Econ 144 Project 1

Jordan Youhanaie and Cal Parker

4/17/2023

Loading Libraries

```
#Import potentially relevant libraries
library(tseries, quietly = T)
```

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

```
library(forecast, quietly = T)
#library(DAAG)
require("datasets")
library(tis)
```

```
## Warning: package 'tis' was built under R version 4.3.3
```

```
##
## Attaching package: 'tis'
```

```
## The following object is masked from 'package:forecast':
##
##     easter
```

```
library(vars)
```

```
## Loading required package: MASS
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

```
library(dynlm)
library(fpp3)
```

```
## ── Attaching packages ──────────────────────────────────────────── fpp3 0.5 ──
```

```
## ✓ tibble      3.2.1     ✓ tsibble     1.1.3
## ✓ dplyr       1.1.2     ✓ tsibbledata 0.4.1
## ✓ tidyr       1.3.0     ✓ feasts      0.3.1
## ✓ lubridate   1.9.2     ✓ fable       0.3.3
## ✓ ggplot2     3.5.1     ✓ fabletools  0.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## ── Conflicts ───────────────────────────────────────────────── fpp3_conflicts ──
## ✗ dplyr::between()     masks tis::between()
## ✗ lubridate::date()    masks base::date()
## ✗ lubridate::day()     masks tis::day()
## ✗ dplyr::filter()      masks stats::filter()
## ✗ lubridate::hms()     masks tis::hms()
## ✗ tsibble::index()     masks zoo::index()
## ✗ tsibble::intersect() masks base::intersect()
## ✗ tsibble::interval()  masks lubridate::interval()
## ✗ dplyr::lag()         masks stats::lag()
## ✗ lubridate::month()   masks tis::month()
## ✗ lubridate::period()  masks tis::period()
## ✗ lubridate::POSIXct() masks tis::POSIXct()
## ✗ lubridate::quarter() masks tis::quarter()
## ✗ dplyr::select()      masks MASS::select()
## ✗ tsibble::setdiff()   masks base::setdiff()
## ✗ lubridate::today()   masks tis::today()
## ✗ tsibble::union()     masks base::union()
## ✗ fable::VAR()         masks vars::VAR()
## ✗ lubridate::year()    masks tis::year()
## ✗ lubridate::ymd()     masks tis::ymd()
```

```
library(seasonal)
```

```
##
## Attaching package: 'seasonal'
```

```
## The following object is masked from 'package:tibble':
##
##     view
```

```
library(stats)
```

Section I

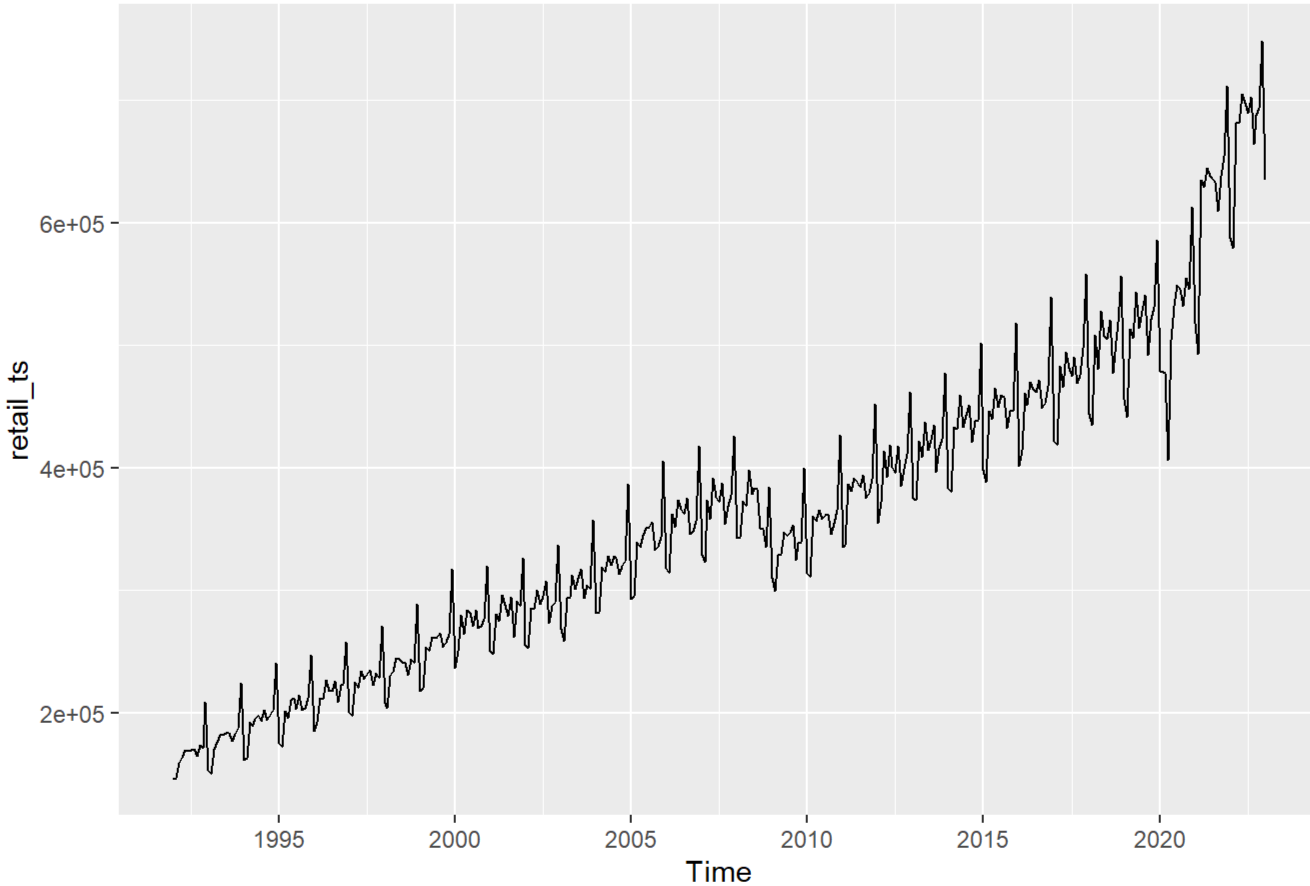
For our project, we decided to use the retail sales recordings of the United States, more specifically, FRED's account of "advance retail sales: retail trade and food services." It is one of the most important economic indicators that national publications use to provide investors and consumers with an understanding of domestic consumption habits. Last month, the value of advanced retail sales was around $691.7 billion. This economic indicator is composed of various retail transactions, such as gasoline station sales, motor vehicle sales, and general merchandise sales. There are many other different components that make up this measurement.

Section II

Part 1

(a)

