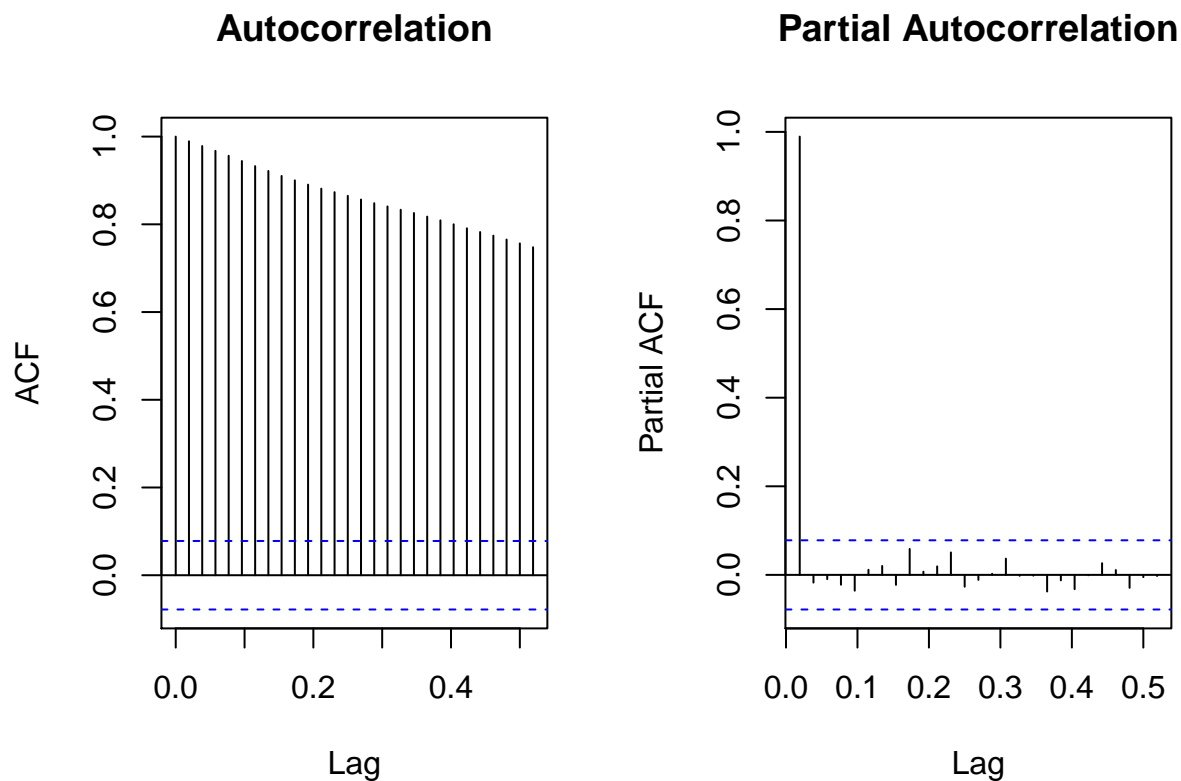
D) Linear & Nonlinear Models

```
t <- seq(2010, 2022, length = length(appl_ts))
m1 <- lm(appl_ts ~ t + on_ts + smh_ts + tsm_ts)
m2 <- lm(appl_ts ~ t + on_ts + smh_ts + I(smh_ts^2) + tsm_ts + I(t ^ 2))
m3 <- lm(log(appl_ts) ~ t + on_ts + smh_ts + tsm_ts)
ds <- data.frame(x = t, y = appl_ts)
m4 <- nls(y ~ exp(a + b * t), data = ds, start = list(a = 0, b = 0))

summary(m1)
```

```
##
## Call:
## lm(formula = appl_ts ~ t + on_ts + smh_ts + tsm_ts)
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -18.1310  -3.4409  -0.7659   2.6238  24.6829
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2435.64373  276.49923   8.809   <2e-16 ***
## t             -1.21193    0.13739  -8.821   <2e-16 ***
## on_ts         -0.98350    0.07615 -12.915   <2e-16 ***
## smh_ts         0.74629    0.03397  21.968   <2e-16 ***
## tsm_ts         0.06751    0.05098   1.324    0.186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.34 on 625 degrees of freedom
## Multiple R-squared:  0.9818, Adjusted R-squared:  0.9817
## F-statistic:  8414 on 4 and 625 DF,  p-value: < 2.2e-16
```

```
summary(m2)
```

```
##
## Call:
## lm(formula = appl_ts ~ t + on_ts + smh_ts + I(smh_ts^2) + tsm_ts +
##     I(t^2))
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -18.2093  -2.8608  -0.6775   2.1633  26.1931
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.313e+05  1.928e+05  -0.681    0.496
## t            1.304e+02  1.917e+02   0.681    0.496
## on_ts       -1.177e+00  8.278e-02 -14.220  < 2e-16 ***
## smh_ts       6.055e-01  6.294e-02   9.621  < 2e-16 ***
## I(smh_ts^2)  5.415e-04  9.129e-05   5.932 4.97e-09 ***
## tsm_ts       8.689e-03  5.544e-02   0.157    0.876
```

