Economics 144: Project 1

Yihao Li
Claudia Huey
Sijia Hua

April 25, 2019

1 Introduction

The data used in this project was the total number of motor vehicles sold in the United States, which was retrieved from the Federal Reserve Bank.[1] Detailed information is provided as following table.

Data Description	
Title	Total Vehicle Sales
Time Period	1976-01-01 - 2019-03-01
Frequency	Monthly
Units	Thousands of Units
Seasonal Adjusted	Not

Table 1: Source Data Information

Based on this regularly updated motor vehicle sales data, we simulated the trend and created seasonal dummies with a monthly cycle to forecast whether sales over the next two years(h = 24) will conform to the waveform fluctuations of historical data. According to the data, about every 10 years, starting from 1981, the United States has experienced a national recession, which is reflected by a large jump in the volatility of sales. In this project, we focus on forecasting the future value of motor vehicle sales and its performance in next several recession periods based on the forecasting model we constructed.

2 Results

2.1 Modeling and Forecasting Trend

(a) Time Series Plot
The time series plot of total vehicle sales and its ACF,PACF are shown in Figure 1.

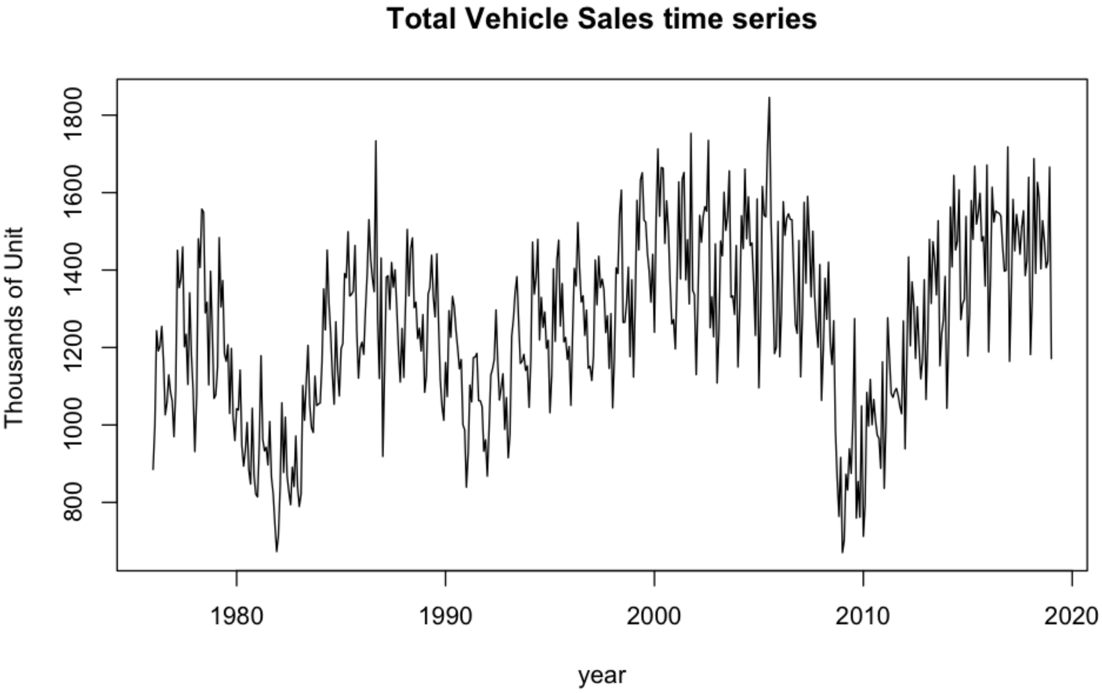


Figure 1: Time Series Display of Data Source

(b) Covariance Stationary Test
This time series is not covariance stationary. There is not constant mean or variance. It can also been seen from the ACF plot (Figure 1), that the autocovariance function is not stable.

(c) ACF and PACF
The fact that there are spikes in the PACF plot suggests that a linear model with three lags may serve as an appropriate representation of the time series.

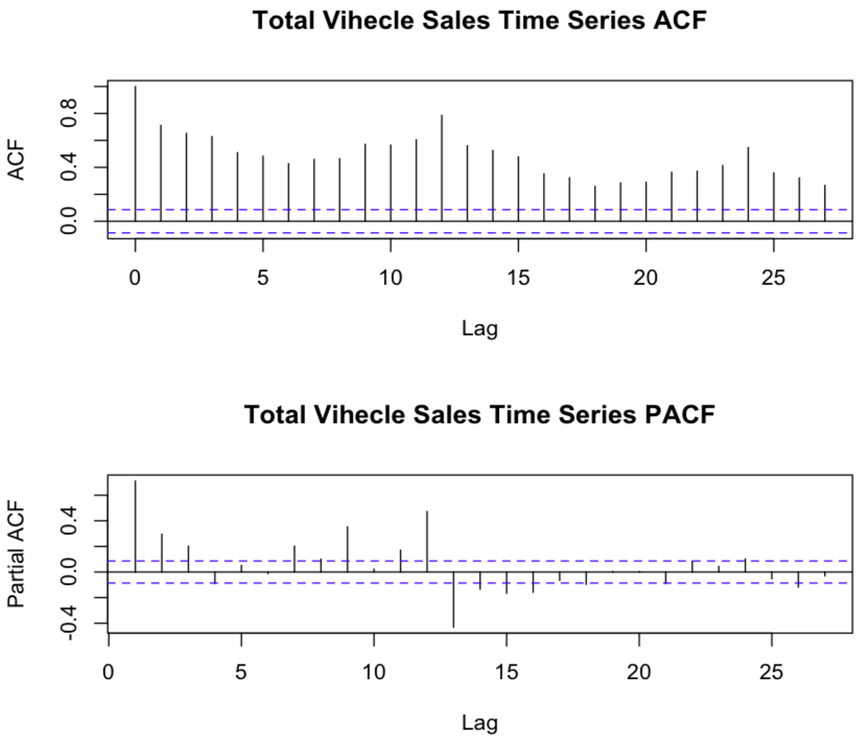


Figure 2: ACF and PACF plot of Data Source

(d) Liner and Non-linear Model
Linear with 3 lags:
Let X be the Total Vehicle Sales. We modeled the data with the following equation:

$$X_t = b_0 + b_1 X_{t-1} + b_2 X_{t-2} + b_3 X_{t-3} + u_t \tag{1}$$

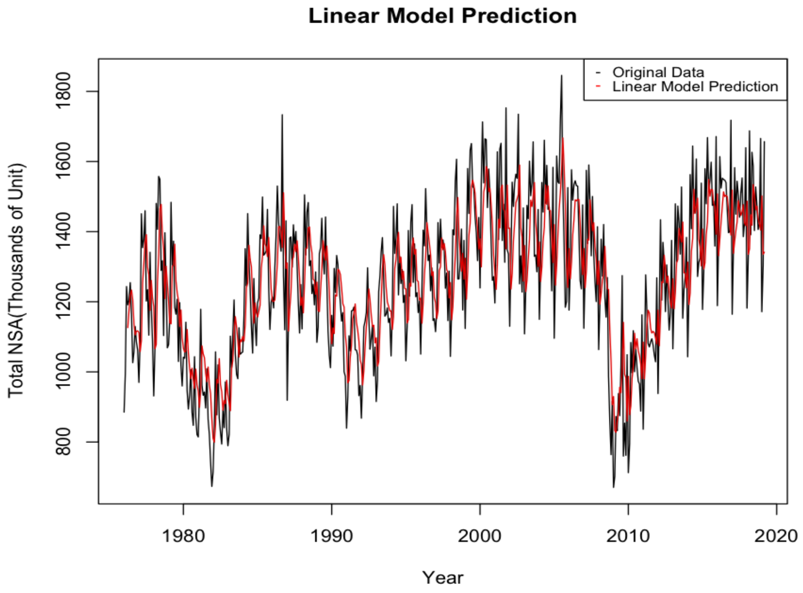


Figure 3: Original Data Plot and Linear Model Prediction

Nonlinear(exponential):
In equation(2),t is the time component, a and b are variables:

$$Y_t = e^{a+bt} \tag{2}$$

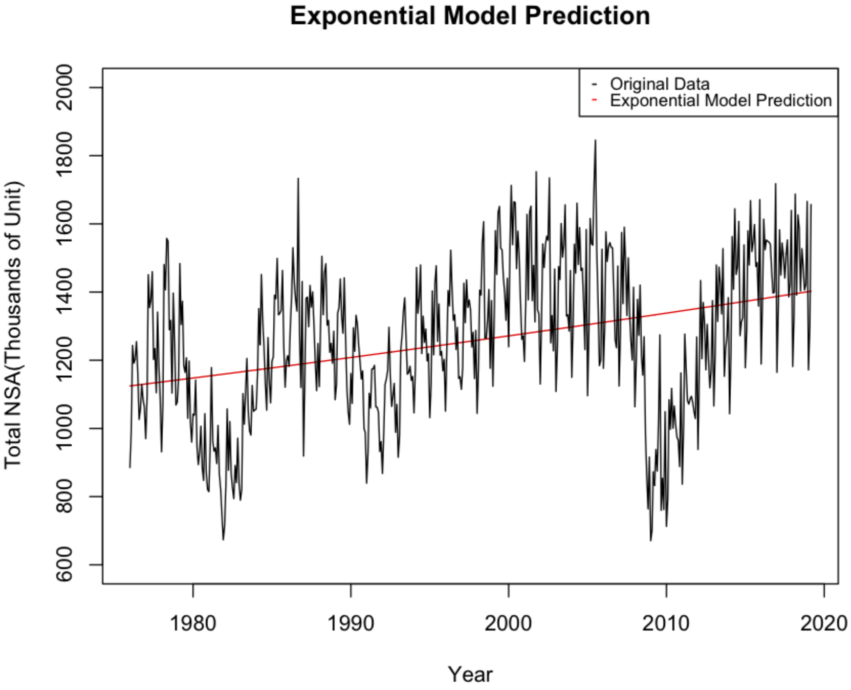


Figure 4: Original Data Plot and Exponential Model Prediction

(e) Residual and Fitted Values Plots
Linear:

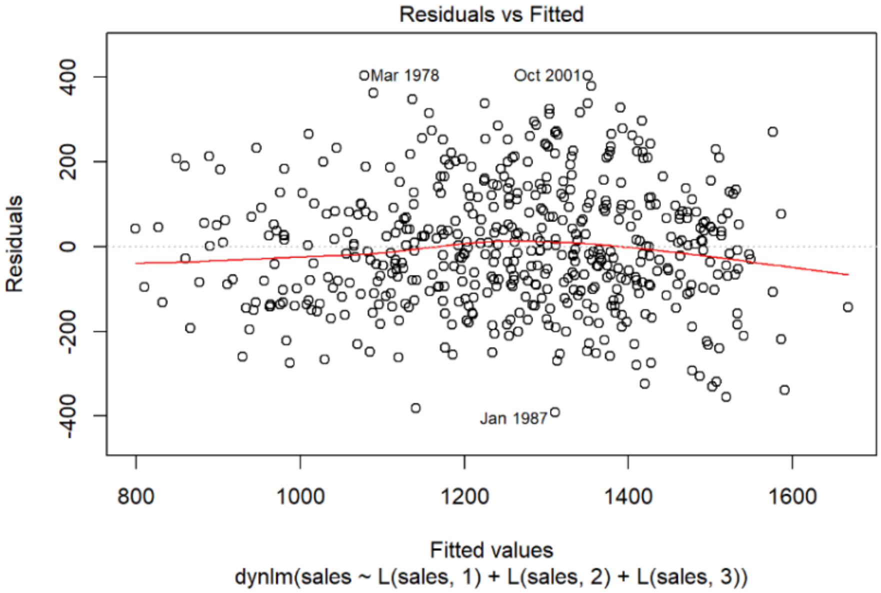


Figure 5: Residuals vs. Fitted Values for the Linear Model

Exponential:

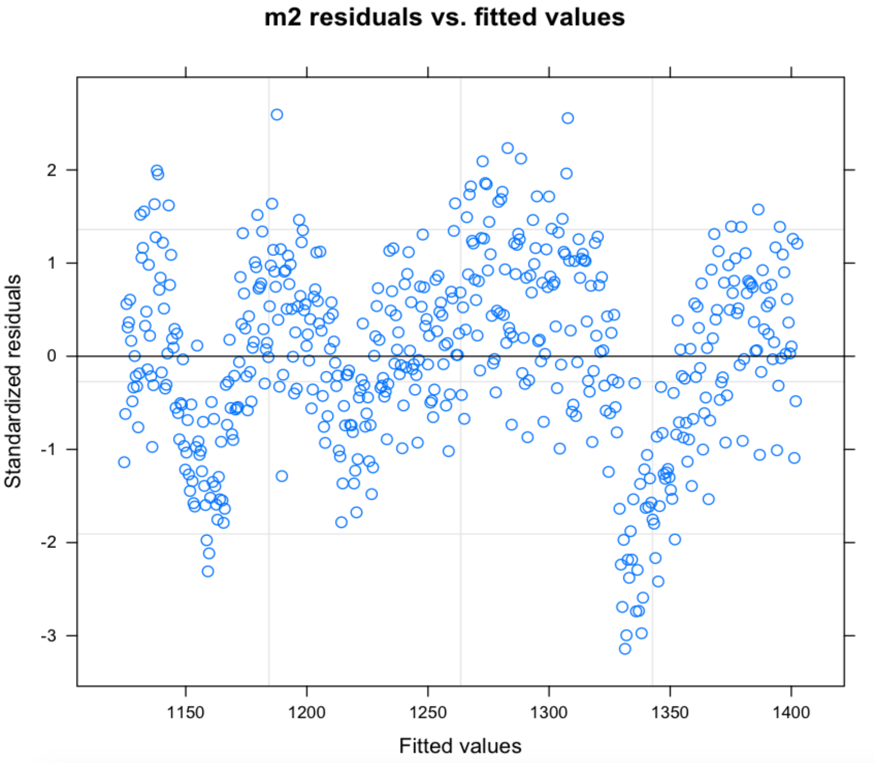


Figure 6: Residuals vs. Fitted Values for the Exponential Model

The residual plot for the linear model shows that there is almost perfectly normal scatter. The residuals for the exponential model, however, shows signs of a cyclical pattern, which suggests that the exponential model is not a good fit for this data.

