

# ECON 430: Project 2

*Roya Latifi, Cristian Martinez, Aneri Patel, Anshika Sharma*

*December 9, 2020*

## Contents

<b>I. Introduction</b>	<b>2</b>
<b>II. Results</b>	<b>2</b>
1. Time-series plots and respective ACF and PACF plots. . . . .	3
2. Fitting an ARIMA model to both series using auto.arima . . . . .	8
3. Fitting a model that includes, trend, seasonality and cyclical components . . . . .	14
4. Plotting residuals vs. fitted values . . . . .	43
5. Plotting the ACF and PACF of the respective residuals . . . . .	47
6. Plot the respective CUSUM and interpret the plot. . . . .	49
7. Plot the respective Recursive Residuals and interpret the plot. . . . .	53
8. Diagnostic statistics . . . . .	55
9. Forecasting 12-steps ahead with the respective error bands . . . . .	59
10. Fitting an appropriate VAR model . . . . .	61
11. Impulse response functions . . . . .	68
12. Granger-Causality test . . . . .	70
13. Using VAR model to forecast 12-steps ahead with respective error bands . . . . .	71
14. Backtesting the ARIMA model . . . . .	73
<b>III. Conclusion and Future Work</b>	<b>84</b>
<b>IV. References:</b>	<b>84</b>

# I. Introduction

For the purpose of this project, the two variables we will be using are the University of Michigan's Consumer Sentiment and Total Vehicle Sales in United States from January 1978 to October 2020.

The University of Michigan's Consumer Sentiment (measured in ratios) is a statistical measurement and economic indicator of the overall health of the economy as determined by consumer opinion. Consumer sentiment takes into account an individual's feelings toward his or her current financial health, the health of the economy in the short-term, and the prospects for longer-term economic growth. The questionnaire is randomly conducted monthly and by phone.

Total Vehicle Sales (measured in thousands of units) comprises of all classes of vehicles (commercial to C-Class), both imported and domestically made.

These two variables were selected due to the possible link between consumers perceived level of confidence and their decision of purchasing a new vehicle. Many believe that sales of vehicles are driven by the level of confidence that consumers have on the outlook of the macro-economy.

By using these two variables together, we will not only be able to forecast the level of confidence that individuals will have, but also forecast the future number of vehicle sales, which will be of importance not only from an investment perspective, but also from an academic perspective. Of course there are other factors that influence individuals to purchase a new vehicle, but one can argue that individuals would not commit to a large purchase if they did not believe the economy was operating poorly due to the risk of becoming unemployed.

The purpose of this research is to develop a forecast of both economic variables and also derive statistical support that vehicle sales are influenced by consumer confidence. The study also aims to show how both economic variables are closely linked.

# II. Results

```
#Calling our data
library(readxl)

data <- read_excel("Consumer_confidence.xlsx")
head(data)
```

```
## # A tibble: 6 x 3
##   Date                Vehicle_Sold Sentiment
##   <dtm>                <dbl>      <dbl>
## 1 1978-01-01 00:00:00      931.       83.7
## 2 1978-02-01 00:00:00     1071.       84.3
## 3 1978-03-01 00:00:00     1481.       78.8
## 4 1978-04-01 00:00:00     1406.       81.6
## 5 1978-05-01 00:00:00     1558.       82.9
## 6 1978-06-01 00:00:00     1549.       80
```

```
vehicle = cbind(data[,1], data[,2])
sentiment = cbind(data[,1], data[,3])
```

```
#Saving our data as ts
```

```

vehicle.ts = ts(vehicle$Vehicle_Sold, start = 1978, frequency = 12)
sentiment.ts = ts(sentiment$Sentiment, start = 1978, frequency = 12)

#Creating time dummy
t<-seq(1978, 2020,length=length(vehicle.ts))
t2 <- t^2

```

## 1. Time-series plots and respective ACF and PACF plots.

- Vehicles Sold:

```

#Plotting our time series
#frequency = 12 as this is a monthly data
plot(vehicle.ts , main = 'Time Series Plot of Vehicle Sold',
      ylab= "Vehicle Sold")
abline(h=mean(vehicle$Vehicle_Sold), col="red", lwd=2)

```

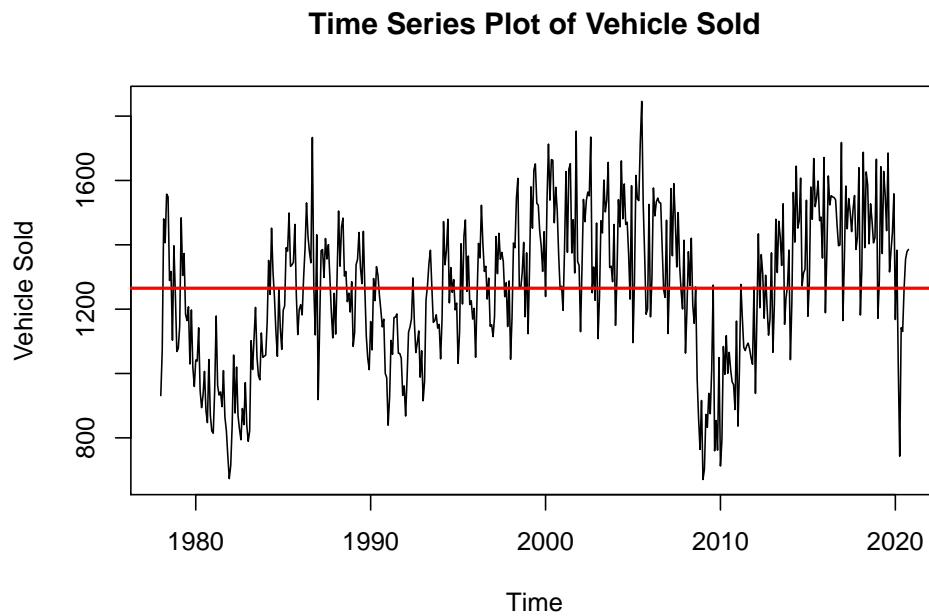


Figure 1: Time Series Plot for Vehicles Sold

```
#ACF and PACF
```

```
acf(vehicle$Vehicle_Sold, type = "correlation", main="Autocorrelation", ylab="ACF", lag.max = 60)  
pacf(vehicle$Vehicle_Sold, main="Partial Autocorrelation", ylab="PACF", lag.max = 60)
```

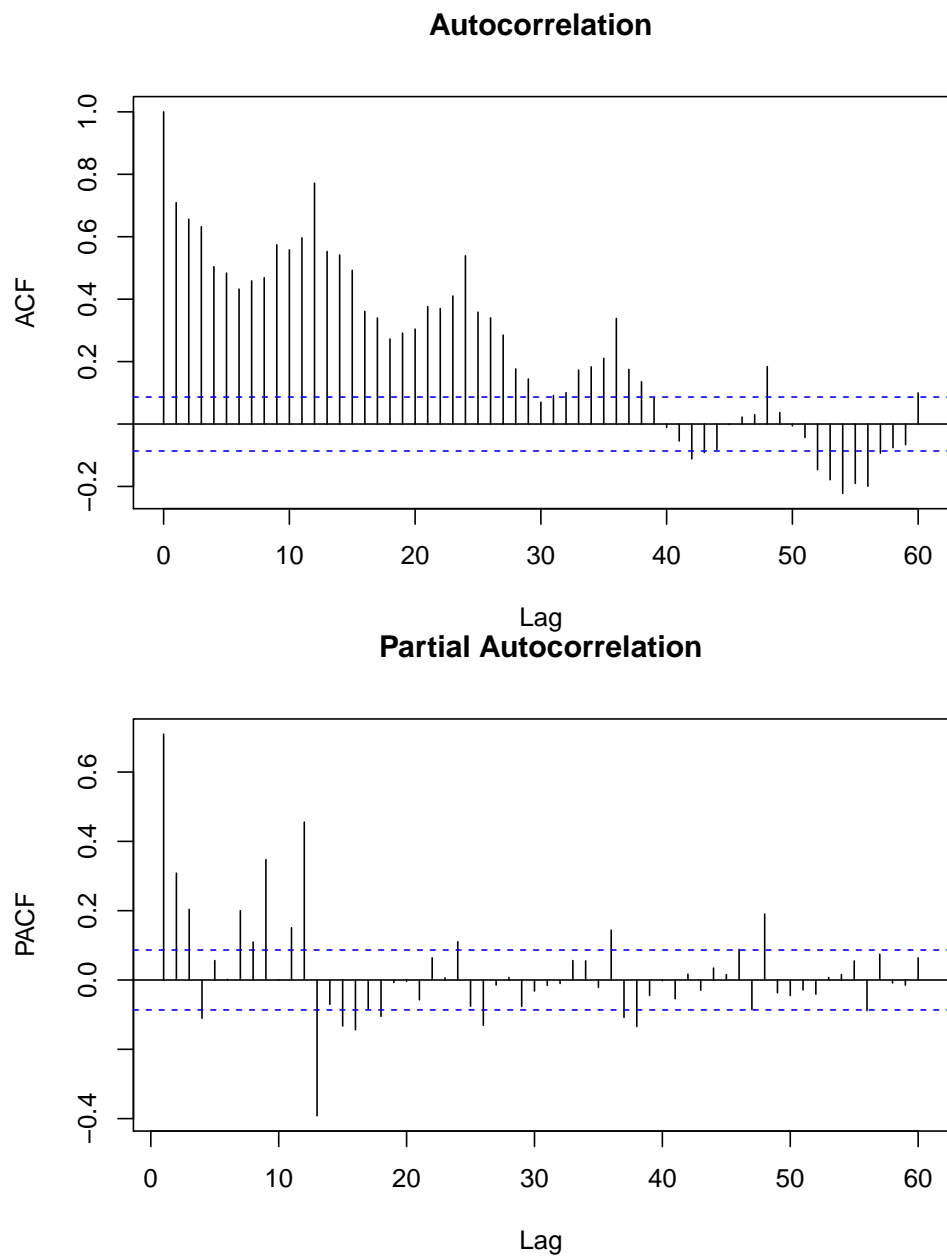


Figure 2: Time Series Plot for Vehicles Sold

The autocorrelation function measures the correlation between observations of a time series. The ACF plot in the given figure shows a significant correlation at lag 1 that gradually declines to 0, while continuing to spike at seasonality.

The partial autocorrelation function also measures the ACF, after adjusting for the presence of all other terms of smaller lags. PACF spike at 1 and then at 12, before decaying to zero indicating that we might have seasonal MA(1) process. The PACF plot has significant spikes till lag 3.

ACF and PACF suggest an MA(1) process, and an AR(3) process.

ACF spike at 1, PACF oscillating and decaying to 0 appears to look like a MA(1) process. PACF has significant spikes till lag 3 signaling and ACF oscillating and decaying to 0 indicates AR(3). PACF has largest spike at 12 = 12x1 signaling seasonal AR(1) process.

```
adf.test(vehicle$Vehicle_Sold,
        k = trunc((length(vehicle$Vehicle_Sold)-1)^(1/3)))
```

```
##
## Augmented Dickey-Fuller Test
##
## data: vehicle$Vehicle_Sold
## Dickey-Fuller = -2.4731, Lag order = 8, p-value = 0.378
## alternative hypothesis: stationary
```

We will now test if our data for Total Vehicle Sales is stationary using the Augmented Dickey-Fuller Test. Based on a p-value > 0.05, we fail to reject the null hypothesis and conclude that Total Vehicle Sales is Non-Stationary.

- Sentiment :

```
#Plotting our time series
#frequency = 12 as this is a monthly data
plot(sentiment.ts , main = 'Time Series Plot of Sentiment',
     ylab= "Vehicle Sold")
abline(h=mean(sentiment$Sentiment), col="red", lwd=2)
```

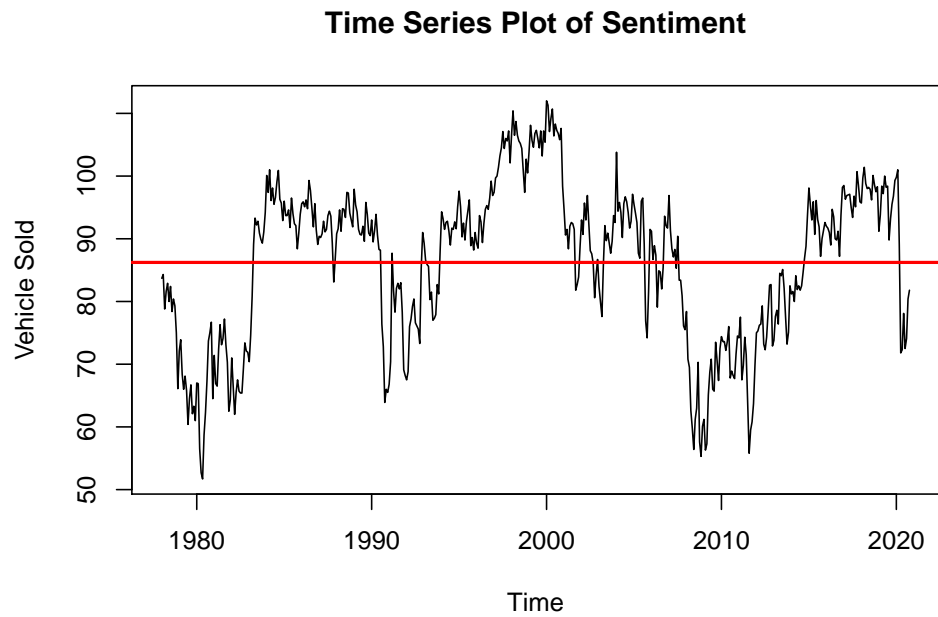


Figure 3: Time Series Plot for Sentiment

```
#ACF and PACF  
acf(sentiment$Sentiment, type = "correlation", main="Autocorrelation", ylab="ACF", lag.max = 60)  
pacf(sentiment$Sentiment, main="Partial Autocorrelation",  
      ylab="PACF", lag.max = 60)
```

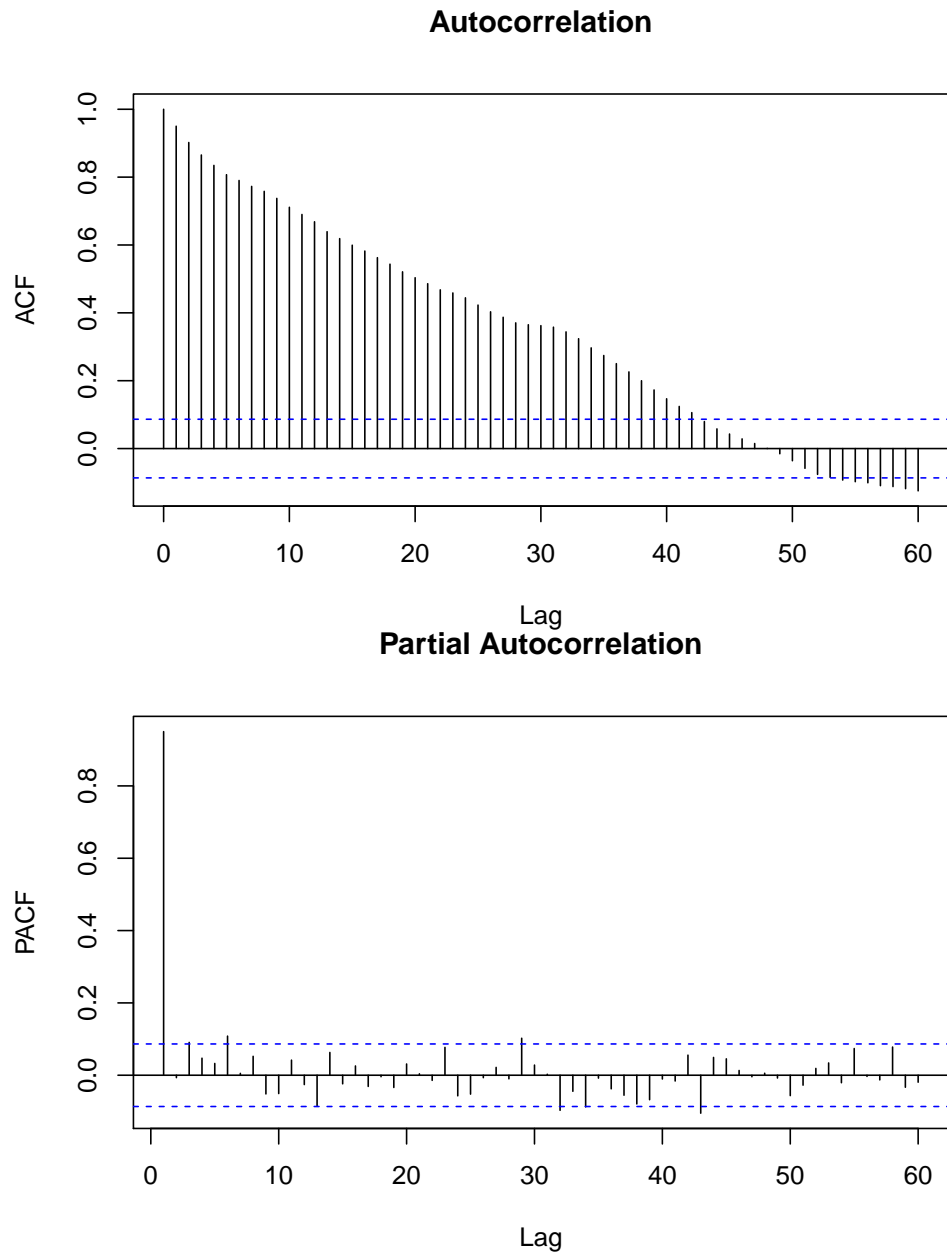


Figure 4: Time Series Plot for Sentiment

PACF has significant spike at lag 1 and ACF oscillating and decaying to 0 indicates AR(1) process.

```
adf.test(sentiment$Sentiment,  
        k = trunc((length(sentiment$Sentiment)-1)^(1/3)))
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: sentiment$Sentiment  
## Dickey-Fuller = -2.4694, Lag order = 8, p-value = 0.3796  
## alternative hypothesis: stationary
```

Similar to what we did for Total Vehicle Sales, we will also test our Consumer Sentiment data to determine if the data is stationary using the Augmented Dickey-Fuller test. With a p-value > 0.05, we fail to reject the null hypothesis and conclude that Consumer Sentiment is Non-Stationary. Thus, we conclude that the data has unit root.

## 2. Fitting an ARIMA model to both series using auto.arima

- Total vehicle sales

```
# STL decomposition  
plot(stl(vehicle.ts, s.window = "periodic"), main="Total Vehicle Sales")  
auto.arima(vehicle.ts)
```

```
## Series: vehicle.ts  
## ARIMA(3,0,1)(2,1,1)[12]  
##  
## Coefficients:  
##          ar1      ar2      ar3      ma1      sar1      sar2      sma1  
##          1.1063 -0.1994  0.0625 -0.6370  0.2527 -0.1448 -0.8184  
## s.e.    0.1098  0.0805  0.0634  0.1022  0.0628  0.0558  0.0435  
##  
## sigma^2 estimated as 9642: log likelihood=-3017.25  
## AIC=6050.51   AICc=6050.8   BIC=6084.26
```



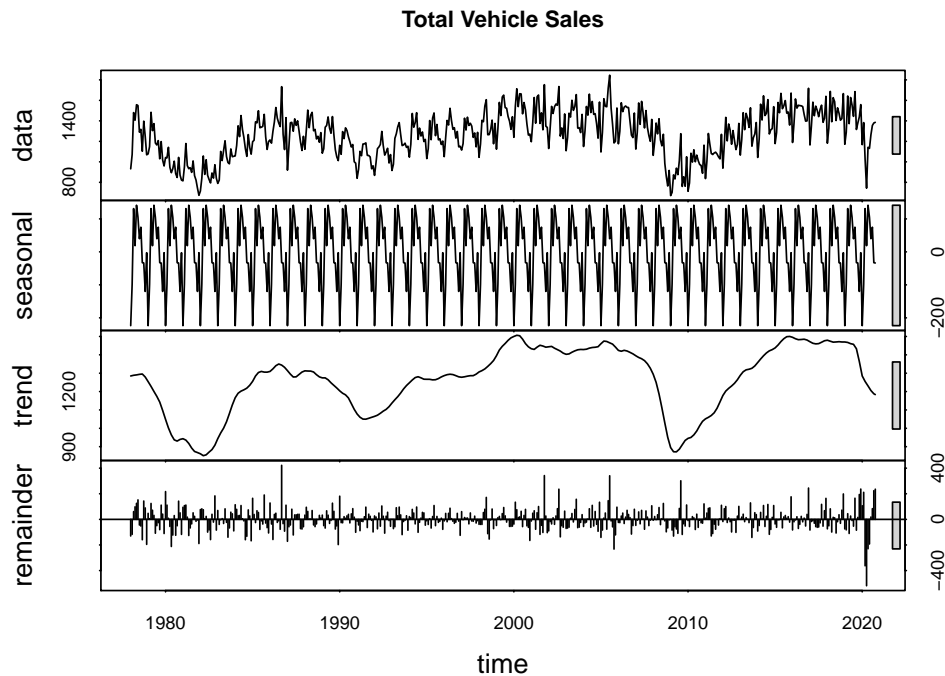


Figure 5: STL Decomposition of Vehicles Sold

From the STL decomposition plot above, it is evident that there are trend and seasonality components. When we look at the remainder component, the magnitude signals that there might be a cyclical component in our data as well.

According to the result of `auto.arima`, the fit model has AR(3), MA(1), Seasonal-AR(2) and Seasonal-MA(1).

```
#Fitting Arima Model
model_vehicle = Arima(vehicle.ts,order=c(3,0,1),seasonal=list(order=c(2,1,1)))
summary(model_vehicle)
```

```
## Series: vehicle.ts
## ARIMA(3,0,1)(2,1,1)[12]
##
## Coefficients:
##          ar1      ar2      ar3      ma1      sar1      sar2      sma1
##      1.1063 -0.1994  0.0625 -0.6370  0.2527 -0.1448 -0.8184
## s.e.  0.1098  0.0805  0.0634  0.1022  0.0628  0.0558  0.0435
##
## sigma^2 estimated as 9642:  log likelihood=-3017.25
## AIC=6050.51   AICc=6050.8   BIC=6084.26
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.8827013 96.36307 68.3354 -0.4307902 5.642638 0.6289508
##              ACF1
## Training set 0.002326596
```

```
coeftest(model_vehicle)
```

```
##
```

```
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1    1.106320   0.109833 10.0727 < 2.2e-16 ***
## ar2   -0.199444   0.080462 -2.4787  0.01318 *
## ar3    0.062451   0.063373  0.9855  0.32440
## ma1   -0.636957   0.102152 -6.2354 4.507e-10 ***
## sar1    0.252748   0.062838  4.0222 5.766e-05 ***
## sar2   -0.144754   0.055814 -2.5935  0.00950 **
## sma1   -0.818353   0.043525 -18.8020 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Arima(vehicle.ts,order=c(2,0,1),seasonal=list(order=c(2,1,1)))
```

```
## Series: vehicle.ts
## ARIMA(2,0,1)(2,1,1)[12]
##
## Coefficients:
##          ar1          ar2          ma1          sar1          sar2          sma1
##          1.1639   -0.1885   -0.7003   0.2548   -0.1400   -0.8211
## s.e.    0.0790    0.0730    0.0600    0.0625    0.0555    0.0430
##
## sigma^2 estimated as 9642:  log likelihood=-3017.73
## AIC=6049.46   AICc=6049.68   BIC=6078.99
```

```
plot(vehicle.ts, ylab = "Vehicles Sold", main="ARIMA Model For Total Vehicles Sold")
lines(fitted(model_vehicle), col = "red")
```

## ARIMA Model For Total Vehicles Sold

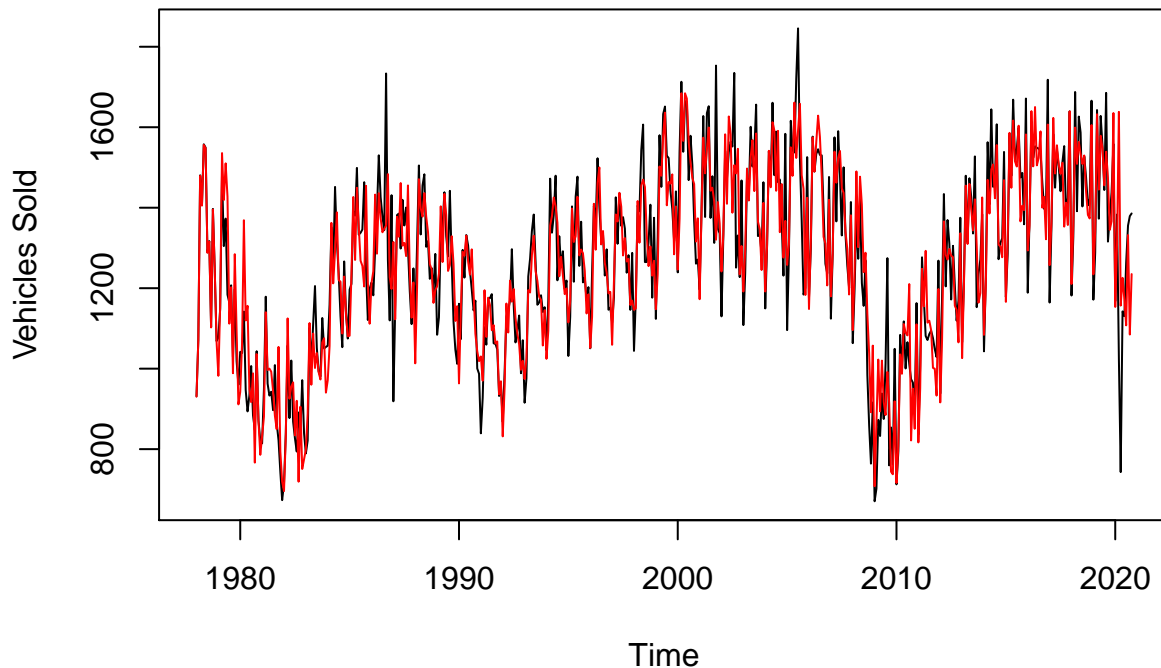


Figure 6: Arima Model Fit

Looking at the plot above, The Arima derived using the Auto.Arima command seems to fit the vehicle data quite well. However, it does not appear to account for the larger spikes. While the auto.arima function suggests AR(3), the coeftest shows that AR(3) is not significant, even at the 10% level. This suggests that there may be room for further improvement.

- Consumer Sentiment

```
# STL decomposition
plot(stl(sentiment.ts, s.window = "periodic"), main = "Consumer Sentiment")
```

```
#Using auto.arima
auto.arima(sentiment.ts)
```

```
## Series: sentiment.ts
## ARIMA(2,1,1)(1,0,1)[12]
##
## Coefficients:
##          ar1          ar2          ma1          sar1          sma1
##          0.7172   -0.0989   -0.7554    0.6540   -0.5758
## s.e.    0.1363    0.0508    0.1307    0.4108    0.4441
##
## sigma^2 estimated as 15.35:  log likelihood=-1426.06
## AIC=2864.12   AICc=2864.28   BIC=2889.56
```

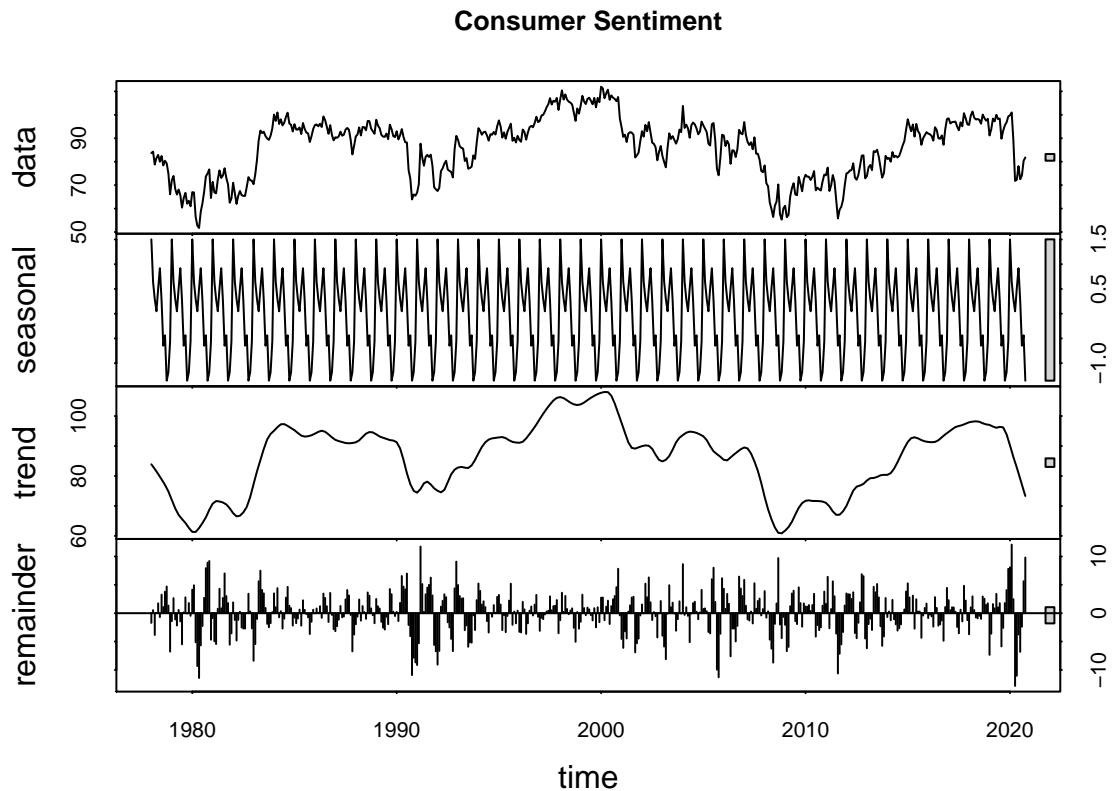


Figure 7: STL Decomposition of Sentiments

From the STL decomposition plot above, it is evident that there are trend and seasonality components. When we look at the remainder component, the magnitude signals that there might be a cyclical component in our data as well. The order of magnitude for the trend is larger, indicating that there might be a stronger trend component than seasonal component in the data.