5

```
## I(t^2)      -3.238e-02  4.762e-02  -0.680     0.497
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.134 on 623 degrees of freedom
## Multiple R-squared:  0.9832, Adjusted R-squared:  0.983
## F-statistic:  6078 on 6 and 623 DF,  p-value: < 2.2e-16
```

```r
summary(m3)
```

```
##
## Call:
## lm(formula = log(appl_ts) ~ t + on_ts + smh_ts + tsm_ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41510 -0.09562 -0.01306  0.08093  0.44964
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.807e+02  7.762e+00 -36.163  < 2e-16 ***
## t            1.408e-01  3.857e-03  36.499  < 2e-16 ***
## on_ts       -1.014e-02  2.138e-03  -4.743 2.61e-06 ***
## smh_ts       4.283e-03  9.537e-04   4.491 8.45e-06 ***
## tsm_ts       3.692e-03  1.431e-03   2.580   0.0101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1499 on 625 degrees of freedom
## Multiple R-squared:  0.963,  Adjusted R-squared:  0.9628
## F-statistic:  4070 on 4 and 625 DF,  p-value: < 2.2e-16
```

```r
summary(m4)
```

```
##
## Formula: y ~ exp(a + b * t)
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## a -6.372e+02  9.074e+00  -70.22   <2e-16 ***
## b  3.176e-01  4.491e-03   70.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.12 on 628 degrees of freedom
##
## Number of iterations to convergence: 9
## Achieved convergence tolerance: 8.782e-06
```

```r
par(mfrow = c(2, 2))
plot(appl_ts,
     xlab = "Time (Year)",
```

```
      ylab = "Price USD",
      main = "Linear Model")
lines(t, m1$fit, col = "skyblue", lwd = 2)

plot(appl_ts,
      xlab = "Time (Year)",
      ylab = "Price USD",
      main = "Quadratic Model")
lines(t, m2$fit, col = "skyblue", lwd = 2)

plot(log(appl_ts),
      xlab = "Time (Year)",
      ylab = "Price USD",
      main = "Log Linear Model")
lines(t, m3$fit, col = "skyblue", lwd = 2)

plot(appl_ts,
      xlab = "Time (Year)",
      ylab = "Log Price USD",
      main = "Exponential Model")
lines(ds$x, predict(m4, list(x = ds$x)), col = "skyblue", lwd = 2)
```
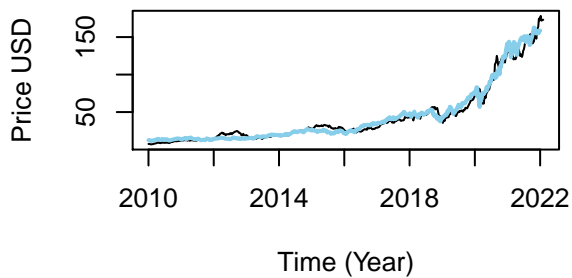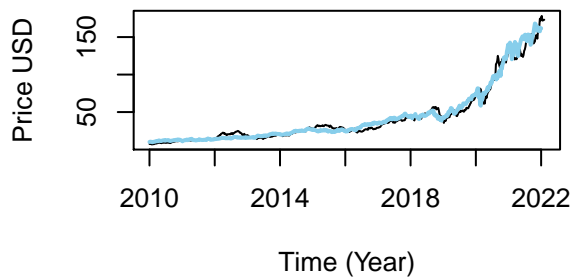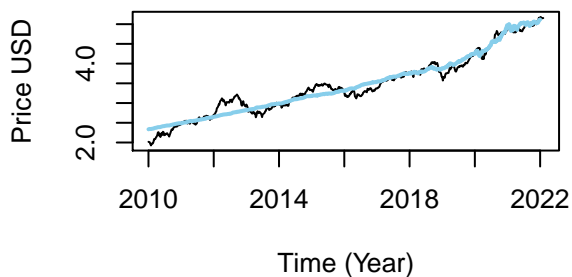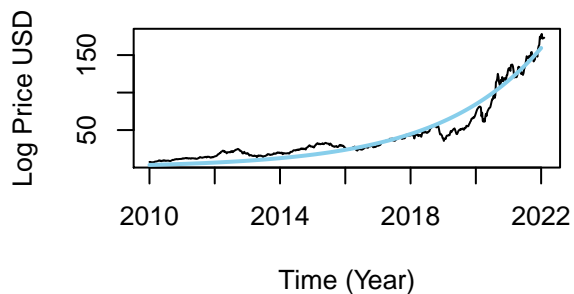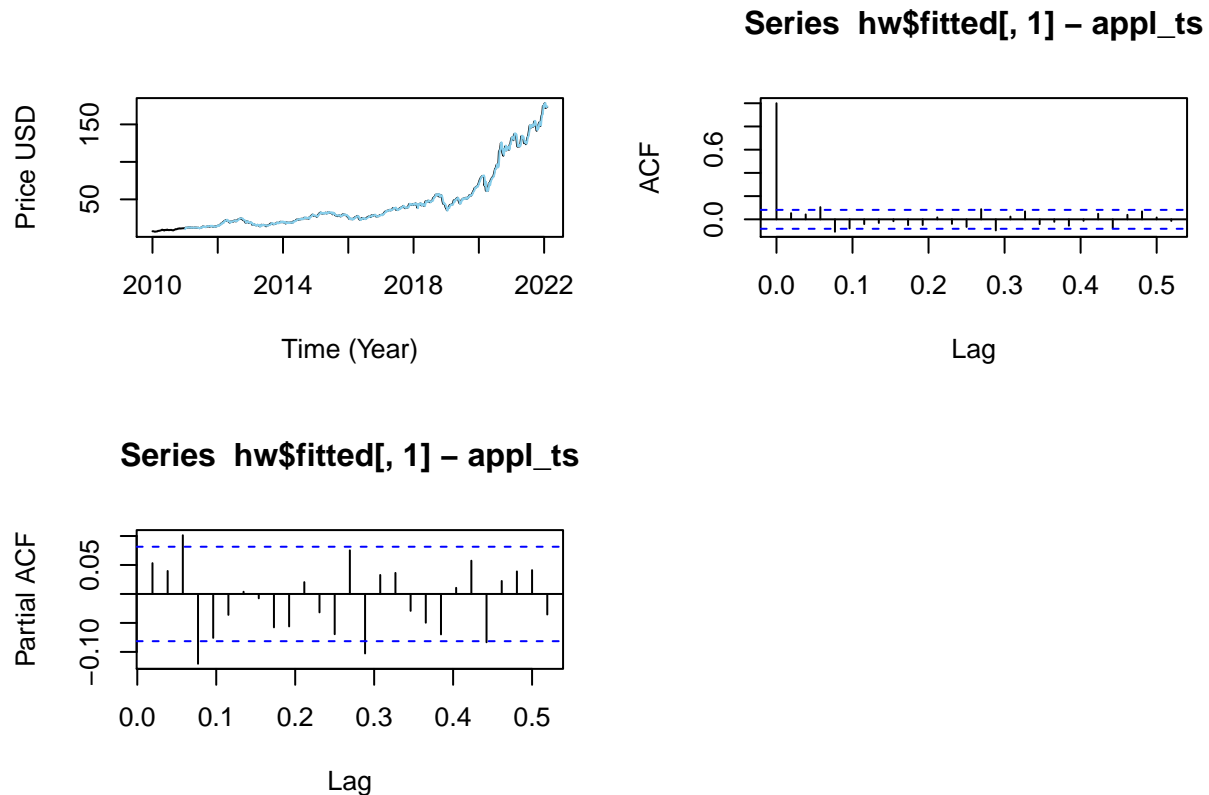


```
hw <- HoltWinters(appl_ts)
plot(appl_ts, xlab = "Time (Year)", ylab = "Price USD")
lines(hw$fitted[, 1], col = "skyblue")
```

```
hw_res <- appl_ts - hw$fitted[, 1]
acf(hw$fitted[, 1] - appl_ts)
pacf(hw$fitted[, 1] - appl_ts)
```


E) Residuals vs. Fitted Values Plots for all Models

Looking at the residuals versus fitted values plots reveals strong patterns for the log-linear model, and exponential model. In each case, it is clear the residuals are dependent on x as well as the error term.

