

# ECON 144: Project 1

Zach Wrubel, Austin Pham, Shannon

## 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 see 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 electronic 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

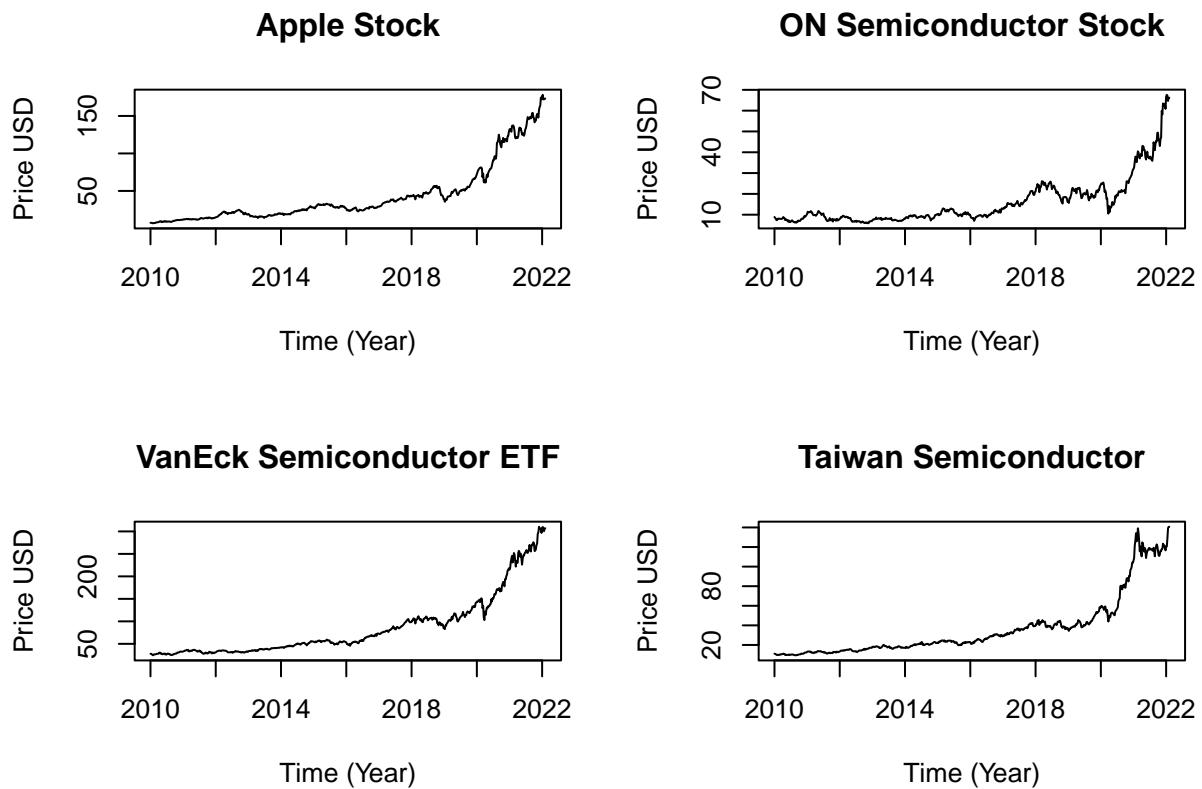
#### A) Time-Series Plot

```
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

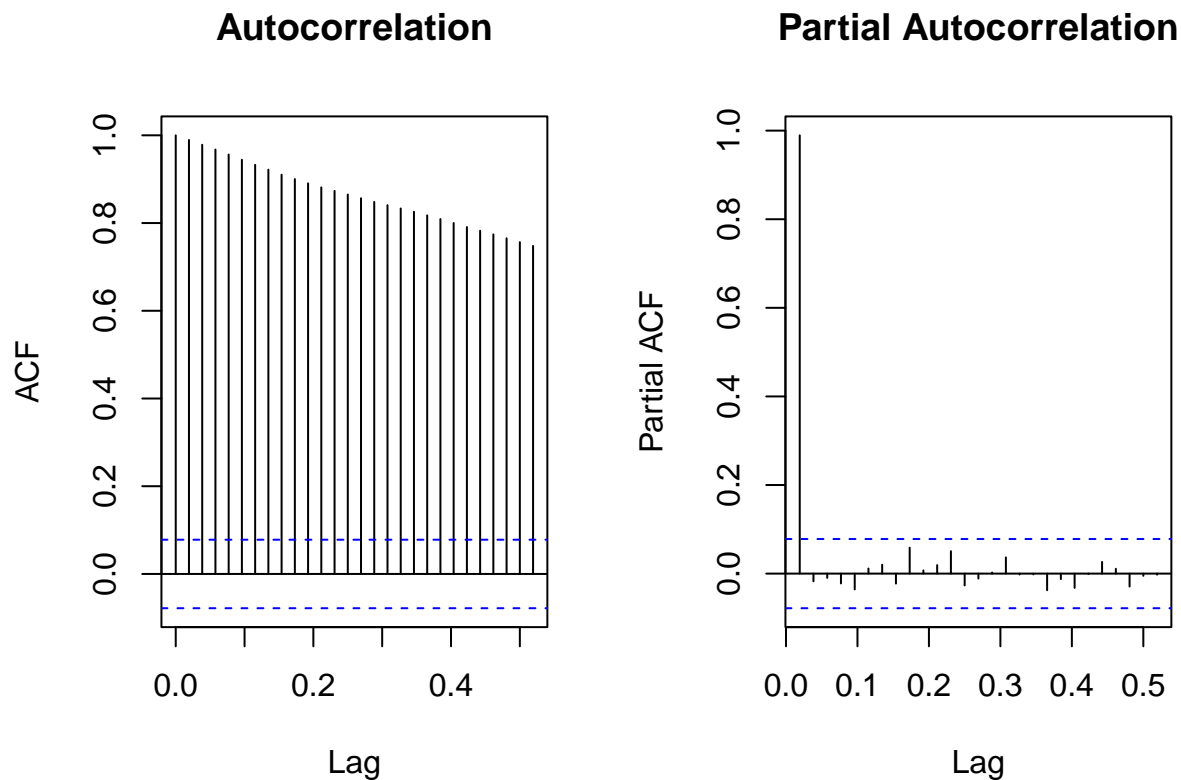
The Apple time series does not appear to be covariance stationary. The object is not mean reverting as each random variable appears to have non-constant means. This can be seen in the positive linear trend in the time series. Additionally, it appears that the variance structure is not constant over time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

### C) Plot & Discuss ACF & PACF

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 it is so hard to model.