```
#Import data
retail <- read.csv("RSAFSNA.csv")
#Convert to time series
retail_ts <- ts(retail$RSAFSNA,start=1992, freq=12)
#Create time object for plotting and modeling
t<-seq(1992, 2023,length=length(retail_ts))
#Plot time series
autoplot(retail_ts)
```

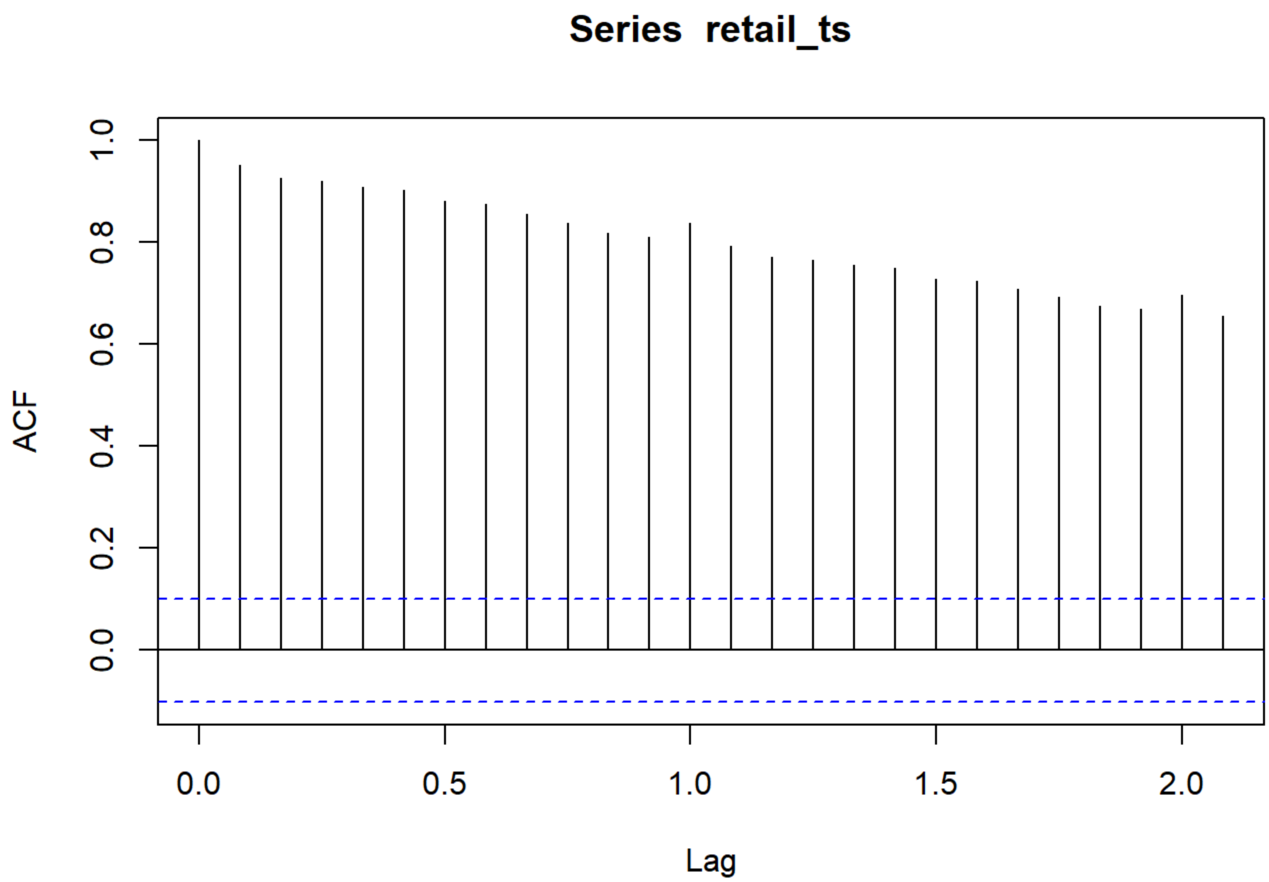


(b)

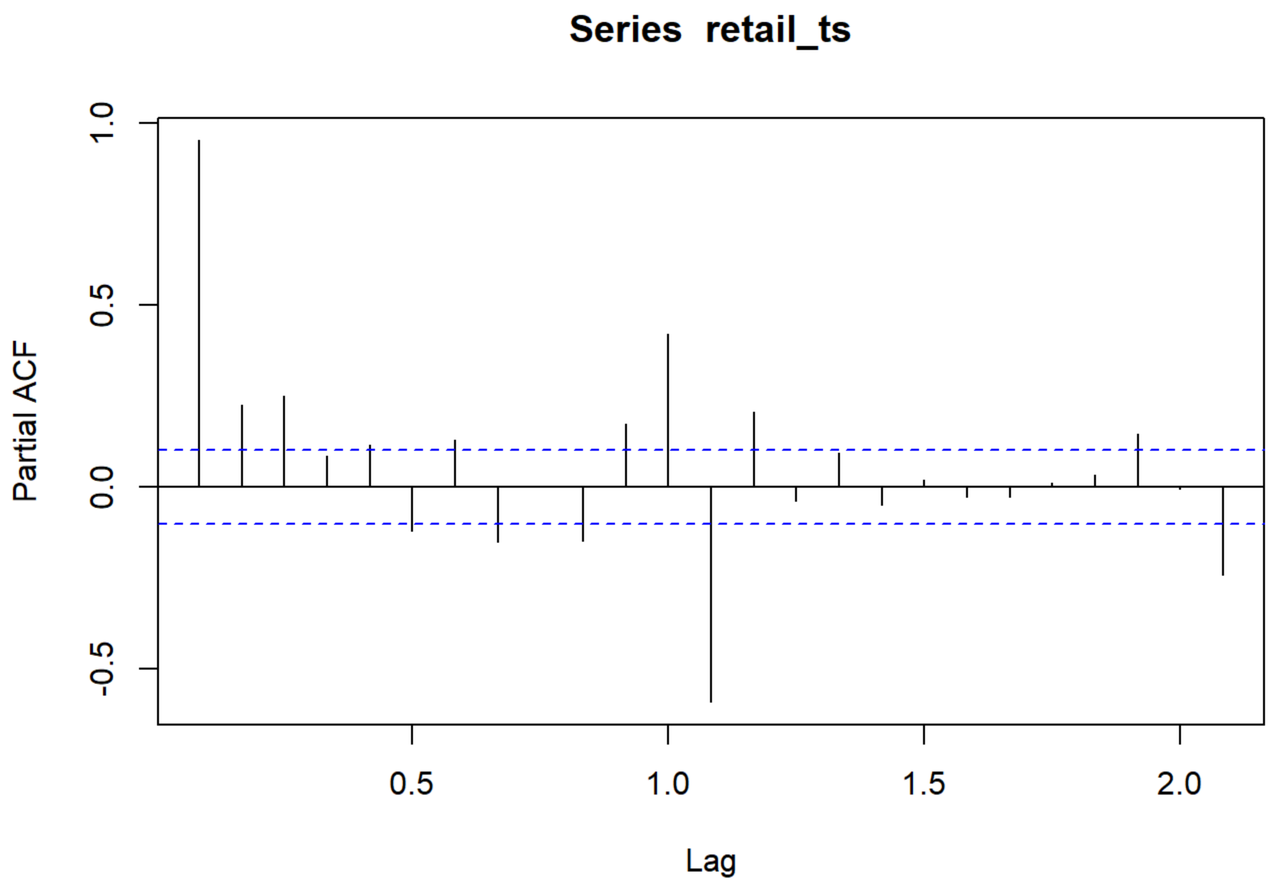
The seasonality of this data appears to increase in volatility as time goes on as opposed to staying constant. Thus, the data are not covariance stationary.

(c)

```
#ACF of time series
acf(retail_ts)
```




```
#PACF of time series
pacf(retail_ts)
```



The ACF plot shows that the ACF value for every amount of lag is well above the threshold, indicating that the data are highly dependent on past values and thus not covariance stationary. The PACF similarly shows several values above the threshold and appears to potentially have some sort of cyclical nature to it, further supporting our conclusion.

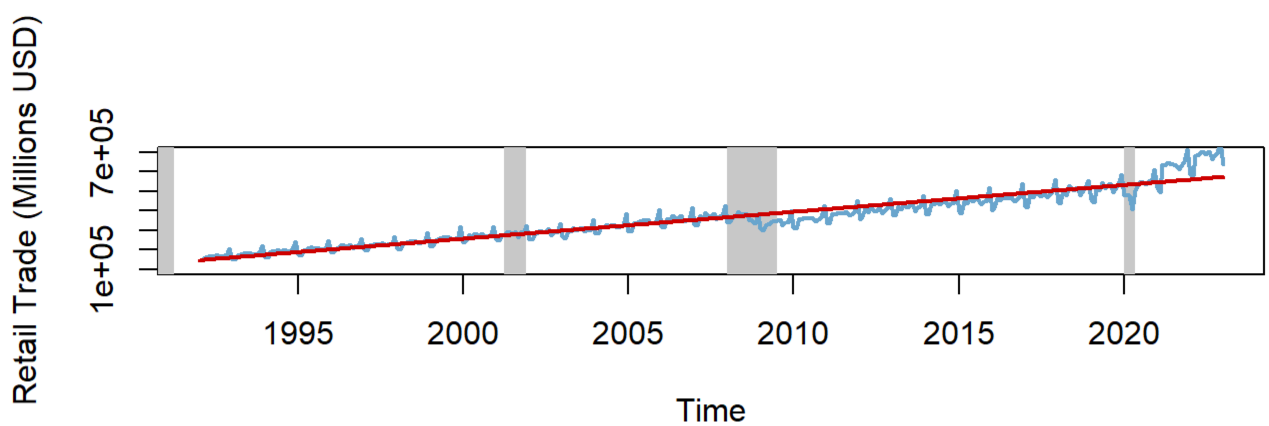
(d)

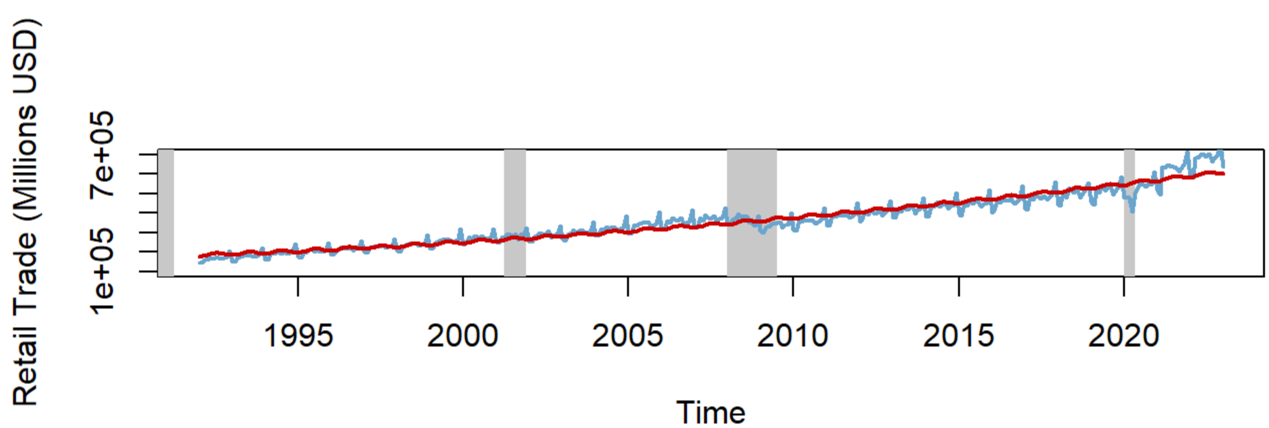
Model Fitting

```
#Linear Fit
m1 <- lm(retail_ts~t)

#Non-Linear Fit
#Quadratic-periodic
#In testing several non-linear models, we found that a quadratic-periodic model that included co
sine but not sine was best, as sine was not statistically signnificant
#Create necessary variables for developing model
t2<-t^2
cos_t<-cos(2*pi*t)

#Fit Model
m2 <- lm(retail_ts~t+t2+cos_t)
```

Plot

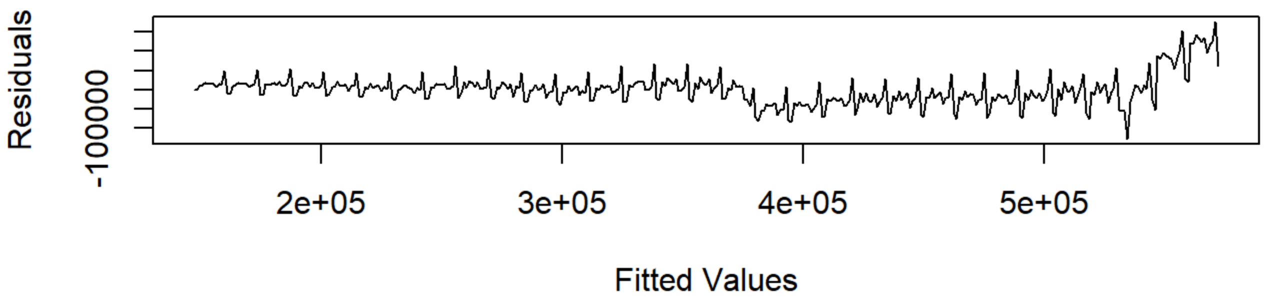
```
par(mfrow=c(2,1))