In the linear model, the residuals demonstrate somewhat of a pattern at the beginning, then disperse as the fitted values increase. This is similar to the pattern shown in the quadratic model.

The Log-Linear Model suggests strong cycles in the data. Although it has weaker patterns, the exponential model also suggests the presence of strong cycles along with displaying a pattern between the residuals and x.

The Holt-Winters residual plot is the most well behaved with the residuals clustered around 0 and no apparent pattern, suggesting the residuals mostly depend on the error term and the model's assumptions are not violated.

```
par(mfrow = c(3, 2))
plot(m1$fit,m1$res,
     main="Linear Model",
     ylab="Residuals",
     xlab="Fitted Values")
```

```r
plot(m2$fit,m2$res,
     main="Quadratic Model",
     ylab="Residuals",
     xlab="Fitted Values")

plot(m3$fit,m3$res,
     main="Log-Linear Model",
     ylab="Residuals",
     xlab="Fitted Values)")

plot(predict(m4, list(x = ds$x)),residuals(m4),
     main="Exponential Model",
     ylab="Residuals",
     xlab="Fitted Values")

plot(hw$fitted[,1], hw_res,
     main = "Holt-Winters",
     ylab = "Residuals",
     xlab = "Fitted Values")
```

Linear Model

Quadratic Model

Log–Linear Model

Exponential Model
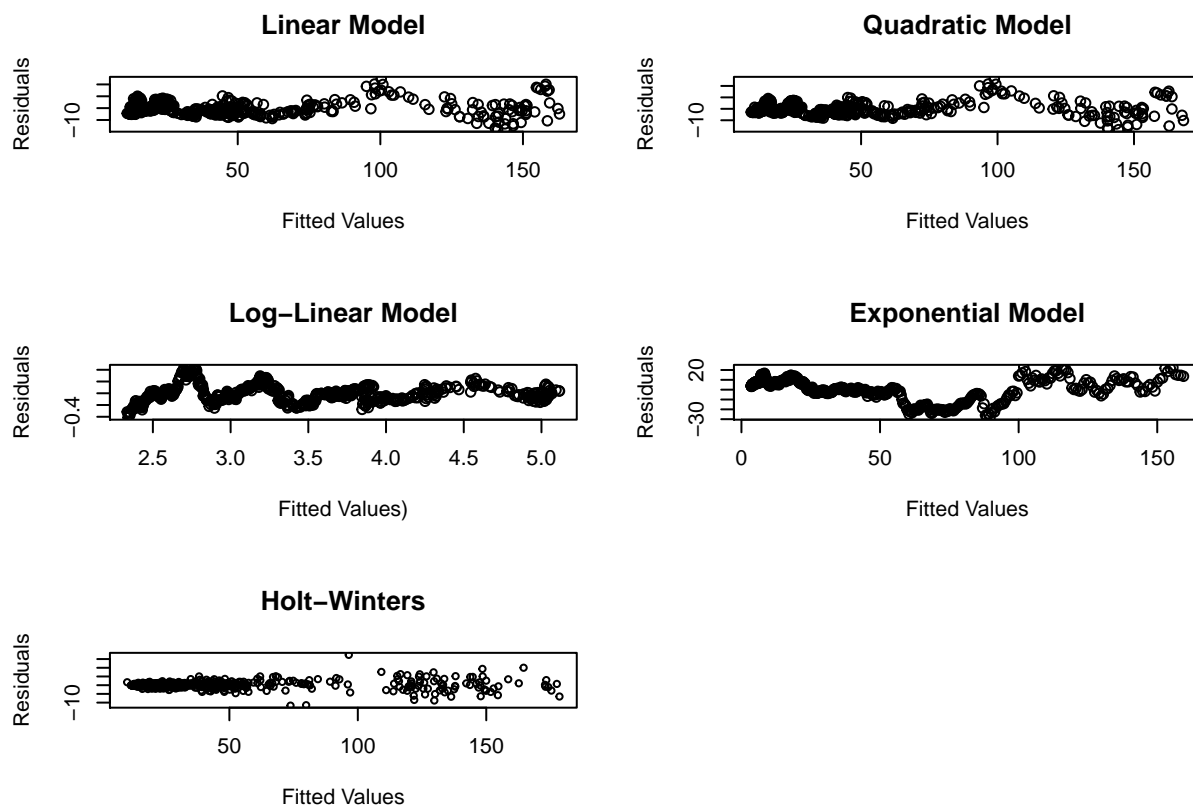
Holt–Winters

F) Histogram of Residuals

Similarly to the residual versus fitted values plots, these histograms reinforce Holt-Winters being the best model. The linear model and quadratic model's residuals are both skewed right and not centered around 0.