According to the result of `auto.arima`, the fit model has AR(2), MA(1), Seasonal-AR(1) and Seasonal-MA(1) and one-period difference cycle.

```
#Fitting the model
model_sentiment = Arima(sentiment.ts, order=c(2,1,1), xreg = cbind(t), seasonal=list(order=c(1,0,1)))
summary(model_sentiment)
```

```
## Series: sentiment.ts
## Regression with ARIMA(2,1,1)(1,0,1)[12] errors
##
## Coefficients:
##      ar1      ar2      ma1      sar1      sma1      t
##      0.7180 -0.0988 -0.7561  0.6534 -0.5750 -0.1658
## s.e.  0.1361  0.0508  0.1305  0.4065  0.4394  1.6391
##
## sigma^2 estimated as 15.38:  log likelihood=-1426.05
## AIC=2866.11   AICc=2866.33   BIC=2895.79
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.004300743 3.894863 2.962178 -0.1468642 3.619064 0.390569
```

```
##                               ACF1
## Training set -0.0001031855
```

```
coeftest(model_sentiment)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.718035   0.136116  5.2752 1.326e-07 ***
## ar2  -0.098838   0.050807 -1.9454  0.05173 .
## ma1  -0.756148   0.130485 -5.7949 6.836e-09 ***
## sar1  0.653373   0.406482  1.6074  0.10797
## sma1 -0.574979   0.439416 -1.3085  0.19070
## t    -0.165849   1.639123 -0.1012  0.91941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(sentiment.ts, ylab= "Consumer Sentiment Level",
main = "ARIMA Model For Consumer Sentiment")
lines(fitted(model_sentiment), col = "red")
```

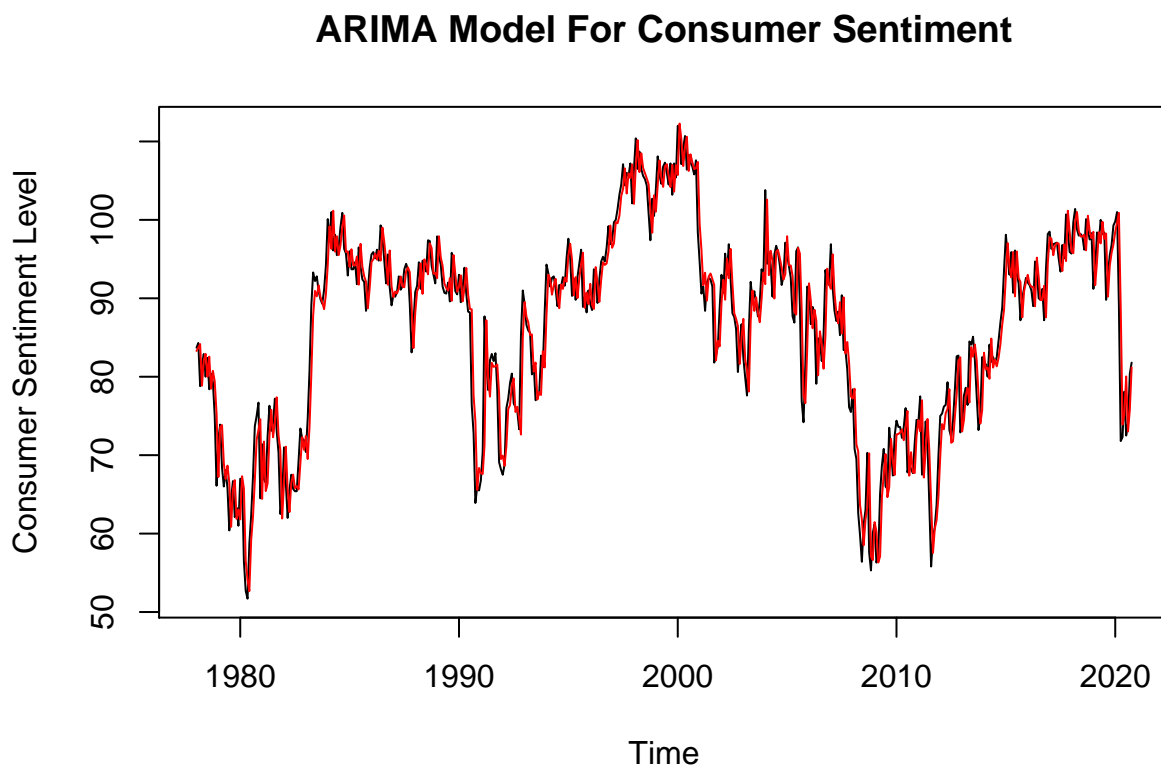


Figure 8: Arima Model Fit

The Auto.Arima model has been fitted on the data above. It seems to fit the vehicle data quite well. However, the `coefTest` shows that S-AR(1), S-MA(1) and `t` are not significant, even at the 10% level. Only the beta coefficients AR(1) and MA(1) are statistically significant. This suggests that there may be room for further improvement.

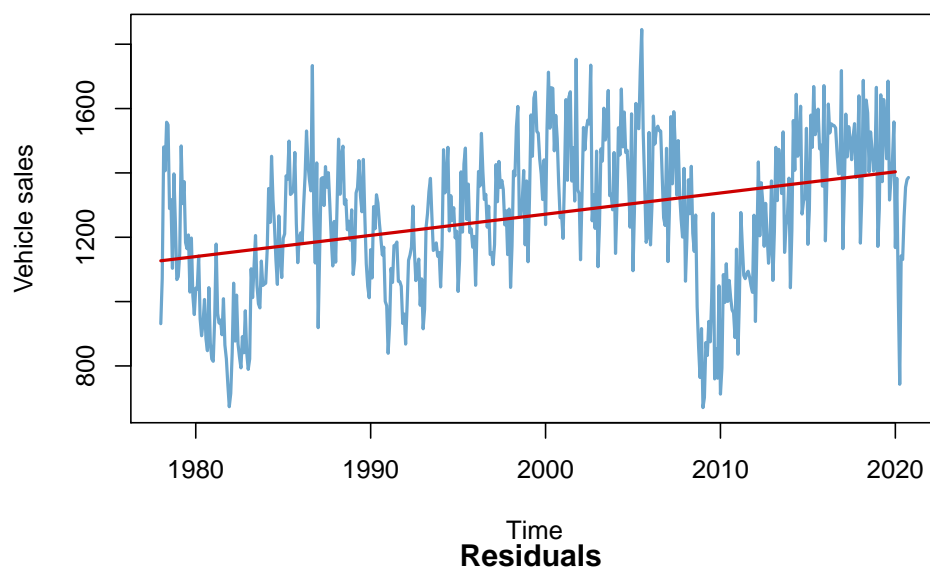
### 3. Fitting a model that includes, trend, seasonality and cyclical components

Vehicle

```
# Linear trend model #
m=lm(vehicle.ts~t)

plot(vehicle.ts,ylab="Vehicle sales", xlab="Time", lwd=2, col='skyblue3',
main = "Linear Trend Fit (Total Vehicles Sold)")
lines(t,m$fit,col="red3",lwd=2)
plot(t,m$res, ylab="Residuals",type='l',xlab="Time", main = "Residuals")
abline(h = 0, col = "red", lwd = 2)
```

**Linear Trend Fit (Total Vehicles Sold)**



**Residuals**

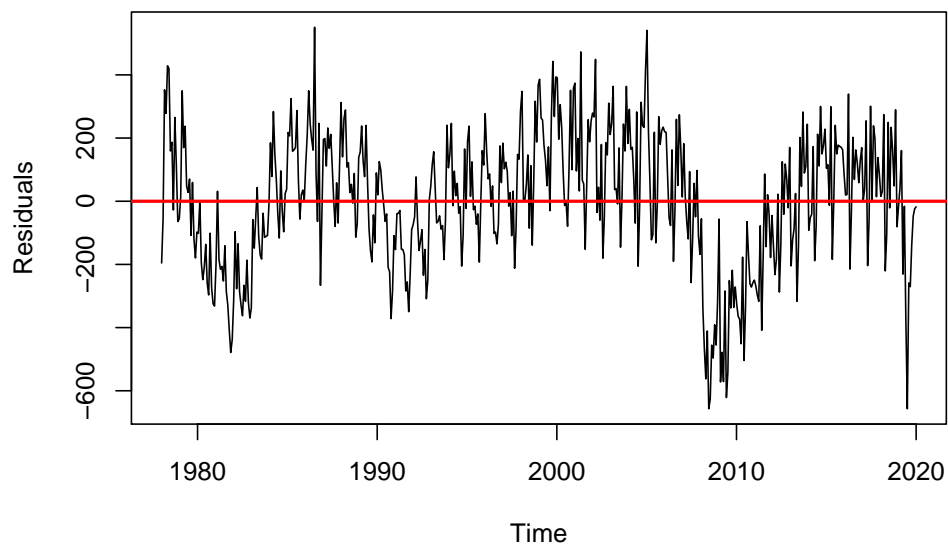


Figure 9: Linear

```
acf(m$res,lag=36,main="Residual Sample Autocorrelations")
pacf(m$res,lag=36,main="Partial Residual Sample Autocorrelations")
```

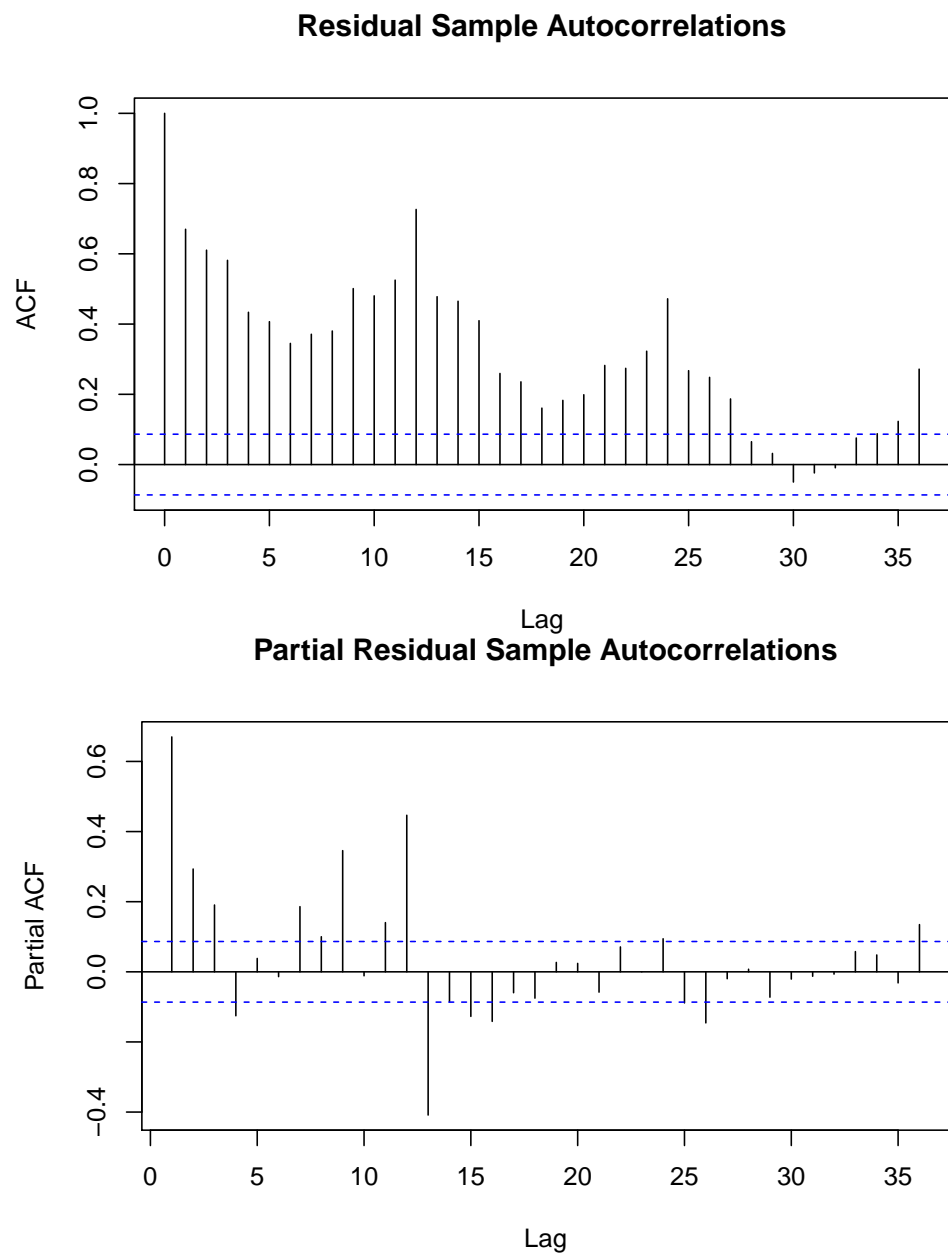


Figure 10: Linear

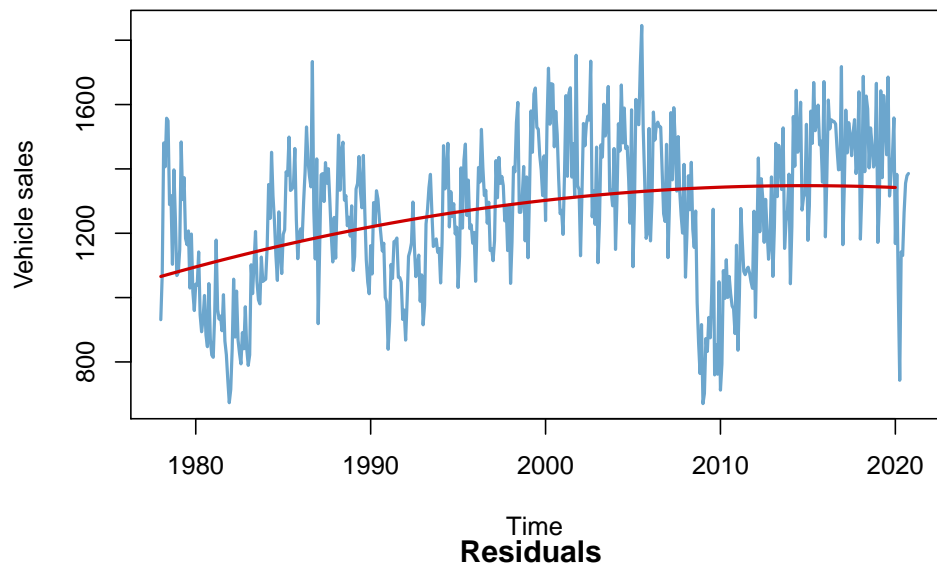


The resulting residuals from the linear trend fit do not appear to look like white noise. The residuals indicate the presence of a distinct pattern and seasonality. Since there are still dynamics left in acf and pacf residual graphs, we can say that this model does not capture all the dynamics of the data. So we will add a seasonal component later to the model to account for this seasonality and observe if the fit improves.

```
#Quadratic trend model  
m1=lm(vehicle.ts~t+t2)
```

```
plot(vehicle.ts,ylab="Vehicle sales", xlab="Time", lwd=2, col='skyblue3',  
main = "Quadratic Trend Fit (Total Vehicles Sold)")  
lines(t,m1$fit,col="red3",lwd=2)  
plot(t,m1$res, ylab="Residuals",type='l',xlab="Time", main = "Residuals")  
abline(h = 0, col = "red", lwd = 2)
```

**Quadratic Trend Fit (Total Vehicles Sold)**



**Residuals**

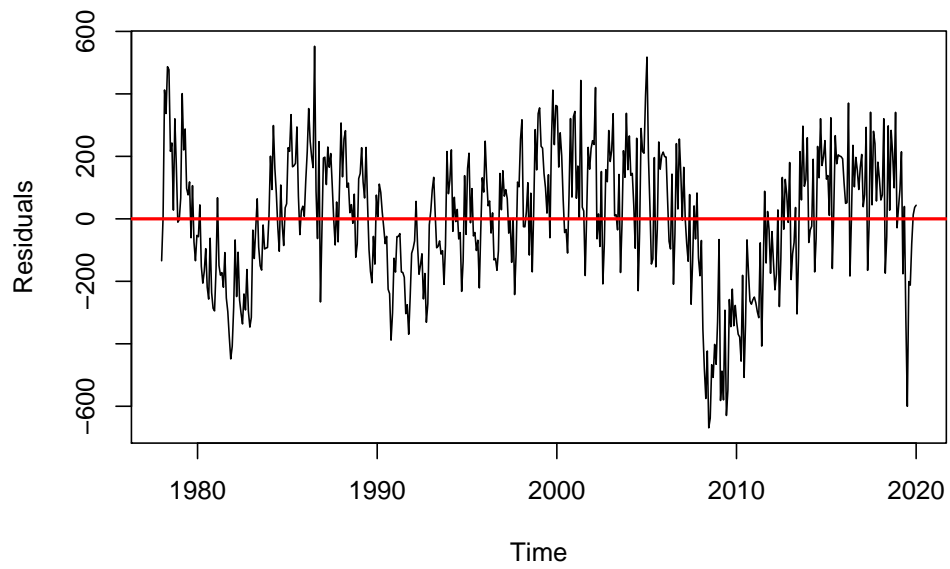


Figure 11: Quadratic

```
acf(m1$res,lag=36,main="Residual Sample Autocorrelations")
pacf(m1$res,lag=36,main="Partial Residual Sample Autocorrelations")
```

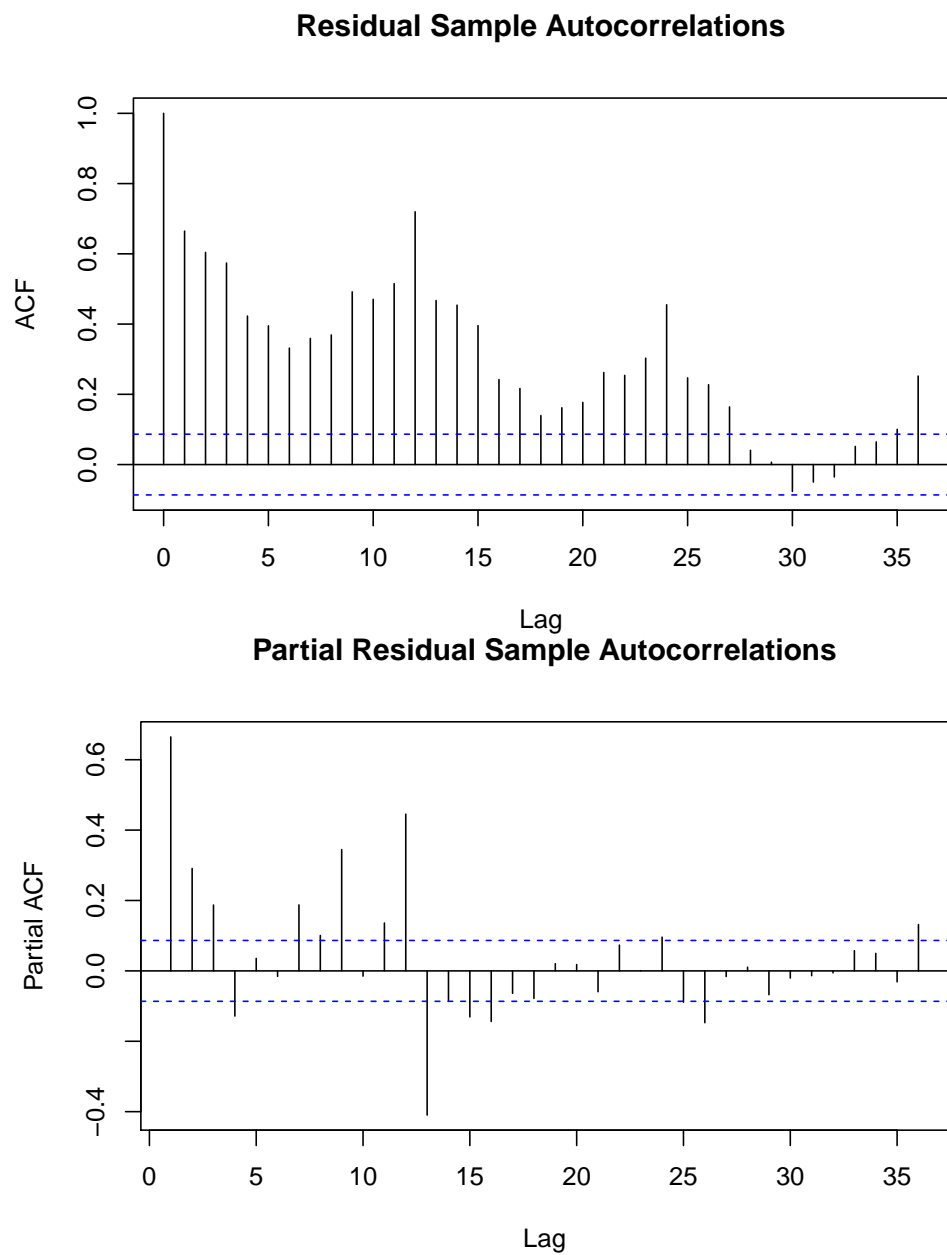


Figure 12: Quadratic

Before we account for seasonality in the vehicle model, we attempted to fit a quadratic trend model as plotted above. It seems to be the best-fit for the sequence since it captures the overall trend of total vehicle sales.

The resulting residuals from the quadratic trend fit do not appear to look like white noise. The residuals indicate the presence of a distinct pattern and seasonality. Since there are still dynamics left in acf and pacf residual graphs, we can say that this model does not capture all the dynamics of the data. So now we will add a seasonal component to the model to account for this seasonality.

```
#Trend + Season
m2=tslm(vehicle.ts~ t+t2+season)

#par(mfrow=c(4,1))
plot(vehicle.ts,ylab="Vehicle sales", xlab="Time", lwd=2, col='skyblue3',
main = "Trend & Seasonality (Total Vehicles Sold)")
lines(t,m2$fit,col="red3",lwd=2)
plot(t,m2$res, ylab="Residuals",type='l',xlab="Time", main = "Residuals")
abline(h = 0, col = "red", lwd = 2)
```

**Trend & Seasonality (Total Vehicles Sold)**

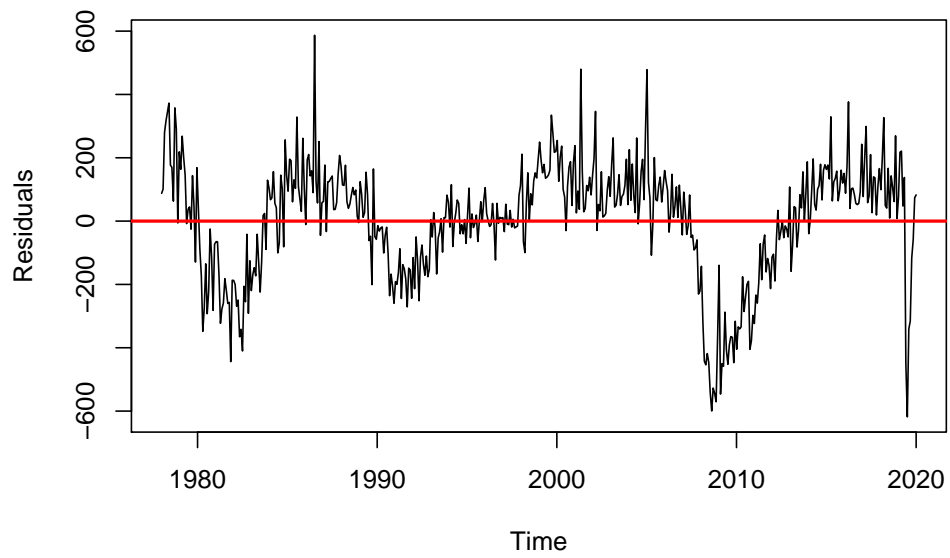
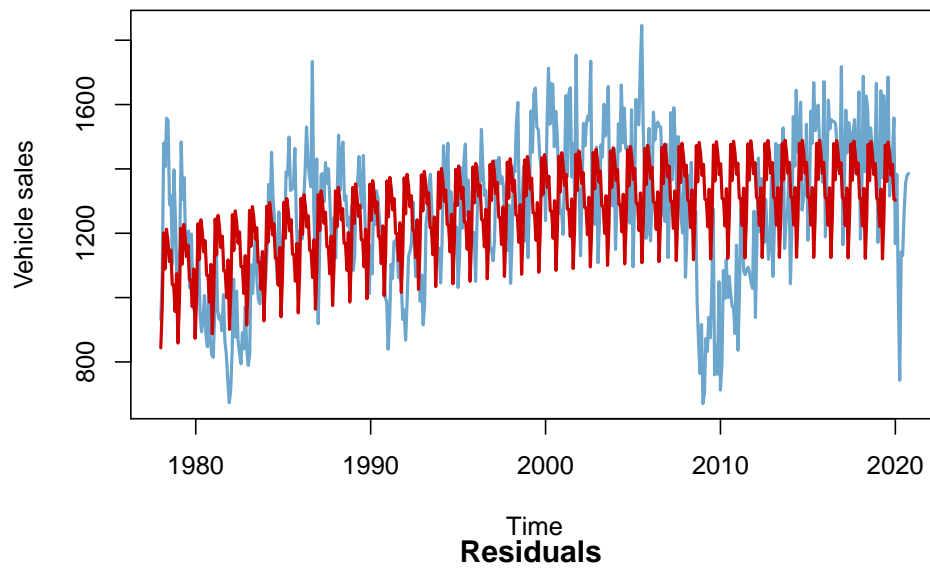