#Plot time series data
plot(retail_ts, ylab="Retail Trade (Millions USD)", xlab="Time", lwd = 2, col='skyblue3', ylim=c
(100000,700000), xlim=c(1992,2023))

#The next commands add the U.S. recession bands and re layer the original plot so that the reces
sion bands do not cover it
#Recession bands
nberShade()
#Replot time series data
lines(retail_ts, lwd = 2, col='skyblue3')
#Add linear trend line
lines(t, m1$fit, col="red3", lwd = 2)

#Second plot of time series data
plot(retail_ts, ylab="Retail Trade (Millions USD)", xlab="Time", lwd = 2, col='skyblue3', ylim=c
(100000,700000), xlim=c(1992,2023))
#Recession bands
nberShade()
#Replot time series data
lines(retail_ts, lwd = 2, col='skyblue3')
#Add non-linear trend line
lines(t, m2$fit, col="red3",lwd=2)
```

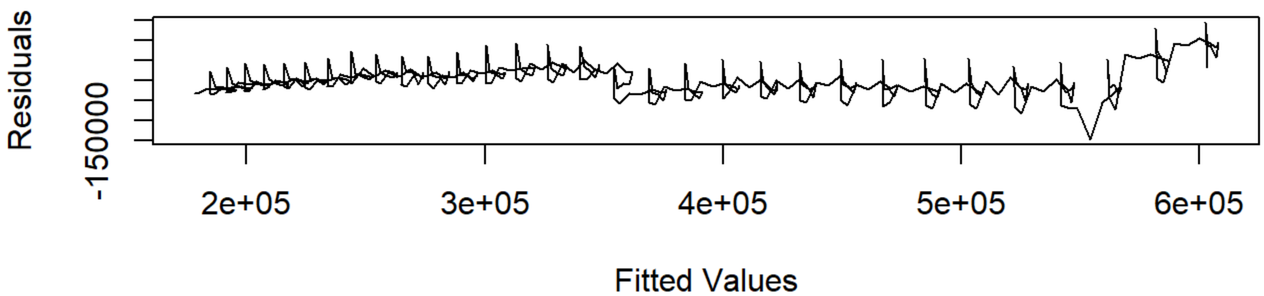




(e)

```
par(mfrow=c(2,1))
#Plot fitted values vs. residuals of linear model
plot(m1$fitted.values,m1$res, ylab="Residuals",type='l',xlab="Fitted Values")
#Plot fitted values vs. residuals of non-linear model
plot(m2$fitted.values,m2$res, ylab="Residuals",type='l',xlab="Fitted Values")
```

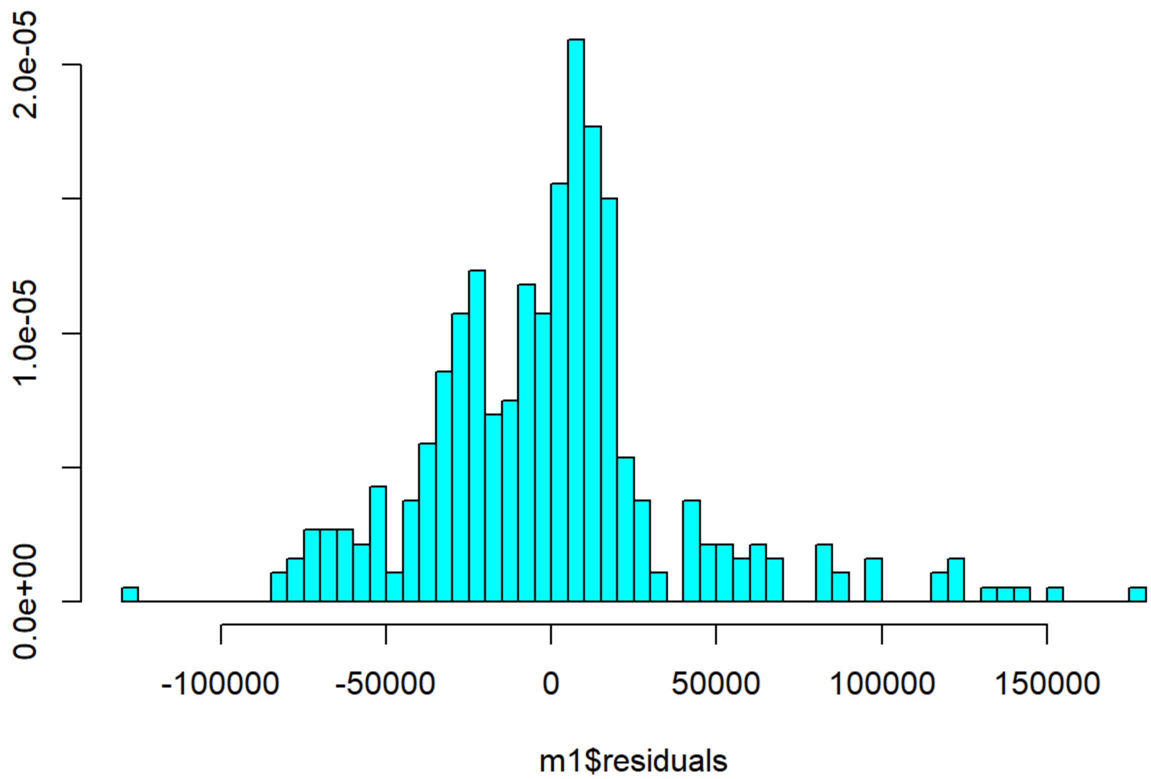




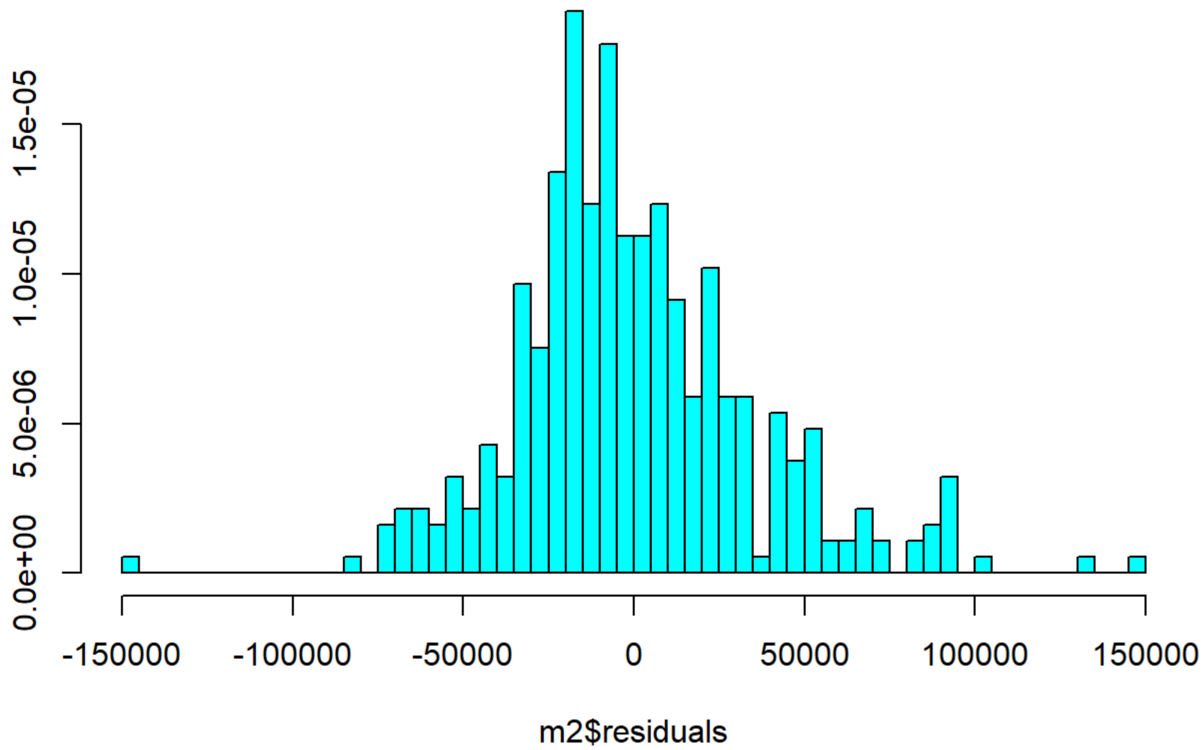
Part E contains two plots; one with the fitted values of our linear model against its residuals (top), and the other with the fitted values of our non-linear model against its residuals (bottom). Since our linear model is non-decreasing, the fitted values will appear in the same order that is observed with our x-variable. This will give us a plot that shows a clear difference between the linear model fitted values for every index of time. Meanwhile, our non-linear model has an unusual appearance because of its cosine component. This is expected considering that our plot incorporates a parametric function/multidimensional output component since our fitted values and residuals both depend on the moment in time that they were produced. In other words, our fitted values contains a parameter (time) to determine its order in the plot. Since our non-linear function is an increasing and decreasing pattern, from the cosine function, the fitted values will fluctuate back and forth resulting in increasing and decreasing parts of our x-axis variable and present us with a function that has overlapping values on our x-axis.

(f)