The exponential model's residuals are skewed left. Both the Holt-Winters and the Log-Linear Model's residuals have an approximately normal distribution. The Holt-Winters model has the most normal distribution of residuals with both a mean of 0.11 and median of 0.021. Thus, the Holt-Winters residuals appear to show the variance is normally distributed and the model's assumptions have not been violated.
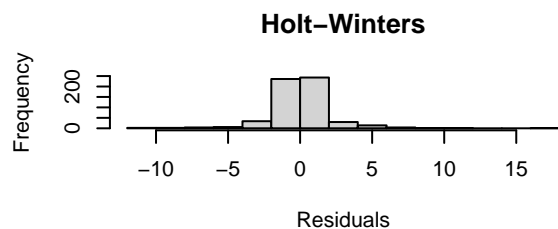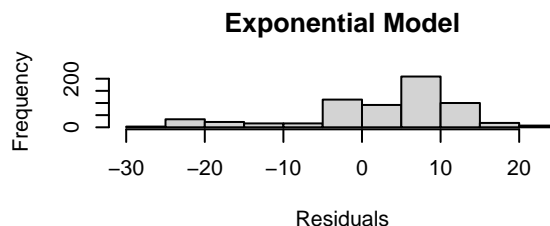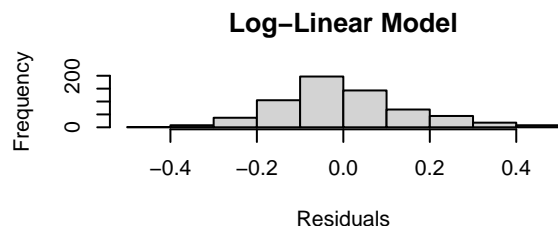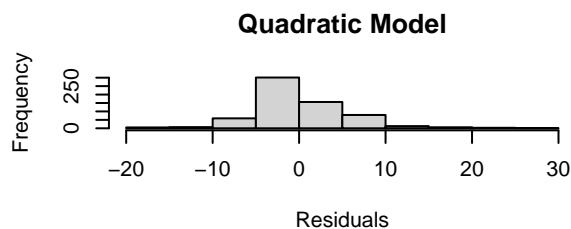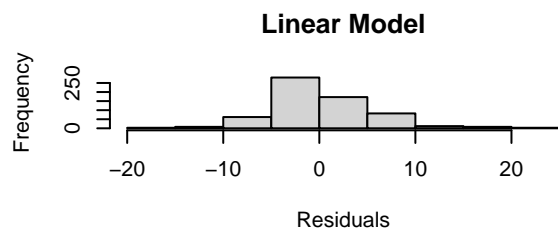
```r
par(mfrow = c(3, 2))
hist(m1$residuals,
     main="Linear Model",
     xlab="Residuals")

hist(m2$residuals,
     main="Quadratic Model",
     xlab="Residuals")

hist(m3$residuals,
     main="Log-Linear Model",
     xlab="Residuals")

hist(residuals(m4),
     main="Exponential Model",
     xlab="Residuals")

hist(hw_res,
     main="Holt-Winters",
     xlab="Residuals")
```

G) Discuss Model Diagnostics

The Linear model has an adjusted R-squared of 0.9817, a MSR of 28.29044, and the slope coefficient is statistically significant. The Quadratic model has an adjusted R-squared of 0.89, a MSR of 26.06315, and both slope coefficients are statistically significant. The Log-Linear Model has an adjusted R-squared of 0.9241, a MSR of 0.02229463, and the slope coefficient is statistically significant. The Exponential Model has a residual standard error of 10.12 on 628 degrees of freedom, a MSR of 102.144, and 9 iterations to convergence. The Holt-Winters Model has a MSR of .531779

Based on these summary statistics, the Log-Linear Model and the Holt-Winters Model are the most appropriate.

```
#Compare adjusted R-Squared of each model
summary(m1)
```

```
##
## Call:
## lm(formula = appl_ts ~ t + on_ts + smh_ts + tsm_ts)
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -18.1310  -3.4409  -0.7659   2.6238  24.6829
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2435.64373  276.49923   8.809   <2e-16 ***
## t             -1.21193    0.13739  -8.821   <2e-16 ***
## on_ts         -0.98350    0.07615 -12.915   <2e-16 ***
## smh_ts         0.74629    0.03397  21.968   <2e-16 ***
## tsm_ts         0.06751    0.05098   1.324    0.186
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.34 on 625 degrees of freedom
## Multiple R-squared:  0.9818, Adjusted R-squared:  0.9817
## F-statistic:  8414 on 4 and 625 DF,  p-value: < 2.2e-16
```

```
summary(m2)
```

```
##
## Call:
## lm(formula = appl_ts ~ t + on_ts + smh_ts + I(smh_ts^2) + tsm_ts +
##     I(t^2))
##
## Residuals:
##      Min      1Q   Median      3Q      Max
## -18.2093  -2.8608  -0.6775   2.1633  26.1931
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.313e+05  1.928e+05  -0.681    0.496
## t            1.304e+02  1.917e+02   0.681    0.496
## on_ts       -1.177e+00  8.278e-02 -14.220  < 2e-16 ***
```

```
## smh_ts       6.055e-01  6.294e-02   9.621  < 2e-16 ***
## I(smh_ts^2)  5.415e-04  9.129e-05   5.932 4.97e-09 ***
## tsm_ts       8.689e-03  5.544e-02   0.157    0.876
## I(t^2)      -3.238e-02  4.762e-02  -0.680    0.497
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.134 on 623 degrees of freedom
## Multiple R-squared:  0.9832, Adjusted R-squared:  0.983
## F-statistic:  6078 on 6 and 623 DF,  p-value: < 2.2e-16
```

summary(m3)

```
##
## Call:
## lm(formula = log(appl_ts) ~ t + on_ts + smh_ts + tsm_ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41510 -0.09562 -0.01306  0.08093  0.44964
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.807e+02  7.762e+00 -36.163  < 2e-16 ***
## t            1.408e-01  3.857e-03  36.499  < 2e-16 ***
## on_ts       -1.014e-02  2.138e-03  -4.743 2.61e-06 ***
## smh_ts       4.283e-03  9.537e-04   4.491 8.45e-06 ***
## tsm_ts       3.692e-03  1.431e-03   2.580   0.0101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1499 on 625 degrees of freedom
## Multiple R-squared:  0.963,  Adjusted R-squared:  0.9628
## F-statistic:  4070 on 4 and 625 DF,  p-value: < 2.2e-16
```

summary(m4)

```
##
## Formula: y ~ exp(a + b * t)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## a -6.372e+02  9.074e+00  -70.22   <2e-16 ***
## b  3.176e-01  4.491e-03   70.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.12 on 628 degrees of freedom
##
## Number of iterations to convergence: 9
## Achieved convergence tolerance: 8.782e-06
```

```r
summary(hw$fitted)
```

```
##       xhat              level            trend              season
##  Min.   : 10.22   Min.   :  9.438   Min.   :-0.04341   Min.   :-1.37478
##  1st Qu.: 20.68   1st Qu.: 20.861   1st Qu.: 0.05601   1st Qu.:-0.35031
##  Median : 30.11   Median : 29.537   Median : 0.10482   Median : 0.01235
##  Mean   : 46.56   Mean   : 46.375   Mean   : 0.17543   Mean   : 0.00851
##  3rd Qu.: 51.53   3rd Qu.: 51.902   3rd Qu.: 0.15242   3rd Qu.: 0.46741
##  Max.   :178.51   Max.   :177.127   Max.   : 0.85194   Max.   : 1.10385
```

```r
#Compare MSR
mean((m1$residuals)^2)
```

```
## [1] 28.29044
```

```r
mean((m2$residuals)^2)
```

```
## [1] 26.06315
```

```r
mean((m3$residuals)^2)
```

```
## [1] 0.02229463
```

```r
mean((residuals(m4))^2)
```

```
## [1] 102.144
```

```r
mean((hw_res)^2)
```

```
## [1] 4.531779
```