(f) Histograms
Linear:

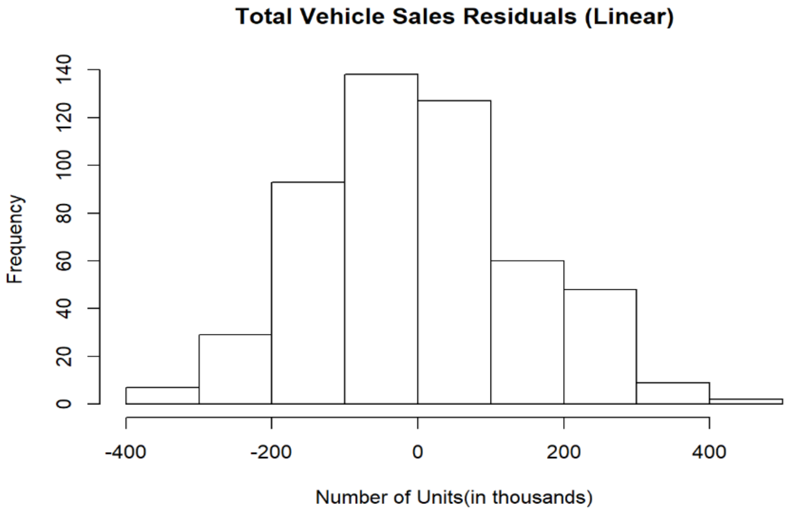


Figure 7: Histogram for residual of the Linear Model

Exponential:

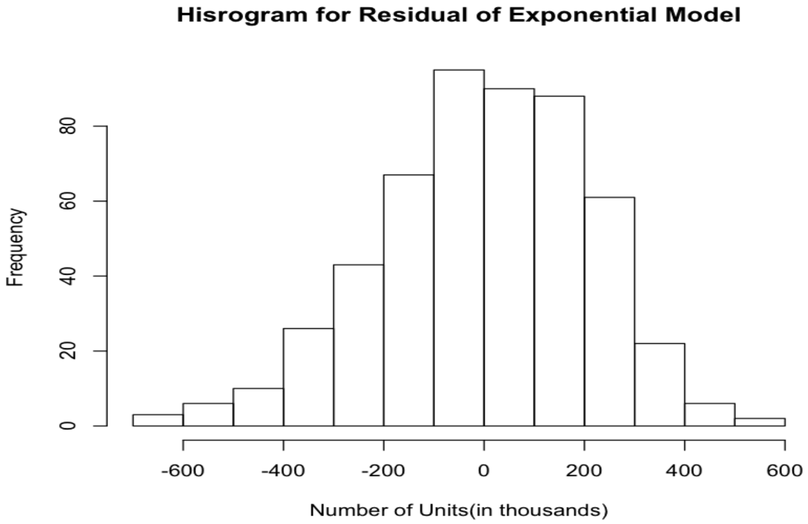


Figure 8: Histogram for residual of the Exponential Model

The histogram from the linear model is relatively normally distributed. The histogram from the exponential model is also relatively normally distributed, but there seems to be a slight left skew as well. When comparing the two histograms side by side, the residuals of the linear model seem to be closer to normal distribution than the residuals of the exponential model.

(g) The R-squared value from the linear model is around 0.5709, which is high enough to be considered acceptable. The F statistic is also small enough so that the null hypothesis stating that all the coefficients are equal to 0 can be rejected at any significance level. In our case, we chose a significant level of 0.01. For the exponential model, all the t-values are far greater than any usually accepted confidence interval, so we can reject the null hypothesis that any coefficient is 0 as well.

(h) AIC and BIC
The AIC for the linear model is 6608.876 and the BIC is 6634.341, while the AIC for the exponential model is 7002.632 with a BIC of 7015.376. Both the AIC and the BIC agree that the linear model is a better fit.

(i) Forecasting
In forecast function, only h=519 can be used, otherwise there would be Error in variables. We can zoom in the forecasting with a smaller step size (h=129) by using the code below instead of 101st and 102rd lines:

```
quartz("linear model(m1) forecasting")
plot(forecast(m1,h = 519,xlim=c(1976,2030)))
```

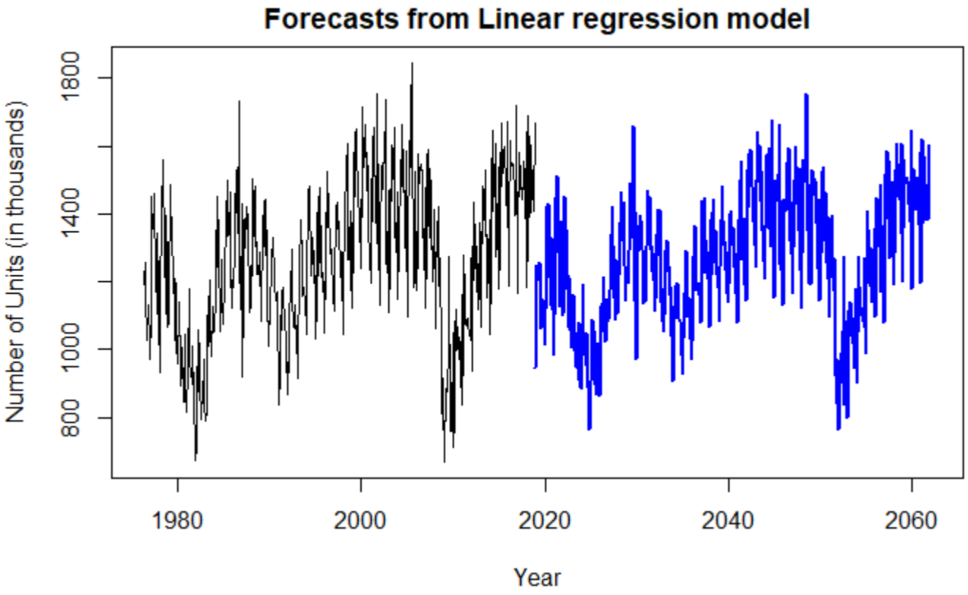


Figure 9: Forecast for the Linear Regression Model

2.2 Modeling and Forecasting Seasonality

(a) Constructing and Testing Seasonal Dummies

We constructed a set of seasonal dummies in order to represent monthly seasonality through the 519 × 12 matrix shown below.

$$\begin{aligned}
D_i1 &= \{0,1,0,0,0,0,0,0,0,0,0,0,0,1,0...\}\\
D_i2 &= \{0,0,1,0,0,0,0,0,0,0,0,0,0,0,1...\}\\
&...\\
D_i11 &= \{0,0,0,0,0,0,0,0,0,0,0,1,0,0,0...\}\\
D_i12 &= \{1,0,0,0,0,0,0,0,0,0,0,0,0,0,0...\}
\end{aligned}$$

Where as the row index i, from 1 to 519, indicates the number of entries, the column index j, from 1 to 12, indicates the 12 months in a year.

From Figure 10, it is observed that the residual plot does not look random; there are several outliers that are in different years at top and bottom. The normal Q-Q plot, which is a normal probability plot, provides a relatively straight line with no deviation points. This shows that the residuals are almost normally distributed. The Scale-Location graph does not look random as well. In the residuals vs. Leverage graph, there are no leverage points shown.

