ECON 144: Project 1

Zach Wrubel, Austin Pham, Shannan Liu

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 the volatile nature of 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.

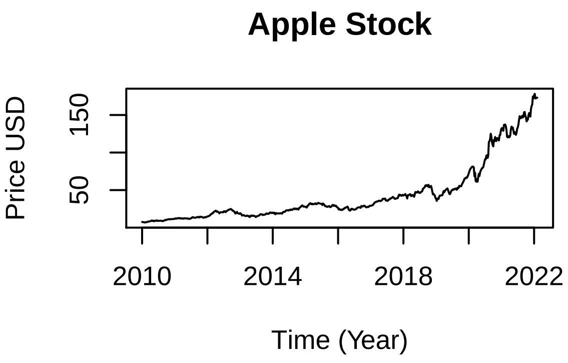
```
appl <-  read.csv("AAPL.csv", sep = ",")
on <-  read.csv("ON.csv", sep = ",")
smh <-  read.csv("SMH.csv", sep = ",")
tsm <-  read.csv("TSM.csv", sep = ",")

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)
```

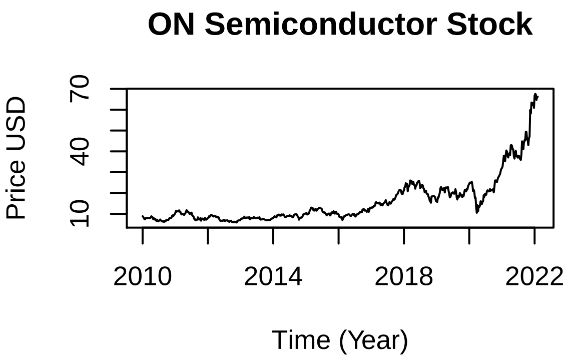
II) Results

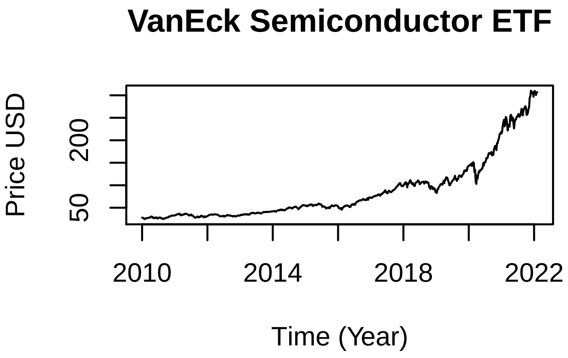
1) Modeling and Forecasting Trend

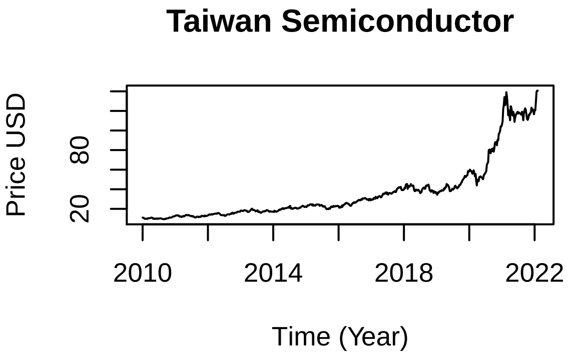
A) Time-Series Plot

```
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")
```