H) Select Trend Model using AIC and BIC

The Log-Linear has the lowest AIC (-596.2854) and lowest BIC (-569.6111), therefore it is the best Trend model out of the rest. However, as seen in the graphs above, it doesn't model Apple's stock price information as accurately as the Linear, Quadratic, or Holt-Winters models.

```r
AIC(m1,m2,m3,m4)
```

```
##    df       AIC
## m1  6 3905.6526
## m2  8 3857.9916
## m3  6 -596.2854
## m4  3 4708.4842
```

```r
BIC(m1,m2,m3,m4)
```

```
##    df       BIC
## m1  6 3932.3269
## m2  8 3893.5573
## m3  6 -569.6111
## m4  3 4721.8214
```

I) Forecast 16 Weeks Ahead Using Preferred Model

We will now forecast the next year of weekly APPL closing prices using Holt-Winters model. This model was preferred because it had residuals that most resembled white noise, meaning that it captured the structure of our data the best. In addition to that, Holt-Winters can generate predictions without requiring future stock price data from Taiwan Semiconductor, ON Semiconductor, and VanEck Semiconductor ETF, making it an easier option to plot.

```
#Holt-Winters Filtering
#quartz()
hwpred <- predict(hw, 52, prediction.interval = TRUE,level=0.5)
plot(hw,hwpred,
     main="Weekly APPL Closing Price Prediction using Holt-Winters Filtering",
     ylab="Price (USD)",
     xlab="Time (Year)",
     xlim=c(2010,2024))
```

Weekly APPL Closing Price Prediction using Holt–Winters Filtering
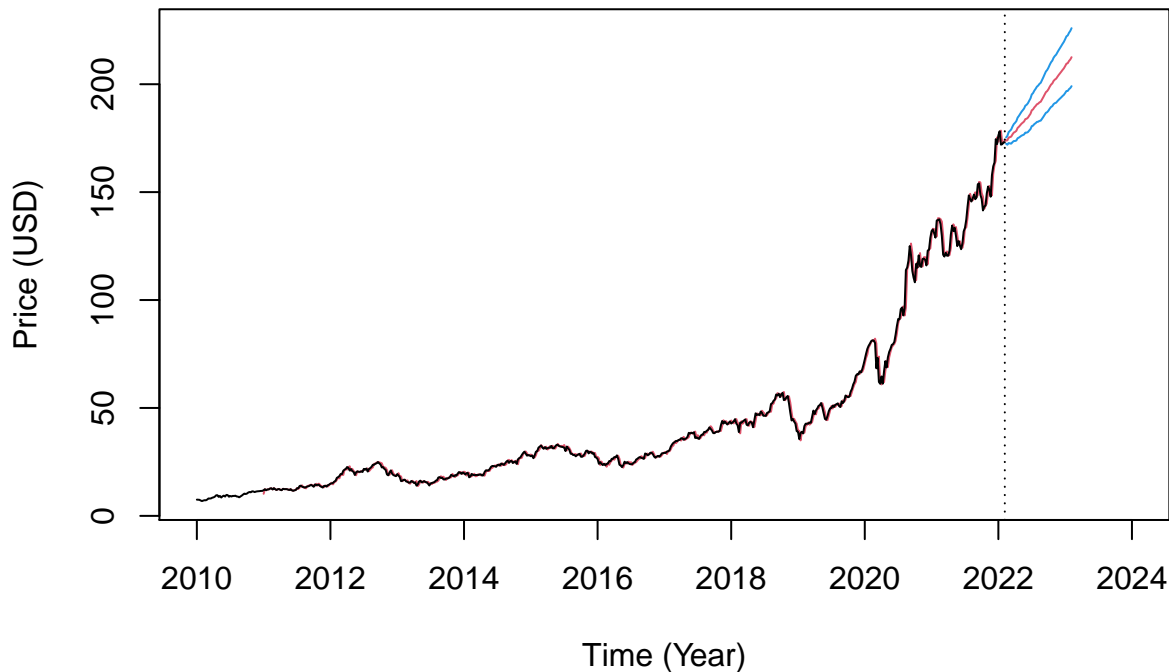

```
#Holt-Winters Forecast
plot(forecast(hw,h = 52),
     main="Weekly APPL Closing Price Forecast using Holt-Winters Forecast",
     xlab="Time (Year)",
     ylab="Price (USD)")
```