Figure 13: Trend + Season

```

acf(m2$res,lag=36,main="Residual Sample Autocorrelations")
pacf(m2$res,lag=36,main="Partial Residual Sample Autocorrelations")

summary(m2)

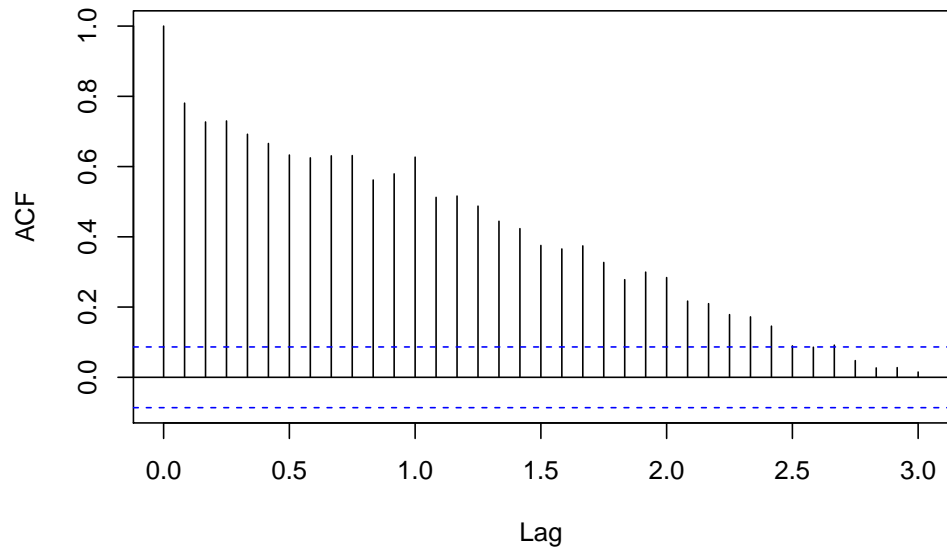
```

```

##
## Call:
## tslm(formula = vehicle.ts ~ t + t2 + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -618.15  -89.11   34.84  114.68  586.90
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.609e+05  2.480e+05  -3.471 0.000563 ***
## t            8.558e+02  2.481e+02   3.449 0.000610 ***
## t2          -2.124e-01  6.207e-02  -3.423 0.000671 ***
## season2      1.253e+02  4.005e+01   3.130 0.001851 **
## season3      3.543e+02  4.005e+01   8.847 < 2e-16 ***
## season4      2.415e+02  4.005e+01   6.030 3.19e-09 ***
## season5      3.638e+02  4.005e+01   9.083 < 2e-16 ***
## season6      3.267e+02  4.005e+01   8.158 2.77e-15 ***
## season7      2.611e+02  4.005e+01   6.518 1.74e-10 ***
## season8      2.951e+02  4.005e+01   7.367 7.25e-13 ***
## season9      1.866e+02  4.005e+01   4.660 4.07e-06 ***
## season10     1.843e+02  4.005e+01   4.602 5.30e-06 ***
## season11     1.005e+02  4.029e+01   2.496 0.012896 *
## season12     2.169e+02  4.029e+01   5.384 1.12e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 185.7 on 500 degrees of freedom
## Multiple R-squared:  0.3513, Adjusted R-squared:  0.3344
## F-statistic: 20.83 on 13 and 500 DF,  p-value: < 2.2e-16

```

### Residual Sample Autocorrelations



### Partial Residual Sample Autocorrelations

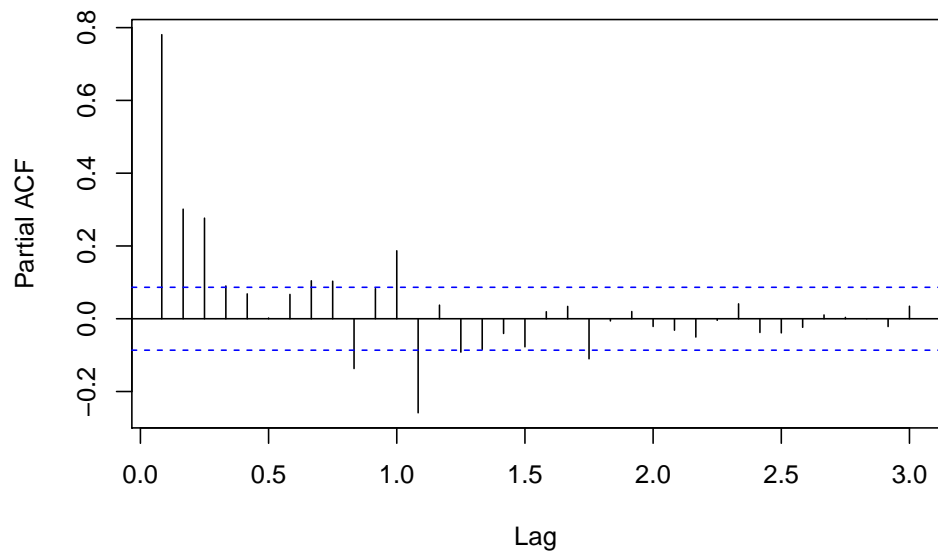


Figure 14: Trend + Season

The plot above accounts for both trend and seasonality. The residuals indicate the presence of a distinct pattern and cycles. The trend and seasonal components of the model are all significant. However, we saw from the residuals that they are not yet white noise. Additionally, we can see from the ACF and the PACF that the residuals are not white noise. This means that there is still some structure left that we can incorporate into our model to improve the fit. So we add a cyclical component to the model to account for this seasonality.

From ACF and PACF, PACF spike at 3 and ACF decaying to 0 indicating AR(3) process. ACF spike at 1, PACF oscillating and decaying to 0 so looks like MA(1) process. PACF has largest spike at 12 = 12x1 signaling seasonal AR(1) process

```
model_vehicle_ = Arima(vehicle.ts, order=c(3,0,1),
                        xreg = cbind(t, t2), seasonal=list(order=c(2,1,1)))
summary(model_vehicle_)

## Series: vehicle.ts
## Regression with ARIMA(3,0,1)(2,1,1)[12] errors
##
## Coefficients:
##          ar1          ar2          ar3          ma1          sar1          sar2          sma1          t
##          1.1009    -0.1969    0.0637   -0.6318    0.2542   -0.1443   -0.8199    -4.2578
## s.e.    0.1138     0.0811    0.0647    0.1066    0.0632    0.0559    0.0437   2052.2096
##          t2
##          0.0023
## s.e.    0.5132
##
## sigma^2 estimated as 9677:  log likelihood=-3017.16
## AIC=6054.32   AICc=6054.77   BIC=6096.51
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.900332 96.33899 68.26139 -0.5808559 5.644095 0.6282696
##              ACF1
## Training set 0.002259069
```

After careful consideration, the revised model was derived which includes all three components and the quadratic trend effect.

```
# plot of fit and residuals
#par(mfrow=c(4,1))
plot(vehicle.ts, main = "Total Vehicles Sold")
lines(fitted(model_vehicle_), col = "red")

plot(t,model_vehicle_$res, ylab="Residuals",type='l',xlab="Time",lwd=2,
main = "Residuals")
abline(h = 0, col = "red", lwd = 2)
```



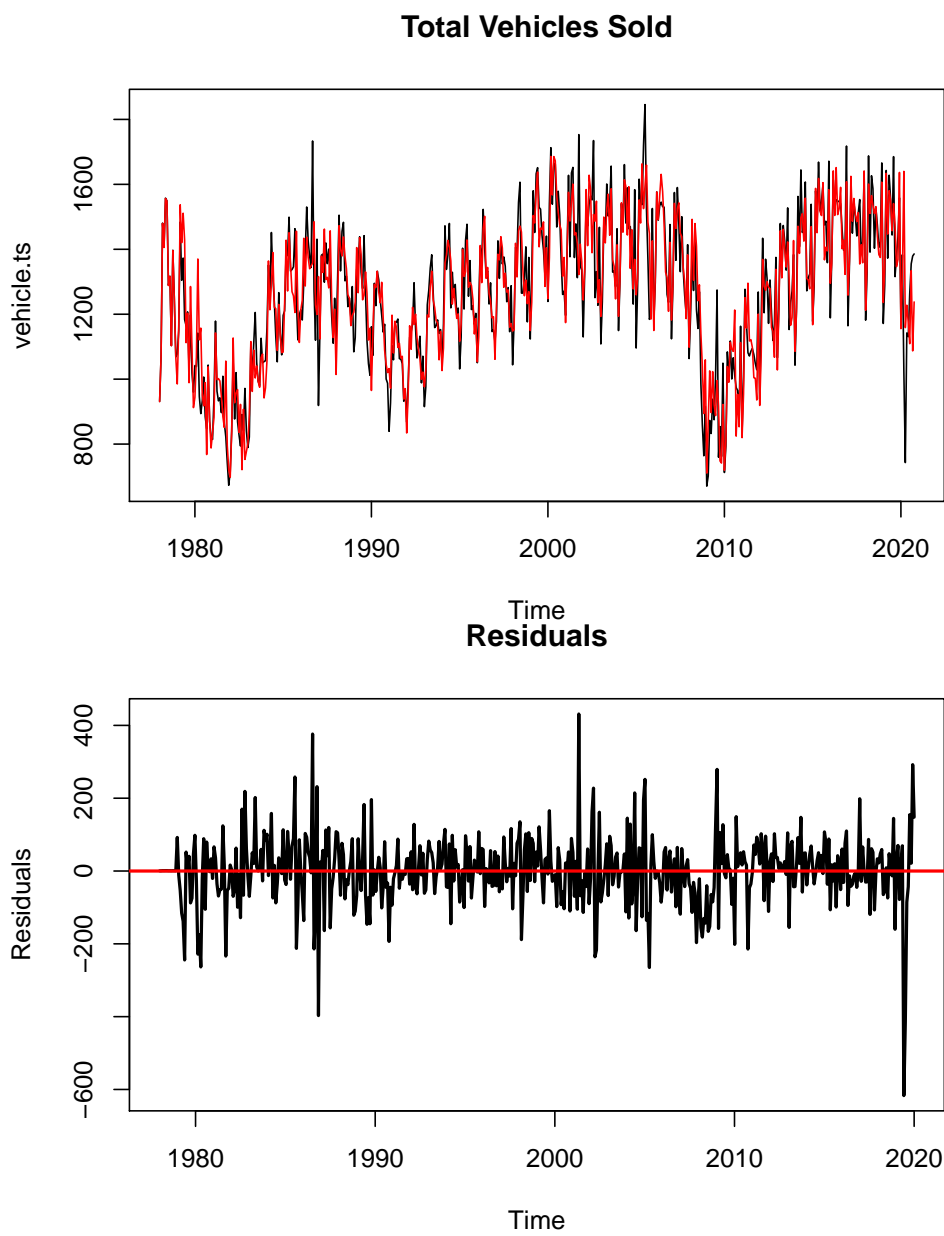


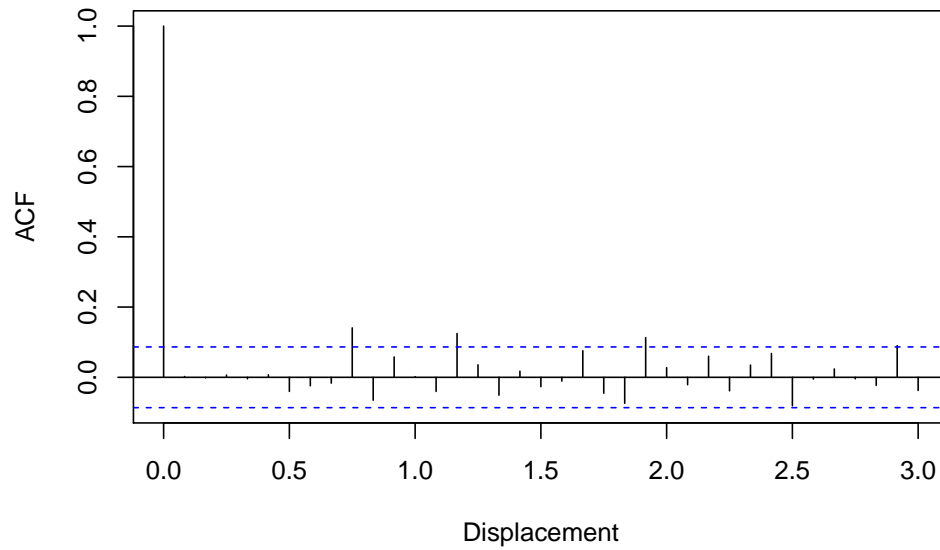
Figure 15: Trend + Season + Cycle

```
acf(model_vehicle$res,lag=36,main="Residual Sample Autocorrelations",xlab="Displacement")
pacf(model_vehicle$res,lag=36,main="Residual Sample Partial Autocorrelations", xlab="Displacement")

Box.test(model_vehicle$residuals, type="Box-Pierce")
```

```
##
## Box-Pierce test
##
## data: model_vehicle$residuals
## X-squared = 0.0026231, df = 1, p-value = 0.9592
```

### Residual Sample Autocorrelations



### Residual Sample Partial Autocorrelations

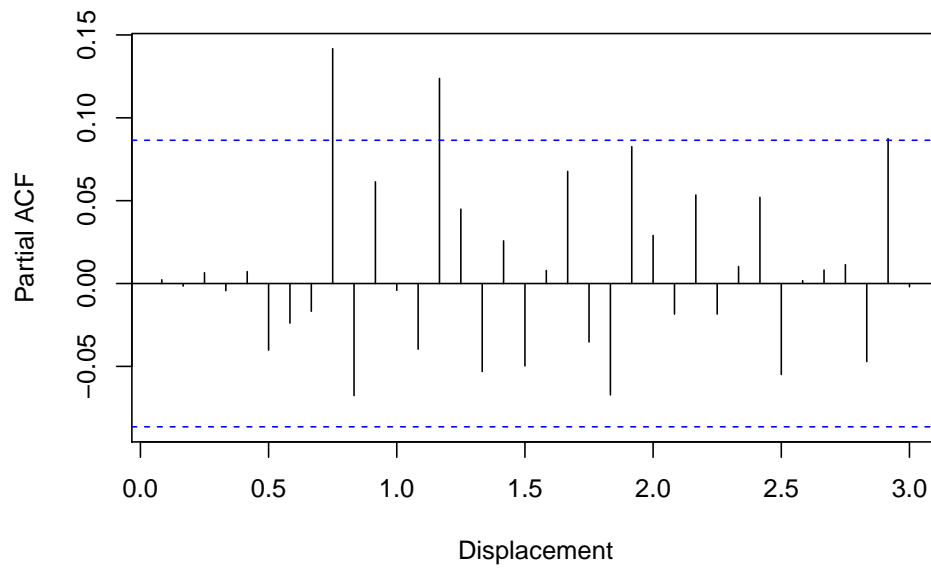


Figure 16: Trend + Season + Cycle

We will now perform a Box-Pierce test to determine if our residuals from the model we created are white noise or if there are still correlations we are not accounting for. The Box-Pierce Test has a null hypothesis that the data is white noise, compared to the alternative that they are not and correlations exist. With a p-value  $> 0.05$ , we fail to reject the null and conclude that the data are independent and White Noise exists. To visually support our conclusion, the ACF and PACF plots support that no structure are left in the residuals.

- Sentiment

Similar to the process that we performed for our total vehicle sales data, we will now repeat for our consumer sentiment data to observe the results.

```
#Linear trend model
m=lm(sentiment.ts~t)

#par(mfrow=c(4,1))
plot(sentiment.ts,ylab="Sentiment", xlab="Time", lwd=2, col='skyblue3',
main = "Linear Trend (Consumer Sentiment)")
lines(t,m$fit,col="red3",lwd=2)
plot(t,m$res, ylab="Residuals",type='l',xlab="Time", main = "Residuals")
abline(h = 0, col = "red", lwd = 2)
```

**Linear Trend (Consumer Sentiment)**

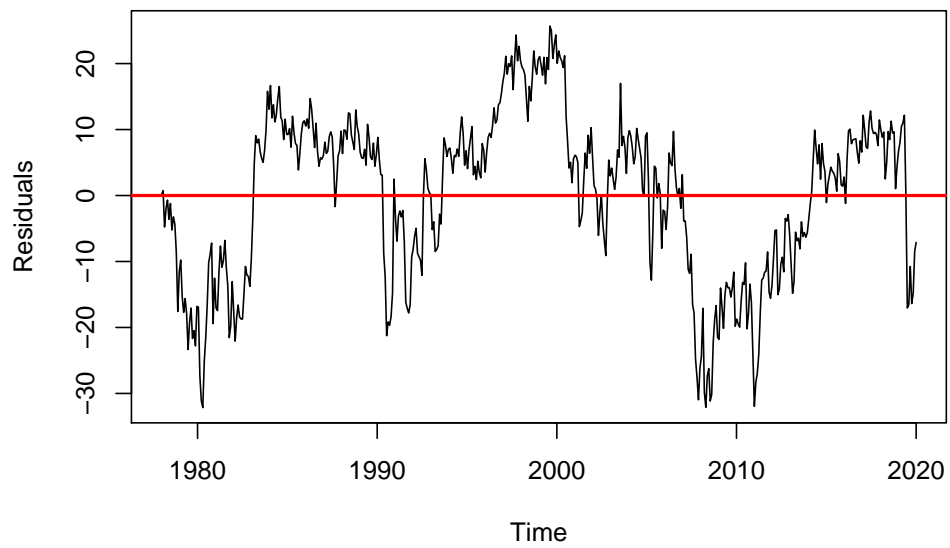
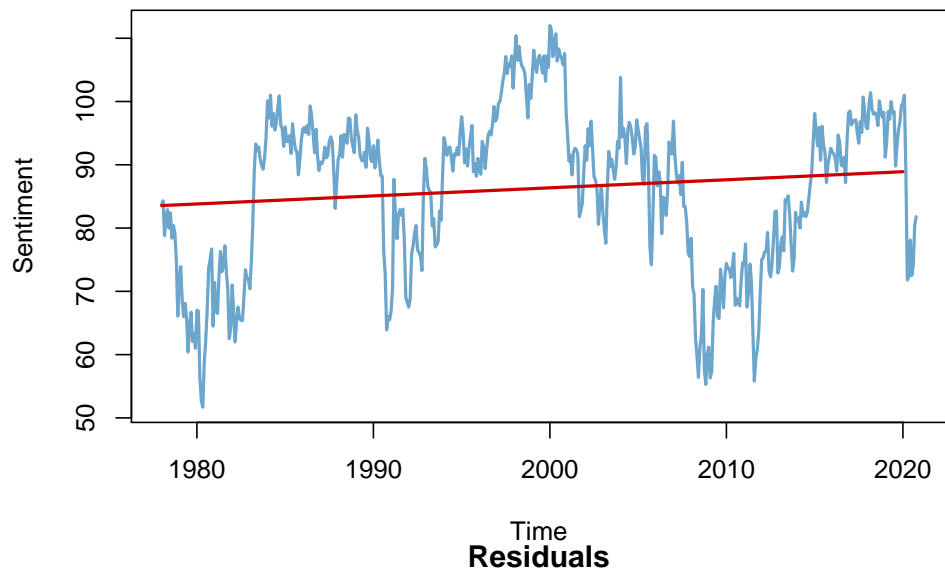


Figure 17: Linear

```
acf(m$res,lag=36,main="Residual Sample Autocorrelations")
pacf(m$res,lag=36,main="Partial Residual Sample Autocorrelations")
```

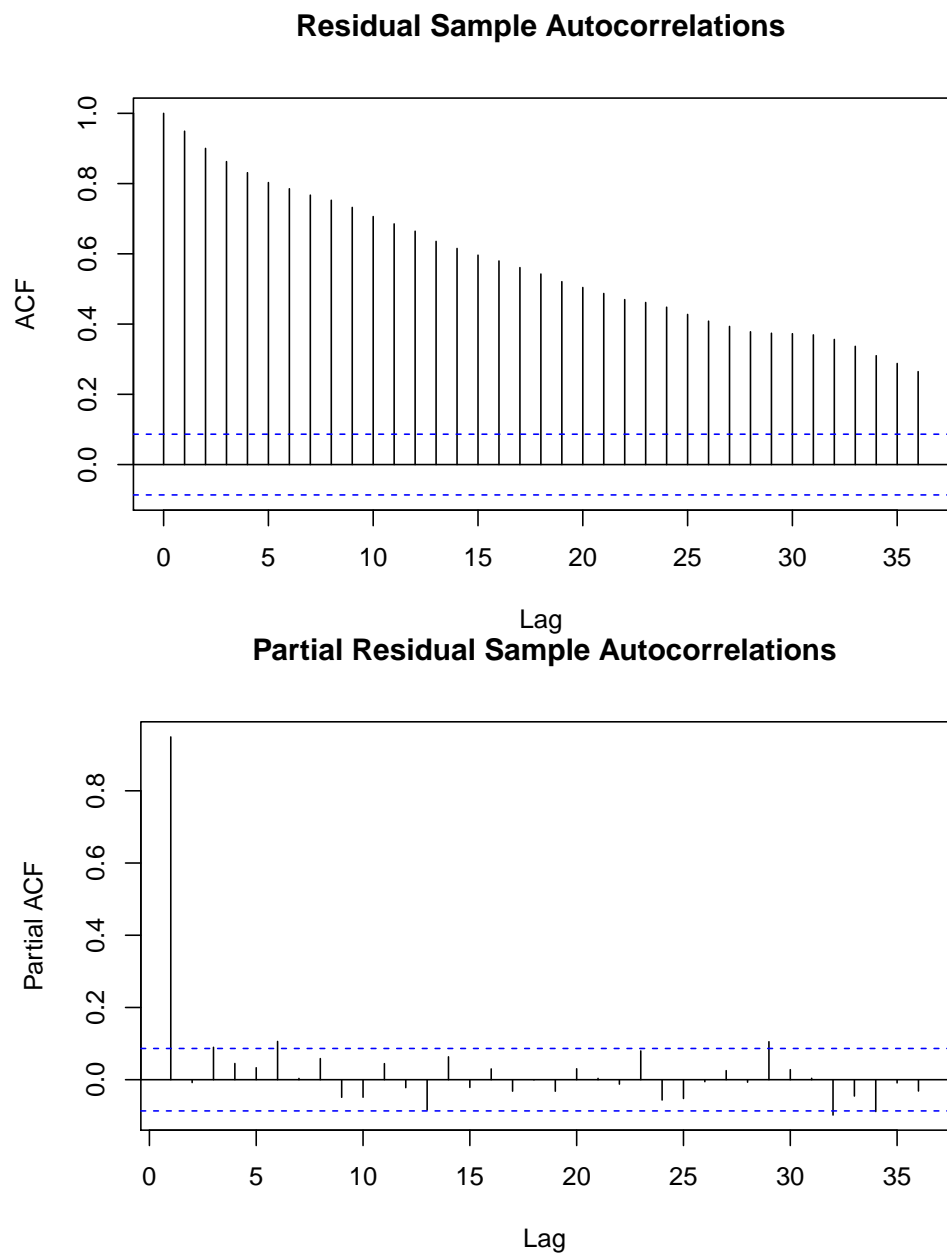


Figure 18: Linear

The resulting residuals from the linear trend fit do not appear to look like white noise. Since there are still dynamics left in acf and pacf residual graphs, we will attempt to fit a quadratic next to see if the fit improves.

```
#Quadratic Fit#  
mod.1=lm(sentiment.ts~t+t2)  
  
#par(mfrow=c(4,1))  
plot(sentiment.ts,ylab="Consumer Sentiment Level", xlab="Time", lwd=2, col='skyblue3',  
main = "Quadratic Fit (Consumer Sentiment)")  
lines(t,mod.1$fit,col="red3",lwd=2)  
plot(t,mod.1$res, ylab="Residuals",type='l',xlab="Time")
```

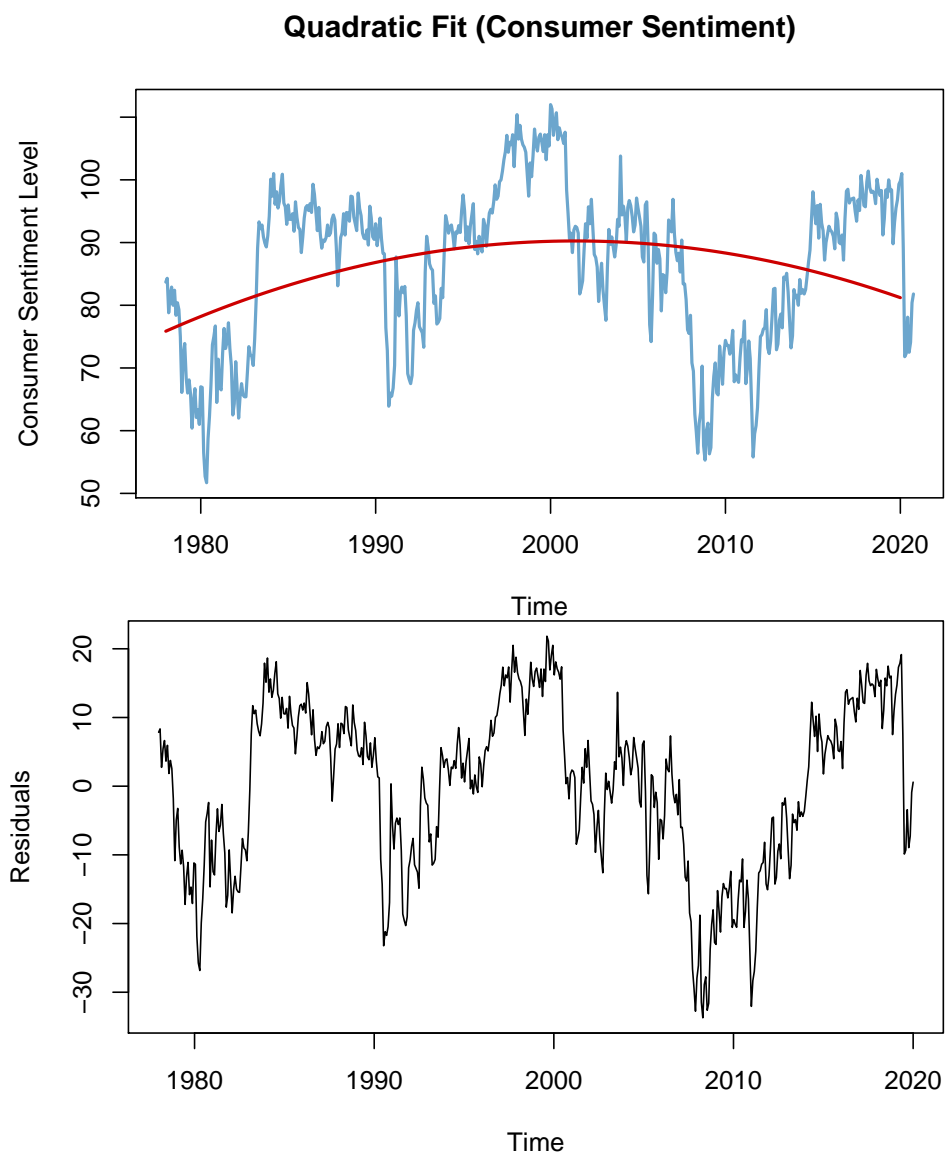


Figure 19: Quadratic

```
acf(mod.1$res,lag=36,main="Residual Sample Autocorrelations",xlab="Displacement")
pacf(mod.1$res,lag=36,main="Residual Sample Partial Autocorrelations", xlab="Displacement")
```