```
#Histogram of linear residuals
truehist(m1$residuals, nbins = 50)
```




```
#Histogram of non-linear residuals
truehist(m2$residuals, nbins = 50)
```



Part F contains two histograms of the frequencies of our residuals for both our linear and non-linear functions. We can see that our histogram for the linear model has a greater maximum value than a minimum value, resulting in a slightly skewed distribution that seems to be normal but with residuals that correspond to a model that is less likely to accurately fit our data. On the other hand, our non-linear model has a histogram frequency distribution that more closely matches the distribution of a normal distribution. This presents us with a beginning piece of evidence in favor of our non-linear model to be selected as a better representation of a trend to predict our data.

(g)

```
#Descriptive statistics for linear model
summary(m1)
```

```
##
## Call:
## lm(formula = retail_ts ~ t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -128199  -23937    1113   14483  177668
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.718e+07  4.623e+05  -58.79   <2e-16 ***
## t            1.372e+04  2.303e+02   59.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39900 on 371 degrees of freedom
## Multiple R-squared:  0.9053, Adjusted R-squared:  0.9051
## F-statistic:  3548 on 1 and 371 DF,  p-value: < 2.2e-16
```

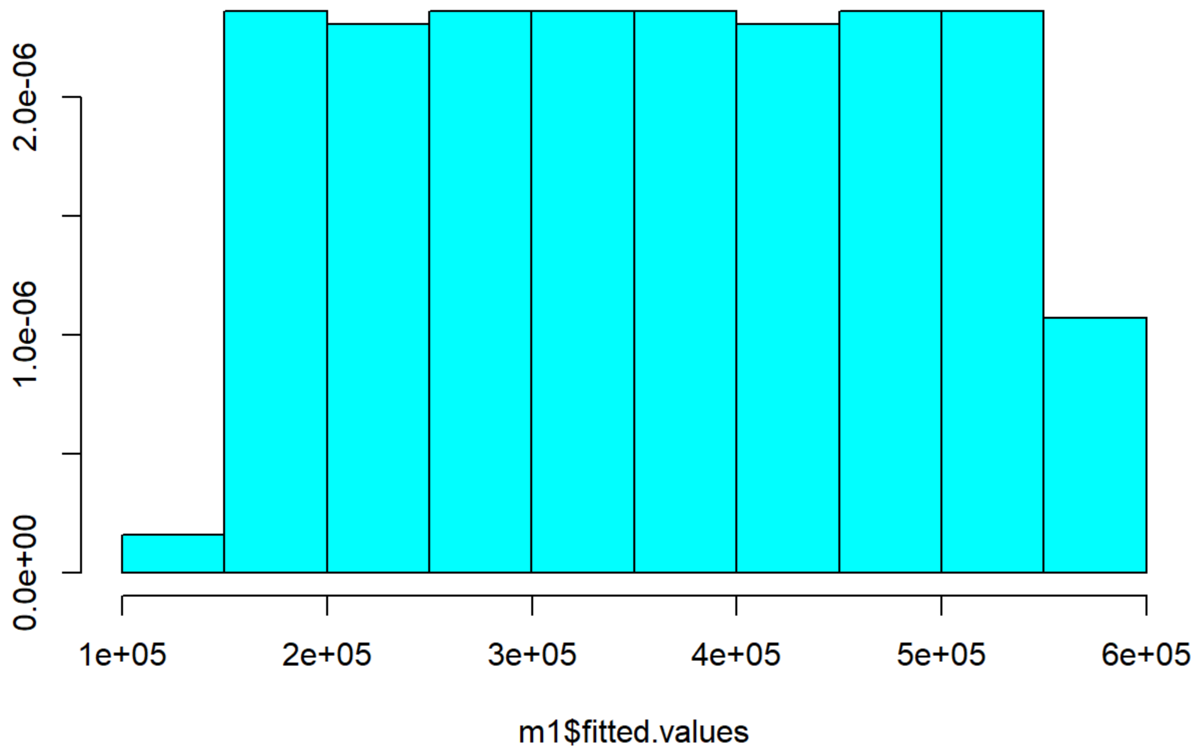
```
mean(m1$fitted.values)
```

```
## [1] 359723.9
```

```
var(m1$fitted.values)
```

```
## [1] 15188973435
```

```
truehist(m1$fitted.values)
```




```
#Descriptive statistics for non-linear model
summary(m2)
```

```
##
## Call:
## lm(formula = retail_ts ~ t + t2 + cos_t)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -147764  -20850   -5504   19256  146435
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.328e+08  1.036e+08   9.001  < 2e-16 ***
## t           -9.427e+05  1.032e+05  -9.131  < 2e-16 ***
## t2           2.382e+02  2.572e+01   9.264  < 2e-16 ***
## cos_t       -7.187e+03  2.615e+03  -2.748  0.00629 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35760 on 369 degrees of freedom
## Multiple R-squared:  0.9244, Adjusted R-squared:  0.9238
## F-statistic:  1504 on 3 and 369 DF,  p-value: < 2.2e-16
```

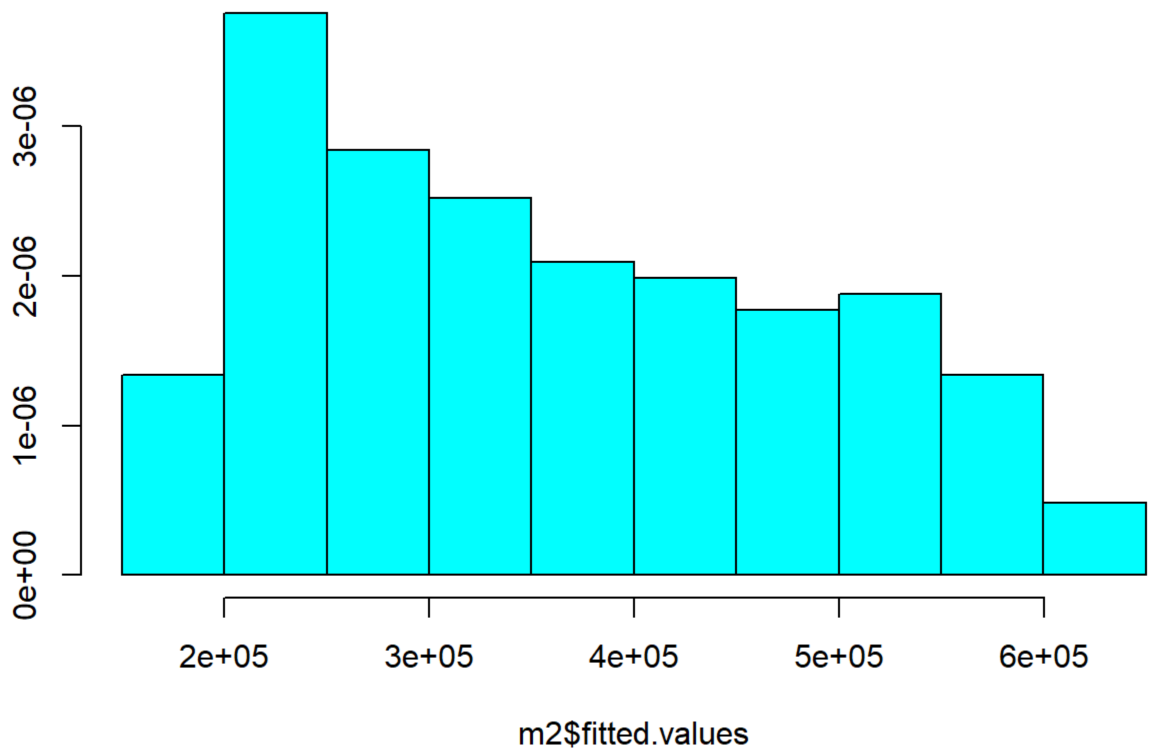
```
mean(m2$fitted.values)
```