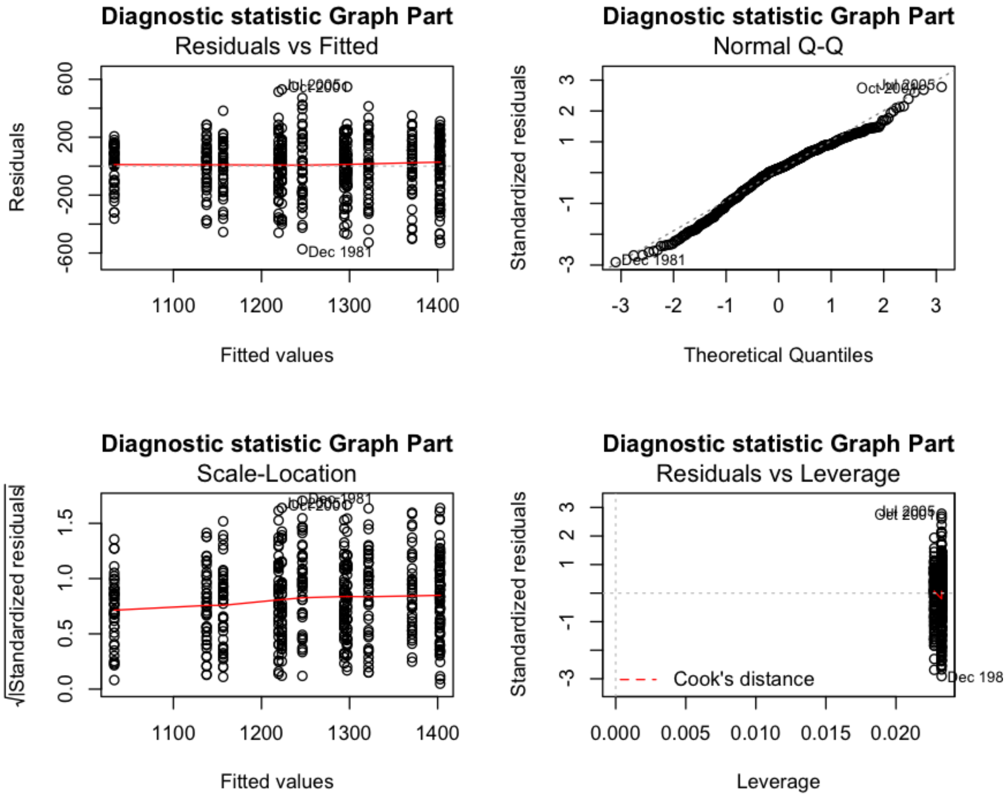
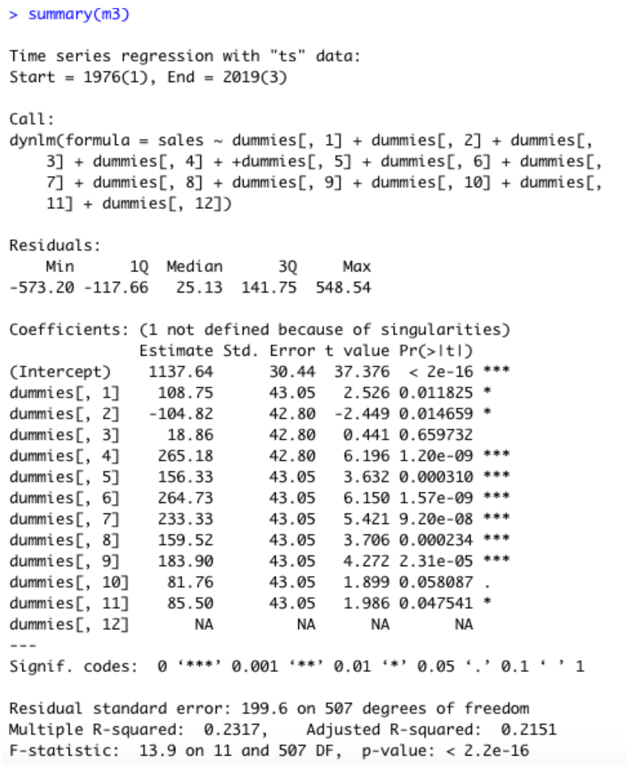


Figure 10: Diagnostic statistic Graph Part

Figure 11 provides specific information on this model, which consists purely of seasonal dummies. The variables dummies[, 3], dummies[, 10], and dummies[, 11] are not significant in this model; while dummies[, 12] is NA. It can be interpreted that dummies[, 12] is linearly related to the other variables.



```
> summary(m3)

Time series regression with "ts" data:
Start = 1976(1), End = 2019(3)

Call:
dynlm(formula = sales ~ dummies[, 1] + dummies[, 2] + dummies[,
    3] + dummies[, 4] + +dummies[, 5] + dummies[, 6] + dummies[,
    7] + dummies[, 8] + dummies[, 9] + dummies[, 10] + dummies[,
    11] + dummies[, 12])

Residuals:
    Min      1Q  Median      3Q     Max
-573.20 -117.66   25.13  141.75  548.54

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    1137.64      30.44  37.376  < 2e-16 ***
dummies[, 1]    108.75      43.05   2.526 0.011825 *
dummies[, 2]   -104.82      42.80  -2.449 0.014659 *
dummies[, 3]     18.86      42.80   0.441 0.659732
dummies[, 4]    265.18      42.80   6.196 1.20e-09 ***
dummies[, 5]    156.33      43.05   3.632 0.000310 ***
dummies[, 6]    264.73      43.05   6.150 1.57e-09 ***
dummies[, 7]    233.33      43.05   5.421 9.20e-08 ***
dummies[, 8]    159.52      43.05   3.706 0.000234 ***
dummies[, 9]    183.90      43.05   4.272 2.31e-05 ***
dummies[, 10]    81.76      43.05   1.899 0.058087 .
dummies[, 11]    85.50      43.05   1.986 0.047541 *
dummies[, 12]       NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 199.6 on 507 degrees of freedom
Multiple R-squared:  0.2317,    Adjusted R-squared:  0.2151
F-statistic:  13.9 on 11 and 507 DF,  p-value: < 2.2e-16
```

Figure 11: Diagnostic statistic Numerical Part

(b) Seasonal Factor Plot

Figure 13 is the plot of the coefficients in the pure seasonal dummy model we constructed in (a). The intercept seems to have the largest value, which may affect our prediction, but the intercept is not considered a seasonal factor. Besides the intercept, March and May are in the peak of dummies, while after May, the dummies' influence decays until November in the same year.

Note: actually we considered to set intercept = 0 to obtain a pure seasonal dummy model and we obtained adjusted R-squared 0.99 with all variables significant. However, it behaves as if the value of the removed intercept is equally assigned to each variable. Due to the possible issue of biased prediction, we decided to leave intercept in this pure seasonal dummy model.

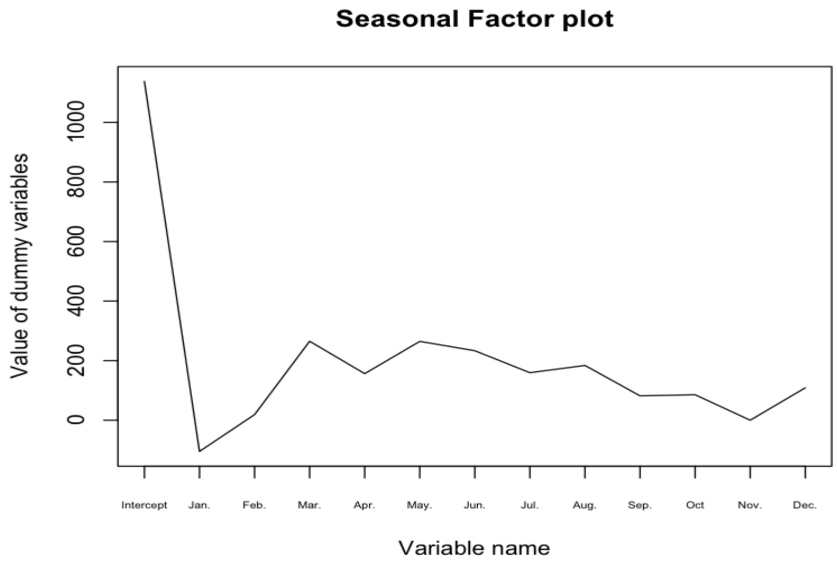


Figure 12: Seasonal Factor Plot

(c) Model Improvement

After combining the previous linear model(equation(1)) with the pure seasonal dummies, the model was improved by omitting dummies[,10] due to its lack of significance, which can be seen in Figure 14. As the intercept is not statistically significant (as shown in Figure 15), we chose to delete the intercept factor for achieving a higher Adjusted R Square($\sim$ 99.4%) in Figure 16. We chose this as the full model to discuss in the following subsections.