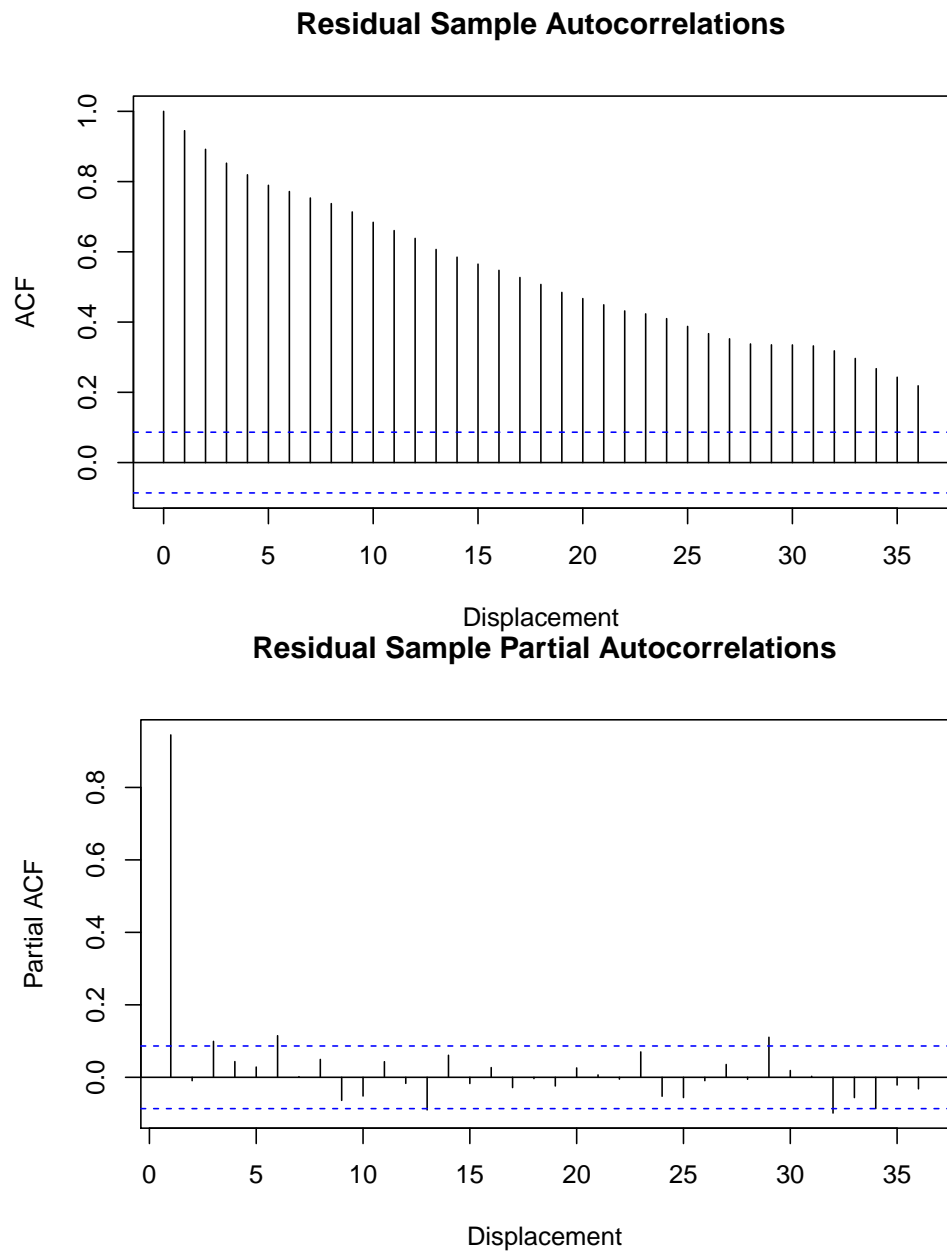


Figure 20: ACF - PACF Quadratic



After fitting a quadratic, the residuals show a pattern and so do the acf and pacf plots. Perhaps seasonality and cyclical components are still remaining in the residuals. We will attempt to correct for seasonality by adding a seasonal component to our model.

```
#Trend + Season
mod.2=tslm(sentiment.ts~ t + t2 + season)

#par(mfrow=c(4,1))
plot(sentiment.ts,ylab="Sentiments", xlab="Time", lwd=2, col='skyblue3',
, main = "Trend & Seasonality (Consumer Sentiment)")
lines(t,mod.2$fit,col="red3",lwd=2,lty=2)
plot(t,mod.2$res, ylab="Residuals",type='l',xlab="Time",lwd=2,
main = "Residuals")

acf(mod.2$res,lag=36,main="Residual Sample Autocorrelations",xlab="Displacement")
pacf(mod.2$res,lag=36,main="Residual Sample Partial Autocorrelations", xlab="Displacement")

plot(stl(mod.2$residuals, s.window = "periodic"), main="Decomposition of Consumer Sentiment")

summary(mod.2)
```

```
##
## Call:
## tslm(formula = sentiment.ts ~ t + t2 + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -33.687  -9.070   2.344   9.193  20.366
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.052e+05  1.623e+04  -6.482 2.18e-10 ***
## t            1.052e+02  1.624e+01   6.479 2.21e-10 ***
## t2          -2.628e-02  4.061e-03  -6.471 2.32e-10 ***
## season2      -8.933e-01  2.620e+00  -0.341  0.733
## season3     -1.277e+00  2.620e+00  -0.487  0.626
## season4     -1.549e+00  2.620e+00  -0.591  0.555
## season5     -1.115e+00  2.620e+00  -0.426  0.671
## season6     -7.305e-01  2.620e+00  -0.279  0.781
## season7     -1.415e+00  2.620e+00  -0.540  0.589
## season8     -2.362e+00  2.620e+00  -0.901  0.368
## season9     -2.190e+00  2.621e+00  -0.836  0.404
## season10    -3.164e+00  2.621e+00  -1.208  0.228
## season11    -2.830e+00  2.636e+00  -1.074  0.284
## season12    -2.229e+00  2.636e+00  -0.845  0.398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.15 on 500 degrees of freedom
## Multiple R-squared:  0.09552,    Adjusted R-squared:  0.072
## F-statistic: 4.062 on 13 and 500 DF,  p-value: 2.222e-06
```

### Trend & Seasonality (Consumer Sentiment

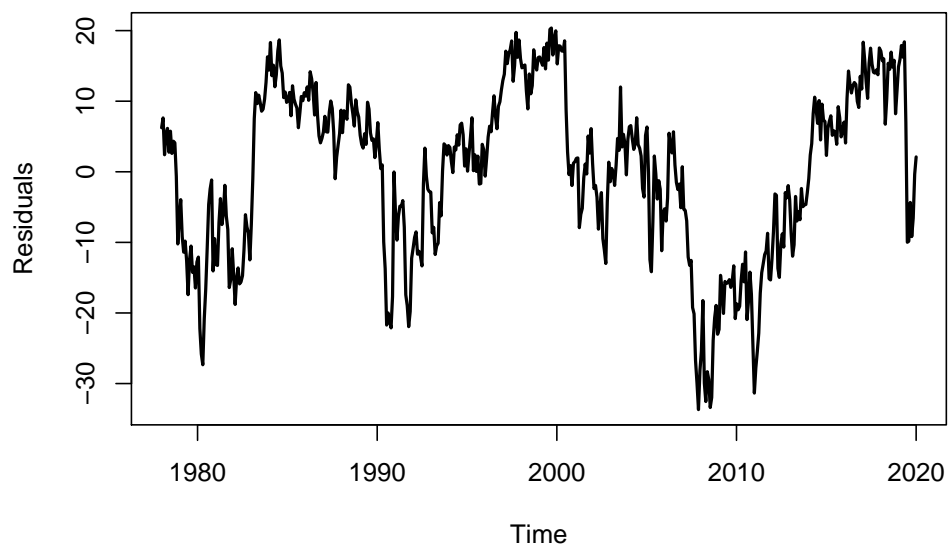
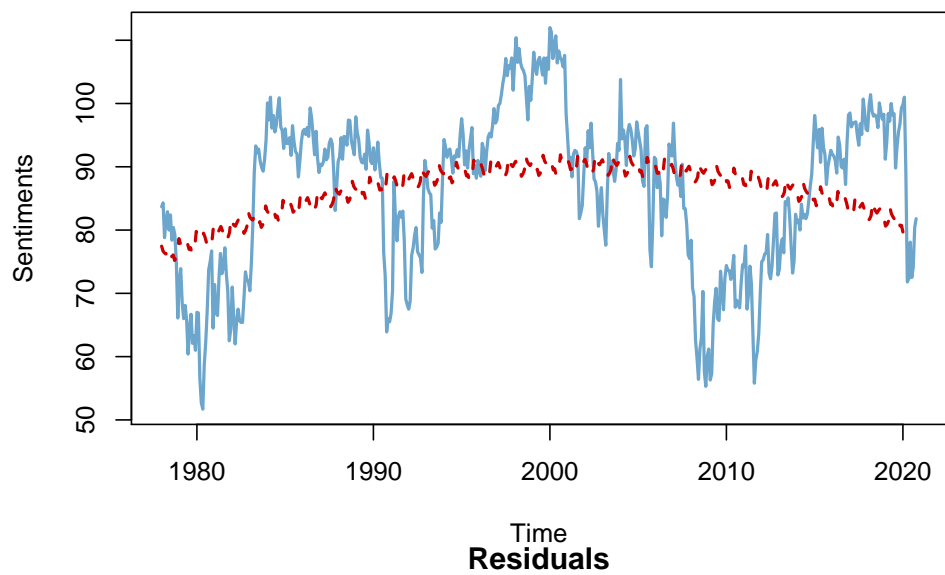
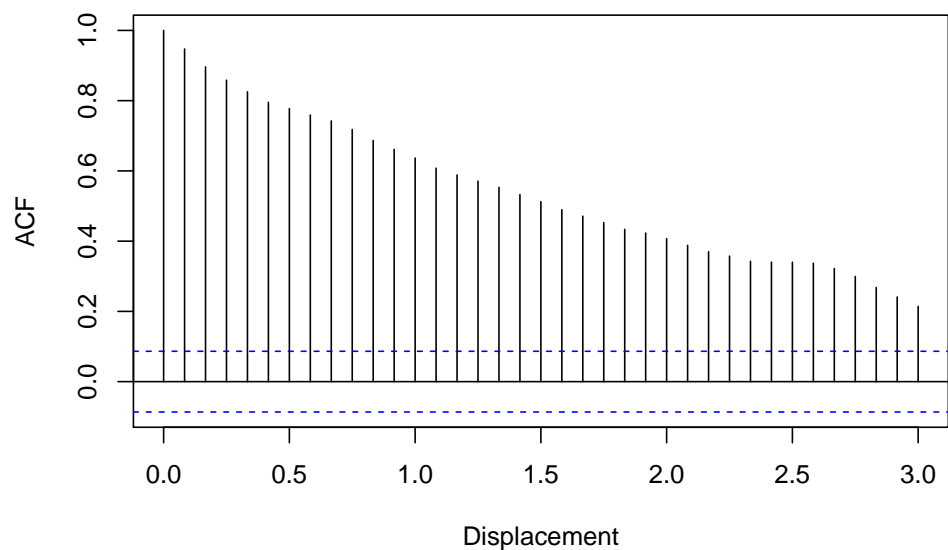


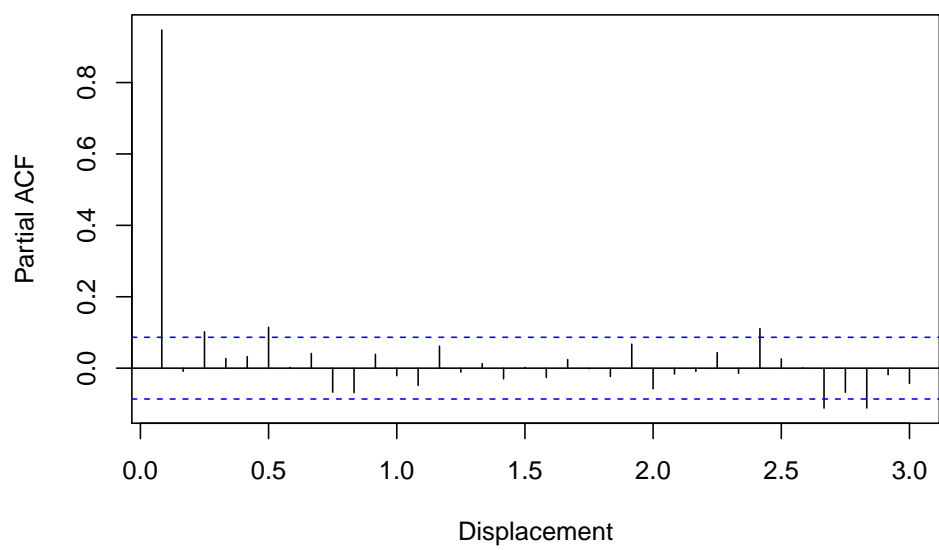
Figure 21: Trend + Season

*#all coefficients are not significant*

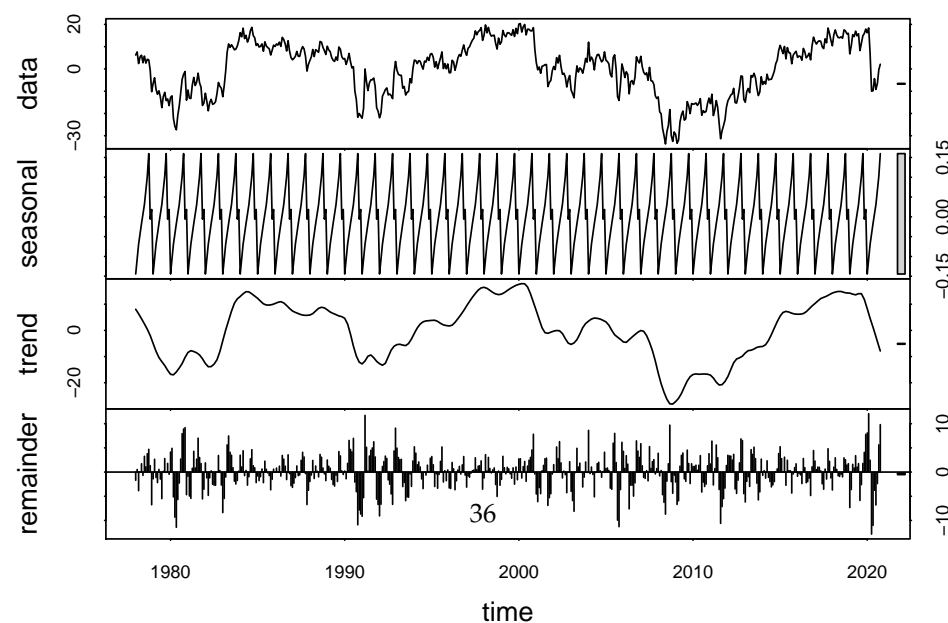
**Residual Sample Autocorrelations**



**Residual Sample Partial Autocorrelations**



**Decomposition of Consumer Sentiment**



The plot above accounts for both trend and seasonality now. The residuals indicate the presence of a distinct pattern and cycles. The trend component are statistically significant, but the seasonal components of the model are all insignificant. We believe that the cyclical component has a dominating effect and accounts for more of the variation in the data compared to the variation explained by a seasonality component. By looking at the remainder from the decomposition above, it supports our stance that a cyclical component is still remaining and stronger in magnitude compared to seasonality. Thus, will attempt to add a cyclical component to the model to account for this seasonality.

From ACF and PACF, PACF spike at 1 and ACF decaying to 0 indicating AR(1) process. ACF spike at 1, PACF oscillating and decaying to 0 so looks like MA(1) process. PACF has spikes at 12xi seasonal AR(1) process.

We want to bring your attention to the quadratic variable we fitted in our model before we continue on with the analysis. Originally we included the variable because we believed the quadratic term would improve the model, but it was discovered that when we plotted the CUSUM for the arima consumer sentiment model, a pronounced structural break was discovered. After evaluating the model further, we determined that the quadratic variable was the source of the problem. As a collective group, we opted to remove the variable. Below is the respective Recursive CUSUM plot.

```
# Discarded Model #
Discarded_Model=Arima(sentiment.ts,order=c(2,1,1),
                      xreg = cbind(t, t2),
                      seasonal=list(order=c(1,0,1)))

# Recursive Residuals plot For Consumer Sentiment Model #
library(strucchange)
y=recresid(Discarded_Model$res~1)
plot(y, pch=16,ylab="Recursive Residuals")
plot(efp(Discarded_Model$residuals~1, type = "Rec-CUSUM"))
```

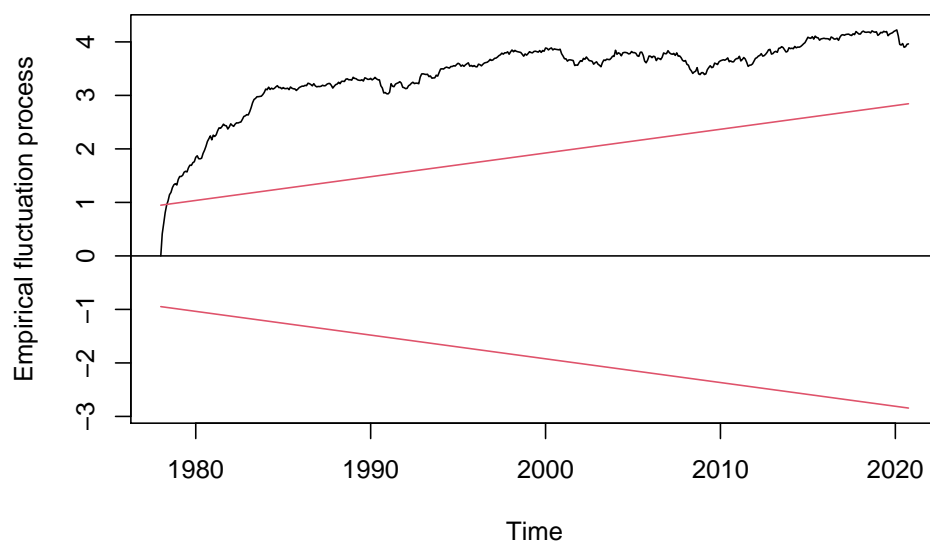
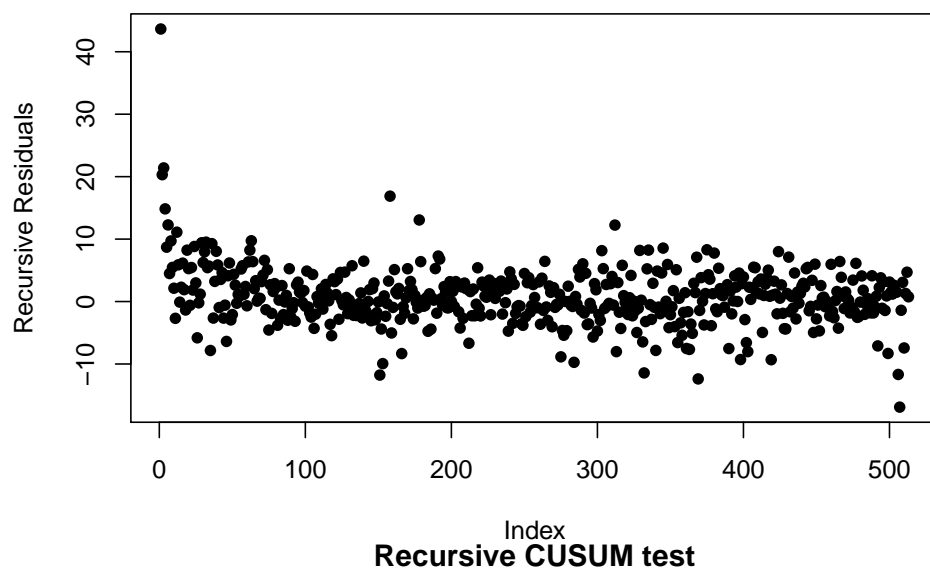


Figure 23: Discarded Model

```

# plot of fit and residuals
model_sentiment_=Arima(sentiment.ts,order=c(2,1,1),
                        xreg = cbind(t),
                        seasonal=list(order=c(1,0,1)))

summary(model_sentiment_)

## Series: sentiment.ts
## Regression with ARIMA(2,1,1)(1,0,1)[12] errors
##
## Coefficients:
##          ar1      ar2      ma1      sar1      sma1          t
##          0.7180 -0.0988 -0.7561  0.6534 -0.5750 -0.1658
## s.e.    0.1361   0.0508   0.1305  0.4065   0.4394   1.6391
##
## sigma^2 estimated as 15.38:  log likelihood=-1426.05
## AIC=2866.11   AICc=2866.33   BIC=2895.79
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.004300743 3.894863 2.962178 -0.1468642 3.619064 0.390569
##              ACF1
## Training set -0.0001031855

#par(mfrow=c(4,1))
plot(sentiment.ts, main ="ARIMA Consumer Sentiment Model")
lines(fitted(model_sentiment_), col = "red")
plot(t,model_sentiment_$res, ylab="Residuals",type='l',xlab="Time",lwd=2,
main = "Residuals")

```

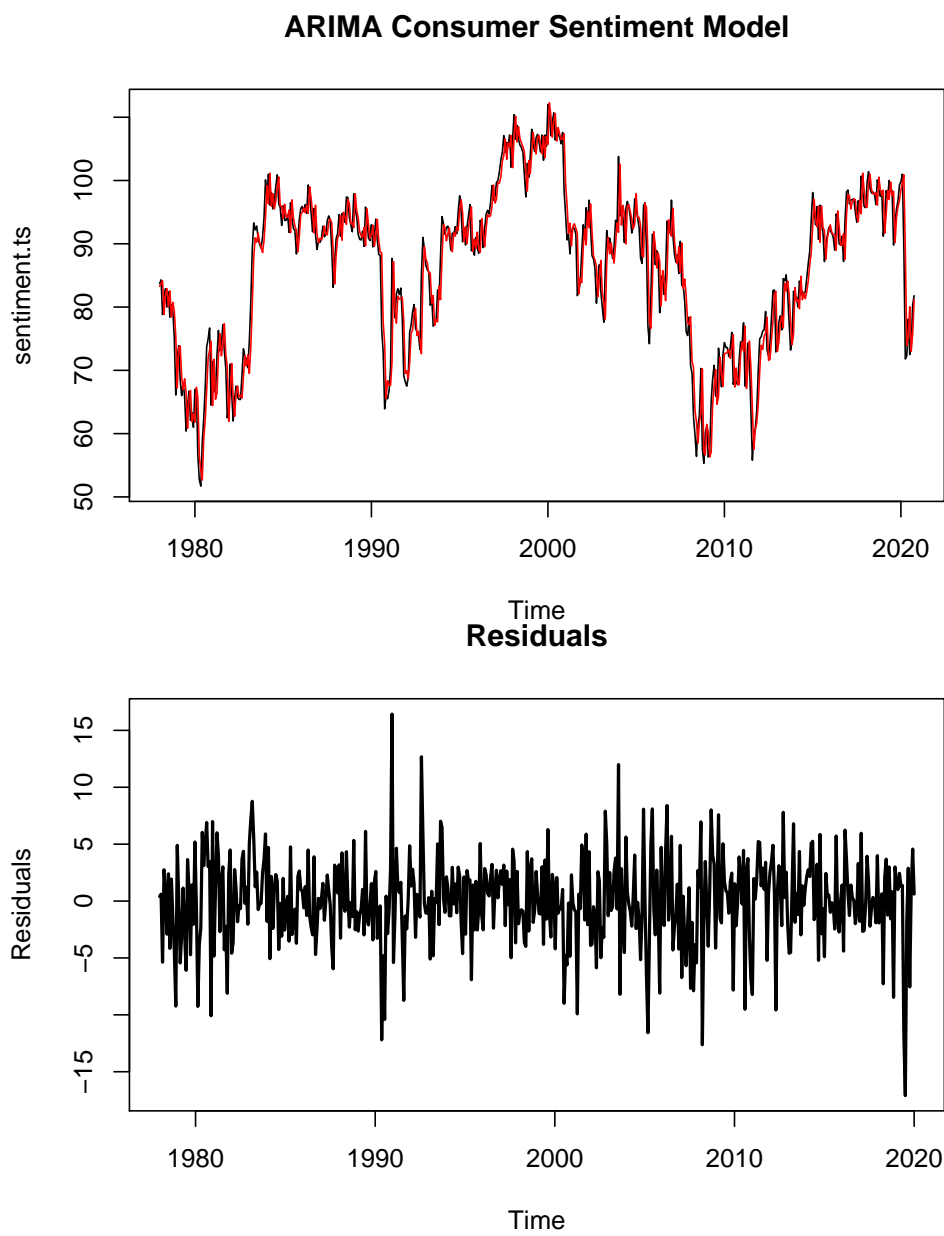


Figure 24: Trend + Season + Cycle

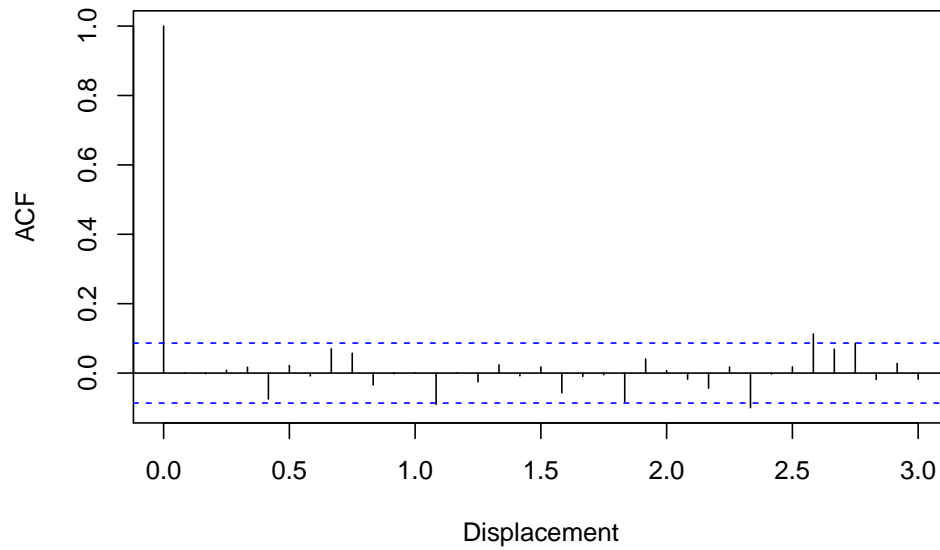


```
acf(model_sentiment_$res,lag=36,main="Residual Sample Autocorrelations",xlab="Displacement")
pacf(model_sentiment_$res,lag=36,main="Residual Sample Partial Autocorrelations", xlab="Displacement")
Box.test(model_sentiment_$residuals)
```

```
##
## Box-Pierce test
##
## data: model_sentiment_$residuals
## X-squared = 5.4727e-06, df = 1, p-value = 0.9981
```

```
#plot(forecast(model_sentiment_, h = 12))
```

### Residual Sample Autocorrelations



### Residual Sample Partial Autocorrelations

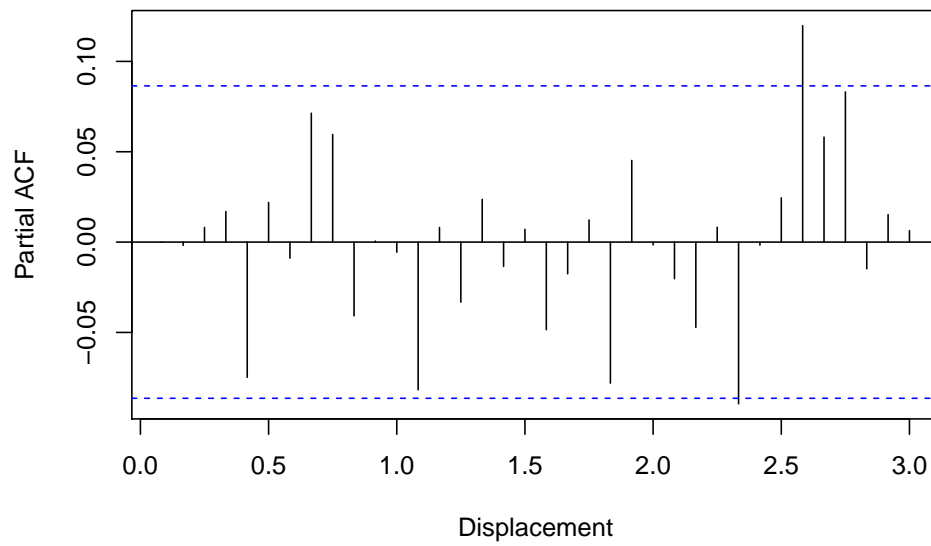


Figure 25: Trend + Season + Cycle

We will now perform a Box-Pierce test, which has as a null hypothesis that the residuals are white noise and an alternative hypothesis is that the residuals are correlated. With a p-value > 0.05, we fail to reject the null hypothesis and conclude that the residuals are independent and there is evidence of White Noise. The ACF and PACF plots support that no structure is left in the residuals.