Weekly APPL Closing Price Forecast using Holt–Winters Forecast
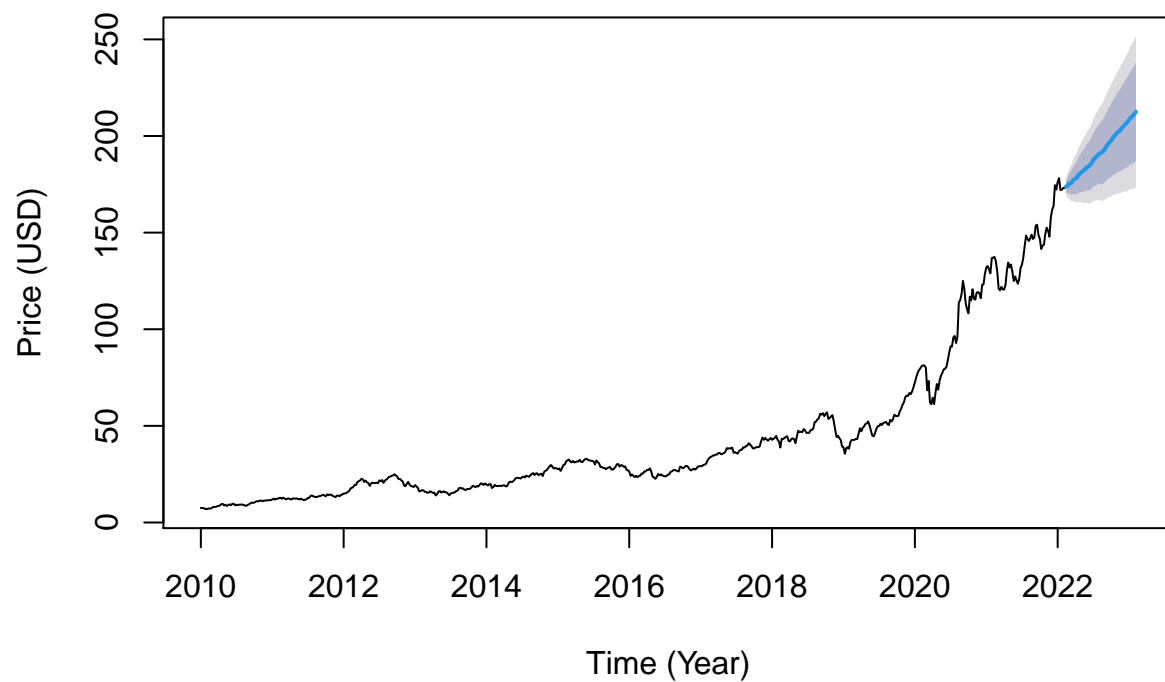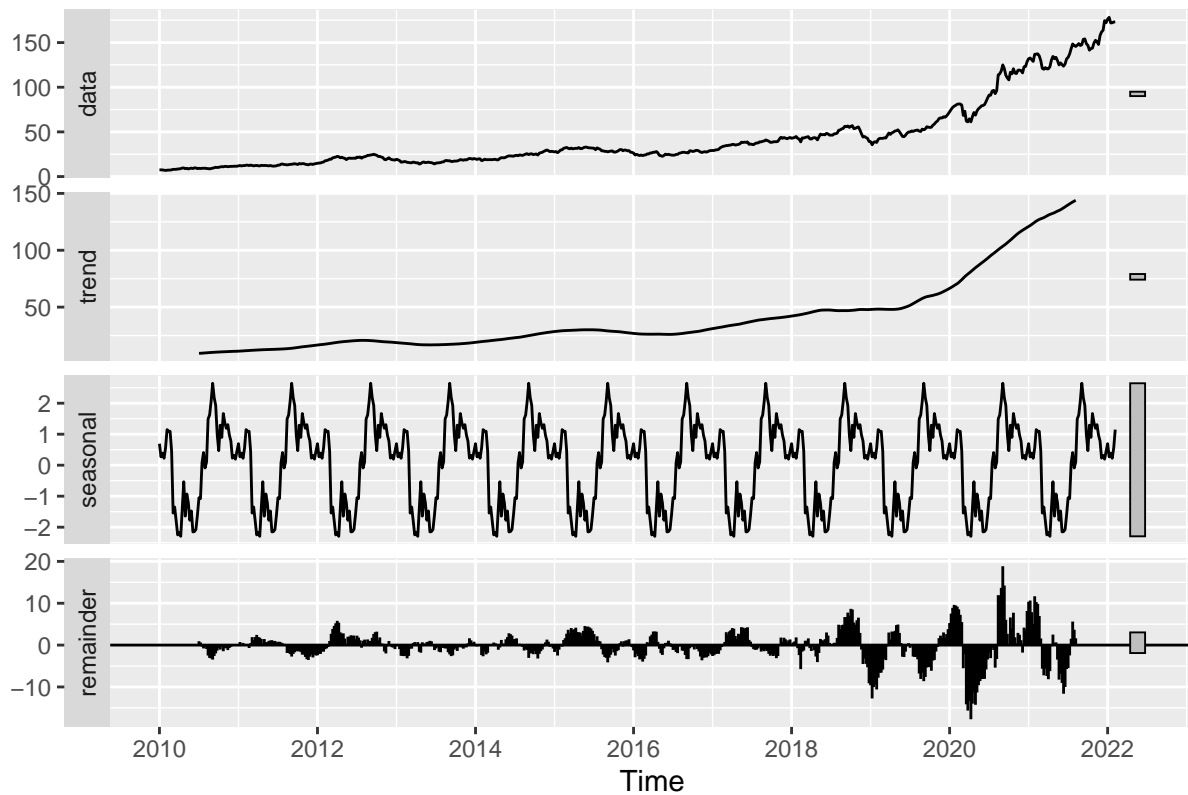

2) Trend and Seasonal Adjustments

A) Perform an additive decomposition of your series

```
# Perform additive decomposition and plot the decomposition
a_dcmp = decompose(appl_ts, "additive")
autoplot(a_dcmp)
```

Decomposition of additive time series


```
# Detrend and Seasonally Adjust the series
detrend_seas_adj_appl = appl_ts - a_dcmp$trend - a_dcmp$seasonal

# Comment on the ACF and PACF of the residuals
tsdisplay(detrend_seas_adj_appl,
          main = "Apple's Adjusted Closing Price
(Detrended and Seasonally Adjusted)")
```

**Apple's Adjusted Closing Price
(Detrended and Seasonally Adjusted)**

The residuals plot exhibits heteroskedasticity, and it looks like it can be modelled to a certain extent. The ACF and PACF plots of the residuals affirm this visual intuition, showing that there is still structure to our data. In particular, the ACF correlogram shows evidence of significant autocorrelation beyond 100 weeks. Meanwhile the PACF also shows some significant correlations, albeit not as strong as the ACF graph. These graphs indicate that additive decomposition does not fully capture the patterns exhibited in Apple's weekly adjusted closing price.