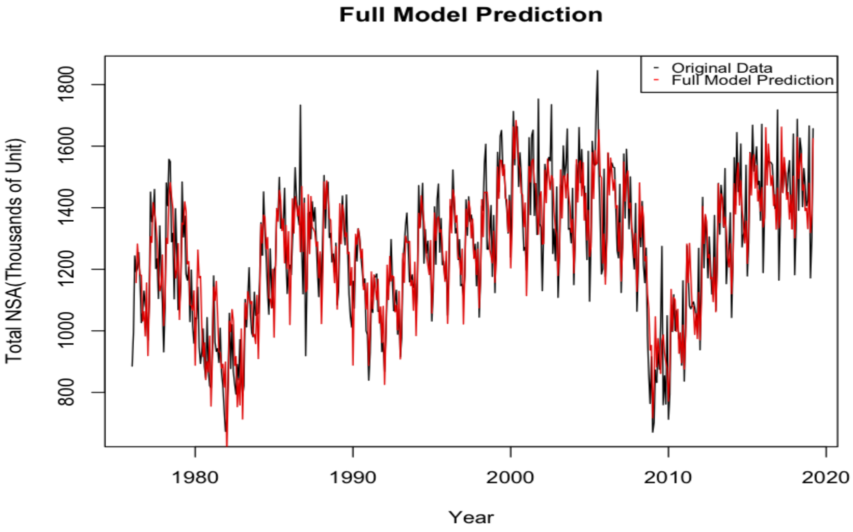
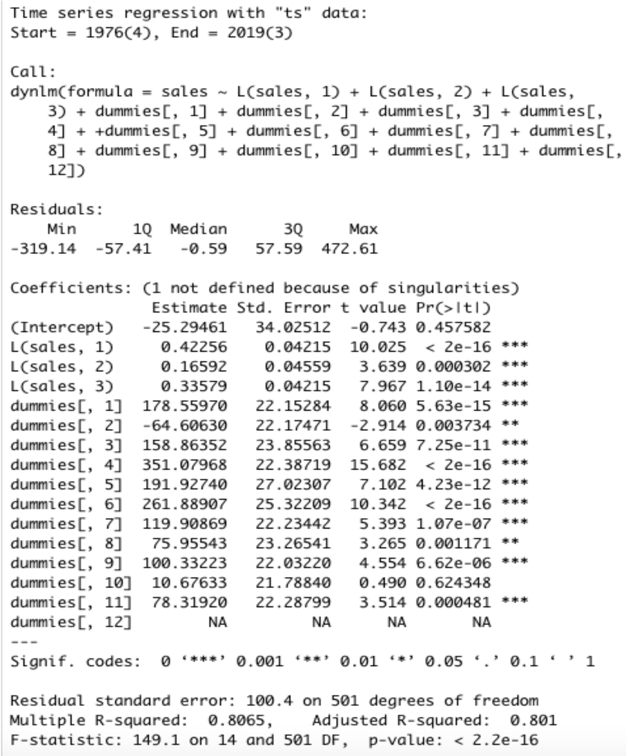


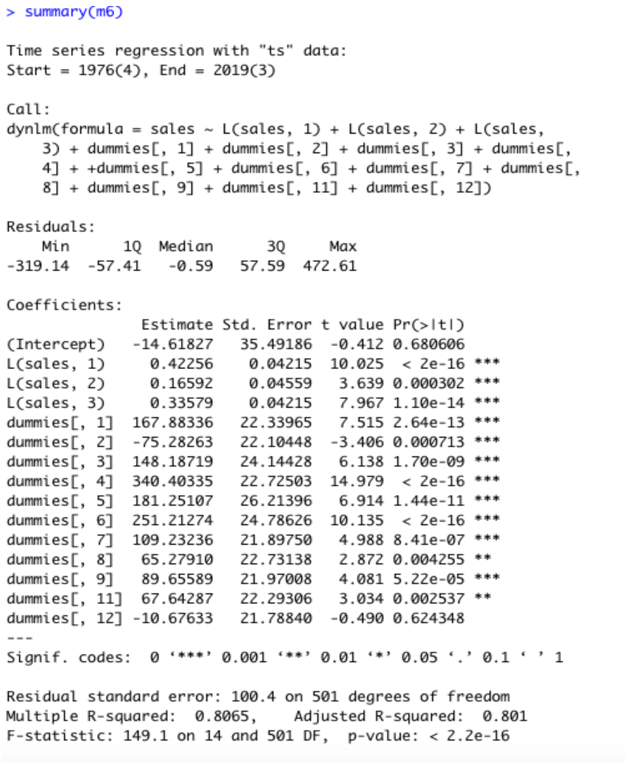
Figure 13: Original Data Plot and Full Model Prediction

In the fitted value vs. residual graph of the full model(Figure 17), the residual bounce randomly around 0, but with several outliers. The relationship of variables in the full model is reasonable. There is a horizontal bond formed by residuals, which indicates that the variances of error terms are nearly equal.



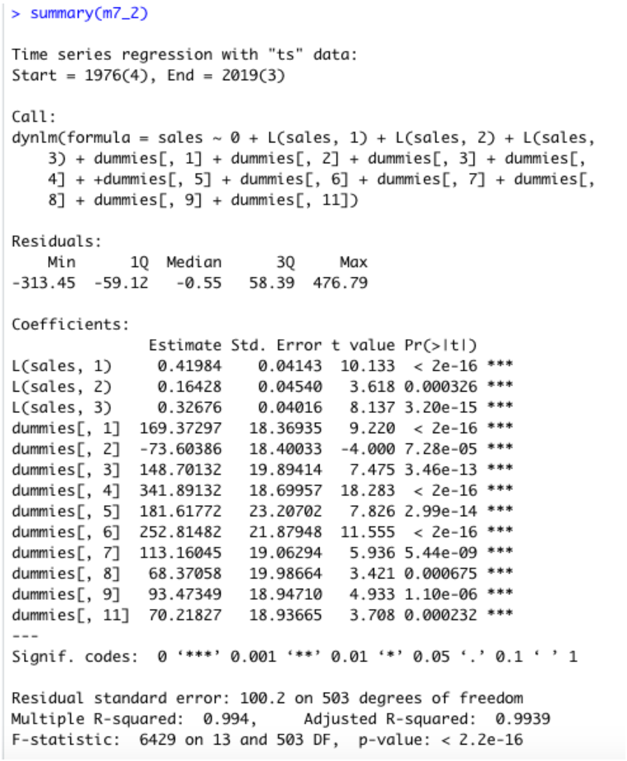
```
Time series regression with "ts" data:
Start = 1976(4), End = 2019(3)

Call:
dynlm(formula = sales ~ L(sales, 1) + L(sales, 2) + L(sales,
    3) + dummies[, 1] + dummies[, 2] + dummies[, 3] + dummies[,
    4] + +dummies[, 5] + dummies[, 6] + dummies[, 7] + dummies[,
    8] + dummies[, 9] + dummies[, 10] + dummies[, 11] + dummies[,
    12])

Residuals:
    Min      1Q  Median      3Q     Max
-319.14  -57.41   -0.59   57.59  472.61

Coefficients: (1 not defined because of singularities)
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -25.29461   34.02512  -0.743 0.457582
L(sales, 1)     0.42256    0.04215  10.025  < 2e-16 ***
L(sales, 2)     0.16592    0.04559   3.639 0.000302 ***
L(sales, 3)     0.33579    0.04215   7.967 1.10e-14 ***
dummies[, 1]  178.55970   22.15284   8.060 5.63e-15 ***
dummies[, 2]  -64.60630   22.17471  -2.914 0.003734 **
dummies[, 3]  158.86352   23.85563   6.659 7.25e-11 ***
dummies[, 4]  351.07968   22.38719  15.682  < 2e-16 ***
dummies[, 5]  191.92740   27.02307   7.102 4.23e-12 ***
dummies[, 6]  261.88907   25.32209  10.342  < 2e-16 ***
dummies[, 7]  119.90869   22.23442   5.393 1.07e-07 ***
dummies[, 8]   75.95543   23.26541   3.265 0.001171 **
dummies[, 9]  100.33223   22.03220   4.554 6.62e-06 ***
dummies[, 10]  10.67633   21.78840   0.490 0.624348
dummies[, 11]  78.31920   22.28799   3.514 0.000481 ***
dummies[, 12]        NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 100.4 on 501 degrees of freedom
Multiple R-squared:  0.8065,	Adjusted R-squared:  0.801
F-statistic: 149.1 on 14 and 501 DF,  p-value: < 2.2e-16
```