#### 4. Plotting residuals vs. fitted values

```
#par(mfrow=c(3,1))
#Respective residuals vs. fitted values
plot(model_vehicle$fitted, model_vehicle$residuals, xlab = "Fitted Values", ylab = "Residuals", main = "Total Vehicles Sold")
abline(h = 0, col = "red", lwd = 2)
```

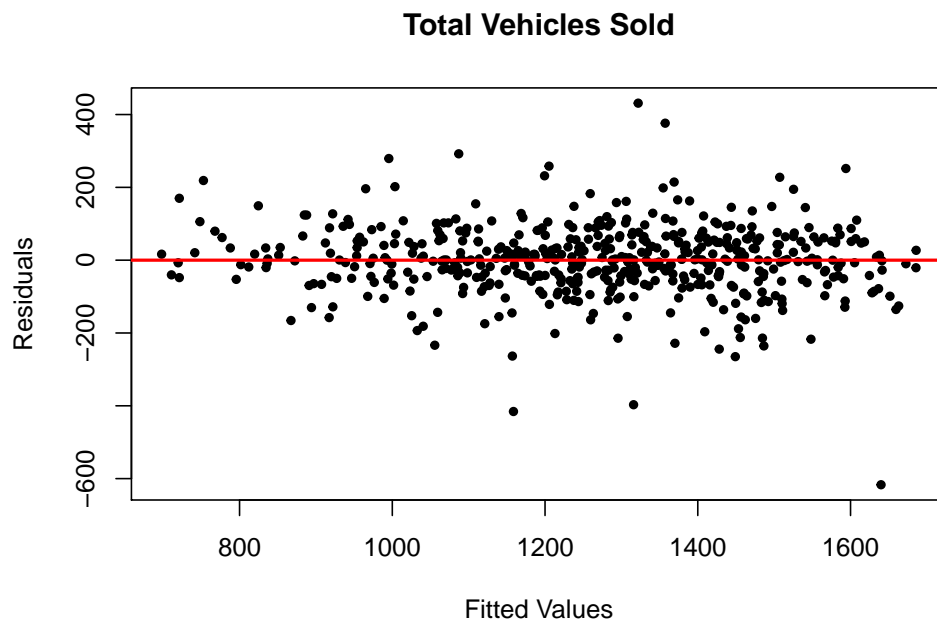


Figure 26: Vehicles Sold Respective residuals vs. fitted values

```
#Respective residuals in time
plot(model_vehicle$residuals, xlab = "Time",
     ylab = "Residuals", main = " Total Vehicles Sold", pch=20)
abline(h = 0, col = "red", lwd = 2)

#Histogram of the residuals
hist(model_vehicle$residuals, xlab = "Residual of Vehicles Sold",
     main="Histogram of Residuals")
```

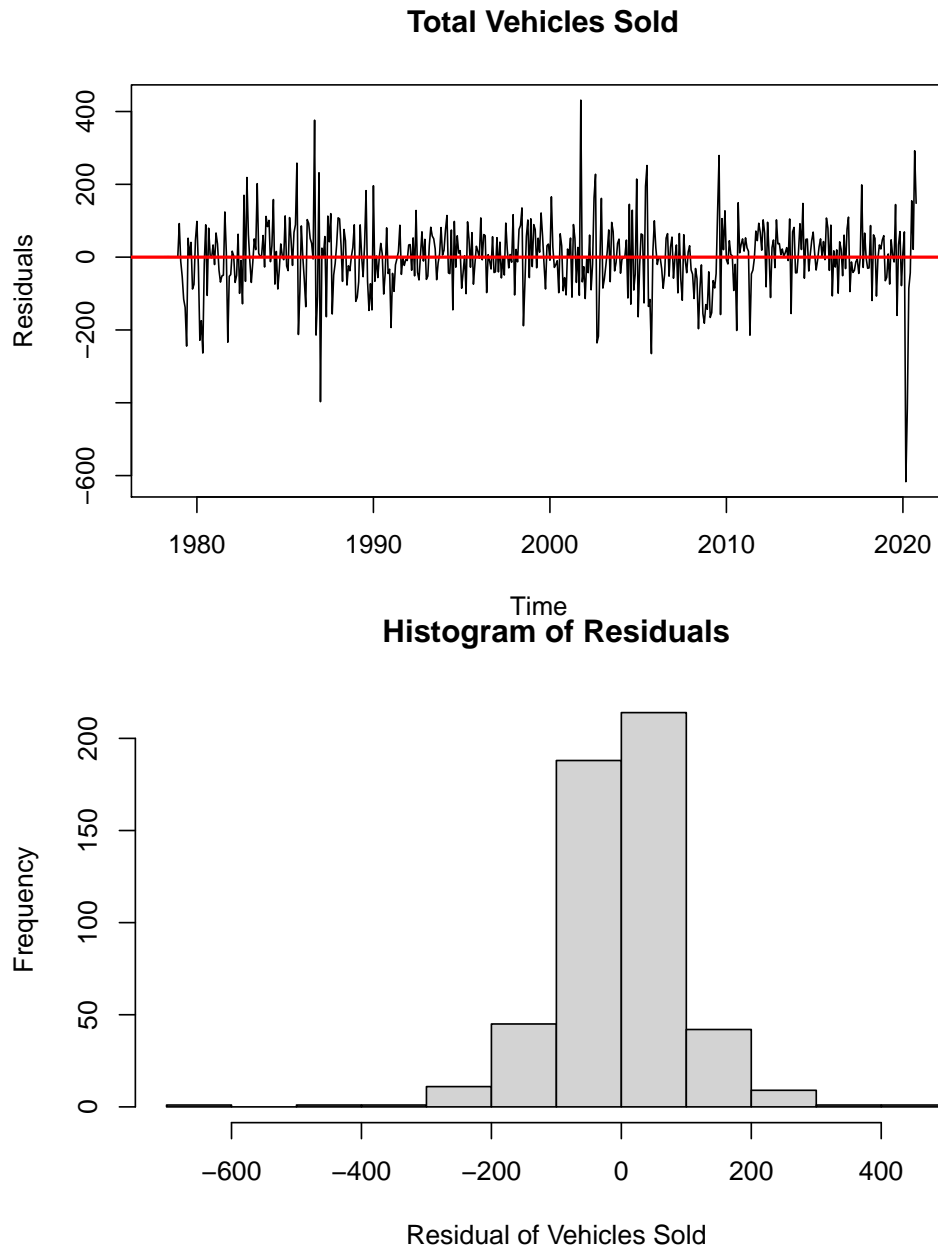


Figure 27: Vehicles Sold Respective residuals vs. fitted values

There isn't a noticeable pattern in the model that we can observe. The residuals against the fitted values appear to be randomly distributed about zero with a constant variance. The histogram of the residuals are showing almost a normal distribution with a slight cluster on the left. To test if the residuals are in fact normally distributed, we will perform a Jarque Bera Test to statistically support if the residuals are in fact normally distributed. The Jarque Bera Test has a null hypothesis that the residuals are normally distributed and the alternative is that the residuals are not normally distributed.

```
jarque.bera.test(model_vehicle$residuals)
```

```
##  
## Jarque Bera Test  
##  
## data: model_vehicle$residuals  
## X-squared = 585.07, df = 2, p-value < 2.2e-16
```

With a P-Value < 0.05, we reject the null hypothesis and conclude that the residuals are not normally distributed, which is not required, but it is a desired result.

```
#Respective residuals vs. fitted values  
plot(model_sentiment$fitted, model_sentiment$residuals, xlab = "Fitted Values", ylab = "Residuals", m  
abline(h = 0, col = "red", lwd = 2)  
  
#Respective residuals in time  
plot(model_sentiment$residuals, xlab = "Time",  
      ylab = "Residuals", main = " Consumer Sentiment", pch=20)  
abline(h = 0, col = "red", lwd = 2)  
  
#Histogram of the residuals  
hist(model_sentiment$resid, xlab = "Residual of Consumer Sentiment",  
      main="Histogram of Residuals")
```

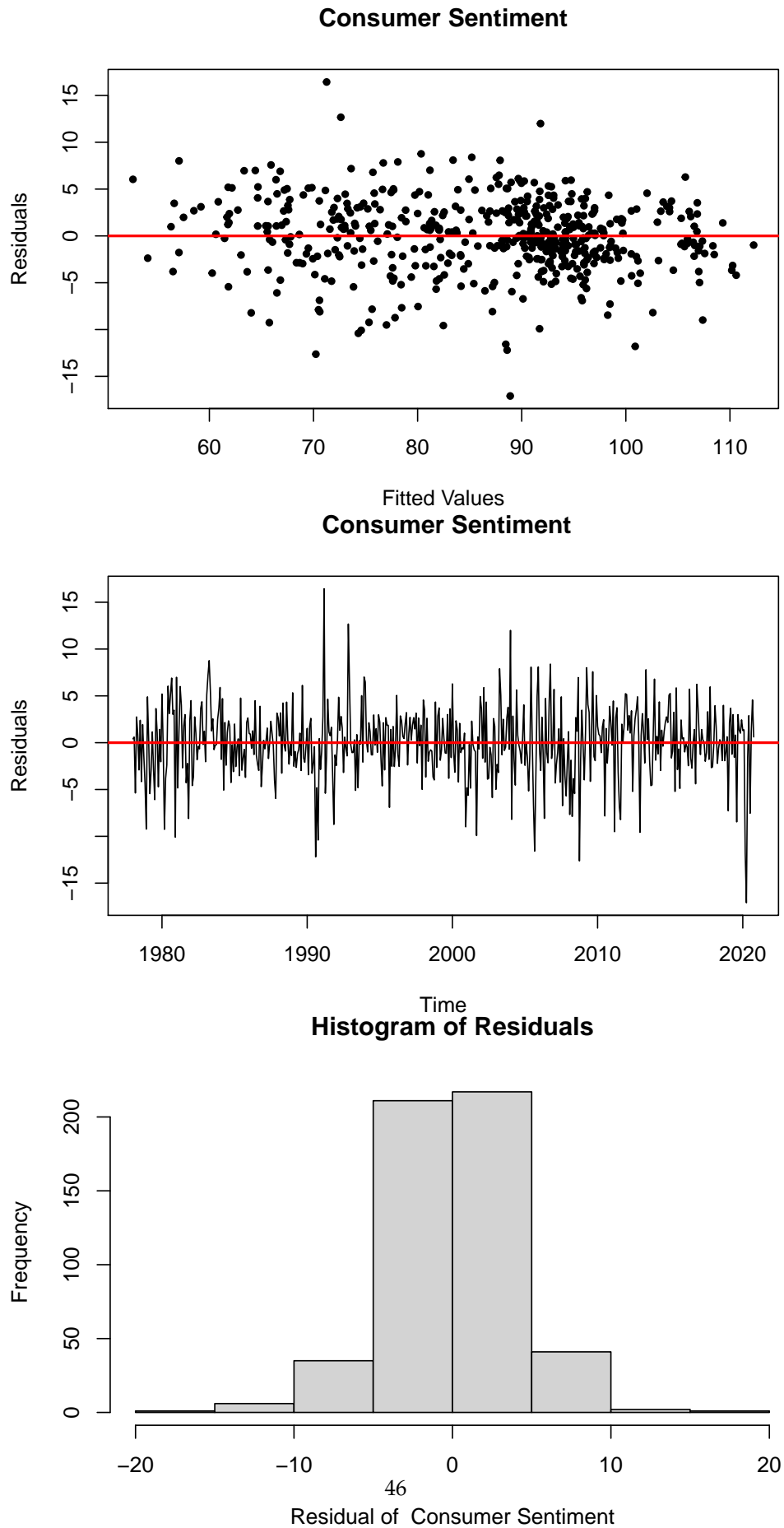


Figure 28: Sentiments Respective residuals vs. fitted values

There isn't a noticeable pattern in the model that we can observe. The residuals against the fitted values appear to be randomly distributed about zero with a constant variance. The residuals are distributed around zero (mean of residuals close to zero) without a pattern indicating that no heteroskedasticity exists.

The histogram of the residuals are showing almost a normal distribution.

Again, to test if the residuals are in fact normally distributed, we will perform a Jarque Bera Test to statistically support if the residuals are in fact normally distributed. The Jarque Bera Test has a null hypothesis that the residuals are normally distributed and the alternative is that the residuals are not normally distributed.

```
jarque.bera.test(model_sentiment_$residuals)

##
##   Jarque Bera Test
##
## data:  model_sentiment_$residuals
## X-squared = 58.412, df = 2, p-value = 2.071e-13
```

With a P-Value  $< 0.05$ , we reject the null hypothesis and conclude that the residuals are not normally distributed, which is not required, but it is a desired result.

## 5. Plotting the ACF and PACF of the respective residuals

```
par(mfrow=c(2,1))
acf(model_vehicle_$residuals, type = "correlation", main="Autocorrelation",
lag.max=60, ylab= "Vehicle residuals ACF")
pacf(model_vehicle_$residuals,main="Partial Autocorrelation",lag.max=60, ylab="Vehicle residuals PACF")

# null hypothesis is that residuals are white noise
Box.test(model_vehicle_$residuals, type="Box-Pierce")

##
##   Box-Pierce test
##
## data:  model_vehicle_$residuals
## X-squared = 0.0026231, df = 1, p-value = 0.9592
```

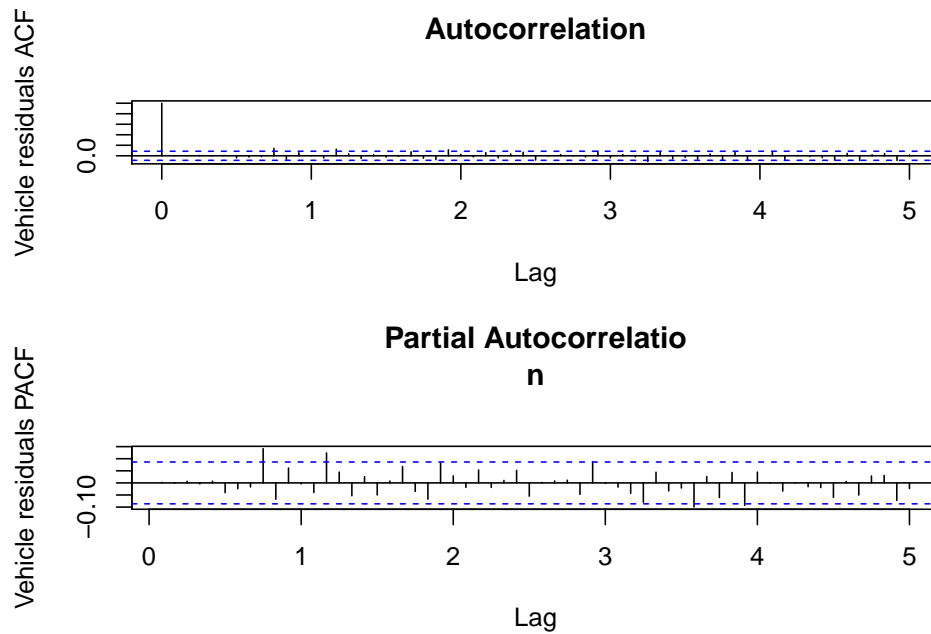


Figure 29: ACF and PACF of the respective residuals - Vehicles

When we look at our ACF and PACF of the residuals, we see some values are inside the Barlett Window and some values close to the 95% significance bands. In order to test if our residuals are white noise, we also conducted a Box-Pierce Test. The Box-Pierce Test states that the p-value is  $> 0.05$  indicating that we fail to reject the null hypothesis and conclude that our residuals are white noise. Thus, there is no dynamic left that our model.

```
par(mfrow=c(2,1))
acf(model_sentiment_$residuals, type = "correlation", main="Autocorrelation",
lag.max=60, ylab= "Vehicle residuals ACF")
pacf(model_sentiment_$residuals, main="Partial Autocorrelation",
lag.max=60, ylab="Vehicle residuals PACF")

# null hypothesis is that residuals are white noise
Box.test(model_sentiment_$residuals, type="Box-Pierce")
```

```
##
## Box-Pierce test
##
## data: model_sentiment_$residuals
## X-squared = 5.4727e-06, df = 1, p-value = 0.9981
```



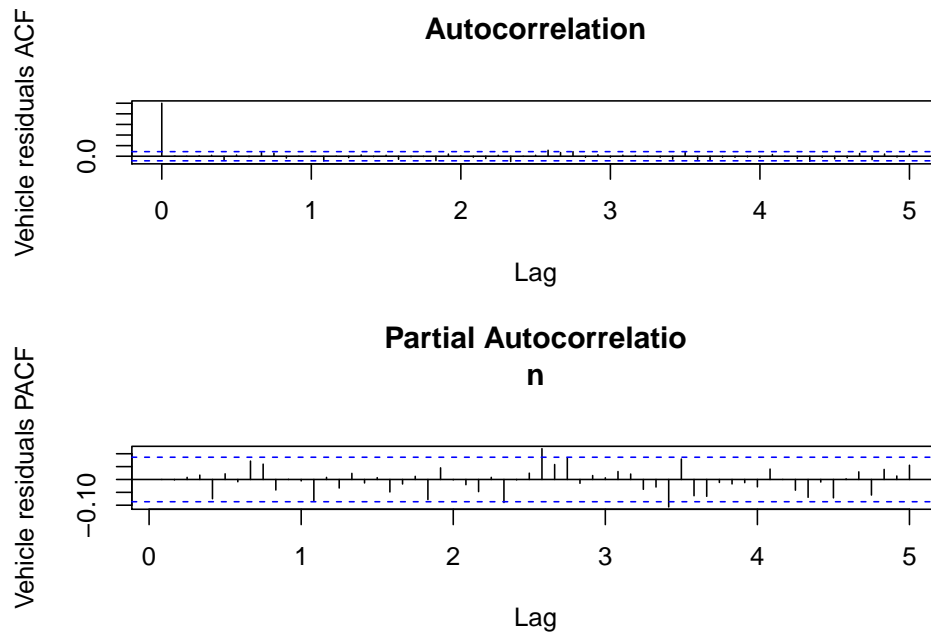


Figure 30: ACF and PACF of the respective residuals - Sentiment

When we look at our ACF and PACF of the residuals, we see some values inside the Barlett Window for the sentiment data as well and some values are close to the 95% significance bands. In order to test if our residuals are white noise, we also conducted a Box-Pierce Test. The Box-Pierce Test states that the p-value is  $> 0.05$  indicating that we fail to reject the null hypothesis and conclude that our residuals are white noise. Thus, there is no dynamic left that our model.

## 6. Plot the respective CUSUM and interpret the plot.

```
#Respective CUSUM plot
library(strucchange)
y=recresid(model_vehicle$res~1)
plot(y, pch=16,ylab="Recursive Residuals")
plot(efp(model_vehicle$residuals~1, type = "Rec-CUSUM"))

#Structural change test
sctest(model_vehicle$residuals ~ 1)

##
## Recursive CUSUM test
##
## data: model_vehicle$residuals ~ 1
## S = 0.53725, p-value = 0.543
```

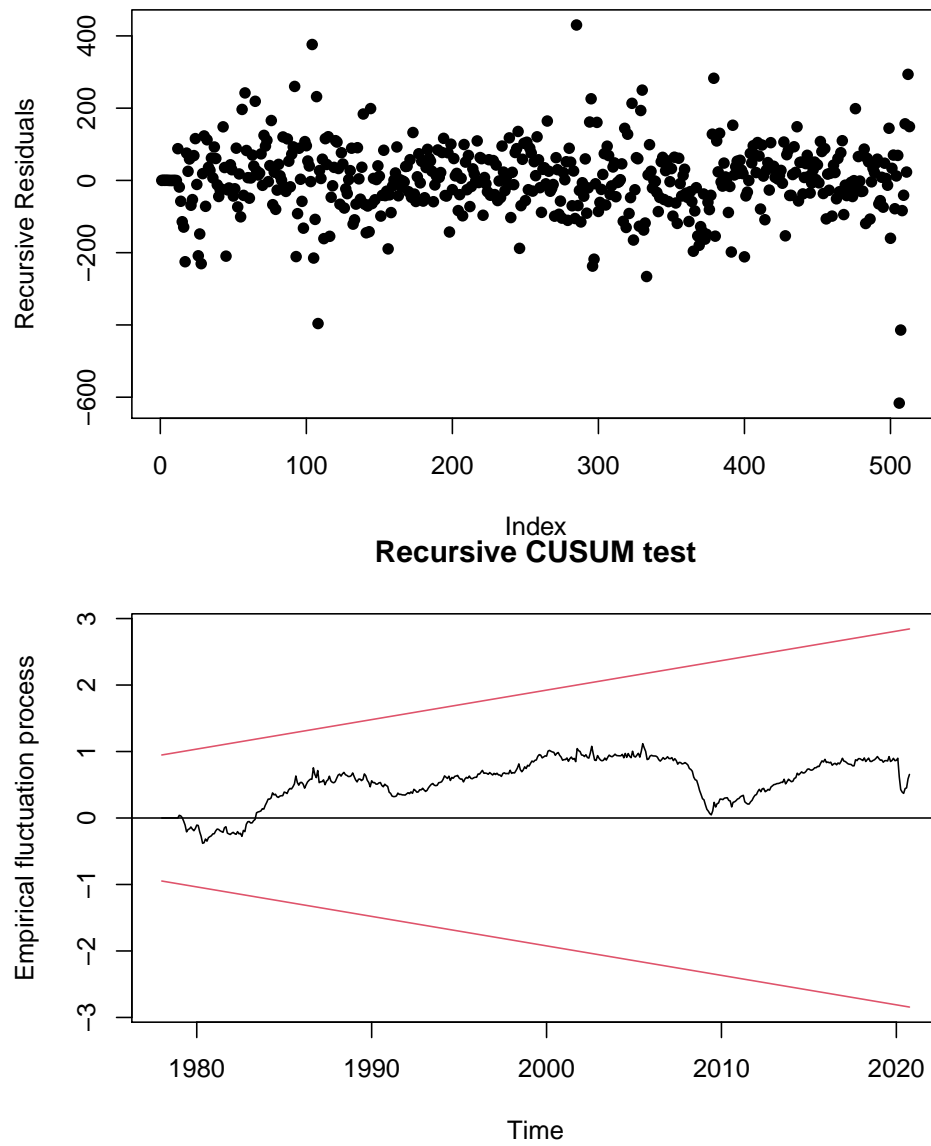


Figure 31: Respective CUSUM for Vehicles

Cumulative Sum is used for parameter stability. As we can see from the Recursive CUSUM plot since they are within the confidence band there are no structural breaks in the data. (Although the residuals does not equal to 0 all the time) We further conduct structural test to test our observation. From the structural change test, since we get p-value > 0.05 we fail to reject the null hypothesis and conclude that there are no structural breaks.

```
#Respective CUSUM plot
library(strucchange)
y=recresid(model_sentiment_$res~1)
plot(y, pch=16,ylab="Recursive Residuals")
plot(efp(model_sentiment_$residuals~1, type = "Rec-CUSUM"))

#Structural change test
sctest(model_sentiment_$residuals ~ 1)
```

```
##
## Recursive CUSUM test
##
## data: model_sentiment_$residuals ~ 1
## S = 0.58563, p-value = 0.4415
```

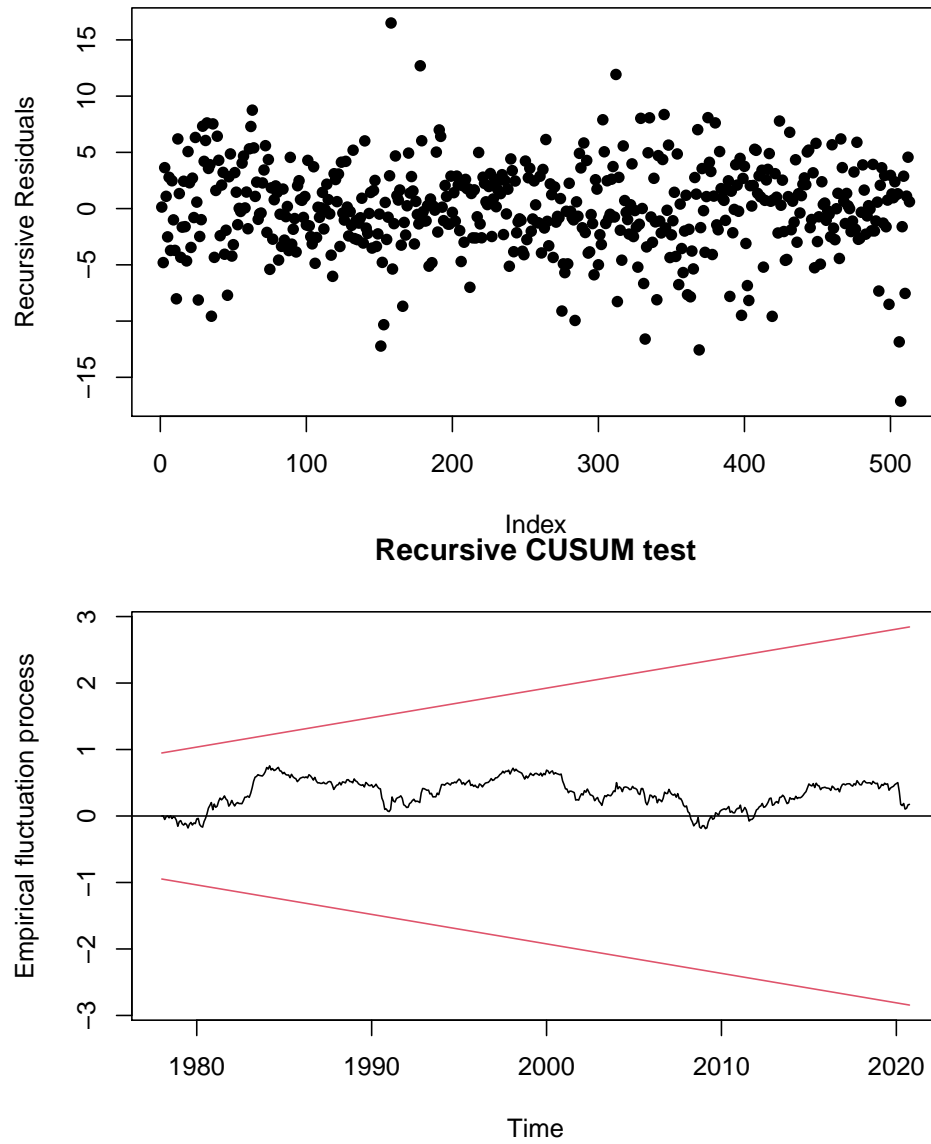


Figure 32: Respective CUSUM for Sentiment

As we can see from the Recursive CUSUM plot of the Sentiment Model residuals, since they are within the confidence band there are no structural breaks in the data, (Although the residuals does not equal to 0 all the time) We further conduct structural test to test our observation. From the structural change test, since we get  $p\text{-value} > 0.05$  we fail to reject the null hypothesis and conclude that there no structural breaks.