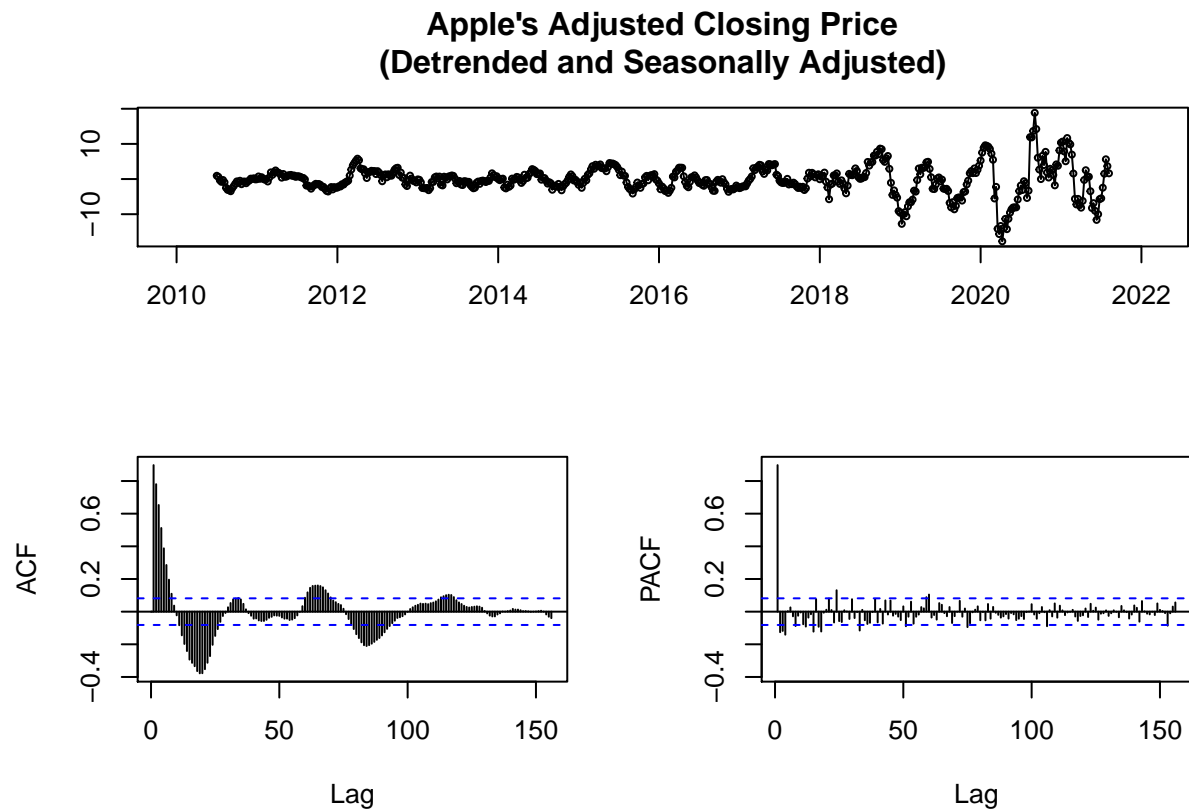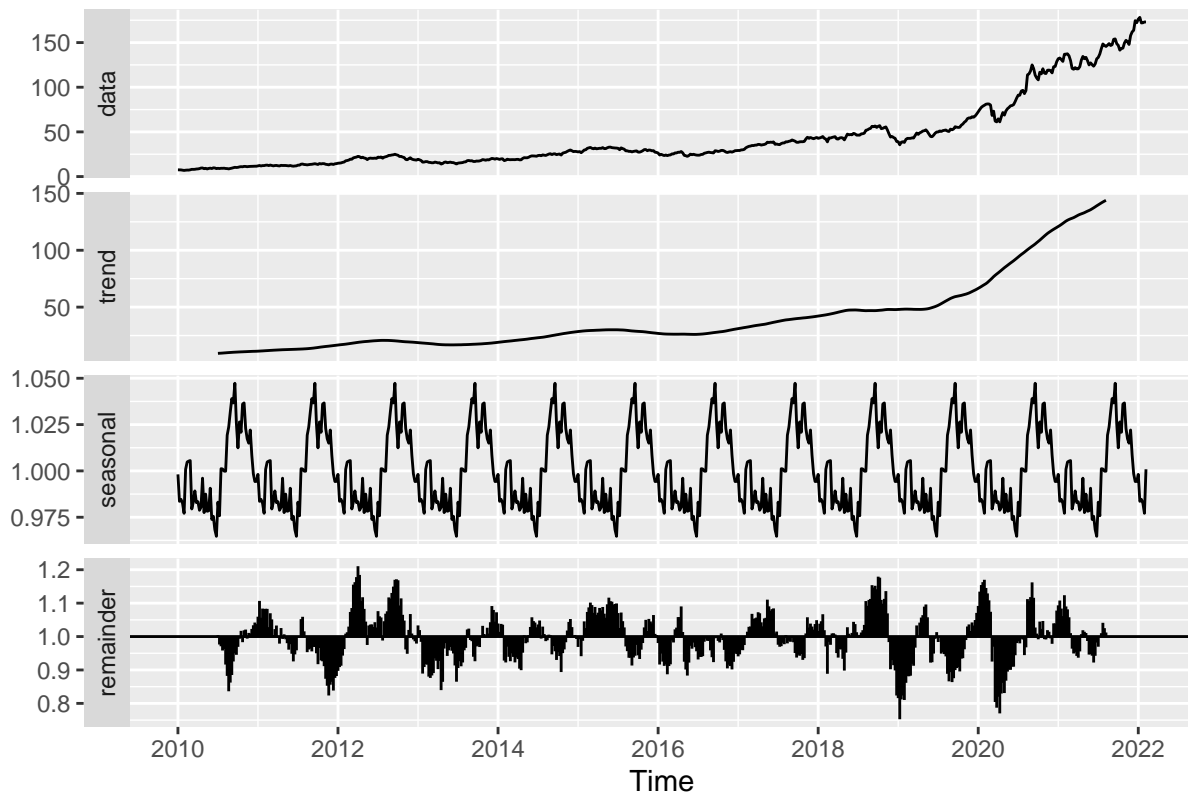**A) Perform a multiplicative decomposition of your series**

```
# Perform additive decomposition and plot the decomposition
m_dcmp = decompose(appl_ts, "multiplicative")
autoplot(m_dcmp)
```

Decomposition of multiplicative time series


```r
# Detrend and Seasonally Adjust the series
detrend_seas_adj_appl = (appl_ts / m_dcmp$trend) / m_dcmp$seasonal

# Comment on the ACF and PACF of the residuals
tsdisplay(detrend_seas_adj_appl,
          main = "Apple's Adjusted Closing Price
(Detrended and Seasonally Adjusted)")
```