```
par(mfrow = c(1, 2))
acf(appl_ts, main = "Autocorrelation")
pacf(appl_ts, main = "Partial Autocorrelation")
```



### D) Fit a Linear & Nonlinear Model

```
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
```

```
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
```

```
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)
```

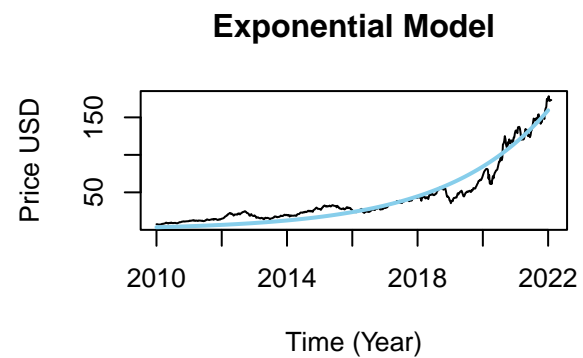
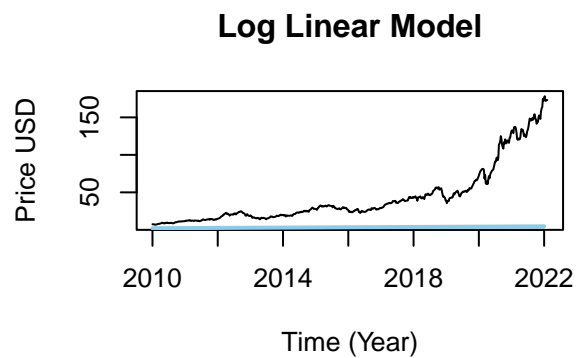
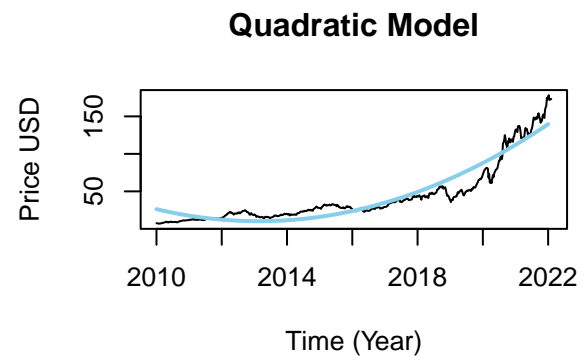
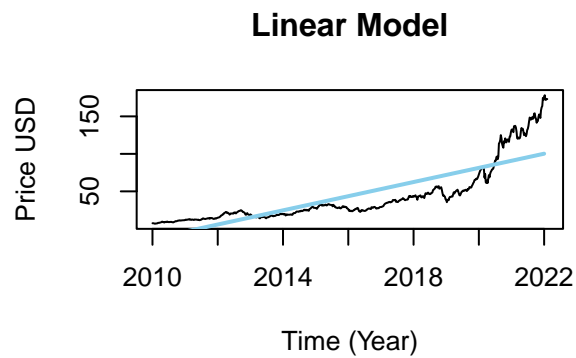
```
##
## Formula: y ~ exp(a + b * t)
##
## Parameters:
##             Estimate Std. Error t value Pr(>|t|)
## a -6.372e+02  9.074e+00  -70.22  <2e-16 ***
## b  3.176e-01  4.491e-03   70.73  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: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

```
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)

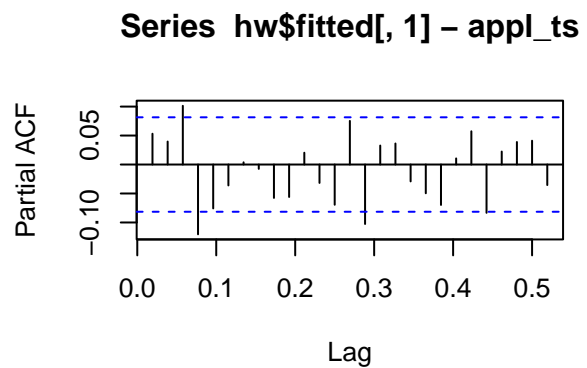
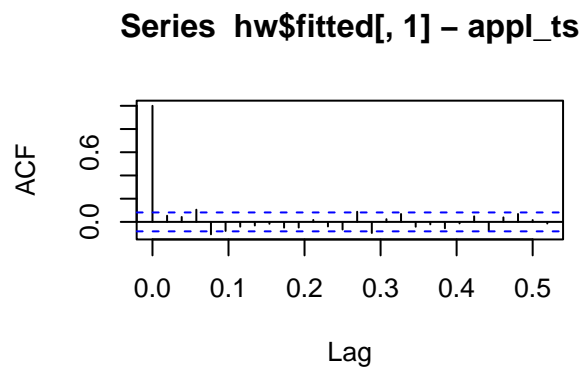
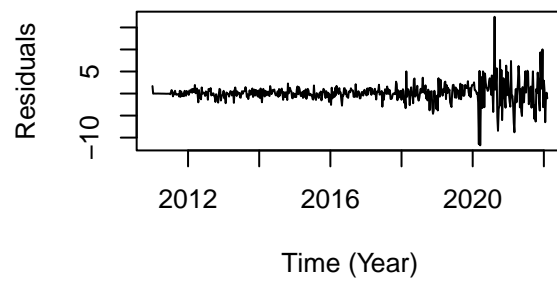
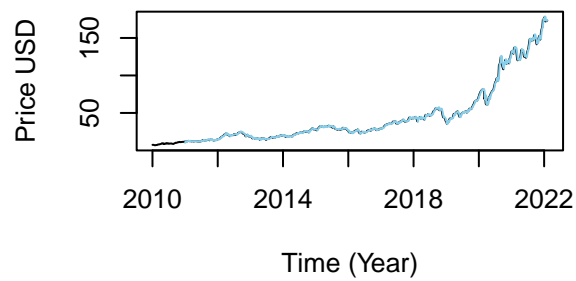
plot(appl_ts,
     xlab = "Time (Year)",
     ylab = "Price USD",
     main = "Quadratic Model")
lines(t, m2$fit, col = "skyblue", lwd = 2)

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)
```



```
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]
plot(hw_res,
      ylab = "Residuals",
      type = 'l',
      xlab = "Time (Year)")
acf(hw$fitted[, 1] - appl_ts)
pacf(hw$fitted[, 1] - appl_ts)
```



## 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