## 7. Plot the respective Recursive Residuals and interpret the plot.

```
#recursive residuals
y = recresid(model_vehicle$residuals~1)
plot(y, pch=16, ylab="Recursive Residuals", main="Vehicle Residuals")
abline(h = 0, col = "red", lwd = 2)
```

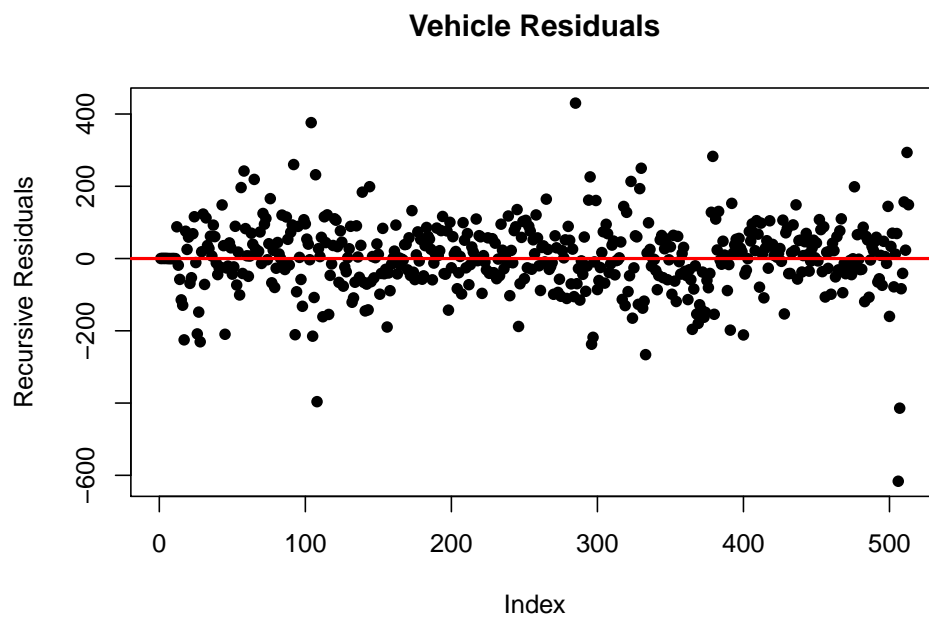


Figure 33: Recursive Residuals for Vehicle

The recursive estimates provide information about the stability of the residuals. They show us if any structural breaks exists. Looking at our Recursive Residuals plot for Vehicle model, we can say that the residuals are pretty stable around zero with a constant variance. We should also note that there are might be a few outliers, but shouldn't cause to much problems when forecasting. We elect to say that there could be due to the distance some of the observations are away from the mean.

```
#recursive residuals
y = recresid(model_sentiment_$residuals~1)
plot(y, pch=16, ylab="Recursive Residuals", main = "Sentiment Residuals")
abline(h = 0, col = "red", lwd = 2)
```

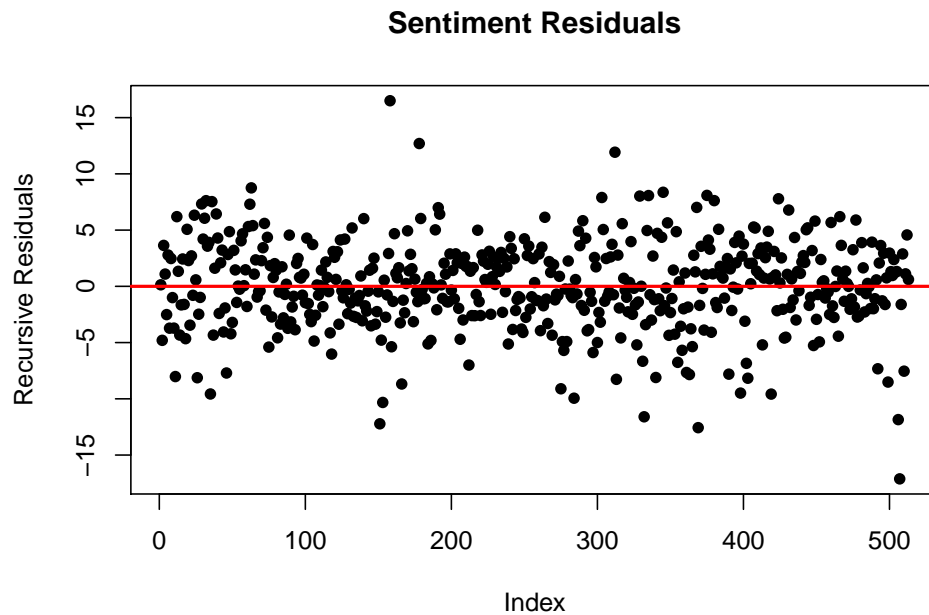
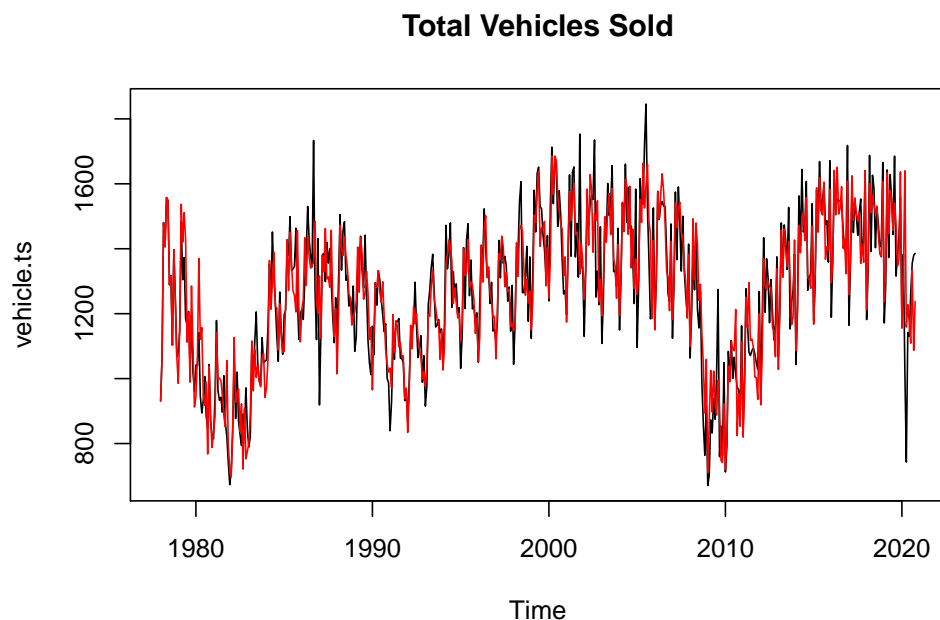


Figure 34: Recursive Residuals for Sentiment

Looking at our Recursive Residuals plot for Sentiment model, we can say that the residuals are pretty stable and are around the value zero with a constant variance. Similar to the Vehicle model residuals plot

## 8. Diagnostic statistics

```
#fitting model on the data
plot(vehicle.ts, main = "Total Vehicles Sold")
lines(fitted(model_vehicle_), col = "red")
```



```
#associated diagnostic statistics
summary(model_vehicle_)
```

```
## Series: vehicle.ts
## Regression with ARIMA(3,0,1)(2,1,1)[12] errors
##
## Coefficients:
##      ar1      ar2      ar3      ma1      sar1      sar2      sma1          t
##      1.1009 -0.1969  0.0637 -0.6318  0.2542 -0.1443 -0.8199   -4.2578
## s.e.  0.1138  0.0811  0.0647  0.1066  0.0632  0.0559  0.0437  2052.2096
##      t2
##      0.0023
## s.e.  0.5132
##
## sigma^2 estimated as 9677:  log likelihood=-3017.16
## AIC=6054.32   AICc=6054.77   BIC=6096.51
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set -0.900332 96.33899 68.26139 -0.5808559 5.644095 0.6282696
##              ACF1
## Training set 0.002259069
```

```
coeftest(model_vehicle_)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  1.1009e+00 1.1379e-01  9.6748 < 2.2e-16 ***
## ar2 -1.9689e-01 8.1052e-02 -2.4292 0.015130 *
## ar3  6.3663e-02 6.4683e-02  0.9842 0.324998
## ma1 -6.3178e-01 1.0663e-01 -5.9249 3.126e-09 ***
## sar1 2.5421e-01 6.3189e-02  4.0230 5.745e-05 ***
## sar2 -1.4427e-01 5.5863e-02 -2.5826 0.009805 **
## sma1 -8.1987e-01 4.3674e-02 -18.7726 < 2.2e-16 ***
## t    -4.2578e+00 2.0522e+03 -0.0021 0.998345
## t2    2.2737e-03 5.1323e-01  0.0044 0.996465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

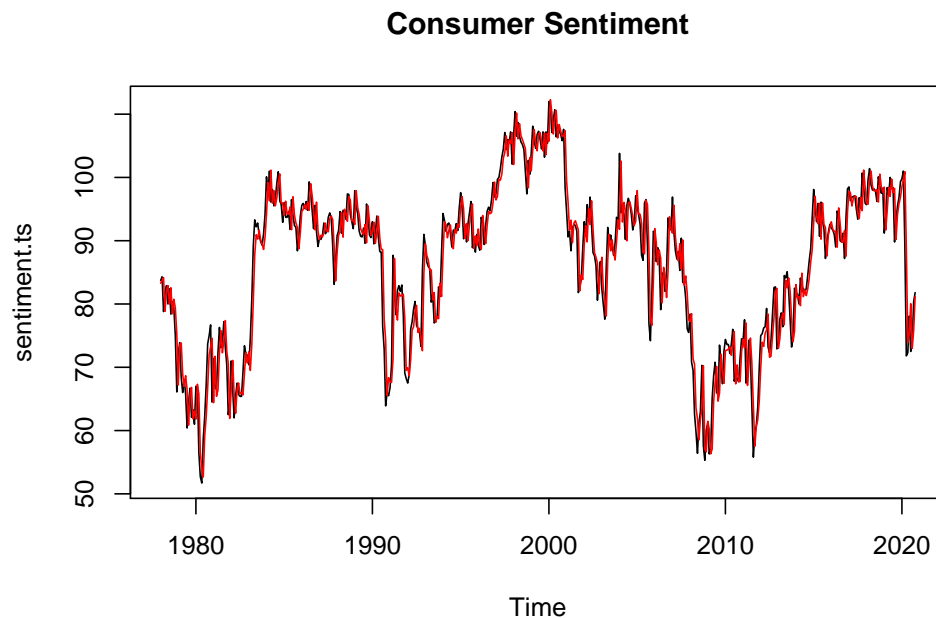


From the summary of the vehicles model, the training set error measures are quite low, indicating a good model fit. Only RMSE and MAE are quite high.

High RMSE could suggest that the data is overfit in the model. On the other hand, high MAE is relative and varies according to the data. There could be various factors contributing toward it.

“High” MAE itself is relative and varies according to the data and there could be multiple factors contributing towards it.

```
plot(sentiment.ts, main = "Consumer Sentiment")
lines(fitted(model_sentiment_), col = "red")
```



```
#associated diagnostic statistics
summary(model_sentiment_)
```

```
## Series: sentiment.ts
## Regression with ARIMA(2,1,1)(1,0,1)[12] errors
##
## Coefficients:
##      ar1      ar2      ma1      sar1      sma1      t
##      0.7180 -0.0988 -0.7561  0.6534 -0.5750 -0.1658
## s.e.  0.1361  0.0508  0.1305  0.4065  0.4394  1.6391
##
## sigma^2 estimated as 15.38:  log likelihood=-1426.05
## AIC=2866.11   AICc=2866.33   BIC=2895.79
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE
## Training set 0.004300743 3.894863 2.962178 -0.1468642 3.619064 0.390569
##              ACF1
## Training set -0.0001031855
```

```
coeftest(model_sentiment_)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1    0.718035   0.136116  5.2752 1.326e-07 ***
## ar2   -0.098838   0.050807 -1.9454  0.05173 .
## ma1   -0.756148   0.130485 -5.7949 6.836e-09 ***
## sar1   0.653373   0.406482  1.6074  0.10797
## sma1  -0.574979   0.439416 -1.3085  0.19070
## t     -0.165849   1.639123 -0.1012  0.91941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now we will analyze the associated diagnostic and error statistics for the consumer sentiment model. We see the model derived a overall variance of 15.38, which in relation to the data is good, but of course, the lower the better. As for error metrics, we want to see how well the model does in terms of how accurate the model performs in predicting values when using a training set. A ME value very close to 0 was derived, which measures the average of all the errors in the model, which the smaller the value, the better it is. A RMSE of 3.894863 was derived, which states that the model does well in predicting within the sample. The MAE of 2.962178 was derived for the model, which states that the model's measurement of errors between paired observations is relatively small, which is a desired result. A MPE of -0.1468642 was derived which measures the average of percentage errors of which the forecast of a model is different compared to the actual values.

Thus, a small value is desired to communicate that the model's forecast ability is good. A MAPE of 3.619064 was derived which measures how accurate a forecast model is and calculates the errors being minimized, thus, the smaller the number, the better the model is at forecasting. Lastly, a MASE of 0.390569 was derived, which measures the mean absolute error produced by the forecast. The lower the value the better since a value greater than 1 means that the actual forecast will do worse out of sample.

## 9. Forecasting 12-steps ahead with the respective error bands

```
#h=12 is forecasting 1 year ahead
m5 = Arima(vehicle.ts, order = c(3,0,1), include.drift = , seasonal = list(order = c(2,1,1)))
plot(forecast(m5, h = 12), shadecols="oldstyle",
main = "Forecast For Total Vehicle Sales")
lines(model_vehicle_$fitted, col = "red")
legend("topleft", c("Actual", "Fitted"), col = c("black", "red"), lty = c(1, 1))
```

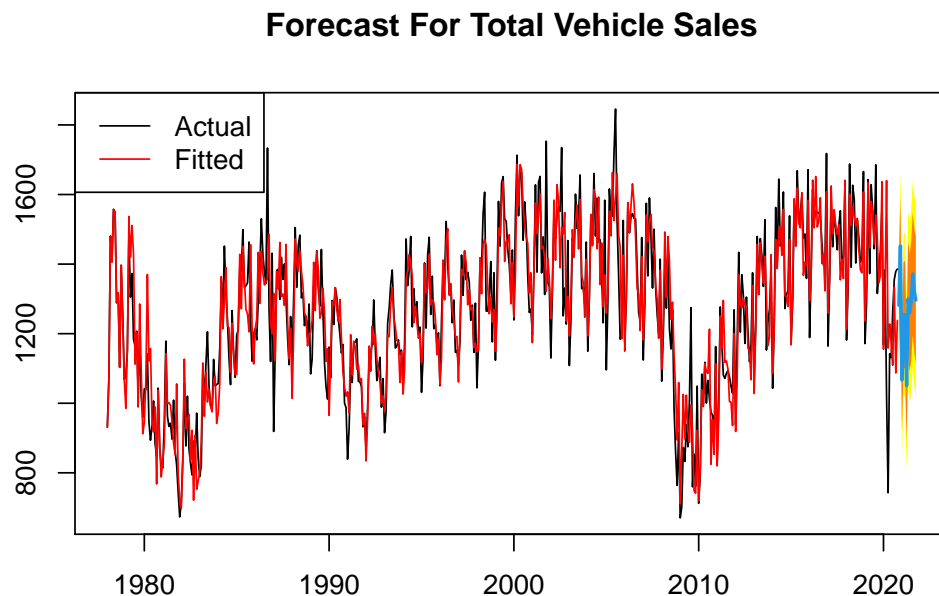


Figure 35: Vehicles Sold Forecast

Forecasting model is plotted for Total vehicle sales from 2020 to 2021 (h=12). The predicted data is shown in the extension of the graph and is color coded. The yellow area on the outside is the prediction interval, while the orange shaded area is the confidence interval and the blue line in the middle is the point forecast. The predicted values show a rise in the the total vehicle in the following 1 year forecast.

```
#h=12 is forecasting 1 year ahead
m6 = Arima(sentiment.ts, order = c(2,1,1), include.drift = , seasonal = list(order = c(1,0,1)))
plot(forecast(m6, h = 12), shadecols="oldstyle", main="Forecast For Consumer Sentiment")
lines(model_sentiment_$fitted, col = "red")
legend("topleft", c("Actual", "Fitted"), col = c("black", "red"), lty = c(1, 1))
```

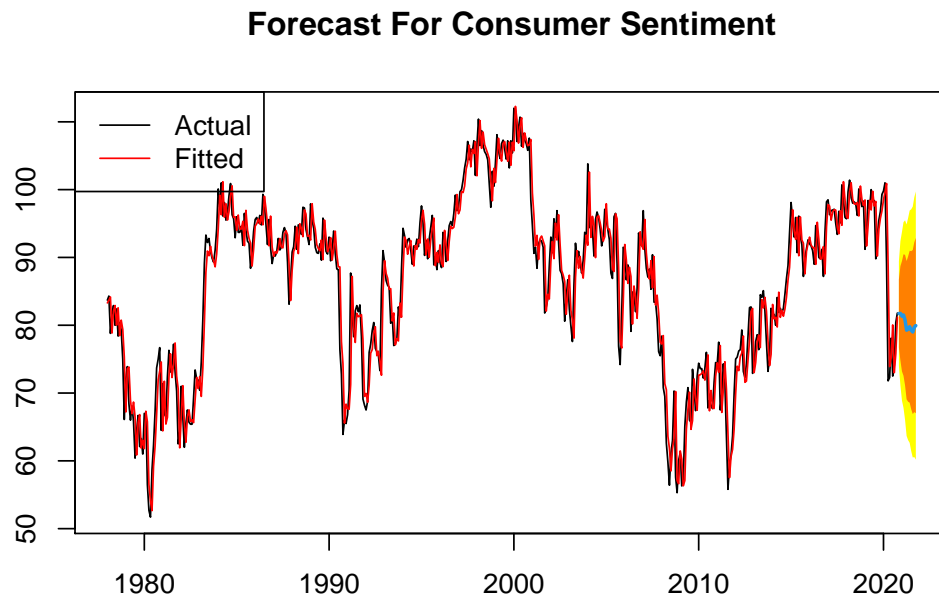


Figure 36: Sentiment Forecast

The plot above is the Forecast for the Consumer Sentiment from 2020 to 2021 (h=12). The predicted data is shown in the extension of the graph and is color coded. The yellow area on the outside is the prediction interval, while the orange shaded area is the confidence interval and the blue line in the middle is the point forecast. The predicted values show a rise in the the total vehicle in the following 1 year forecast.

We want to point out that the forecast for both models appear to be going in different directions with total vehicle sales projected to increase and Consumer Sentiment appearing to be stagnant using the point forecast as reference for both. We will talk in length about this toward the end.

## 10. Fitting an appropriate VAR model

```
# Fitting a VAR model using our two variables
```

```
Model = cbind(vehicle.ts, sentiment.ts)
```

```
Confidence_Model = data.frame(Model)
```

```
VARselect(Confidence_Model, 10)
```

```
## $selection
```

```
## AIC(n) HQ(n) SC(n) FPE(n)
```

```
##      9      9      2      9
```

```
##
```

```
## $criteria
```

```
##              1              2              3              4              5
```

```
## AIC(n)      12.79986      12.73796      12.70953      12.70338      12.71398
```

```
## HQ(n)       12.81958      12.77082      12.75554      12.76253      12.78628
```

```
## SC(n)       12.85013      12.82174      12.82683      12.85418      12.89830
```

```
## FPE(n) 362167.10906 340428.75825 330888.66520 328858.79267 332367.32136
```

```
##              6              7              8              9              10
```

```
## AIC(n)      12.71109      12.69705      12.69857      12.55856      12.56953
```

```
## HQ(n)       12.79654      12.79564      12.81031      12.68345      12.70756
```

```
## SC(n)       12.92892      12.94839      12.98343      12.87693      12.92141
```

```
## FPE(n) 331410.51101 326792.40981 327296.33715 284541.78104 287686.14304
```

```
VAR_Model = VAR(Confidence_Model, p=9)
```

```
summary(VAR_Model)
```

```
##
```

```
## VAR Estimation Results:
```

```
## =====
```

```
## Endogenous variables: vehicle.ts, sentiment.ts
```

```
## Deterministic variables: const
```

```
## Sample size: 505
```

```
## Log Likelihood: -4568.957
```

```
## Roots of the characteristic polynomial:
```

```
## 0.9702 0.9512 0.9207 0.9207 0.8914 0.8914 0.8723 0.8723 0.8601 0.8601 0.7841 0.7638 0.7638 0.7227 0.7227
```

```
## Call:
```

```
## VAR(y = Confidence_Model, p = 9)
```

```
##
```

```
##
```

```
## Estimation results for equation vehicle.ts:
```

```
## =====
```

```
## vehicle.ts = vehicle.ts.l1 + sentiment.ts.l1 + vehicle.ts.l2 + sentiment.ts.l2 + vehicle.ts.l3 + sentiment.ts.l3 + ...
```

```

##
##               Estimate Std. Error t value Pr(>|t|)
## vehicle.ts.l1    0.33796    0.04230   7.990 9.89e-15 ***
## sentiment.ts.l1  3.42886    1.53650   2.232 0.02610 *
## vehicle.ts.l2    0.14016    0.04510   3.108 0.00200 **
## sentiment.ts.l2 -1.11542    2.11167  -0.528 0.59759
## vehicle.ts.l3    0.30321    0.04565   6.642 8.32e-11 ***
## sentiment.ts.l3  1.01793    2.11569   0.481 0.63064
## vehicle.ts.l4   -0.14347    0.04669  -3.073 0.00224 **
## sentiment.ts.l4 -0.93693    2.10986  -0.444 0.65719
## vehicle.ts.l5    0.03334    0.04710   0.708 0.47938
## sentiment.ts.l5  0.79160    2.11281   0.375 0.70807
## vehicle.ts.l6   -0.22539    0.04659  -4.837 1.77e-06 ***
## sentiment.ts.l6  0.29764    2.11267   0.141 0.88802
## vehicle.ts.l7    0.04663    0.04611   1.011 0.31232
## sentiment.ts.l7  0.08804    2.13200   0.041 0.96708
## vehicle.ts.l8   -0.05251    0.04542  -1.156 0.24824
## sentiment.ts.l8 -0.37005    2.12537  -0.174 0.86185
## vehicle.ts.l9    0.36239    0.04192   8.644 < 2e-16 ***
## sentiment.ts.l9 -0.39081    1.60191  -0.244 0.80736
## const           7.43407   45.90955   0.162 0.87143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 132.5 on 486 degrees of freedom
## Multiple R-Squared:  0.674,    Adjusted R-squared:  0.6619
## F-statistic: 55.82 on 18 and 486 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation sentiment.ts:
## =====
## sentiment.ts = vehicle.ts.l1 + sentiment.ts.l1 + vehicle.ts.l2 + sentiment.ts.l2 + vehicle.ts.l3 + s
##
##               Estimate Std. Error t value Pr(>|t|)
## vehicle.ts.l1    3.069e-03  1.248e-03   2.459 0.0143 *
## sentiment.ts.l1  9.509e-01  4.533e-02  20.974 <2e-16 ***
## vehicle.ts.l2   -2.548e-03  1.331e-03  -1.915 0.0561 .
## sentiment.ts.l2 -1.162e-01  6.230e-02  -1.864 0.0629 .
## vehicle.ts.l3   -5.376e-04  1.347e-03  -0.399 0.6900
## sentiment.ts.l3  6.913e-02  6.242e-02   1.107 0.2687
## vehicle.ts.l4   -2.106e-03  1.377e-03  -1.529 0.1269
## sentiment.ts.l4  2.504e-02  6.225e-02   0.402 0.6876
## vehicle.ts.l5    1.090e-03  1.390e-03   0.785 0.4331
## sentiment.ts.l5 -8.130e-02  6.234e-02  -1.304 0.1928
## vehicle.ts.l6    8.126e-04  1.375e-03   0.591 0.5548
## sentiment.ts.l6  1.271e-01  6.233e-02   2.039 0.0420 *
## vehicle.ts.l7   -2.103e-05  1.360e-03  -0.015 0.9877
## sentiment.ts.l7 -4.650e-02  6.290e-02  -0.739 0.4601
## vehicle.ts.l8   -5.325e-04  1.340e-03  -0.397 0.6913
## sentiment.ts.l8  1.071e-01  6.271e-02   1.708 0.0883 .
## vehicle.ts.l9    5.704e-04  1.237e-03   0.461 0.6449
## sentiment.ts.l9 -6.930e-02  4.726e-02  -1.466 0.1432
## const           3.186e+00  1.355e+00   2.352 0.0191 *

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 3.909 on 486 degrees of freedom
## Multiple R-Squared:  0.9087,    Adjusted R-squared:  0.9053
## F-statistic: 268.8 on 18 and 486 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##               vehicle.ts sentiment.ts
## vehicle.ts      17553.93         32.08
## sentiment.ts     32.08         15.28
##
## Correlation matrix of residuals:
##               vehicle.ts sentiment.ts
## vehicle.ts      1.00000         0.06194
## sentiment.ts    0.06194         1.00000

#plot.new()
#plot(VAR_Model)
#or
#tsdisplay(residuals(VAR_Model)[-1],main = "Comps = starts(t-k) + comps(t-k)")
#tsdisplay(residuals(VAR_Model)[-2],main = "Comps = starts(t-k) + comps(t-k)")
```

We will now fit a VAR model using our variables. We will first use the VARselect command to help us derive the correct VAR model to use and the optimal lags for each model. We see that the model selection is suggesting we use 9 based on the results of the AIC, HQ, and FPE tests. Yes, the BIC (SC) suggests we use two, but since the other three tests suggested 9, we are going to create a VAR(9) model.

The VAR results are stated above. The first estimation is for the Consumer Sentiment equation. The model has an adjusted  $R^2$  of 0.9053 and a Residual Standard Error of 3.909, suggesting a good fit to the data. We see that the first lag for Consumer Sentiment (Confidence. 11) is statistically significant at .1%. There are three explanatory variables that are statistically significant at 5% (Vehicle. 11, Confidence. 16, and the Constant intercept) and statistically significant at 10% (Confidence. 12, Vehicle. 12, and Confidence. 18). The remaining variables are insignificant.

We now will observe the second estimation for Total Vehicle Sales. The model derived an adjusted  $R^2$  of 0.6619, lower than the estimation model for Consumer Sentiment. A larger Residual Standard error of 132.50 was derived. However, we see more explanatory variables are statistically significant for this model compared to the first model. Four explanatory variables are statistically significant at .1% (Vehicle. 11, Vehicle. 13, Vehicle. 16, and Vehicle. 19). Two variables are statistically significant at 5% (Vehicle. 12 and Vehicle. 14). Explanatory variable Confidence. 11 is statistically significant at 10%. The remaining explanatory variables are insignificant.

We will now focus on the plots of the VAR model for each variable separately. The first plot is the diagram of fit and residuals for Consumer Sentiment. As you can see, the VAR model fits our data exceptionally well, which is supported by the large adjusted  $R^2$  and small Residual Standard error the model derived. The residuals of the model are plotted below and it appears to be stationary with some slight spikes in certain periods. To confirm if the residuals are in fact stationary, we will conduct an Augmented Dickey Fuller Test and KPSS test. The results will be presented momentarily.

We now want to observe the ACF and PACF plots of the residuals. As you can see, we eliminated a lot of the lags in terms of their correlation strength and it appears to be white noise. To prove this statistically, we will conduct a Ljung-Box Test and Box-Pierce Test.