Apple's Adjusted Closing Price
(Detrended and Seasonally Adjusted)
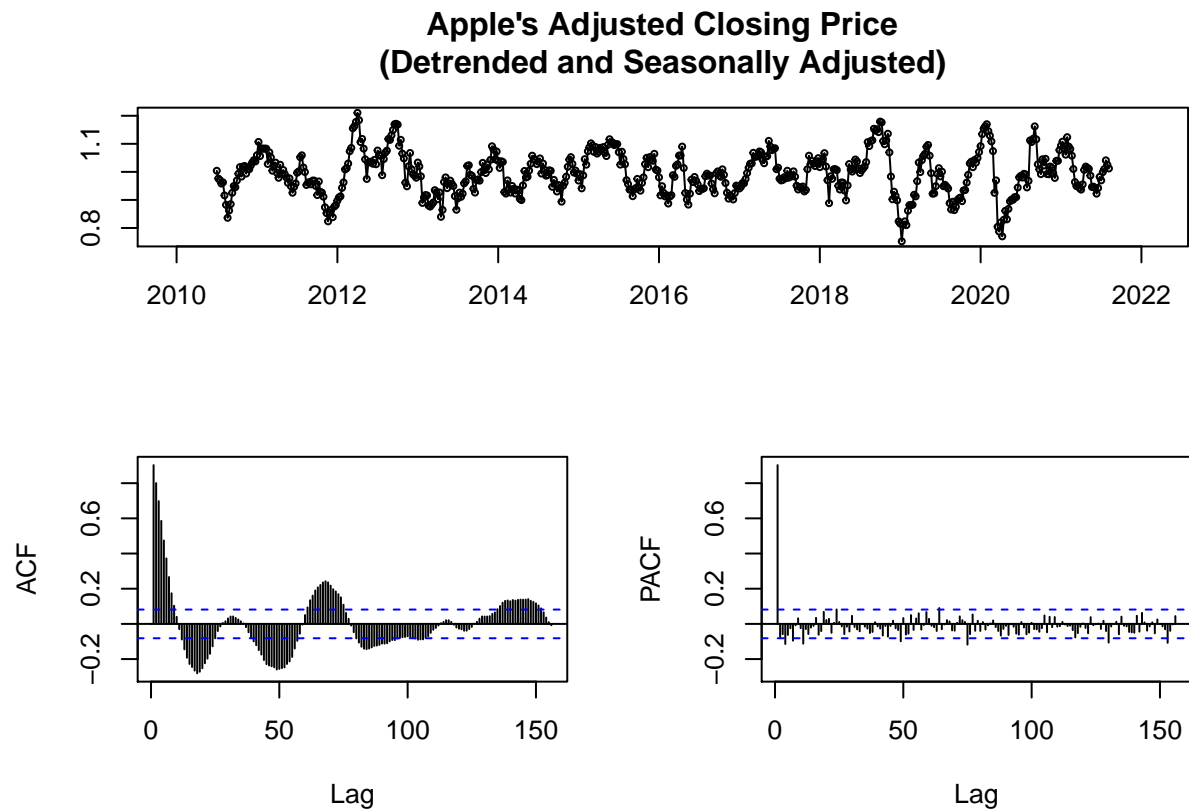
The residuals plot exhibits homoskedasticity, and shows signs of more regularity than the additive decomposition residuals plot. The ACF and PACF plots of the residuals affirm this visual intuition, showing that there is still structure to our data. In particular, the ACF correlogram shows evidence of significant autocorrelation even at 140 weeks beyond the present period. Meanwhile the PACF also shows some significant correlations, albeit much smaller than the ACF graph. These graphs indicate that the multiplicative decomposition did not fully capture the patterns in our data.

C) Which decomposition is better, additive or multiplicative? Why?

The additive decomposition is better.

Although both decompositions' residuals plots show patterns, the multiplicative decomposition's residuals plot exhibits more structure than the graph from the additive decomposition. This is evidenced by the more persistent correlations in the ACF and PACF graphs of the multiplicative decomposition. This result means that the multiplicative decomposition captured less of our data's original structure than the additive decomposition. Therefore, the additive decomposition is better.

D) Based on the two decompositions, and interpretation of the random components, would your models for the cycles be similar (additive vs. multiplicative) or very different? Why?

The models for the cycles would differ between the additive and multiplicative decompositions.

This is because the residuals data under multiplicative decomposition exhibits heteroskedasticity and more regularly spaced oscillations around a mean. On the other hand, the additive decomposition's residuals show heteroskedasticity and oscillations that are less regularly spaced. Also, after observing the scales of the data, it can be seen that the additive decomposition's residuals have a much larger amplitude than the

multiplcative decomposition's residuals. These differences suggest that models for the cycles for the two decompositions would be very different.

III) Conclusions & Future Work

We chose to forecast Apple's adjusted closing price with the Holt-Winters model. However, the linear and quadratic models produced promising visualizations that mapped closely to Apple's stock price. The Holt-Winters model indicates that the price of Apple's stock will continue to rise in the coming year. It also shows that the price will continue to rise without deviating much from a linear upward trajectory. Based on this information, our model suggests that investing in Apple is profitable.

However, this conclusion is limited to the data that we've used to produce our forecasts. Even though we modeled our predictions of Apple's price using the closing prices of semiconductor companies that are highly relevant to the production of Apple's flagship products such as the iPhone and MacBook, our models do not account for how economic shocks, changes in consumer preferences, or other macroeconomic factors could affect Apple's stock performance. For instance, a drastic increase in oil prices could affect Apple's material acquisition costs and also its ability to ship products around the world. This type of impact is not measured in our model. Therefore the use of our model is limited.

We can improve on our models by including more factors in our model to account for exogenous economic shocks or technological advancements. To obtain more accurate predictions, we can also implement different modelling techniques such as Prophet Forecasting developed by Facebook's Core Data Science Team.

IV) References

https://irds.ieee.org/topics/new-semiconductor-technologies-and-applications

https://www.synopsys.com/glossary/what-is-moores-law.html#:~:text=Moore's%20law%20is%20a%20term,doubles%20abo

https://irds.ieee.org/editions/2020

https://rpubs.com/kapage/523169

V) R Source Code

Please find the source code and respective comments included in this document.