Figure 14: Summary of Model consists of linear and all seasonal dummies



```
> summary(m6)

Time series regression with "ts" data:
Start = 1976(4), End = 2019(3)

Call:
dynlm(formula = sales ~ L(sales, 1) + L(sales, 2) + L(sales,
    3) + dummies[, 1] + dummies[, 2] + dummies[, 3] + dummies[,
    4] + +dummies[, 5] + dummies[, 6] + dummies[, 7] + dummies[,
    8] + dummies[, 9] + dummies[, 11] + dummies[, 12])

Residuals:
    Min      1Q  Median      3Q     Max
-319.14  -57.41   -0.59   57.59  472.61

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -14.61827   35.49186  -0.412 0.680606
L(sales, 1)     0.42256    0.04215  10.025  < 2e-16 ***
L(sales, 2)     0.16592    0.04559   3.639 0.000302 ***
L(sales, 3)     0.33579    0.04215   7.967 1.10e-14 ***
dummies[, 1]  167.88336   22.33965   7.515 2.64e-13 ***
dummies[, 2]  -75.28263   22.10448  -3.406 0.000713 ***
dummies[, 3]  148.18719   24.14428   6.138 1.70e-09 ***
dummies[, 4]  340.40335   22.72503  14.979  < 2e-16 ***
dummies[, 5]  181.25107   26.21396   6.914 1.44e-11 ***
dummies[, 6]  251.21274   24.78626  10.135  < 2e-16 ***
dummies[, 7]  109.23236   21.89750   4.988 8.41e-07 ***
dummies[, 8]   65.27910   22.73138   2.872 0.004255 **
dummies[, 9]   89.65589   21.97008   4.081 5.22e-05 ***
dummies[, 11]  67.64287   22.29306   3.034 0.002537 **
dummies[, 12] -10.67633   21.78840  -0.490 0.624348
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 100.4 on 501 degrees of freedom
Multiple R-squared:  0.8065,	Adjusted R-squared:  0.801
F-statistic: 149.1 on 14 and 501 DF,  p-value: < 2.2e-16
```

Figure 15: Summary of Model after deleting dummy[,10]



```
> summary(m7_2)

Time series regression with "ts" data:
Start = 1976(4), End = 2019(3)

Call:
dynlm(formula = sales ~ 0 + L(sales, 1) + L(sales, 2) + L(sales,
    3) + dummies[, 1] + dummies[, 2] + dummies[, 3] + dummies[,
    4] + +dummies[, 5] + dummies[, 6] + dummies[, 7] + dummies[,
    8] + dummies[, 9] + dummies[, 11])

Residuals:
    Min      1Q  Median      3Q     Max
-313.45  -59.12   -0.55   58.39  476.79

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
L(sales, 1)     0.41984    0.04143  10.133  < 2e-16 ***
L(sales, 2)     0.16428    0.04540   3.618 0.000326 ***
L(sales, 3)     0.32676    0.04016   8.137 3.20e-15 ***
dummies[, 1]  169.37297   18.36935   9.220  < 2e-16 ***
dummies[, 2]  -73.60386   18.40033  -4.000 7.28e-05 ***
dummies[, 3]  148.70132   19.89414   7.475 3.46e-13 ***
dummies[, 4]  341.89132   18.69957  18.283  < 2e-16 ***
dummies[, 5]  181.61772   23.20702   7.826 2.99e-14 ***
dummies[, 6]  252.81482   21.87948  11.555  < 2e-16 ***
dummies[, 7]  113.16045   19.06294   5.936 5.44e-09 ***
dummies[, 8]   68.37058   19.98664   3.421 0.000675 ***
dummies[, 9]   93.47349   18.94710   4.933 1.10e-06 ***
dummies[, 11]  70.21827   18.93665   3.708 0.000232 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 100.2 on 503 degrees of freedom
Multiple R-squared:  0.994,     Adjusted R-squared:  0.9939
F-statistic:  6429 on 13 and 503 DF,  p-value: < 2.2e-16
```

Figure 16: Summary of model after deleting dummy[,10] and intercept

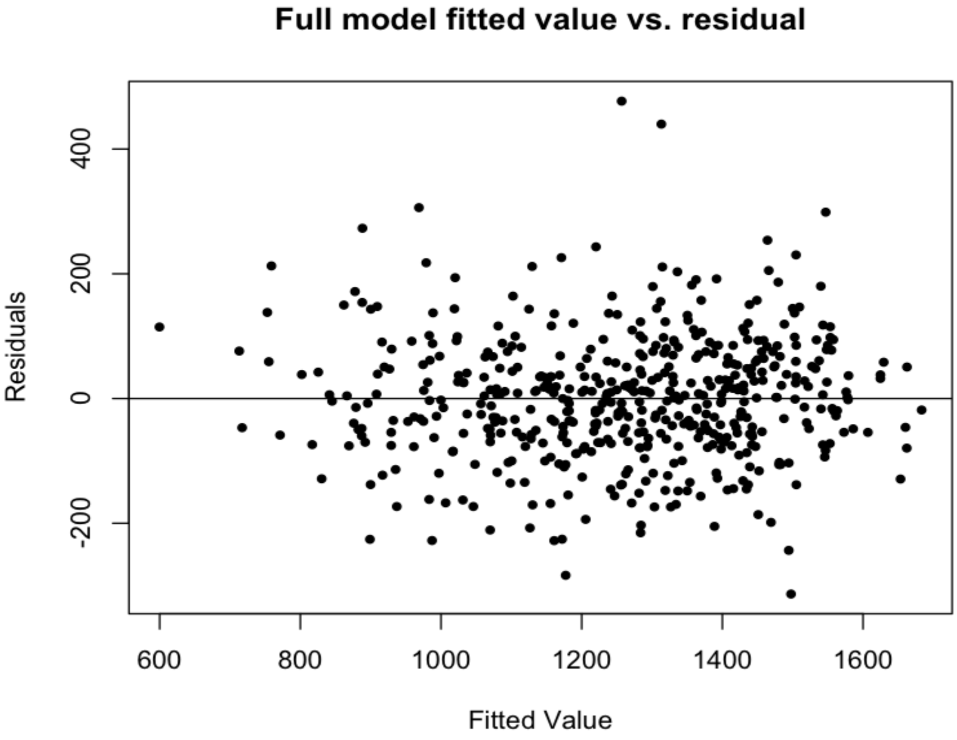


Figure 17: Full model Residuals vs. Fitted Value Plot

(d) Interpret Summary Statistic and Error Matrics

The final statistic result is in Figure 16, all variables are of significance and with a high Adjusted R Square. By look at table 2, even though they improved compared to previous models as we discussed above, AIC, BIC and MSE all perform poorly in this full model(Because we haven't discuss cycles in lectures).