```
## [1] 359723.9
```

```
var(m2$fitted.values)
```

```
## [1] 15508354850
```

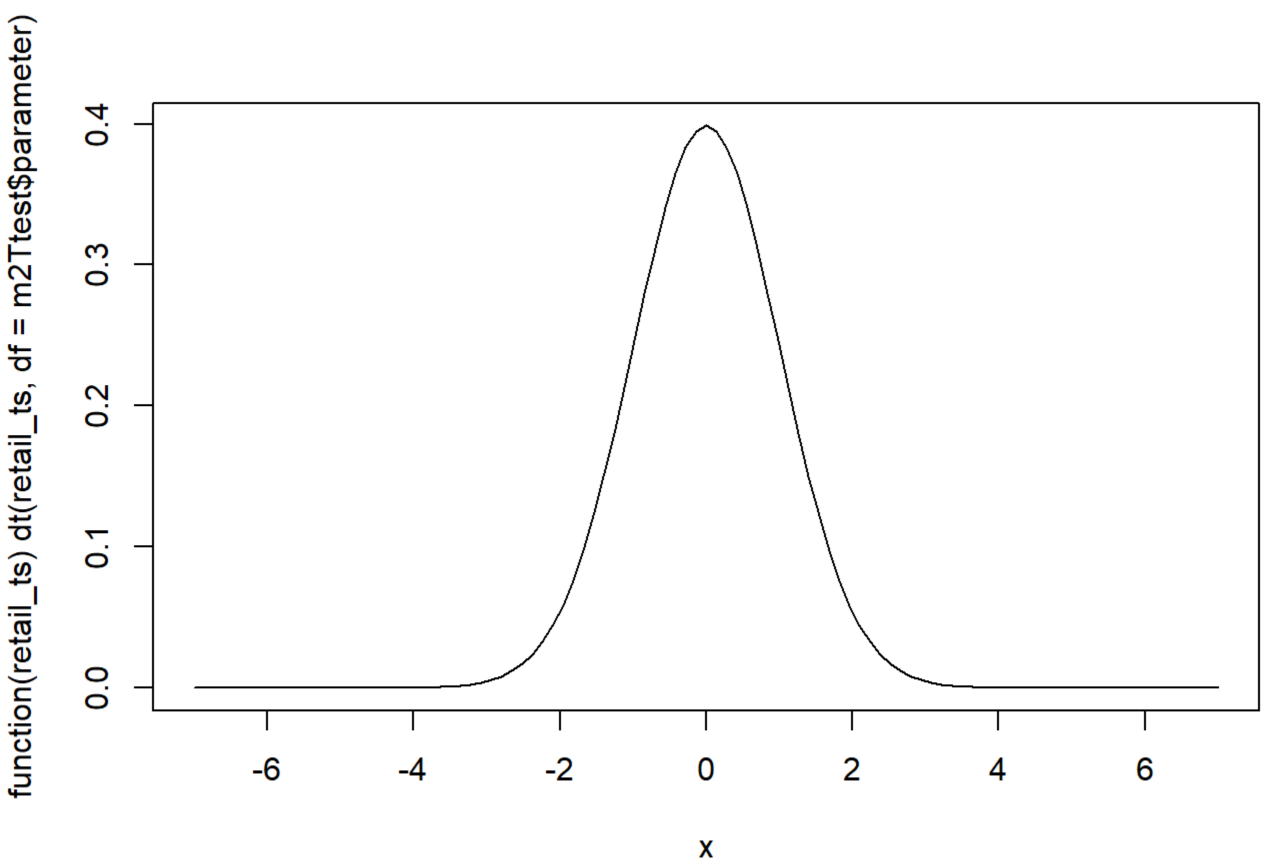
```
truehist(m2$fitted.values)
```



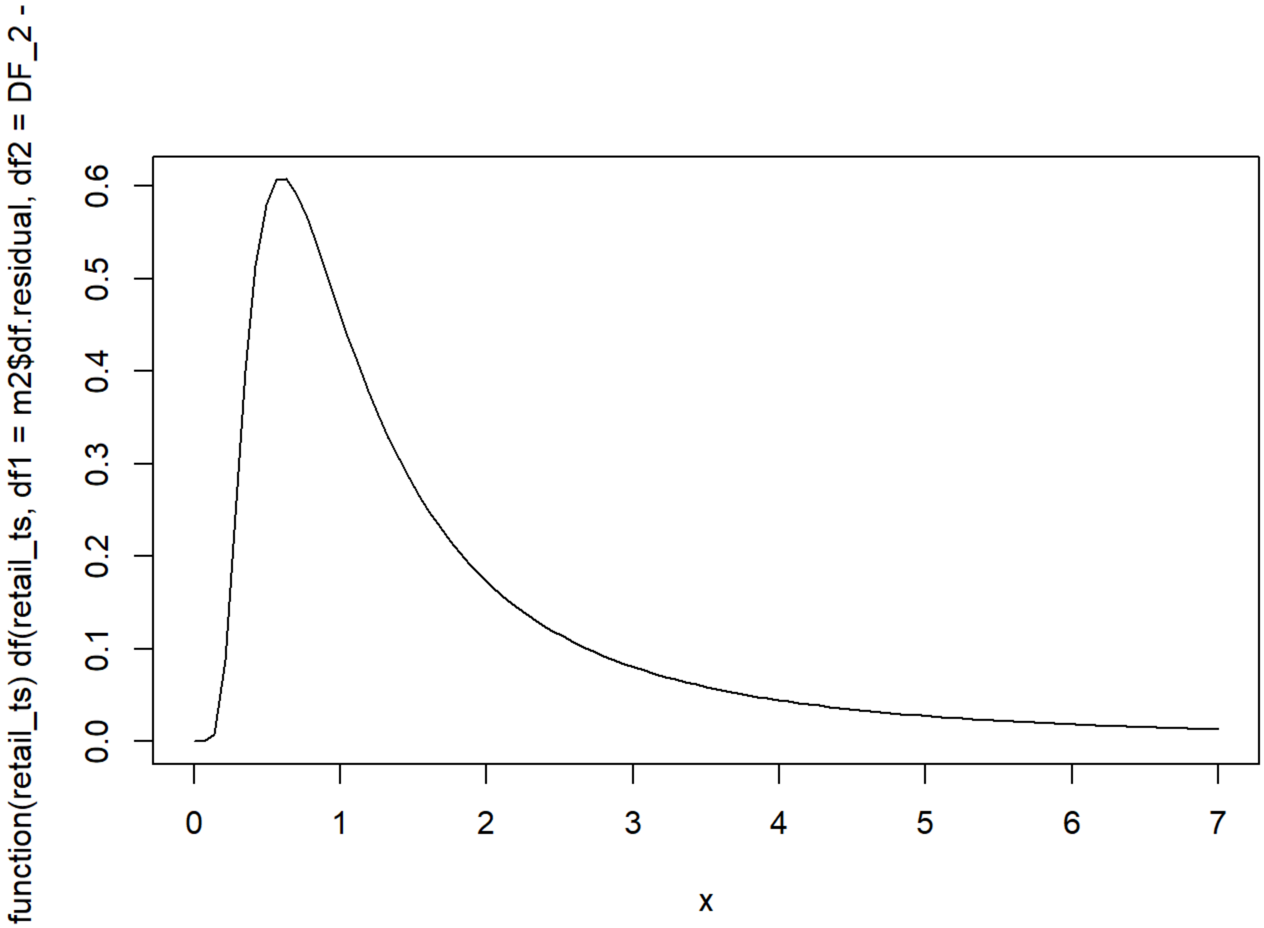

```
m2Ttest<-t.test(retail_ts, data=retail)
m2Ttest
```

```
##
##  One Sample t-test
##
## data:  retail_ts
## t = 53.637, df = 372, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  346536.2 372911.5
## sample estimates:
## mean of x
##  359723.9
```

```
#second valye of degree of freedom (number of parameters -1)
DF_2<-length(m2$coefficients)
#plot of T-distribution of our time series
plot(function(retail_ts) dt(retail_ts, df=m2Ttest$parameter), -7, 7)
```




```
#plot of F-distribution of our time series
plot(function(retail_ts) df(retail_ts, df1=m2$df.residual, df2=DF_2-1), 0,7)
```



As shown by our summaries in part G, the adjusted R-squared and R-squared are both greatest for the non-linear function. This means that there is more explanatory power presented by our function for the non-linear model than our linear model.

(h)

```
#Compare AICs of models
AIC(m1, m2)
```

```
##    df      AIC
## m1  3 8965.827
## m2  5 8886.076
```

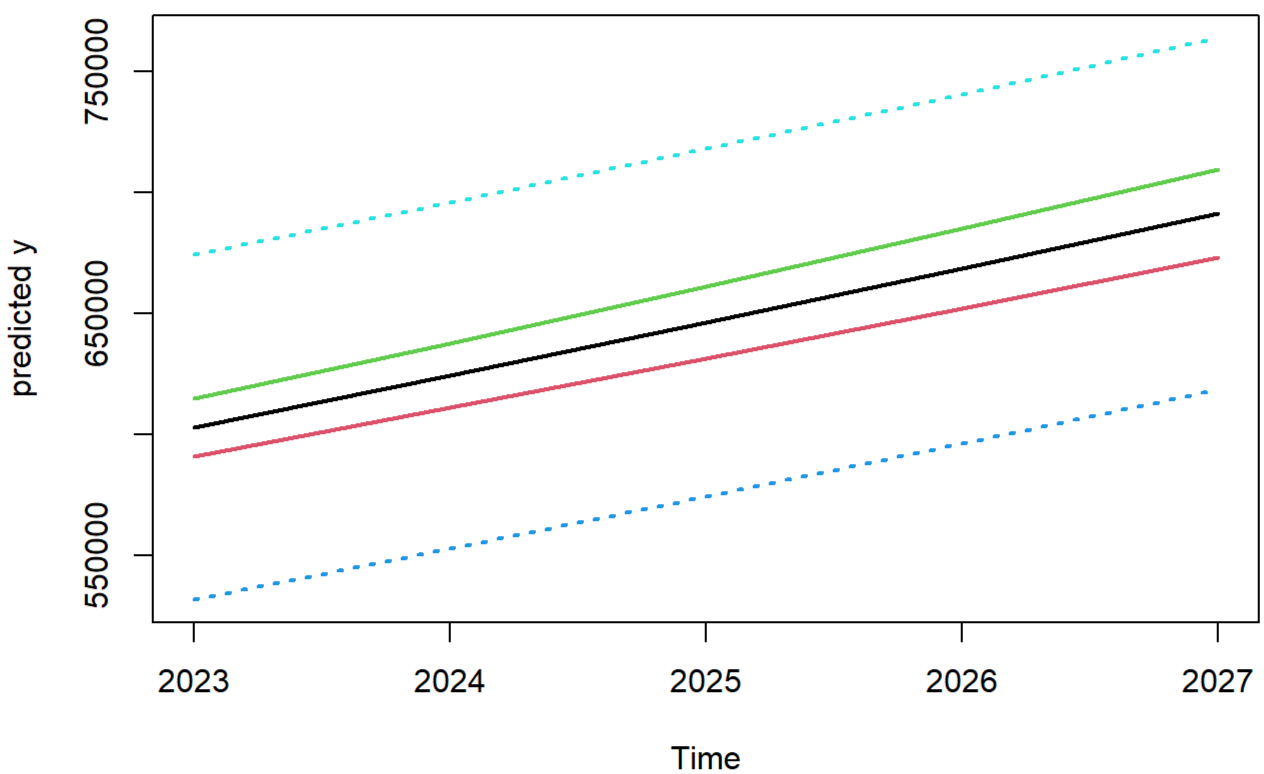
```
#Compare BICs of models
BIC(m1, m2)
```