Diagram of fit and residuals for vehicle.ts

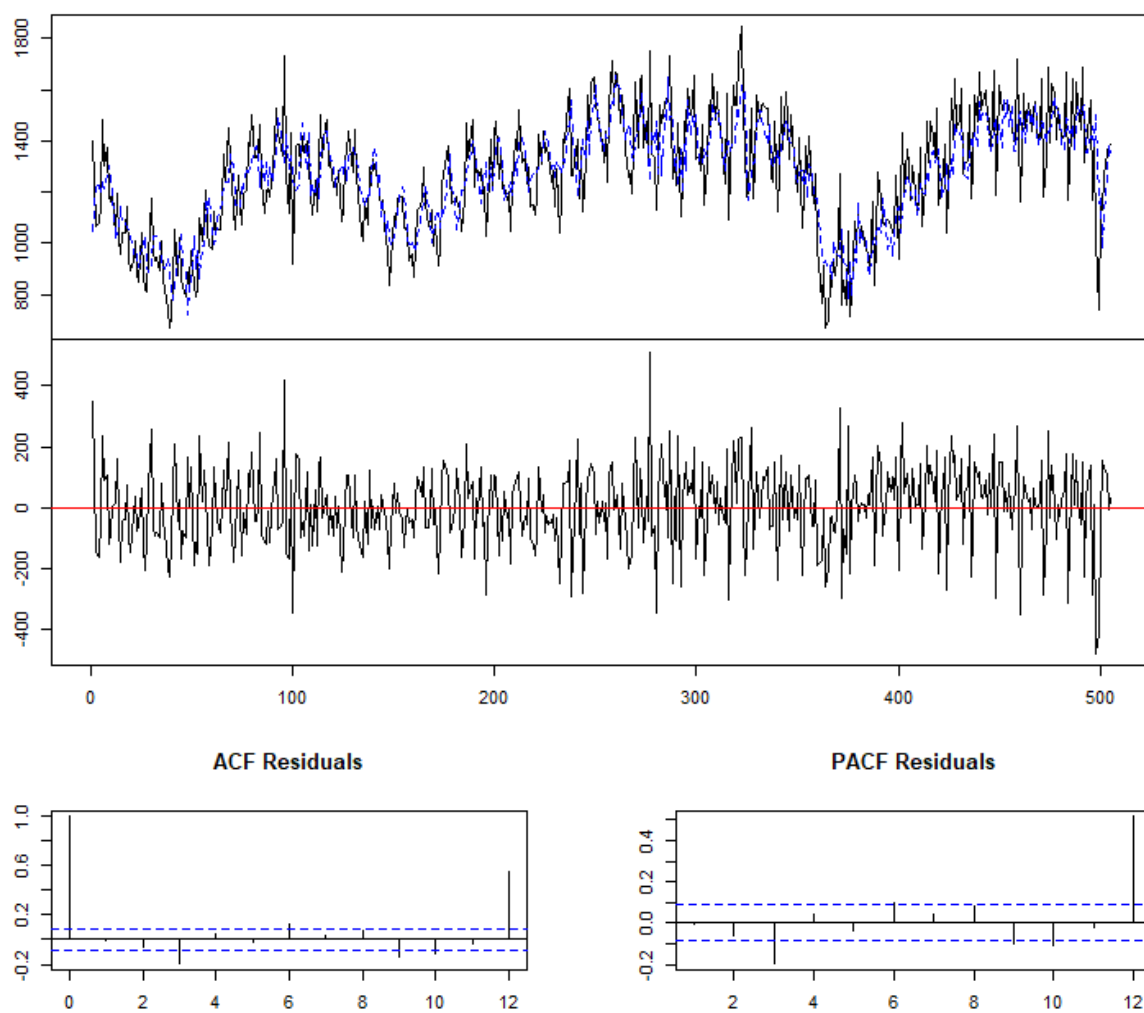


Figure 37: Var Model Plot 2



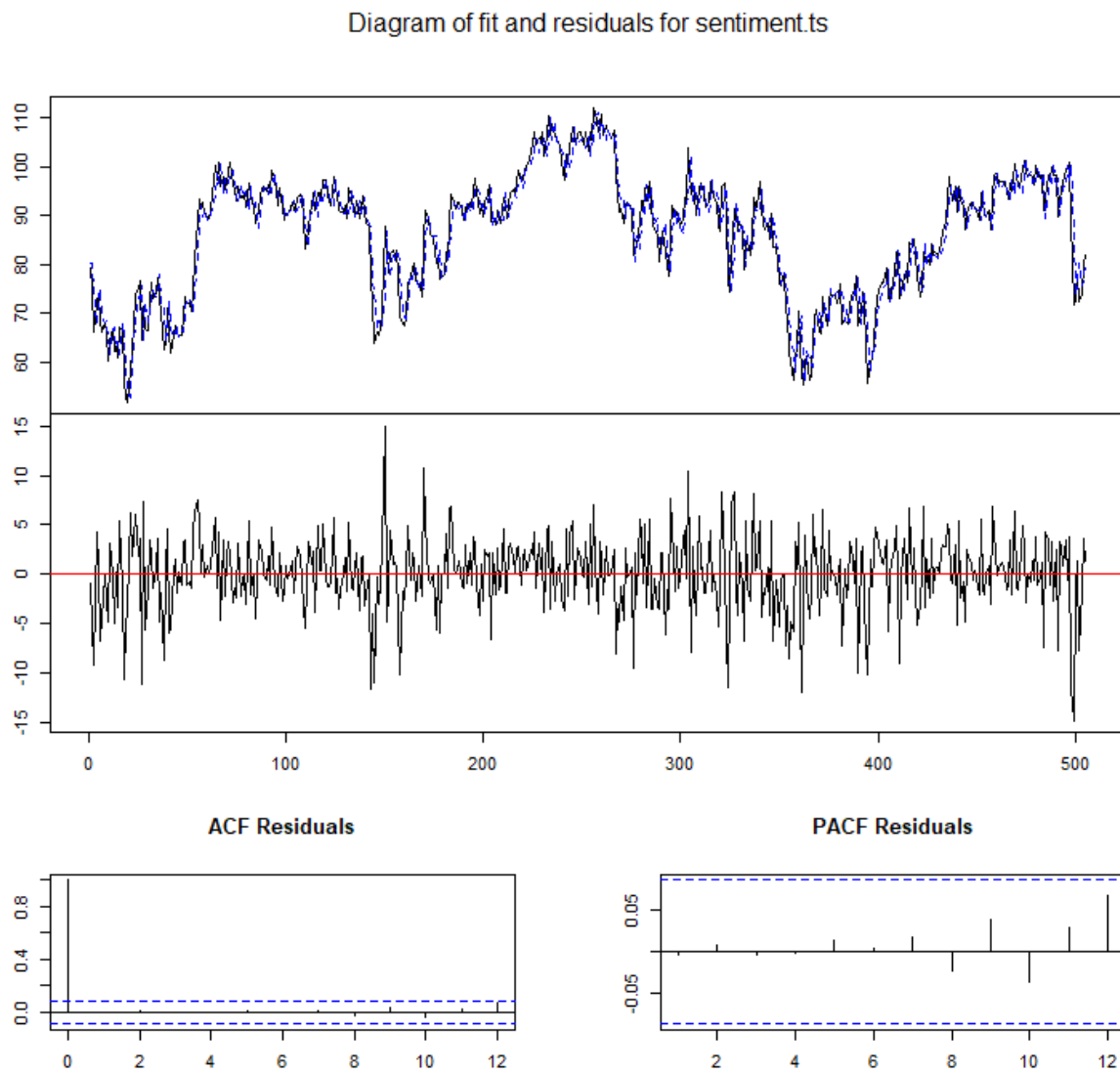


Figure 38: Var Model Plot 2

We also want to see if the residuals are normally distributed. We would desire our residuals to be Normally Distributed, but this isn't necessary to conduct Time Series forecasting. The results of the Jarque Bera Test are included after the Augmented Dickey Fuller Test, KPSS test, Ljung-Box, and Box-Pierce tests.

```
# Augmented Dickey Fuller and KPSS Test for Consumer Sentiment Residuals to Determine Stationary #
adf.test(residuals(VAR_Model)[-1])
```

```
##
## Augmented Dickey-Fuller Test
##
## data: residuals(VAR_Model)[-1]
## Dickey-Fuller = -10.777, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(residuals(VAR_Model)[-1])
```

```
##
## KPSS Test for Level Stationarity
##
## data: residuals(VAR_Model)[-1]
## KPSS Level = 0.42436, Truncation lag parameter = 7, p-value = 0.06665
```

Based on the Augmented Dickey Fuller Test and KPSS test results, the residuals for the Consumer Sentiment model are stationary.

```
# Testing to see if the residuals for the Consumer Sentiment Residuals is White Noise #
# H0: Residuals are White Noise #
# HA: Residuals are not White Noise #
Box.test(residuals(VAR_Model)[-1], type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: residuals(VAR_Model)[-1]
## X-squared = 0.021469, df = 1, p-value = 0.8835
```

```
Box.test(residuals(VAR_Model)[-1], type = "Box-Pierce")
```

```
##
## Box-Pierce test
##
## data: residuals(VAR_Model)[-1]
## X-squared = 0.021405, df = 1, p-value = 0.8837
```

Based on the results from the Ljung-Box and Box-Pierce test, we can statistically conclude that our residuals are classified as white noise.

```
# Testing to see if residuals of Consumer Sentiment are Normally Distributed #
jarque.bera.test(residuals(VAR_Model)[-1])
```

```
##
## Jarque Bera Test
##
## data: residuals(VAR_Model)[-1]
## X-squared = 791.05, df = 2, p-value < 2.2e-16
```

Based on the Jarque Bera Test, the residuals are not normally distributed.

The second plot is the diagram of fit and residuals for Total Vehicle Sales. As you can see, the VAR model fits our data, but it is apparent it is not entirely capturing the entire fit, which makes sense due to an adjusted  $R^2$  of 0.6619 and a large Residual Standard error of 132.50. The residuals of the model are plotted below and it appears to be stationary with some pronounce spikes in certain periods. To confirm if the residuals are in fact stationary, we will conduct an Augmented Dickey Fuller Test and KPSS test.

We now want to observe the ACF and PACF plots of the residuals. As you can see, we eliminated a lot of the lags in terms of their correlation strength and it appears to be white noise. There does appear to be a significant lag at lag 12, but can be contributed as pure white noise. To prove this statistically, we will conduct a Ljung-Box Test and Box-Pierce Test.

Similar to what we did for the Consumer Sentiment Model, We also want to see if the residuals are normally distributed. We would desire our residuals to be Normally Distributed, but this isn't necessary to conduct Time Series forecasting. The results of the Jarque Bera Test are included after the Augmented Dickey Fuller Test, KPSS test, Ljung-Box, and Box-Pierce tests.

```
# Augmented Dickey Fuller and KPSS Test for Consumer Sentiment Residuals to Determine Stationary #
adf.test(residuals(VAR_Model)[-2])
```

```
## Warning in adf.test(residuals(VAR_Model)[-2]): p-value smaller than printed p-
## value
```

```
##
## Augmented Dickey-Fuller Test
##
## data: residuals(VAR_Model)[-2]
## Dickey-Fuller = -10.742, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(residuals(VAR_Model)[-2])
```

```
##
## KPSS Test for Level Stationarity
##
## data: residuals(VAR_Model)[-2]
## KPSS Level = 0.36701, Truncation lag parameter = 7, p-value = 0.09137
```

Based on the Augmented Dickey Fuller Test and KPSS test results, the residuals for the Total Vehicle Sales model are stationary.

```
# Testing to see if the residuals for the Consumer Sentiment Residuals is White Noise #
# H0: Residuals are White Noise #
# HA: Residuals are not White Noise #
Box.test(residuals(VAR_Model)[-2], type = "Ljung-Box")
```

```
##
## Box-Ljung test
##
## data: residuals(VAR_Model)[-2]
## X-squared = 0.075352, df = 1, p-value = 0.7837
```

```
Box.test(residuals(VAR_Model)[-2], type = "Box-Pierce")
```

```
##
## Box-Pierce test
##
## data: residuals(VAR_Model)[-2]
## X-squared = 0.075129, df = 1, p-value = 0.784
```

Based on the results from the Ljung-Box and Box-Pierce test, we can statistically conclude that our residuals are classified as white noise.

```
# Testing to see if residuals of Consumer Sentiment are Normally Distributed #
jarque.bera.test(residuals(VAR_Model)[-2])
```

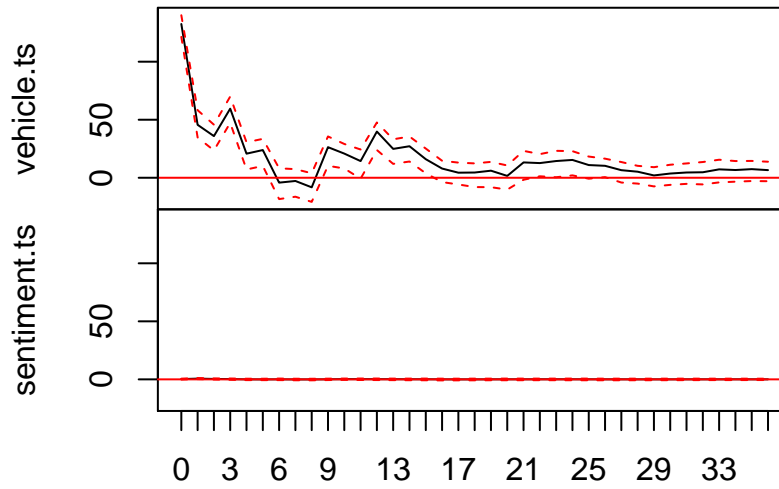
```
##
## Jarque Bera Test
##
## data: residuals(VAR_Model)[-2]
## X-squared = 793.87, df = 2, p-value < 2.2e-16
```

Based on the Jarque Bera Test, the residuals are not normally distributed.

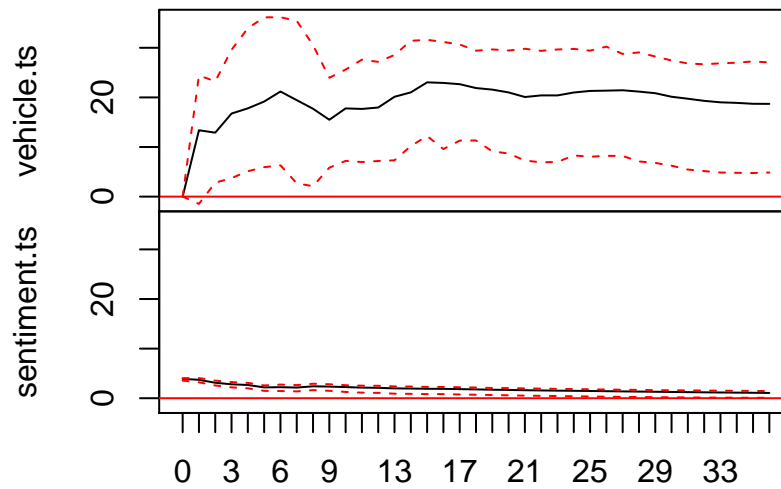
## 11. Impulse response functions

```
# Impulse - Response Functions #
plot(irf(VAR_Model, n.ahead=36))
```

### Orthogonal Impulse Response from vehicle.ts



### 95 % Bootstrap CI, 100 runs Orthogonal Impulse Response from sentiment.ts



### 95 % Bootstrap CI, 100 runs

Figure 39: IRF

We now plot the Impulse Response Functions for our two variables 36 steps away. Since both our variable's data frequencies are monthly, 36 steps is three years. As you can see, there isn't much activity when observing a shock of Consumer Sentiment on Consumer Sentiment. An argument can be made that a minor shock can be observed, but as time goes on it appears to be converging closely to 0. When we observe a shock of Consumer Sentiment on Total Vehicle Sales, we see that Total Vehicle Sales increase and peaks at around 15 (a year and three months) and the effect of the shock lingers for some time. This makes sense because if consumers feel optimistic about the economy, they will partake in purchasing more goods and services like vehicles. We must keep in mind that we designated three years to observe how the Impulse Response Functions would appear, but the sustainability of the shock could linger or not depending on how the economy continues to operate. It is possible that an adverse shock could occur and drastically change our results.

We will now observe how a shock in Total Vehicle Sales will affect Consumer Sentiment. As we can see, the shock is non-existent and is close to 0 for the entire duration. When we observe a shock in Total Vehicle Sales on Total Vehicle Sales, we see a large negative effect and then it gradually begins to decrease, crosses below zero, slightly spikes again and then finally begins to converge close to 0. This scenario could happen where consumers purchase vehicles in large quantities in one month and the next month is followed with a decrease in vehicles purchased.

It is evident that a shock from Consumer Sentiment on Total Vehicle Sales is more apparent and stronger in terms of lingering effects. This makes intuitive sense because if consumer confidence for some reason experience an increase in confidence because their perception of the economy improved, overall consumption will increase and goods like vehicles would be purchased due to the expectation that the economy is operating in a healthy manner.

## 12. Granger-Causality test

```
library(lmtest)
# Granger-Causality #
# Does Confidence Granger cause Vehicle? #
# Ho: Consumer Sentiment does not Granger cause Total Vehicle Sales #
# HA: Confidence Sentiment does Granger cause Total Vehicle Sales #
grangertest(vehicle.ts ~ sentiment.ts, order=9)

## Granger causality test
##
## Model 1: vehicle.ts ~ Lags(vehicle.ts, 1:9) + Lags(sentiment.ts, 1:9)
## Model 2: vehicle.ts ~ Lags(vehicle.ts, 1:9)
##   Res.Df Df       F    Pr(>F)
## 1      486
## 2      495 -9 2.5915 0.006323 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Does Vehicle Granger cause Confidence? #
# Ho: Vehicle does not Granger cause Consumer Sentiment #
# Ha: Vehicle does Granger cause Consumer Sentiment #
grangertest(sentiment.ts ~ vehicle.ts, order=9)

## Granger causality test
##
## Model 1: sentiment.ts ~ Lags(sentiment.ts, 1:9) + Lags(vehicle.ts, 1:9)
```

```
## Model 2: sentiment.ts ~ Lags(sentiment.ts, 1:9)
##   Res.Df Df       F Pr(>F)
## 1     486
## 2     495 -9 1.1628 0.3169
```

Now we are going to perform the Granger-Causality test on our variables to determine if one of our variables can be used to forecast another variable. To further elaborate, we are using Granger-Causality test to measure the ability to predict the future values of a time series variable using prior values from another time series variable. In our case, we are using Total Vehicle Sales and Consumer Sentiment to determine if there is a Granger-Causality. We will first test if Consumer Sentiment Granger causes Total Vehicle Sales, which will we observing if past values of Consumer Sentiment can be used to predict future values of Total Vehicle Sales. The null hypothesis is Consumer Sentiment does not Granger cause Total Vehicle Sales and the alternative is the opposite, it does not Granger cause. With a p-value of 0.006323, we reject the null hypothesis (assuming a critical value of 5%) and conclude that Consumer Sentiment does Granger cause Total Vehicle Sales.

We also wanted to observe if there was a possible Granger-Causality using past Total Vehicle Sales to forecast future Consumer Sentiment. With a p-value of 0.3169, we fail to reject the null hypothesis and conclude that there isn't sufficient evidence to suggest that Total Vehicle Sales does not Granger Cause Consumer Sentiment.

Our statistical conclusions make logical sense. If consumers are confident on their perception on the economy, they will be more inclined to make large purchases like vehicles. We would also expect the opposite to be true that Total Vehicle Sales would decrease when consumer confidence is low, since consumers would perceive the economy to not be performing to their standards. As a reminder, The Impulse Response plot supported that a shock to Consumer Sentiment would cause a response to Total Vehicle Sales depending on the direction of the shock, meaning the shock of Consumer Sentiment may be positive or negative and from that result, Total Vehicle Sales will also change accordingly.

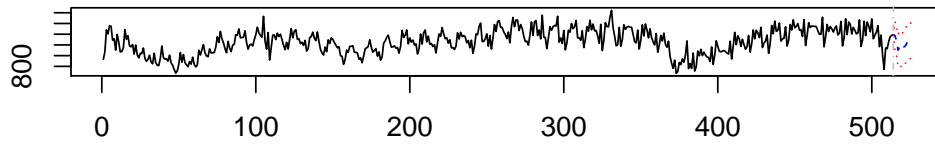
### 13. Using VAR model to forecast 12-steps ahead with respective error bands

```
# Using VAR model to forecast 12-steps ahead #
Prediction_VAR = predict(object=VAR_Model, n.ahead=12)
plot(Prediction_VAR)

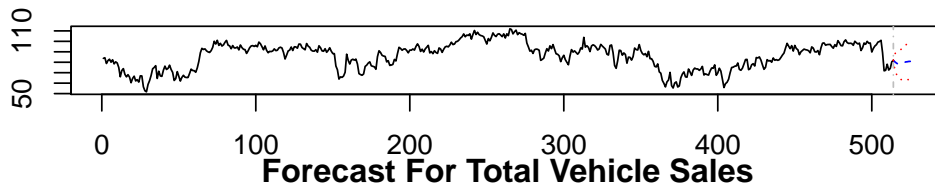
# Using ARIMA model to forecast 12-steps ahead #
m5 = Arima(vehicle.ts, order = c(3,0,1), include.drift = , seasonal = list(order = c(2,1,1)))
plot(forecast(m5, h = 12), main = "Forecast For Total Vehicle Sales")
lines(model_vehicle_$fitted, col = "red")
legend("topleft", c("Actual", "Fitted"), col = c("black", "red"), lty = c(1, 1))

m6 = Arima(sentiment.ts, order = c(2,1,1), include.drift = , seasonal = list(order = c(1,0,1)))
plot(forecast(m6, h = 12), main = "Forecast For Consumer Sentiment")
lines(model_sentiment_$fitted, col = "red")
legend("topleft", c("Actual", "Fitted"), col = c("black", "red"), lty = c(1, 1))
```

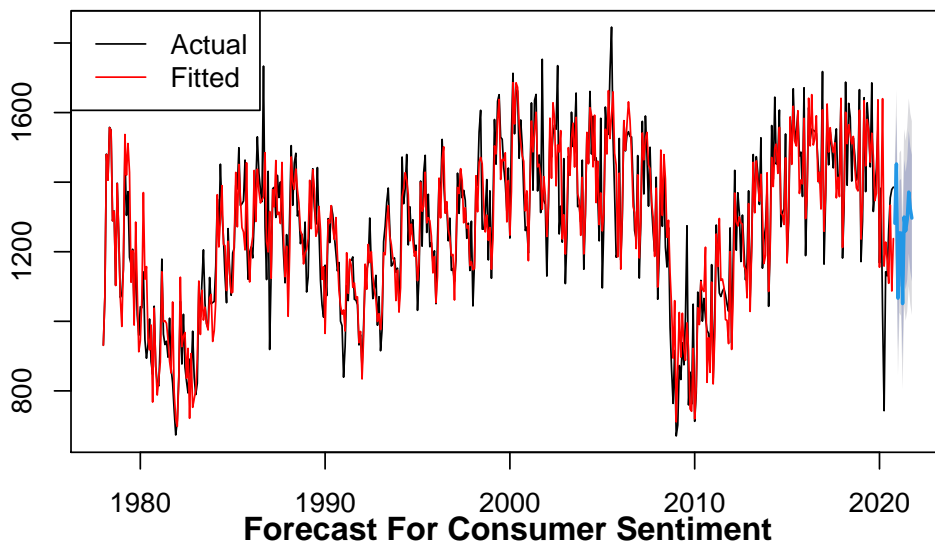
**Forecast of series vehicle.ts**



**Forecast of series sentiment.ts**



**Forecast For Total Vehicle Sales**



**Forecast For Consumer Sentiment**

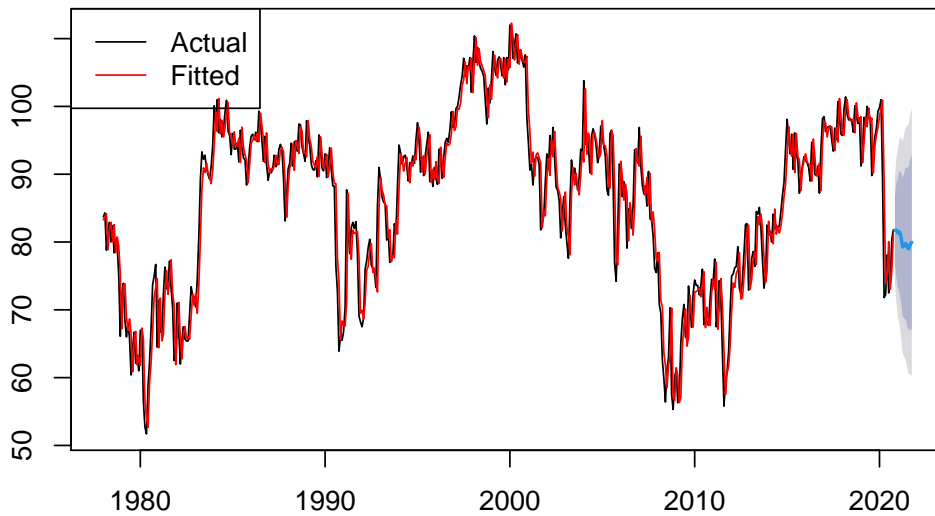


Figure 40: Predicted VAR



The plots above graphically show the forecasts at 12 steps for VAR and Arima models. For the most part from a visual perspective the VAR and Arima model for Total Vehicle Sales seem to be in agreement on the trajectory of the forecast. The VAR model seems to indicate a slight drop, but is for the most part on an upward trajectory. Essentially, both models see Total Vehicle Sales going up one year from now. However, there is a distinct difference for the VAR and Arima model for Consumer Sentiment. The VAR model believes that Consumer Sentiment is going to decrease then slightly increase. The Arima model is predicting Consumer Sentiment to decrease and stay stagnant. This difference is interesting because the VAR model believes that both variables will experience an increase which can be argued can happen if we believe that consumer confidence is a contributing factor in determining total vehicle sales. The explanation for our forecast is referring to the point forecast because many would be interested on a single point forecast then a range (Confidence and Forecast Interval).

## 14. Backtesting the ARIMA model

(a)

Use a recursive backtesting scheme, and forecast 12-steps ahead at each iteration. Compute the mean absolute percentage error at each step. Provide a plot showing the MAPE over each iteration.

```
# Recursive for both variables At 12-Steps #

# Vehicle
library(CombMSC)
c = 1
MAPE = c() #create empty vector to store MAPE
MAPE = as.data.frame(MAPE)
#loop
for (i in 39:1){
  estindex = i*12

  splitdata = splitTrainTest(vehicle.ts,length(vehicle.ts)-estindex) #split data into training and test
  estset = splitdata$train #create estimation set
  predset = splitdata$test #create prediction set
  #fit model to estimation set
  fit <- Arima(estset,order=c(3,0,0), include.drift = TRUE, seasonal=list(order=c(0,1,1)))
  fc <- forecast(fit,h=12) #forecast ahead 12 steps
  j= accuracy(fc,predset) #find MAPE
  #store MAPE
  MAPE[c,1]=i
  MAPE[c,2]=j[2,5]
  c=c+1
}
plot(1:39,MAPE$V2,type='l',xlab="Iterations",ylab="MAPE",
main = "Recursive of Total Vehicle Sales at 12-Steps") #plot

# Consumer Sentiments
c = 1
UCAPE = c() #create empty vector to store MAPE
UCAPE = as.data.frame(UCAPE)

#loop
```

```

for (i in 39:1){
  estindex = i*12
  splitdata = splitTrainTest(sentiment.ts,length(sentiment.ts)-estindex) #split data into training and
  estset = splitdata$train #create estimation set
  predset = splitdata$test #create prediction set
  #fit model to estimation set
  fit <- Arima(estset,order=c(2,1,1), include.drift = FALSE, seasonal=list(order=c(1,0,1)))
  fc <- forecast(fit,h=12) #forecast ahead 12 steps
  j= accuracy(fc,predset) #find MAPE
  #store MAPE
  UMAPE[c,1]=i
  UMAPE[c,2]=j[2,5]
  c=c+1
}
plot(1:39,UMAPE$V2,type='l',xlab="Iterations",ylab="MAPE",
main = "Recursive of Consumer Sentiment at 12-Steps") #plot

```

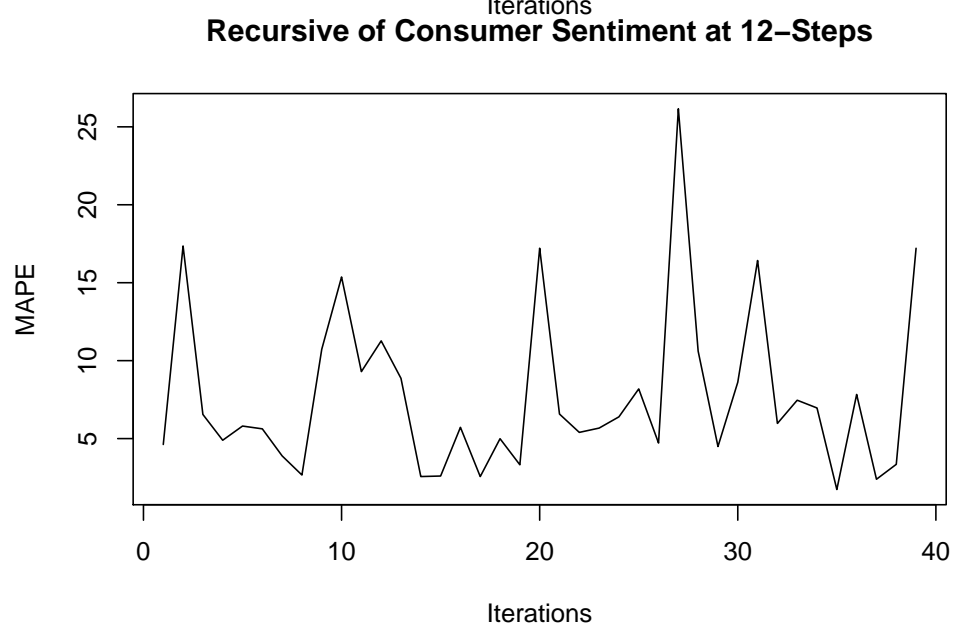
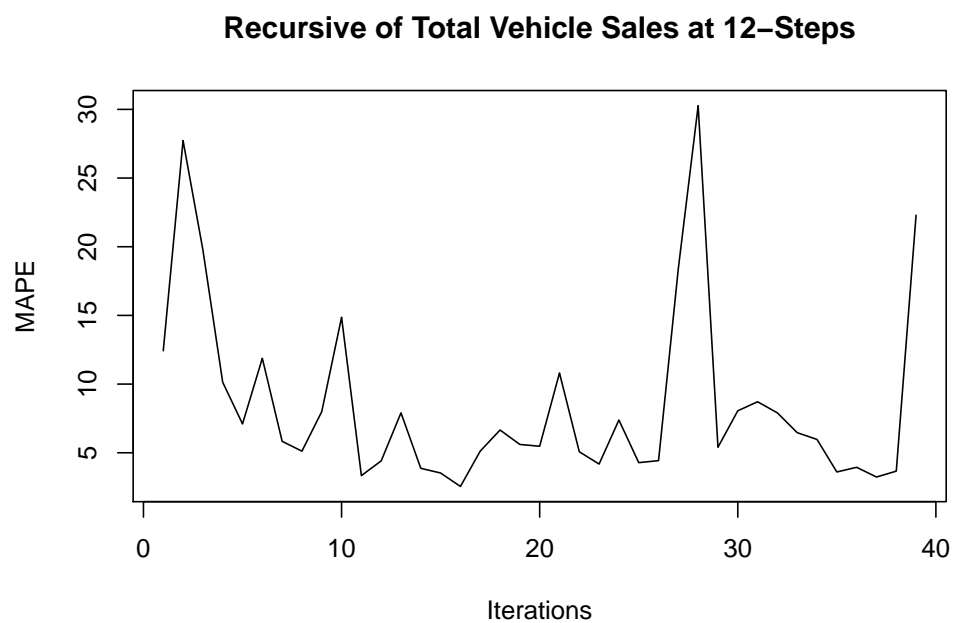


Figure 41: Recursive backtesting

The plots above show the recursive backtest for both consumer sentiment and total vehicle sales. There are some similarities in terms of the magnitude of the MAPE calculations. One noteworthy note is that it appears the MAPE for consumer sentiment has more fluctuations compared to the MAPE of total vehicle sales. The MAPE values for both variables seem rather large, but it can be attributed to the difficulty of forecasting 12-steps away, which for our data is equivalent to one year.