Error Metrics for full model	
AIC	6234.185
BIC	6293.63
MSE	9794.861

Table 2: Error Metrics for full model

(e) Full Model Forecasting

In forecast function, only h=519 can be used, otherwise there would be Error in variables. We can focus on the forecasting with a smaller step size (h=129) by using the code below instead of 195th and 196th lines:

```
quartz()
plot(forecast(m7,h = 519,xlim=c(1976,2030)))
```

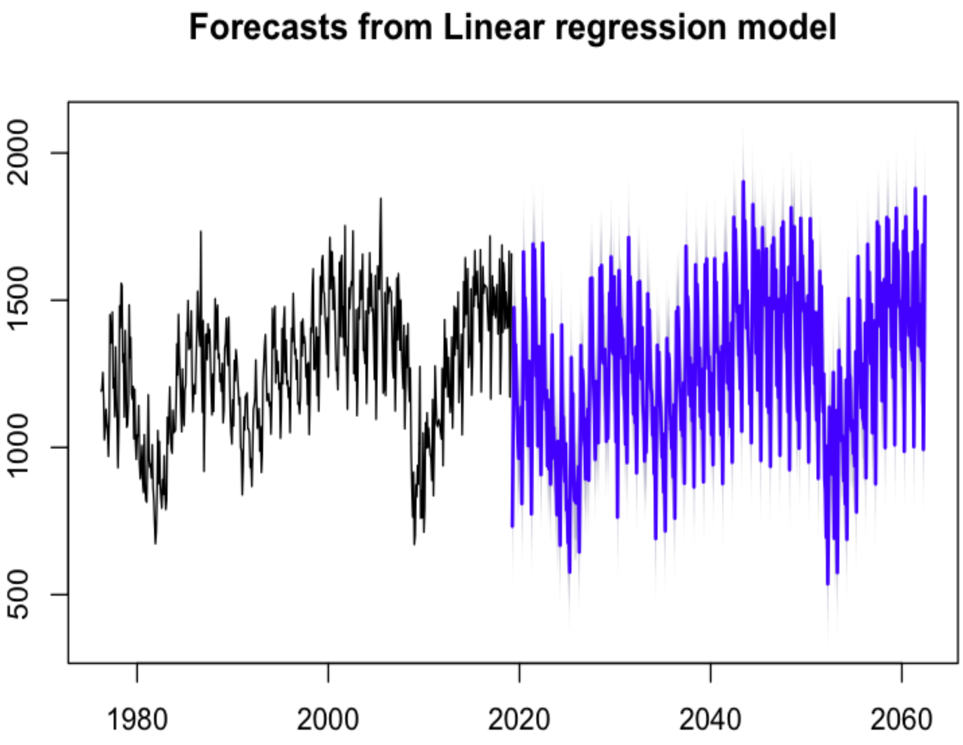


Figure 18: Forecasting with full model with h=519

2.3 Conclusions and Future Work

With the addition of seasonal dummies, the R-squared value improved from the 0.5709 seen in the original linear model, to a very impressive 0.9939. In other words, the new linear model is a near perfect fit for our time series. However, by looking at the ACF and PACF of the residuals shown in the figure below, we notice that there are still spikes in the plots. In other words, there is still room for improvement.

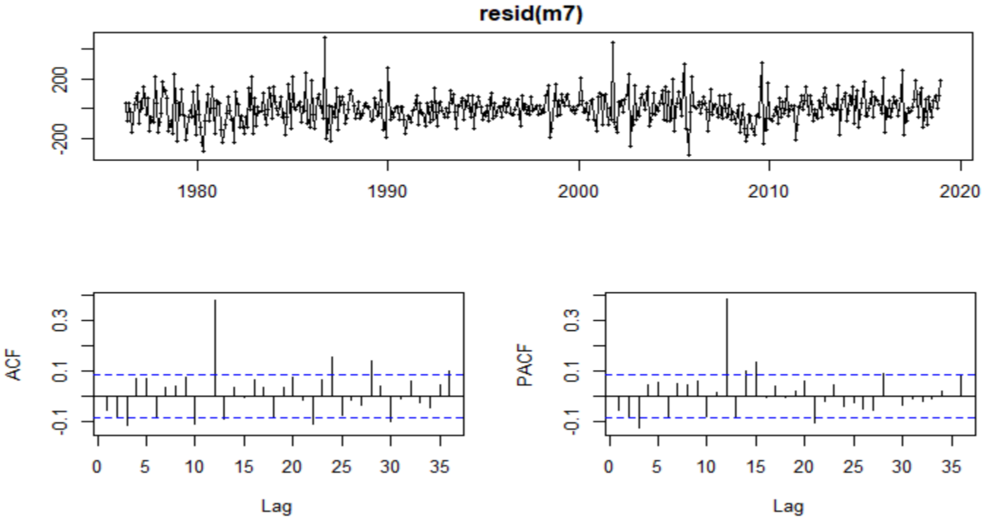


Figure 19: ACF and PACF for the Seasonal Model

When plotting the original time series, the first thing that was noticed was the shape of the trend; in particular, the cycles seen throughout the data. When fitting the exponential model, the cyclical shape was seen in the residual plot as well. Even in the residuals for the seasonally adjusted model, instead of taking on the form of white noise, there is a lingering trace of the original shape. In future works, we may add a cyclical attribute to our model to account for it, and consider using an ARIMA model.

References

[1] U.S. Bureau of Economic Analysis. Total vehicle sales [totalnsa]. April 22, 2019. retrieved from FRED, Federal Reserve Bank of St. Louis https://fred.stlouisfed.org/series/TOTALNSA.

3 R Source code

```r
## R SETUP
library(lubridate)
library(zoo)
library(dynlm)
library(forecast)
library(minpack.lm)
library(numbers)
#read in the data
Data = read.csv("TOTALNSA.csv", header=TRUE, sep=",")

#decode the date
Data$DATE = as.Date(Data$DATE,"%Y-%m-%d")
ts_TOTALNSA = ts(data = Data$TOTALNSA,1976,2019,freq=12)

#II.1
#(a) Show a time-series plot of your data.
quartz()
plot(ts_TOTALNSA,main="Total Vehicle Sales time series",xlab="year",ylab="Thousands of Unit ")

#(b) Does your plot in (a) suggest that the data are covariance stationary? Explain your answer.
#The data points are not covariance stationary because they are not sharing the same mean.

#(c) Plot and discuss the ACF and PACF of your data.

Sales_acf<-acf(Data$TOTALNSA)
Sales_pacf<-pacf(Data$TOTALNSA)
quartz()
par(mfrow=c(2,1))
plot(Sales_acf,main="Total Vihecle Sales Time Series ACF")
plot(Sales_pacf,main="Total Vihecle Sales Time Series PACF")


#(d) Fit a linear and nonlinear (e.g., polynomial, exponential, quadratic + periodic, etc.) model to your series.
# In one window, show both figures of the original times series plot with the respective fit.

#Linear(3 Lag):
sales = ts(Data$TOTALNSA, start = 1976, frequency = 12)
m1=dynlm(sales~L(sales, 1)+L(sales, 2)+L(sales, 3))
t1<-seq(1976, 2019,length=length(m1$fit))
quartz("Linear")
plot(sales, main="Linear Model Prediction", ylab = "Total NSA(Thousands of Unit)", xlab = "Year", lwd = 1, xlim = c(1976,2019))
lines(m1$fitted.values,col = "red3", lwd = 1)
legend("topright",pch= c("-","-"),legend=c("Original Data", "Linear Model Prediction"), col=c("black", "red"), cex=0.8)
summary(m1)