B) Covariance Stationary

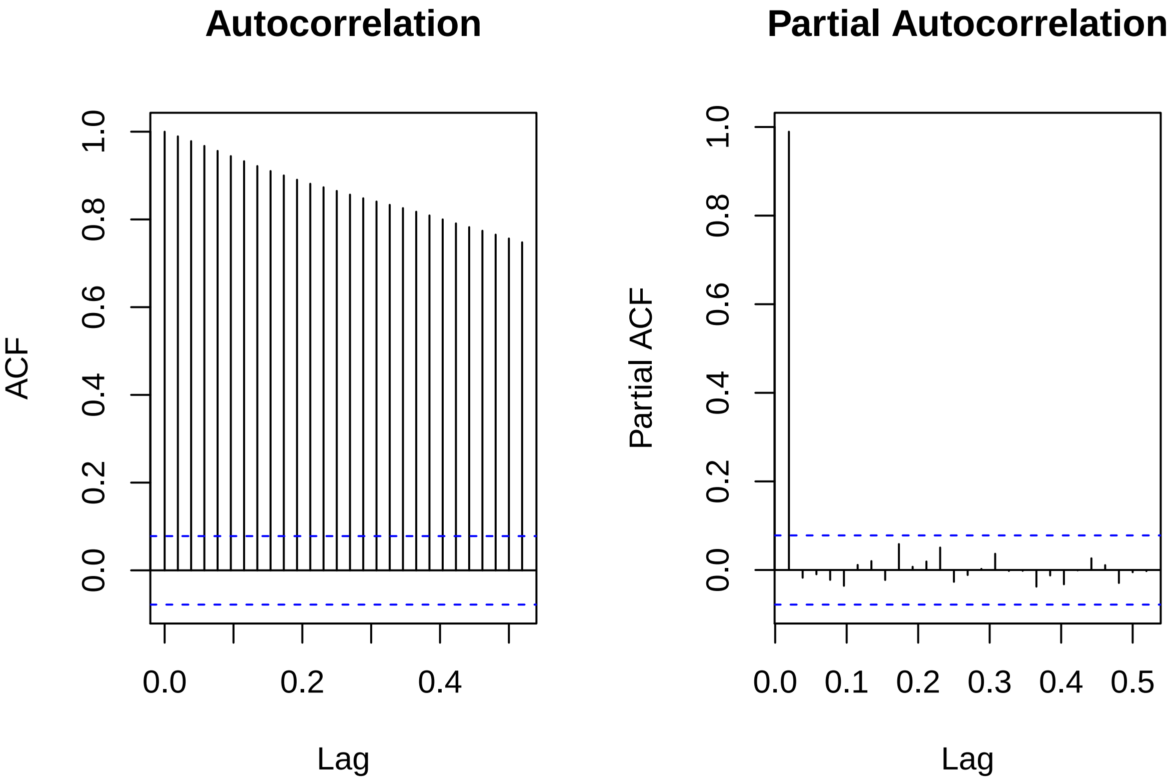
The Apple time series does not appear to be covariance stationary. The object is not mean reverting as each random variable appears to have non-constant means. This can be seen in the positive linear trend in the time series. Additionally, it appears that the variance structure is not constant over time 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)
m2 <- lm(appl_ts ~ t + I(t ^ 2))
m3 <- lm(log(appl_ts) ~ t)
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)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.865 -18.801  -4.753  12.490  78.686
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.899e+04  5.071e+02  -37.45   <2e-16 ***
## t            9.442e+00  2.516e-01   37.54   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 21.91 on 628 degrees of freedom
## Multiple R-squared:  0.6917, Adjusted R-squared:  0.6912
## F-statistic:  1409 on 1 and 628 DF,  p-value: < 2.2e-16
```

```r
summary(m2)
```

```
##
## Call:
## lm(formula = appl_ts ~ t + I(t^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.918  -7.134  -0.080   9.300  40.829
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.655e+06  1.941e+05   34.30   <2e-16 ***
## t           -6.612e+03  1.925e+02  -34.34   <2e-16 ***
## I(t^2)       1.642e+00  4.775e-02   34.39   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 12.9 on 627 degrees of freedom
## Multiple R-squared:  0.8932, Adjusted R-squared:  0.8929
## F-statistic:  2622 on 2 and 627 DF,  p-value: < 2.2e-16
```