```
##    df      BIC
## m1  3 8977.592
## m2  5 8905.684
```

For part H, the AIC and BIC models agree that the more efficient model to use for our data is the non-linear model. Our AIC and BIC are both significantly lower for our non-linear model than our non-linear model because our non-linear trend followed the peaks and troughs of the seasons presented in the data.

(i)

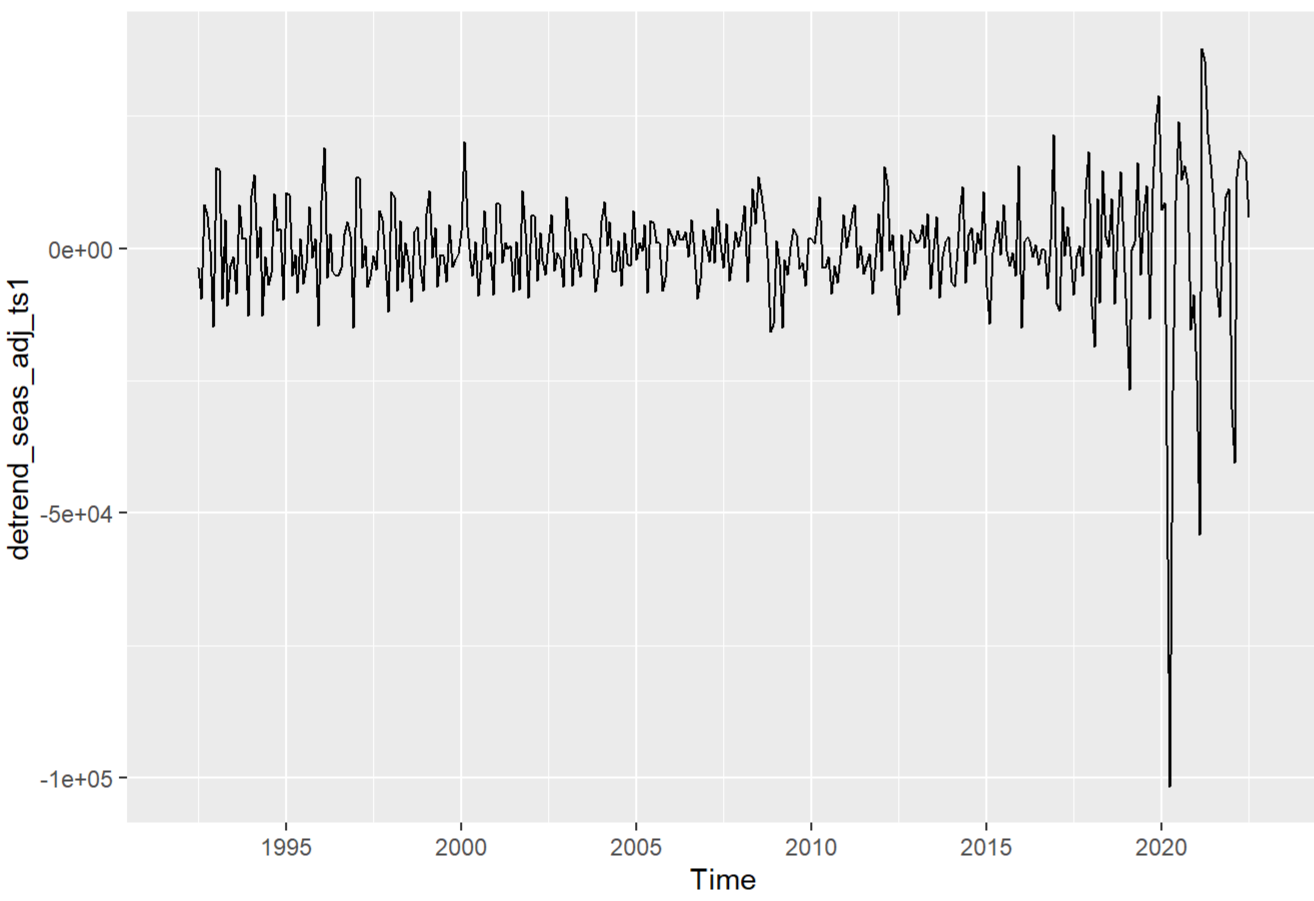
```
#Create time object for future steps to be predicted
#Forecasting 4 years into the future, which should be 48 steps since our data is monthly
tn=data.frame(t=seq(2023,2027))
#Predict / forecast future values of retail price for time range tn based on model developed wit
h original time series data
pred=predict(lm(retail_ts~t+I(t^2)+I(cos(2*pi*t))), tn, se.fit = TRUE)
#Develop prediction interval for forecast
pred.plim = predict(lm(retail_ts~t+I(t^2)+I(cos(2*pi*t))),tn, level =0.95, interval="predictio
n")
#Develop confidence interval for forecast
pred.clim = predict(lm(retail_ts~t+I(t^2)+I(cos(2*pi*t))), tn,level=0.95, interval="confidence")
#Plot forecast and prediction and confidence intervals together
matplot(tn$t,cbind(pred.clim, pred.plim[,-1]),
        lty=c(1,1,1,3,3), type="l", lwd=2, ylab="predicted y",xlab="Time")
```



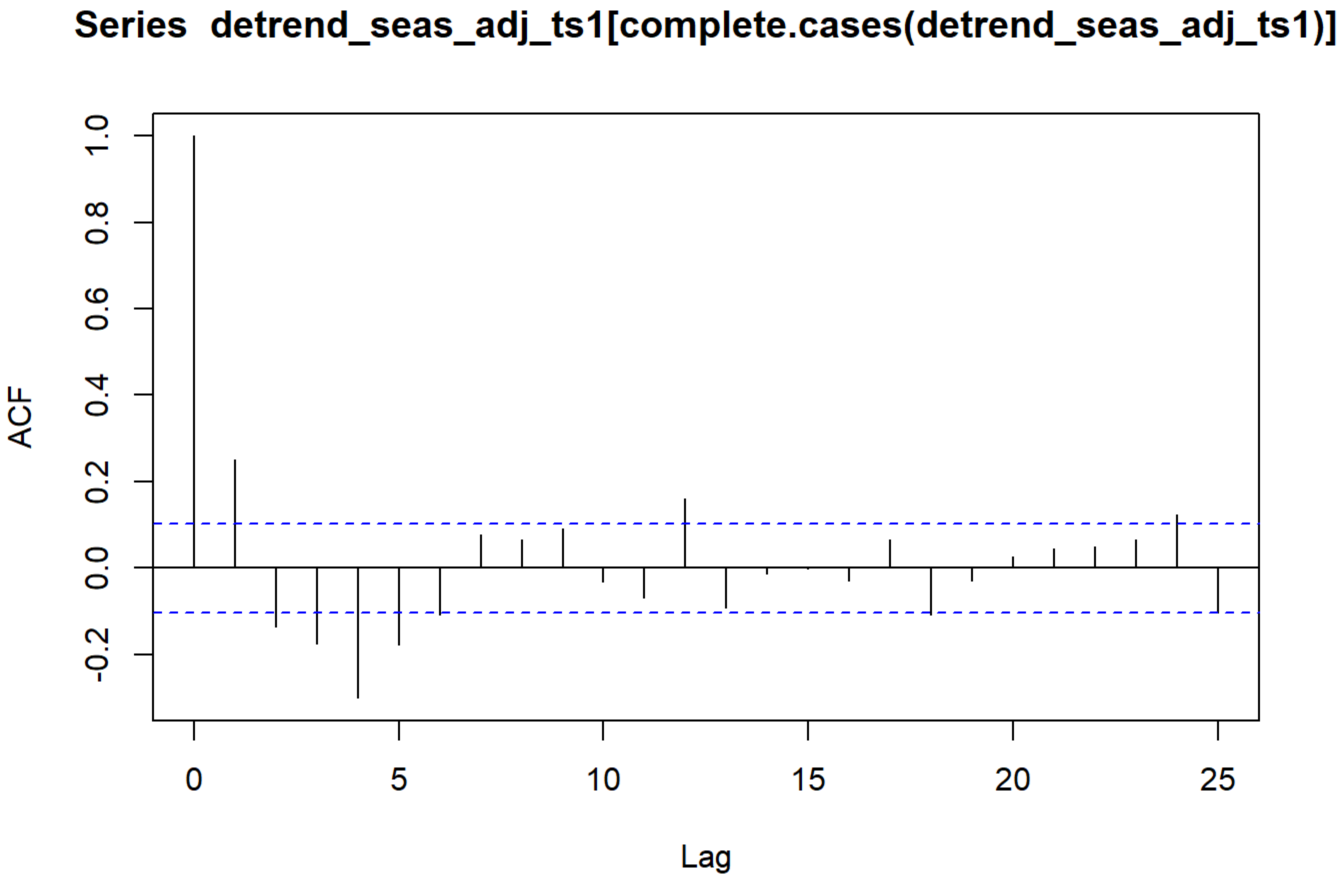
Part 2

(a)

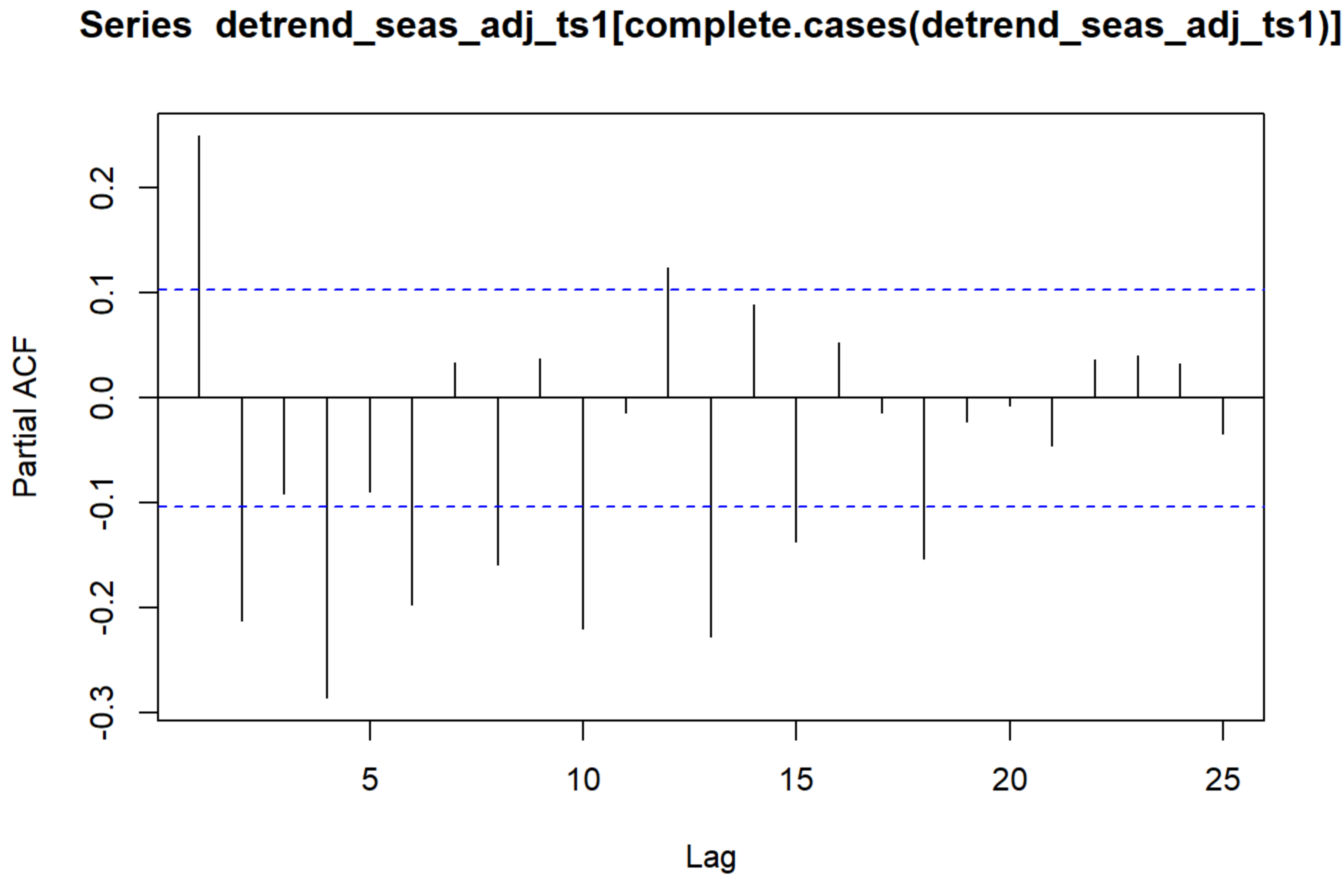
```
#Do additive decomposition of time series
dcmp_add = decompose(retail_ts, "additive")