```

```r
AIC(m1)
BIC(m1)

#Exponential:
t2 = seq(2917,2019,length = length(sales))
ds = data.frame(x = t, y = sales)
m2 = nlsLM(y ~ exp(a+b*t),data = ds, start = list(a = 0, b = 0))
quartz("plots exp")
plot(x = Data$DATE, y= m2$m$fitted()[1:519], type = "l", main="Exponential Model Prediction",col = "red3",ylab = "Total NSA(Thousands of Unit)", xlab = "Year", ylim = c(600,2000))
lines(x = Data$DATE, y = sales, lwd = 1)
legend("topright",pch= c("-","-"),legend=c("Original Data", "Exponential Model Prediction"),col=c("black", "red"), cex=0.8)

quartz("Exponential Model residuals vs. fitted values")
plot(m2)
summary(m2)

#(e) For each model, plot the respective residuals vs. fitted values and discuss your observations.

#for linear:
quartz("e_linear")
plot(x = m1$fitted.values, y = m1$residuals, type = "p", pch = 20, main = "fitted value vs. residual",xlab = "Fitted Value", ylab = "Residuals")
abline(a = 0, b = 0)

#for Exponential:
quartz("e_Exp")
plot(x = m2$m$fitted(), y = m2$m$resid(), type = "p", pch = 20, main = "fitted value vs. residual", xlab = "Fitted Value", ylab = "Residuals")
abline(a = 0, b = 0)

#There’s no obvious shape and generally symmetrically distributed around 0 in both graph plotting
#Thus the variables is linearly related and generally homoscedastic.

#(f) For each model, plot a histogram of the residuals and discuss your observations.
quartz("m1 residuals histogram")
hist(m1$residuals,main="Hisrogram for Residual of Linear Model",xlab="Number of Units(in thousands)")
quartz("m2 residuals histogram")
hist(m2$m$resid(),main="Hisrogram for Residual of Exponential Model",xlab="Number of Units(in thousands)")
#From the histogram, the residuals has a relatively good normal distribution

#(g) For each model, discuss the associated diagnostic statistics (R2 , tdistribution, Fdistribution, etc.)
summary(m1)
#Multiple R-squared: 0.5734, Adjusted R-squared: 0.5709 which is quite acceptable
#t-distribution: all the t-value is far greater than any usually accepted confidence interval,
# so we can reject the null hypothesis that any coefficient is 0
```

```
#F-statistic: 229.4 on 3 and 512 DF, p-value: < 2.2e-16 which is quite acceptable
summary(m2)
#t-distribution: all the t-value is far greater than any usually accepted confidence interval,
# so we can reject the null hypothesis that any coefficient is 0
#The other two data are not provided

#(h) Select a trend model using AIC and one using BIC (show the values obtained from each criterion). Do the selected models agree?
AIC(m1,m2)
BIC(m1,m2)
#Because of the trend, both model have extremely large AIC and BIC, but linear model performs relatively better

#(i) Use your preferred model to forecast h-steps (at least 16) ahead. Your forecast should include the respective uncertainty prediction interval. Depending on your data, h will be in days, months, years, etc.
quartz("m1 forecasting")
plot(forecast(m1,h = 519))

##II.2
#(a) Construct and test (by looking at the diagnostic statistics) a model with a full set of seasonal dummies.
#construct the dummy set, each column is a dummy variable

# Construct seasonal dummies
dummies = matrix(nrow = length(Data$TOTALNSA), ncol = 12)
for (i in 1:12)
{
  for(j in 1:519)
  {
    dummies[j,i] = as.integer(mod(j,12) == i-1)
  }
}
# Construct model with full seasonal dummies
m3=dynlm(sales~dummies[,1]+dummies[,2]+dummies[,3]+dummies[,4]++dummies[,5]+dummies[,6]+dummies[,7]+dummies[,8]+dummies[,9]+dummies[,10]+dummies[,11]+dummies[,12])
#Diagnostic Statistics
quartz("Model diagnostic statistic with seasonal dummies")
par(mfrow=c(2,2))
plot(m3,main="Diagnostic statistic Graph Part")
# numerical diagnostic statistic
summary(m3)
AIC(m3)

#(b) Plot the estimated seasonal factors and interpret your plot.
coef_sdummy <-m3$coefficients
# rearrange order
Dec_sdummy<-coef_sdummy[2]
coef_sdummy <- coef_sdummy[-c(2)]
coef_sdummy[12]<-0 # change na to 0
coef_sdummy <-c(coef_sdummy,Dec_sdummy)
x<-c(1:13) # x-axis
# x-axis name
```

```
x_name <- c("Intercept","Jan.","Feb.","Mar.","Apr.","May.","Jun.","Jul.","Aug.","Sep.","Oct","Nov.","Dec.")
# plot seasonal factor
quartz("seasonal dummies plot")
plot(x,coef_sdummy,xaxt="none",type="l",main="Seasonal Factor plot",ylab="Value of dummy variables",xlab="Variable name")
axis(1, at=1:13, labels=x_name, cex.axis=0.5)

#(c)
m5=dynlm(sales~L(sales, 1)+L(sales, 2)+L(sales, 3)+dummies[,1]+dummies[,2]+dummies[,3]+dummies[,4]++dummies[,5]+dummies[,6]+dummies[,7]+dummies[,8]+dummies[,9]+dummies[,10]+dummies[,11]+dummies[,12])
summary(m5)
AIC(m5)

#have dummies[,10](insignificant) removed
m6=dynlm(sales~L(sales, 1)+L(sales, 2)+L(sales, 3)+dummies[,1]+dummies[,2]+dummies[,3]+dummies[,4]++dummies[,5]+dummies[,6]+dummies[,7]+dummies[,8]+dummies[,9]+dummies[,11]+dummies[,12])
summary(m6)
AIC(m6)

#remove insignificant factor
#Alex's note:
#I hesitate for a while if we should delete the intercept,
#for it's a linear model. If I keep it
m7_1=dynlm(sales~L(sales, 1)+L(sales, 2)+L(sales, 3)+dummies[,1]+dummies[,2]+dummies[,3]+dummies[,4]++dummies[,5]+dummies[,6]+dummies[,7]+dummies[,8]+dummies[,9]+dummies[,11])
summary(m7_1)
AIC(m7_1)

#But if I removed the intercept the R^2 is greatly improved(.99 R^2...)
m7_2=dynlm(sales~0+L(sales, 1)+L(sales, 2)+L(sales, 3)+dummies[,1]+dummies[,2]+dummies[,3]+dummies[,4]++dummies[,5]+dummies[,6]+dummies[,7]+dummies[,8]+dummies[,9]+dummies[,11])
summary(m7_2)
AIC(m7_2)

#For the sake of .99 R^2, I choose the one without intercept as the full model
m7 = m7_2

#plot original data and predction by the full model
quartz("Linear")
plot(sales, main="Full Model Prediction", ylab = "Total NSA(Thousands of Unit)", xlab = "Year", lwd = 1, xlim = c(1976,2019))
lines(m7_2$fitted.values,col = "red3", lwd = 1)
legend("topright",pch= c("-","-"),legend=c("Original Data", "Full Model Prediction"),col=c("black", "red"), cex=0.8)


#for residuals vs. fitted values
quartz("Residuals vs. fitted values for full model")
```

```r
plot(x = m7_2$fitted.values, y = m7$residuals, type = "p", pch = 20, main = "Full model fitted value vs. residual",xlab = "Fitted Value", ylab = "Residuals")
abline(a = 0, b = 0)
#There’s no obvious shape and generally symmetrically distributed around 0 in both graph plotting
#Thus the variables not linearly correlated and generally homoscedastic.

#(d) Interpret the respective summary statistics including the error metrics of your full model.
#Very Impressive R^2, t, and f
summary(m7_2)
#However we are pretty bad at this(Because we haven’t discuss cycles in lectures)
AIC(m7)
BIC(m7)
#Find mse is 9794.861, behave poorly
mean((sales - m7_2$fitted.values)^2)


#(e)Use the full model to forecast h-steps (at least 16) ahead. Your forecast should include the respective prediction interval.

# In forecast function, only h=519 can be used, otherwise there would be Error in variables
quartz()
plot(forecast(m7,h = 519))

#III
#As what I mentioned in II.2.(d), we have a horrible AIC and BIC,
#which is not only because of we have great amount of variables(it’s overfitted for sure)
#but we also have a very stochastic trend, see:
plot(stl(ts(Data$TOTALNSA,freq = 12), s.window = "periodic"))
#as what is discussed in OH, we need more techniques to dealing with the cycles pattern
```