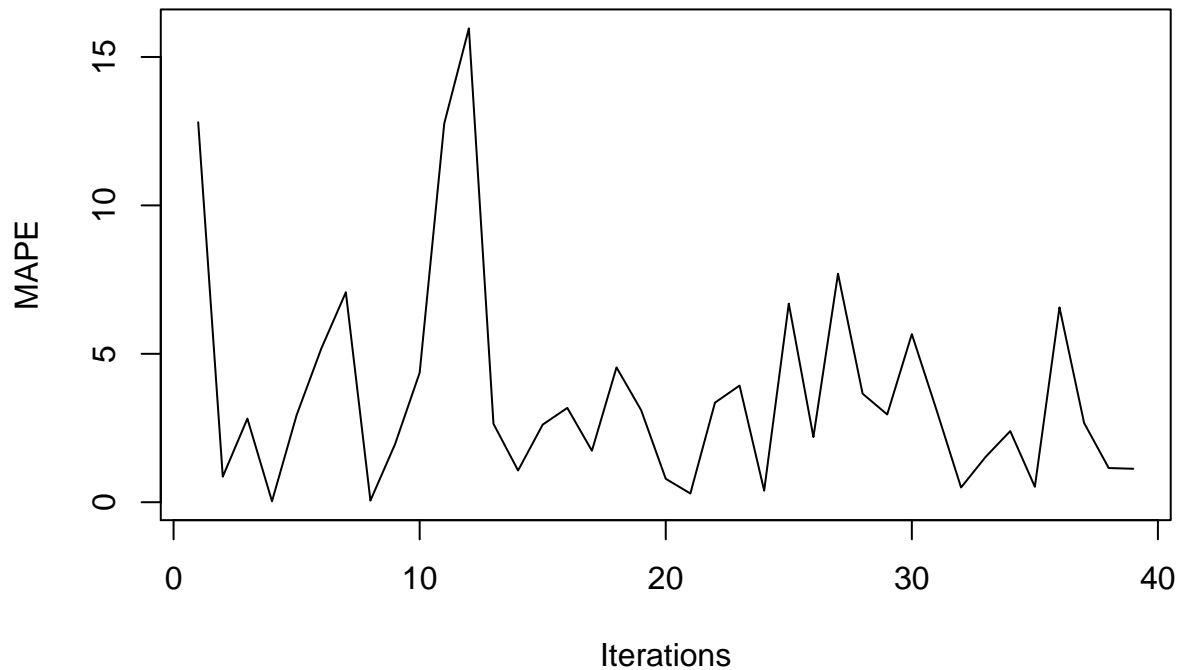
(b)

Shorten your forecast horizon to only 1-step ahead. Compute the absolute percentage error at each iteration, and plot.

```
# Recursive for both variables At 1-Step #

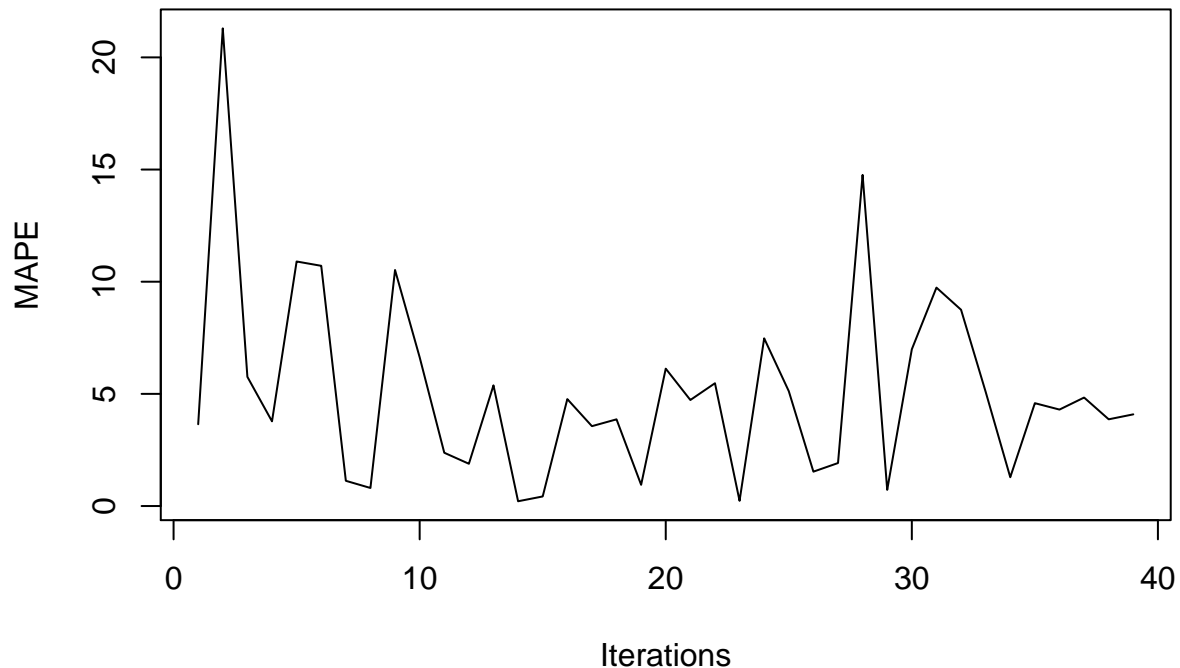
# Consumer Sentiments
c = 1
UCAPE1 = c()
UCAPE1 = as.data.frame(UCAPE1)
for (i in 39:1){
  estindex = i*12
  splitdata = splitTrainTest(sentiment.ts,length(sentiment.ts)-estindex)
  estset = splitdata$train
  predset = splitdata$test
  fit <- Arima(estset,order=c(2,1,1), include.drift = FALSE, seasonal=list(order=c(1,0,1)))
  fc <- forecast(fit,h=1) #forecast 1 step ahead
  j= accuracy(fc,predset)
  UCAPE1[c,1]=i
  UCAPE1[c,2]=j[2,5]
  c=c+1
}
plot(1:39,UCAPE1$V2,type='l',xlab="Iterations",ylab="MAPE",
main = "Recursive of Consumer Sentiment at 1-Step")
```

## Recursive of Consumer Sentiment at 1-Step



```
# Total Vehicles Sold
c = 1
UMAPE1 = c()
UMAPE1 = as.data.frame(UMAPE1)
for (i in 39:1){
  estindex = i*12
  splitdata = splitTrainTest(vehicle.ts,length(vehicle.ts)-estindex)
  estset = splitdata$train
  predset = splitdata$test
  fit <- Arima(estset,order=c(3,1,0), include.drift = TRUE, seasonal=list(order=c(2,0,1)))
  fc <- forecast(fit,h=1) #forecast 1 step ahead
  j= accuracy(fc,predset)
  UMAPE1[c,1]=i
  UMAPE1[c,2]=j[2,5]
  c=c+1
}
plot(1:39,UMAPE1$V2,type='l',xlab="Iterations",ylab="MAPE",
main = "Recursive of Total Vehicles Sales at 1-Step")
```

## Recursive of Total Vehicles Sales at 1-Step



The plots above represent the recursive backtest at now 1-step. Based on the plots, it appears there are some slightly different variations for our variables. Based on the 1-step, it appears that there are more variances in the the MAPE value for both variables, especially for total vehicle sales. The MAPE value is smaller, which signals that perhaps at 1-step forecasts, our variables are better suited to forecast at 1-step ahead.

(c)

Based on your findings above, does your model perform better at longer or shorter horizon forecasts?

Based on the plots we derived for both sections, we collectively agree that our variables do better at 1-step forecasts due to the overall MAPE value being lower in magnitude compared to the magnitude when we were forecasting 12-steps ahead. This conclusion makes logical sense since our accuracy to predict one period way will be be better than forecasting at a larger time horizon. The ability to predict Consumer Sentiment for the next period can be argued is easier to do and the same stance can be made for Total Vehicle Sales. However, forecasting a whole year ahead will be more complex due to the potential shocks that can occur, thus, these results support our position that our variables perform better at forecasting 1-step ahead. It should also be noted that it is possible that within one month a shock can occur, but the likelihood is higher for forecasting a year compared to one month.

The plots above represents a forecast of 12-steps ahead for both variables using a moving window backtesting scheme. What is interesting is the magnitude of the MAPE between our two variables. An interesting observation is that the MAPE for both consumer sentiment and total vehicle sales peaks at a value higher than 25. There are a few similarities in terms of structure of the MAPE calculation, but it appears the MAPE calculations for total vehicle sales has smaller values overall with occassional spikes compared to the consumer sentiment that has more spikes and for the most part overall higher values of MAPE.

The next two plots represents a forecast of 1-step ahead for both variables using a moving window backtesting scheme. Both variables experience a pronounce spike, but the magnitude of the spike is smaller for consumer sentiment at around 15 and total vehicle sales at around 25. The MAPE plot for consumer sentiment overall has lower MAPE, which is desired, but there is a lot more variation in the values of MAPE, which could be explained by the forecasting abilities of the Arima model we developed for the consumer

sentiment variable and also due to the aspect that we are using a moving window. The MAPE for total vehicle sales also can be argued has an overall good MAPE, but what is concerning is the pronounce spike towards the end, which was also the case for consumer sentiment, but it could be explained by the time horizon for forecasting using our models. As the time increases, so does the level of uncertainty, which MAPE captures that result for us.

(d)

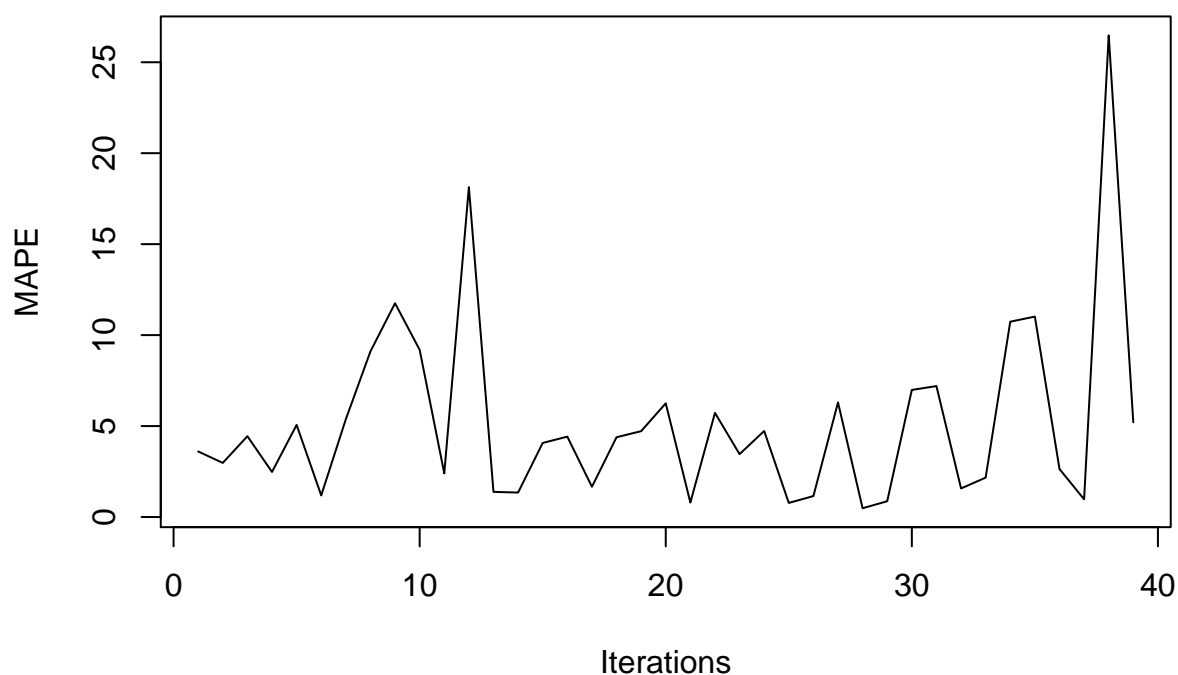
Now test your model using a moving window backtesting scheme. Forecast out 12-steps ahead at each iteration, and plot the forecast errors observed at each iteration. Repeat for a 1-step ahead forecast horizon. Provide plots of both.

```
# Step 12 Total Vehicles Sold
c = 1
UMAPE =c() #create empty vector to store MAPE
UMAPE = as.data.frame(UMAPE)
k=13
newdata = vehicle.ts

#loop
for (i in 39:1){
  estindex = length(newdata)-12
  splitdata = splitTrainTest(vehicle.ts, estindex) #split data into training and testing
  estset = splitdata$train #create estimation set
  predset = splitdata$test #create prediction set
  #fit model to estimation set
  fit <- Arima(estset,order=c(2,1,1), include.drift = T, seasonal=list(order=c(1,0,1)))
  fc <- forecast(fit,h=1) #forecast ahead 12 steps
  j= accuracy(fc,predset) #find MAPE
  #store MAPE
  UMAPE[c,1]=i
  UMAPE[c,2]=j[2,5]
  c=c+1

  #subset new data for new window
  newdata = subset(newdata,start=k,end=514) #setup data for next window
  k=k+12
}
plot(1:39,UMAPE$V2,type='l',xlab="Iterations",ylab="MAPE",
main = "Moving Window Total Vehicles Sales at 12-Step") #plot
```

## Moving Window Total Vehicles Sales at 12-Step



```
#Step 1, Total Vehicles Sold
c = 1
UMAPE = c() #create empty vector to store MAPE
UMAPE = as.data.frame(UMAPE)
k=13
newdata = vehicle.ts

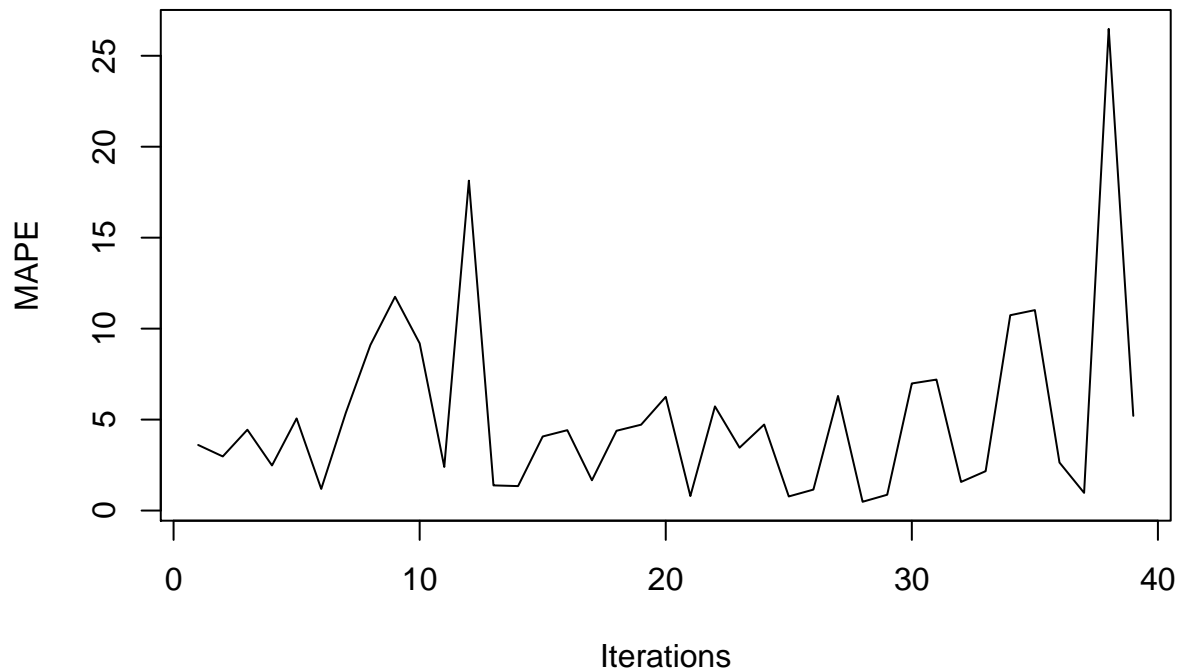
#loop
for (i in 39:1){
  estindex = length(newdata)-12
  splitdata = splitTrainTest(vehicle.ts, estindex) #split data into training and testing
  estset = splitdata$train #create estimation set
  predset = splitdata$test #create prediction set
  #fit model to estimation set
  fit <- Arima(estset,order=c(2,1,1), include.drift = T, seasonal=list(order=c(1,0,1)))
  fc <- forecast(fit,h=1) #forecast ahead 12 steps
  j= accuracy(fc,predset) #find MAPE
  #store MAPE
  UMAPE[c,1]=i
  UMAPE[c,2]=j[2,5]
  c=c+1

  #subset new data for new window
  newdata = subset(newdata,start=k,end=514) #setup data for next window
  k=k+12
}
plot(1:39,UMAPE$V2,type='l',xlab="Iterations",ylab="MAPE",main=
```



```
"Moving Window Total Vehicles Sales at 1-Step") #plot
```

## Moving Window Total Vehicles Sales at 1-Step



```
#Step 12, Consumer Sentiments
c = 1
UMAPE =c() #create empty vector to store MAPE
UMAPE = as.data.frame(UMAPE)
k=13
newdata = sentiment.ts

#loop
for (i in 39:1){
  estindex = length(newdata)-12
  splitdata = splitTrainTest(sentiment.ts, estindex) #split data into training and testing
  estset = splitdata$train #create estimation set
  predset = splitdata$test #create prediction set
  #fit model to estimation set
  fit <- Arima(estset,order=c(2,1,1), include.drift = T, seasonal=list(order=c(1,0,1)))
  fc <- forecast(fit,h=12) #forecast ahead 12 steps
  j= accuracy(fc,predset) #find MAPE
  #store MAPE
  UMAPE[c,1]=i
  UMAPE[c,2]=j[2,5]
  c=c+1

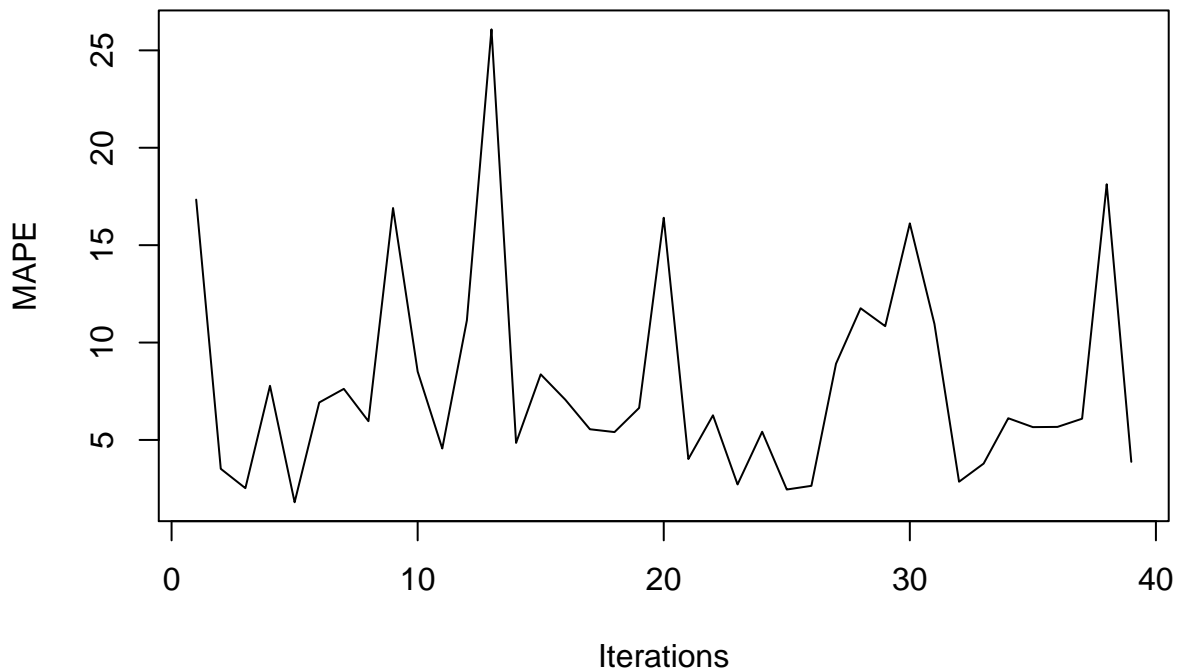
  #subset new data for new window
  newdata = subset(newdata,start=k,end=514) #setup data for next window
  k=k+12
```

```

}
plot(1:39, UMAPE$V2, type='l', xlab="Iterations", ylab="MAPE",
main = "Moving Window Consumer Sentiment at 12-Step") #plot

```

## Moving Window Consumer Sentiment at 12-Step



```

#Step 1, Consumer Sentiments
c = 1
UMAPE =c() #create empty vector to store MAPE
UMAPE = as.data.frame(UMAPE)
k=13
newdata = sentiment.ts

#loop
for (i in 39:1){
  estindex = length(newdata)-12
  splitdata = splitTrainTest(sentiment.ts, estindex) #split data into training and testing
  estset = splitdata$train #create estimation set
  predset = splitdata$test #create prediction set
  #fit model to estimation set
  fit <- Arima(estset,order=c(2,1,1), include.drift = T, seasonal=list(order=c(1,0,1)))
  fc <- forecast(fit,h=1) #forecast ahead 12 steps
  j= accuracy(fc,predset) #find MAPE
  #store MAPE
  UMAPE[c,1]=i
  UMAPE[c,2]=j[2,5]
  c=c+1

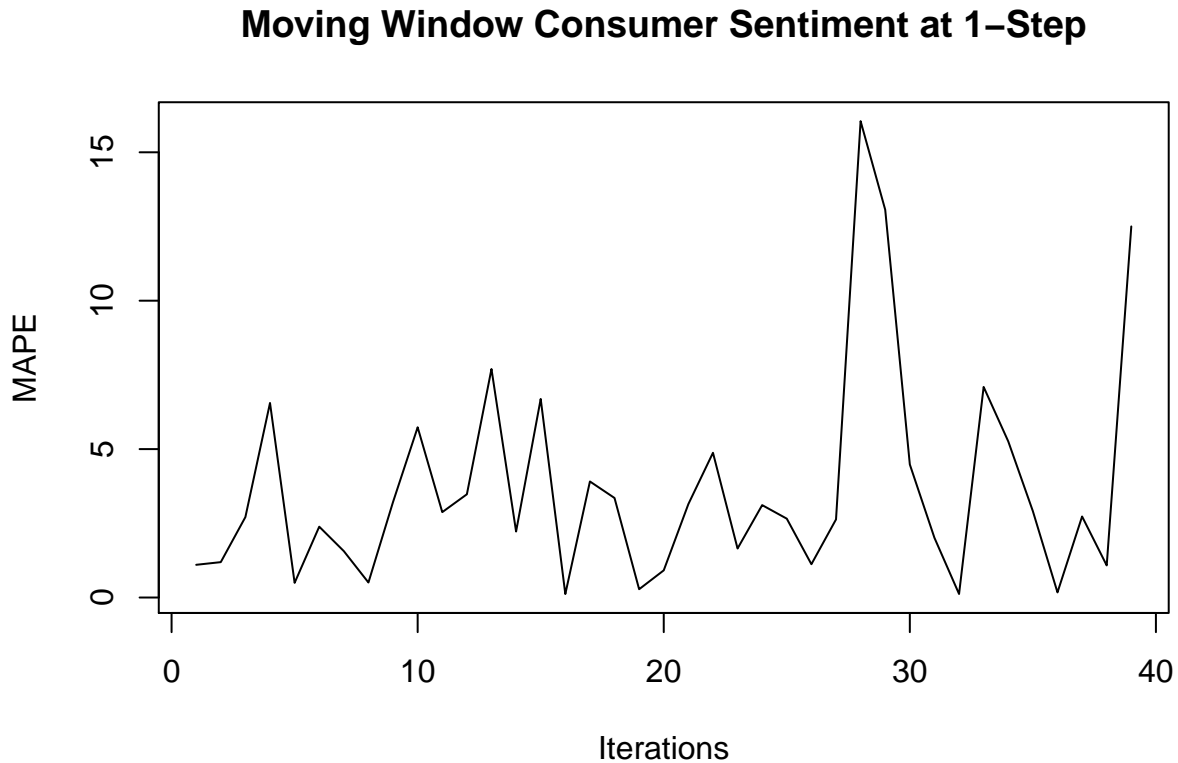
  #subset new data for new window
  newdata = subset(newdata,start=k,end=514) #setup data for next window
}

```

```

k=k+12
}
plot(1:39,UMAPE$V2,type='l',xlab="Iterations",ylab="MAPE",
main = "Moving Window Consumer Sentiment at 1-Step" ) #plot

```



(e)

How do the errors found using a recursive backtesting scheme compare with the errors observed using a moving average backtesting scheme? Which scheme showed higher errors overall, and what does that tell you about your model?

Based on comparing both backtesting schemes, one can argue that for the Arima model we derived for consumer sentiment, the results are different. When comparing 12-steps away for each scheme, the moving average exhibits more fluctuations, but the magnitudes are similar. At one-step, the results are different in terms of the magnitude. As the number of iterations increase, the moving backtesting scheme MAPE values increase, compared to the recursive backtesting where the large MAPE values are at the beginning and gradually begin to decline. However, the errors are smaller for the moving window scheme. For the Arima model for total vehicle sales at 12-steps, the moving backtesting scheme shows a more constant MAPE value compared to the recursive backtesting scheme that exhibit larger magnitude spikes, which could speak against its ability to forecast. At one-step, the MAPE calculation are smaller in terms of magnitude in the moving window scheme compared to the recursive backtesting scheme. The recursive backtest scheme seems to have larger magnitude spikes compared to the moving backtesting scheme, but can be argued as a lower aggregate error. Based on the results we derived, it suggests that our models experience varying levels of errors based on the time horizon. The Arima model for consumer sentiment states we should use the moving backtesting scheme and the model is better to forecast at one-step away. For the Arima model for total vehicle sales, it be argued that the model performs better forecasting one period ahead based on the results from both backtesting schemes. This conclusion is not surprising since forecasting using a larger time horizon is more difficult.

### III. Conclusion and Future Work

In conclusion, after developing, testing the robustness, and checking for the necessary conditions for effective time series forecasting models, we observed that both our ARIMA models and VAR model are in agreement on the direction of the forecasts for both Total Vehicle Sales and University of Michigan's Consumer Sentiment. Both ARIMA and VAR models state that total vehicle sales will increase in the future, while both models see consumer sentiment being leveled with minimal growth, if any, which is interesting since the notion of consumer sentiment influencing total vehicle sales has always been thought of being true and for our case was statistically supported based on the Granger-Causality test. Of course, it should be noted that the Granger-Causality doesn't mean that there is a casual relationship, but it states that we can use past values of consumer sentiment to forecast future sales of vehicles. With the disagreement in terms of the direction of the forecasts of both our variables, it reinforces the idea that there are other economic variables to consider and as our forecast increases, that level of uncertainty is apparent, which follows that forecasting time series has strengths, but also weaknesses.

This analysis once the pandemic is over should be observed again to see whether the lingering effects of the pandemic would greatly alter consumer behavior and ultimately confidence. Since Working From Home Orders are still occurring and for some employers making it permanent, we as a group would be interested to see if preferences shift away from vehicle sales to something completely different since the need to travel to work could no longer be required. This potential phenomena in our opinion is worth exploring in the near future.

### IV. References:

Both datasets were retrieved from FRED. Below are the website sources: University of Michigan's Consumer Sentiment: <https://fred.stlouisfed.org/series/UMCSENT> Total Vehicle Sales: <https://fred.stlouisfed.org/series/TOTALNSA>