#Isolate trend and seasonal components
ts_trend1 = dcmp_add$trend
ts_seasonal1 = dcmp_add$seasonal

#Detrend and seasonally adjust time series
detrend_seas_adj_ts1 = retail_ts - ts_trend1 - ts_seasonal1
#Plot refined time series
autoplot(detrend_seas_adj_ts1)
```




```
#Plot ACF of detrended and seasonally adjusted time series
#Detrending causes NAs to exist at the start and end of the new time series
#To fix this and plot the data, we only keep non NAs (complete cases)
acf(detrend_seas_adj_ts1[complete.cases(detrend_seas_adj_ts1)])
```




```
#Similarly plot PACF
pacf(detrend_seas_adj_ts1[complete.cases(detrend_seas_adj_ts1)])
```

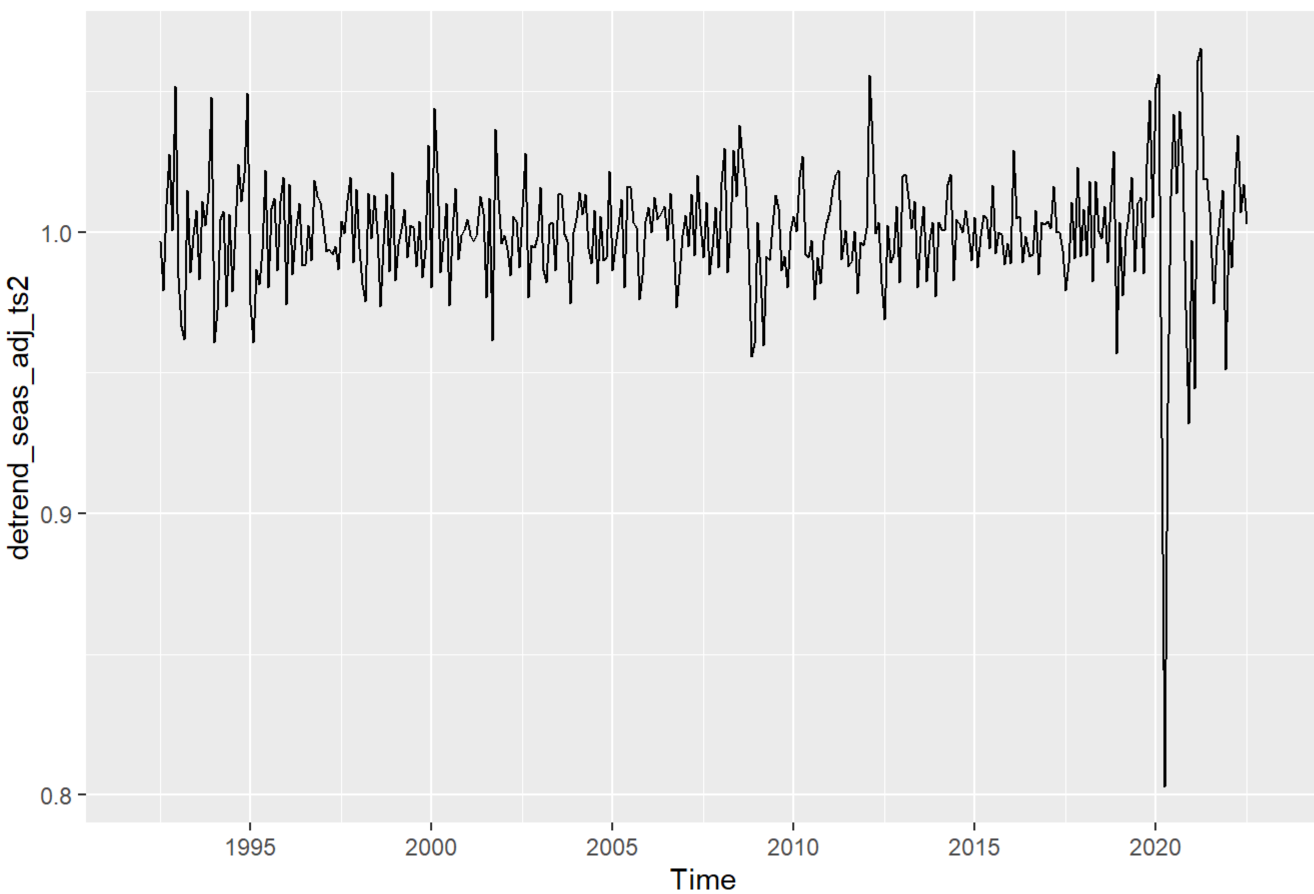


Part

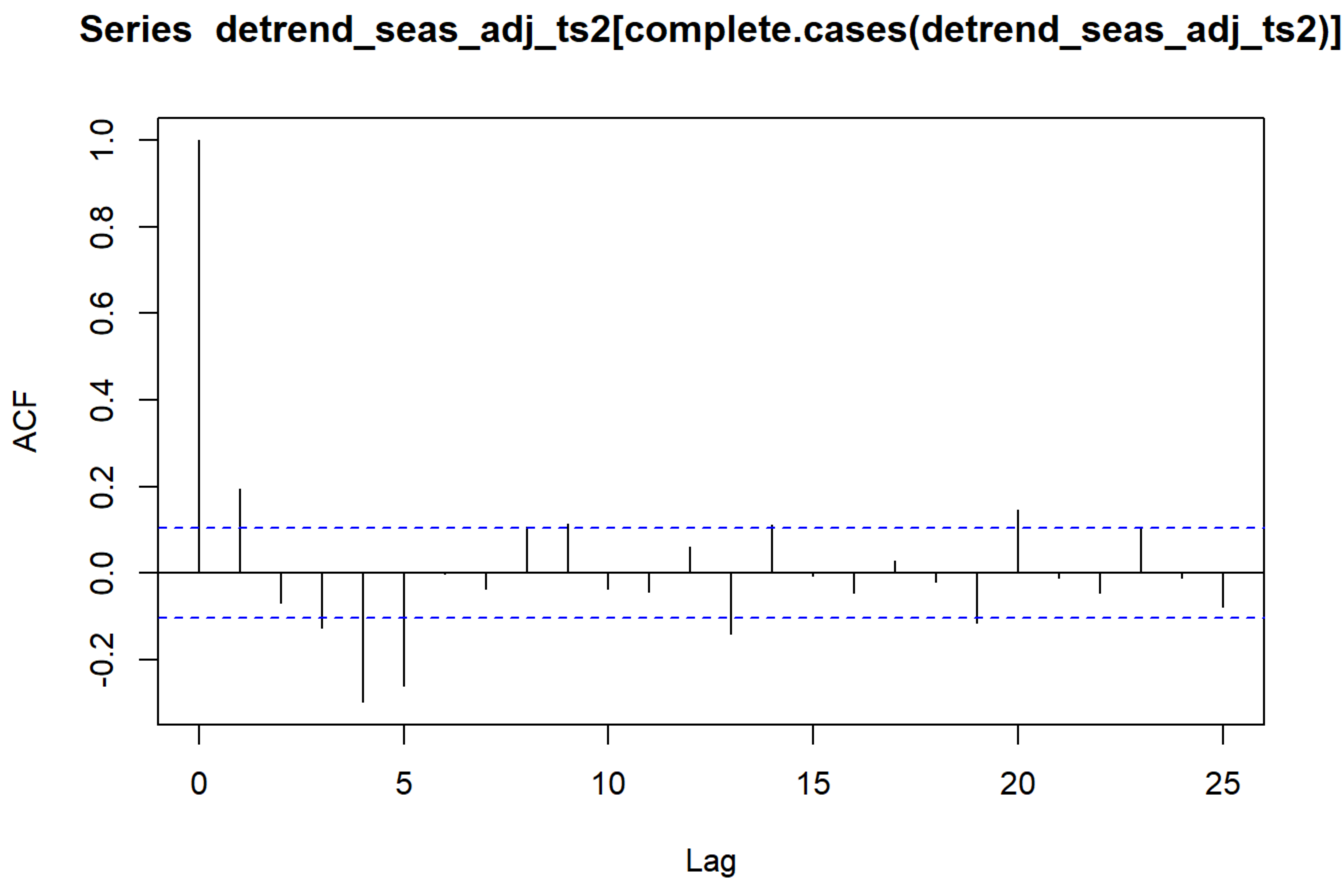
A contains the ACF and PACF for our linear model. Our most significant values For our ACF chart, we see that the most statistically important lags are our 2nd, 4th, and 12th lags. When visually observing the seasonal patterns of a chart provided by our data, we can see that every month following a selected period tends to move in the same direction, resulting in a positively value for our model coefficient. Also, we can see that quarterly increments present a reversal of the gains or losses made by the previous 4 months, coinciding with our ACF plot. Lastly, for each year that passes, every value tends to be slightly larger than the previous year 12 months, as expected by a growing economy like the United States. For our PACF chart, we can see that there are multiple statistically significant lags for our linear model with our 18th lag as our last possible significant lag. Similar to our ACF test, the 2nd, 4th, and 12th lags provide statistically significant correlations between the observed period and previous periods. Also, there are many more other statistically significant periods that affect our model according to the PACF test, such as the 6th, 8th, and 10th lags of our model. These lags are not as intuitively correlated with our non-linear model.

(b)

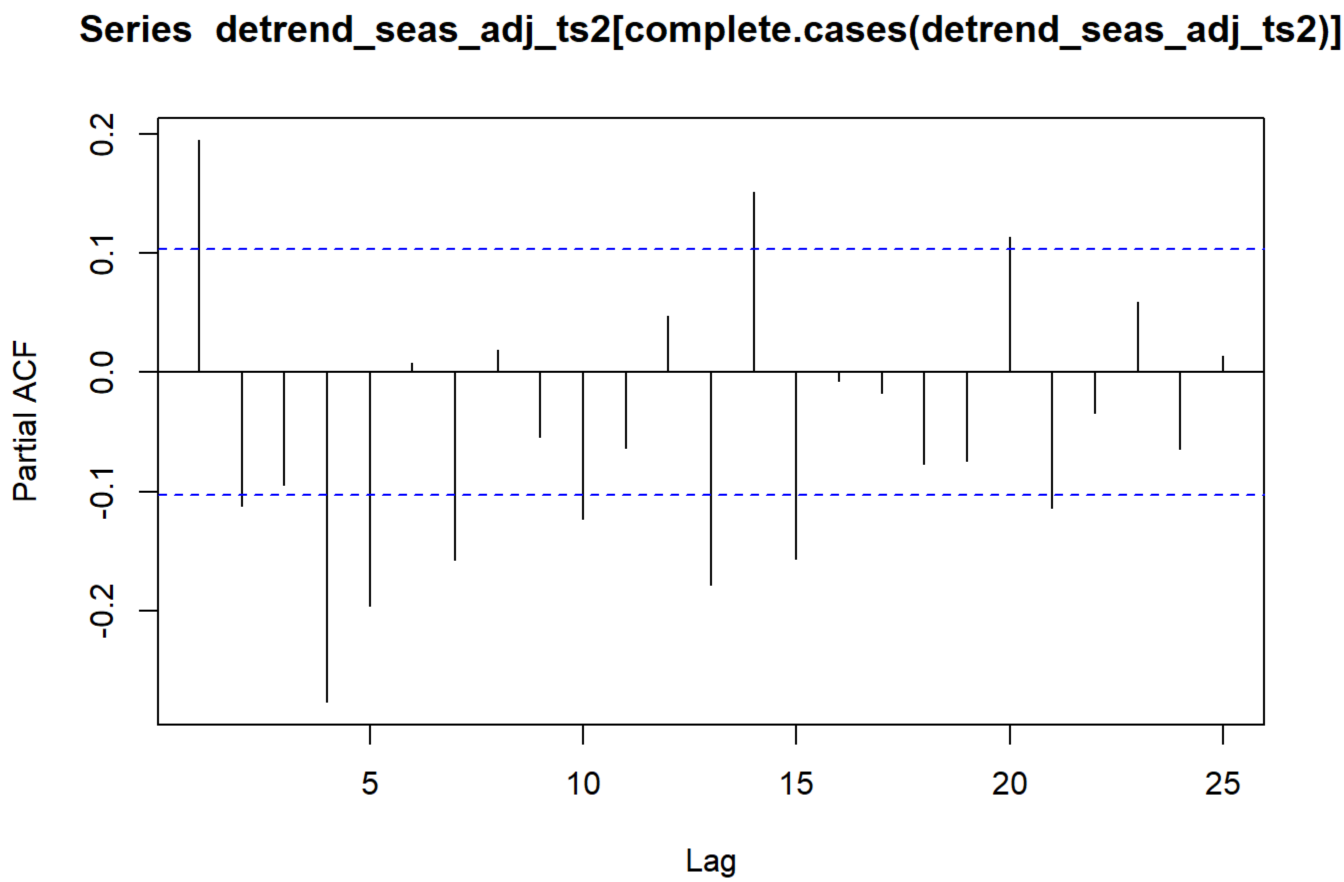
```
#Do multiplicative decomposition of time series
dcmp_mult = decompose(retail_ts, "multiplicative")