```r
summary(m3)
```

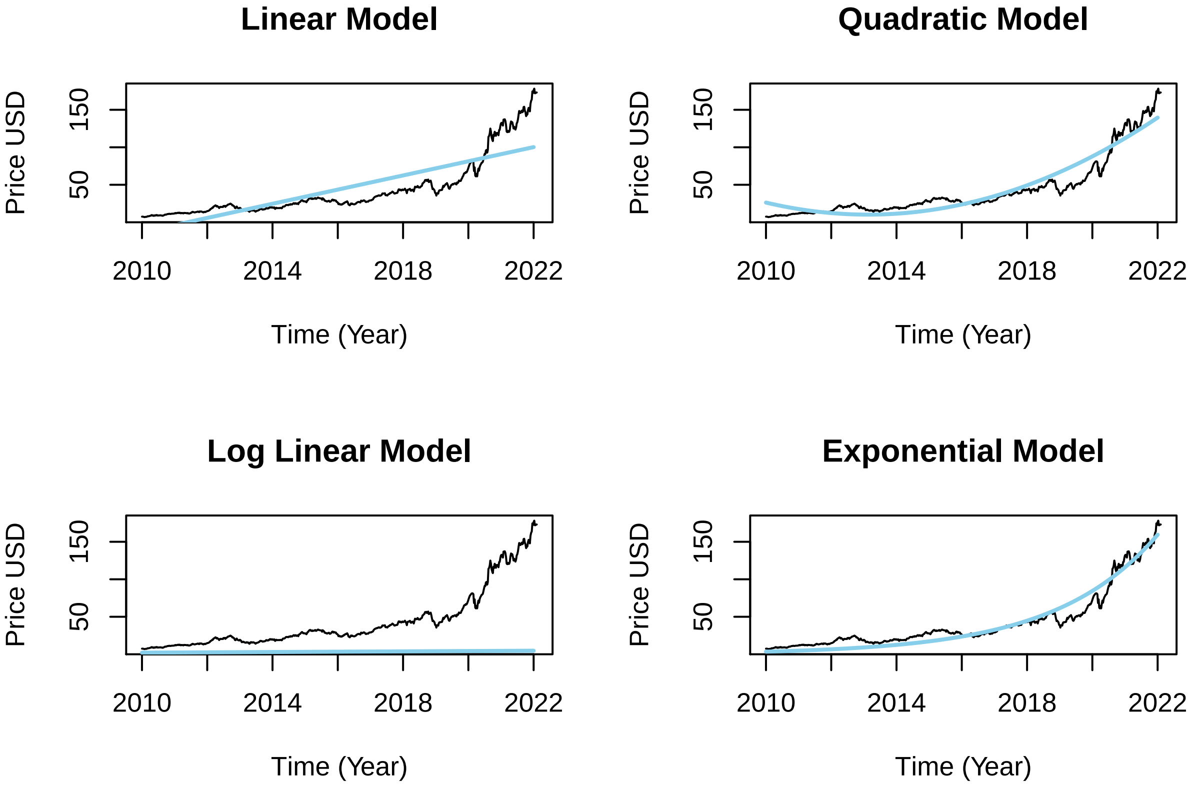
```
##
## Call:
## lm(formula = log(appl_ts) ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51643 -0.14919 -0.03134  0.13836  0.47694
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.303e+02  4.955e+00  -86.85   <2e-16 ***
## t            2.152e-01  2.458e-03   87.54   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Residual standard error: 0.214 on 628 degrees of freedom
## Multiple R-squared:  0.9243, Adjusted R-squared:  0.9241
## F-statistic:  7664 on 1 and 628 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(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 = "Price USD",
     main = "Exponential Model")
lines(ds$x, predict(m4, list(x = ds$x)), col = "skyblue", lwd = 2)
```



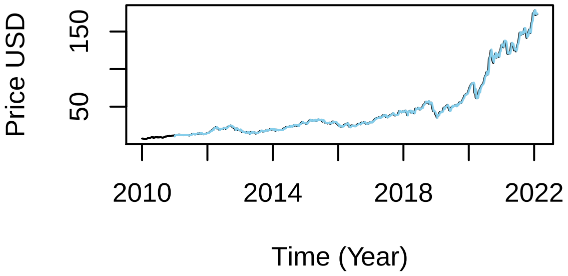
Linear Model

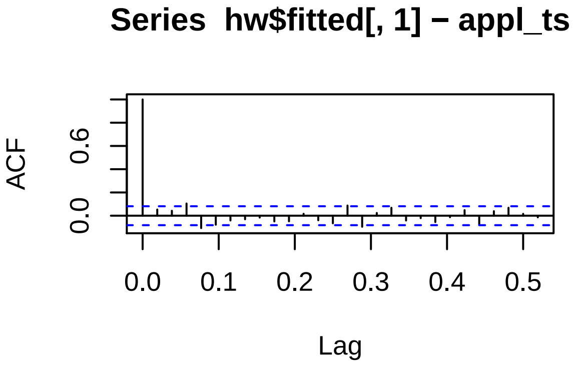
Quadratic Model

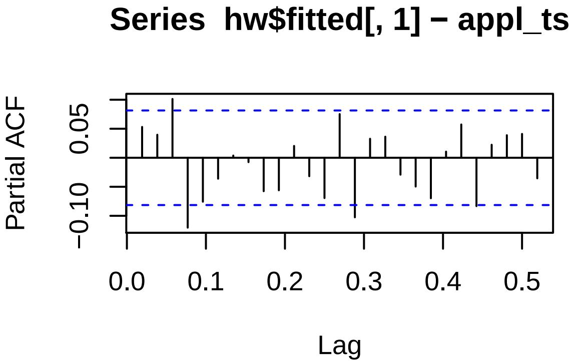
Log Linear Model

Exponential Model

```
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)
```





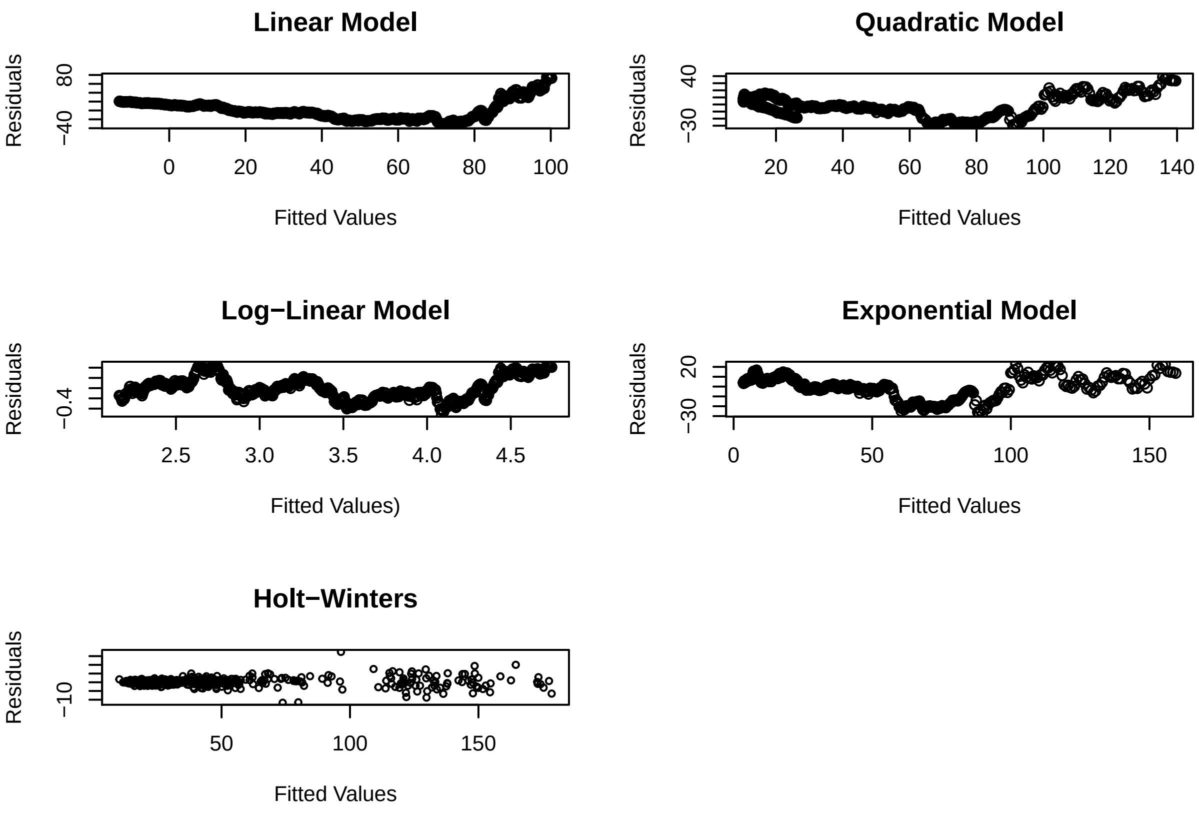


E) Residuals vs. Fitted Values Plots for all Models

Looking at the residuals versus fitted values plots reveals strong patterns for the linear model, quadratic model, 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 a quadratic pattern as the fitted values increase. The quadratic model plot shows less strong of a pattern, but still demonstrates a relationship between the residuals and the fitted values. The Log-Linear Model suggests strong cycles in the data. 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")

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)")
```