#Isolate trend and seasonal components
ts_trend2 = dcmp_mult$trend
ts_seasonal2 = dcmp_mult$seasonal

#Detrend and seasonally adjust time series
detrend_seas_adj_ts2 = (retail_ts / ts_trend2) / ts_seasonal2
#Plot refined time series
autoplot(detrend_seas_adj_ts2)
```




```
#Plot ACF of detrended and seasonally adjusted time series
#Detrending causes NAs to exist at the start and end of the new time series
#To fix this and plot the data, we only keep non NAs (complete cases)
acf(detrend_seas_adj_ts2[complete.cases(detrend_seas_adj_ts2)])
```




```
#Similarly plot PACF
pacf(detrend_seas_adj_ts2[complete.cases(detrend_seas_adj_ts2)])
```



This part contains our PACF and ACF charts for our non-linear model with a multiplicative decomposition. Similar to the PACF and ACF charts of an additive decomposition of our model, the seasonal adjustment appears to be covariance stationary.

(c)

For our data, it appears that our multiplicative and additive decompositions characterize our seasonal adjustments almost equally. Either an additive or multiplicative decomposition will sufficiently transform our data to express accurate changes in trends excluding seasonal fluctuations. This is because either de-trending process will result in covariance stationary data for the non-linear model's seasonal component. Since our data only contains the last few decades of retail sales, and it seems that the variance grows as time proceeds pictured in the charts of the data, it would not be surprising to see a multiplicative decomposition supersede an additive decomposition when more decades of data are collected either in the future, or possibly searched for outside of FRED from the past.

(d)

Our models for cycles would be similar if we additively or multiplicatively decomposed our series. Isolating the cycles would leave us with the same spikes in retail sales. Similar to the last statement of part C, extra data could show that a multiplicative decomposition could beat an additive decomposition if there is more data of a continuously compounded growing economy. Our random components for both decompositions appear to have large spikes at moments of economic deterioration, such as in 2000, 2008, and 2020.

Section III

After evaluating our candidate models to determine the most accurate trend to predict retail sales, we believe that our non-linear model more closely follows the movement in retail sales levels of the United States. One way our model could be improved is by adding more years of data to determine the most effective method of decomposing our model's trend, seasonal, and remainder components. Our additive and multiplicative decomposition presented similar results, but when considering the effect of a continuously compounding economy that has a proportionately growing retail sector, it seems like a multiplicative adjustment would provide a more accurate method of determining covariance stationarity for our trend, which is compatible with a visual examination of our time series data. Also, we could more finely improve our model by selecting better variables and possibly filter the time series. Adding a cosine function significantly improved our linear model, so adjusting our model to match the general pattern of the data would even further improve our model. Lastly, another way to improve our model could be to isolate the cycle to characterize our trend even more accurately. We did not consider supplementing our decomposition by isolating our cycle. The instructions were limited to only seasonal decomposition, but it would be beneficial to include a component to understand the importance of cycles in our time series.

Section IV

Our CSV file contains values for each month's advance retail sales collected by the US Census Bureau and stored by FRED over the last few decades. The US Census Bureau only reports the new advanced retail sales values with the previous two months and the same three months in the previous year. To get a more comprehensive understanding of the economic trend and seasonal fluctuations of advance retail sales, FRED collects the data and keeps an inventory of all values going back to 1992. FRED Advanced Retail Sales Chart: https://fred.stlouisfed.org/series/RSAFSNA (https://fred.stlouisfed.org/series/RSAFSNA) US Census Bureau March 2023 Report: https://www.census.gov/retail/marts/www/marts_current.pdf (https://www.census.gov/retail/marts/www/marts_current.pdf)