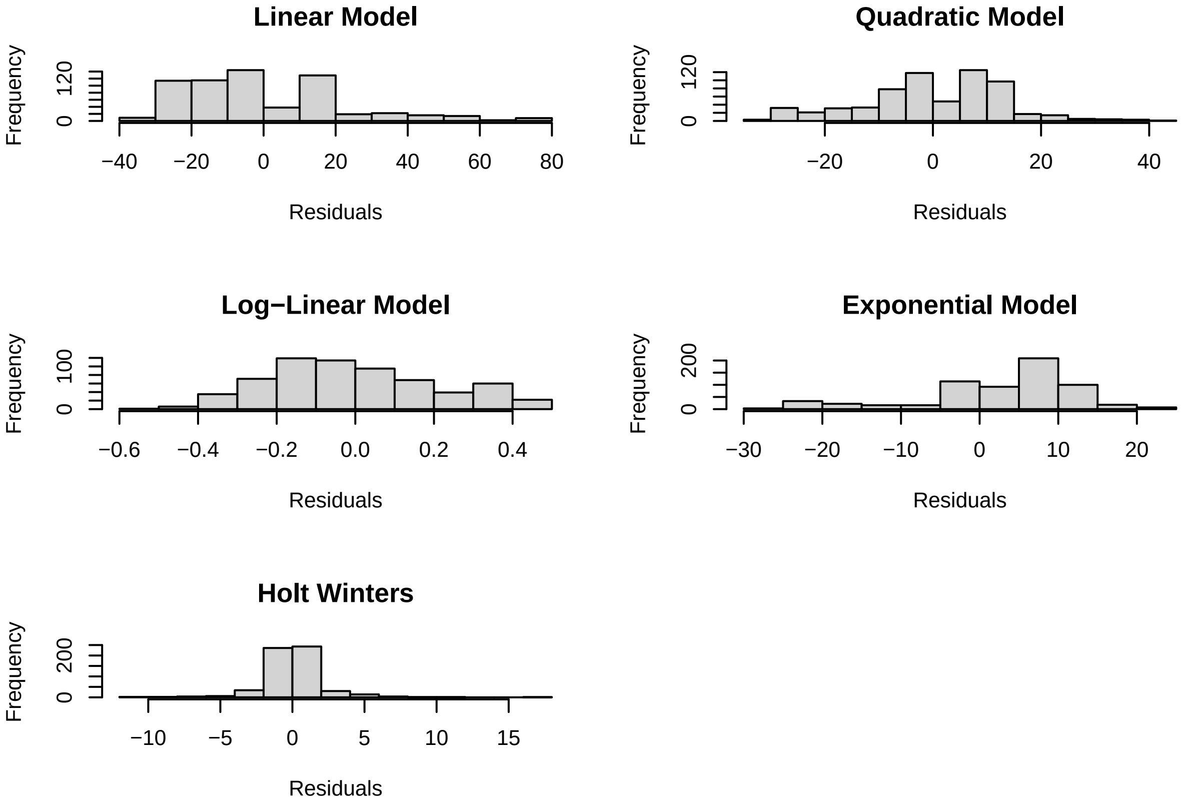
```r
plot(predict(m4, list(x = ds$x)),residuals(m4),
     main="Exponential Model",
     ylab="Residuals",
     xlab="Fitted Values")

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



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 0.6912, a MSR of 478.394, and the slope coefficient is statistically significant. The Quadratic model has an adjusted R-squared of 0.89, a MSR of 165.726, and both slope coefficients are statistically significant. The Log-Linear Model has an adjusted R-squared of 0.9241, a MSR of 0.0457, 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 4.53178.

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

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

```
##
## Call:
## lm(formula = appl_ts ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.865 -18.801  -4.753  12.490  78.686
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.899e+04  5.071e+02  -37.45   <2e-16 ***
## t            9.442e+00  2.516e-01   37.54   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 21.91 on 628 degrees of freedom
## Multiple R-squared:  0.6917, Adjusted R-squared:  0.6912
## F-statistic:  1409 on 1 and 628 DF,  p-value: < 2.2e-16
```

```r
summary(m2)
```

```
##
## Call:
## lm(formula = appl_ts ~ t + I(t^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.918  -7.134  -0.080   9.300  40.829
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.655e+06  1.941e+05   34.30   <2e-16 ***
## t           -6.612e+03  1.925e+02  -34.34   <2e-16 ***
## I(t^2)       1.642e+00  4.775e-02   34.39   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 12.9 on 627 degrees of freedom
## Multiple R-squared:  0.8932, Adjusted R-squared:  0.8929
## F-statistic:  2622 on 2 and 627 DF,  p-value: < 2.2e-16
```

```r
summary(m3)
```

```
##
## Call:
## lm(formula = log(appl_ts) ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51643 -0.14919 -0.03134  0.13836  0.47694
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.303e+02  4.955e+00  -86.85   <2e-16 ***
## t            2.152e-01  2.458e-03   87.54   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.214 on 628 degrees of freedom
## Multiple R-squared:  0.9243, Adjusted R-squared:  0.9241
## F-statistic:  7664 on 1 and 628 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
```

```
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
```

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

```
## [1] 478.394
```

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

```
## [1] 165.726
```

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

```
## [1] 0.04566705
```

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

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

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

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

H)

The Log-Linear has the lowest AIC (-150.5557) and lowest BIC (-137.2186), therefore the models agree and we select the Log-Linear Model for forecasting.

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

```
##    df       AIC
## m1  3 5681.2364
## m2  4 5015.3740
## m3  3 -150.5557
## m4  3 4708.4842
```

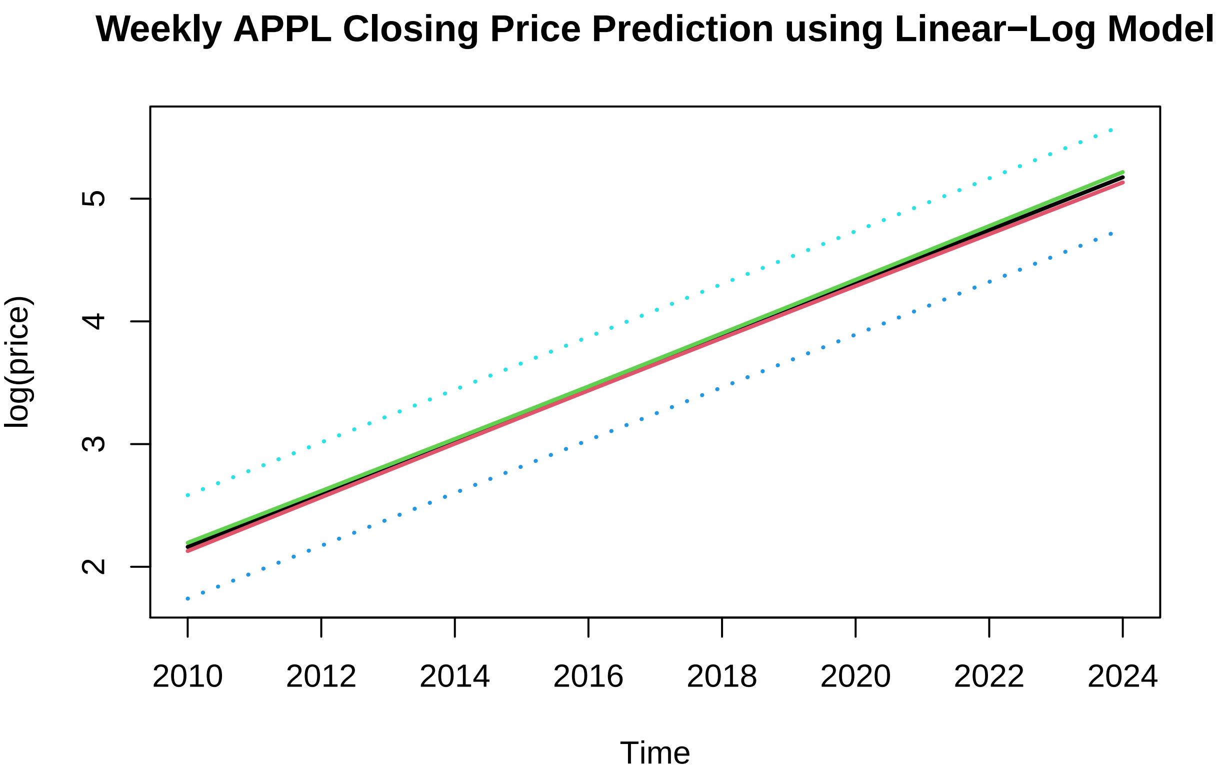
```
BIC(m1,m2,m3,m4)
```

```
##    df       BIC
## m1  3 5694.5735
## m2  4 5033.1569
## m3  3 -137.2186
## m4  3 4721.8214
```

I) Forecast 16 Weeks Ahead Using Preferred Model

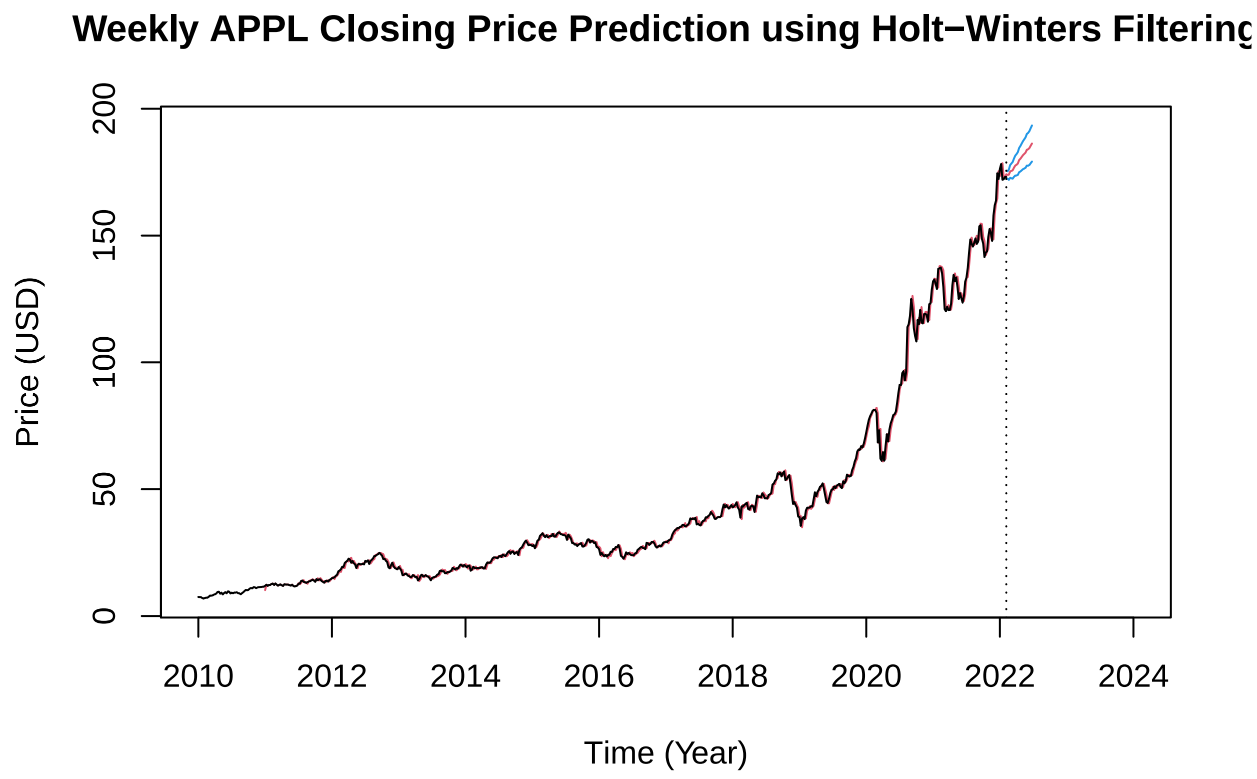
We will now forecast the next two years of weekly APPL closing prices using our Linear-Log model and our Holt-Winters model.

```
#Log-Linear Model Forecast
tn=data.frame(t=seq(2010,2024))
pred=predict(lm(log(appl_ts) ~ t), tn, se.fit = TRUE)
pred.plim = predict(lm(log(appl_ts) ~ t),tn, level =0.95, interval="prediction")
pred.clim = predict(lm(log(appl_ts) ~ t), tn,level=0.95, interval="confidence")
matplot(tn$t,cbind(pred.clim, pred.plim[,-1]),
        lty=c(1,1,1,3,3),
        type="l",
        lwd=2,
        main="Weekly APPL Closing Price Prediction using Linear-Log Model",
        ylab="log(price)",
        xlab="Time")
```

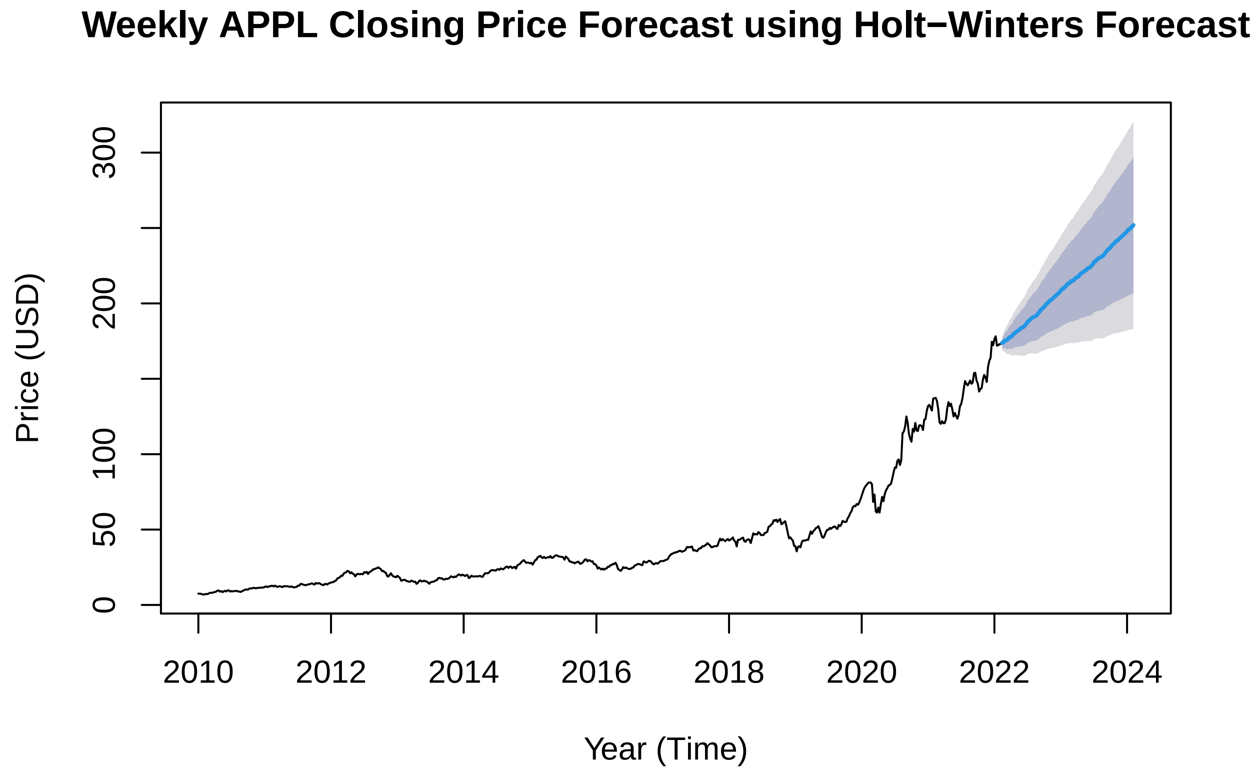


```r
#Holt-Winters Filtering
#quartz()
hwpred <- predict(hw, 20, prediction.interval = TRUE,level=0.5)
plot(hw,hwpred,
     main="Weekly APPL Closing Price Prediction using Holt-Winters Filtering",
     ylab="Price (USD)",
     xlab="Time (Year)",
     xlim=c(2010,2024))

#Holt-Winters Forecast
library(forecast)
```

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



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



2) Trend and Seasonal Adjustments

III) Conclusions & Future Work

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

V) R Source Code