U.S. Monthly Retail Sales Research Project

Yinuo Song

11/24/2019

Introduction

The Retail Trade sector comprises establishments engaged in retailing merchandise, generally without transformation, and rendering services incidental to the sale of merchandise. The retailing process is the final step in the distribution of merchandise; retailers are, therefore, organized to sell merchandise in small quantities to the general public.

I get my data source for retail trade from the U.S. Census Bureau website. It contains monthly sales for retail trade and food services, adjusted and unadjusted for seasonal variations within the U.S. area and has 67 columns in total. In my R project research, I compiled the data from 2000 to 2019 and I only use the total retail sales columns as my main factor. I plan to make the forecasting for the U.S. monthly retail sales in total for the next two years, using functions about time series in R and building models to make trend and seasonality forecast.

Data source: https://www.census.gov/retail/index.html

```
rm(list=ls(all=TRUE))
getwd()

## [1] "/Users/air/Desktop"

library(lattice)
library(foreign)
library(MASS)
library(car)

## Loading required package: carData

require(stats)
require(stats4)

## Loading required package: stats4

library(KernSmooth)
```

```
## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009

library(fastICA)
library(cluster)
library(leaps)
library(mgcv)

## Loading required package: nlme

## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.

library(rpart)
library(pan)
library(mgcv)
library(DAAG)

##
## Attaching package: 'DAAG'

## The following object is masked from 'package:car':
##
##     vif

## The following object is masked from 'package:MASS':
##
##     hills

library("TTR")

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

library(tis)

##
## Attaching package: 'tis'

## The following object is masked from 'package:TTR':
##
##     lags

## The following object is masked from 'package:mgcv':
##
##     ti

require("datasets")
require(graphics)
library("forecast")
```

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

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff

##
## Attaching package: 'forecast'

## The following object is masked from 'package:tis':
##
##     easter

## The following object is masked from 'package:nlme':
##
##     getResponse

require(astsa)

## Loading required package: astsa

##
## Attaching package: 'astsa'

## The following object is masked from 'package:forecast':
##
##     gas

library(RColorBrewer)
library(plotrix)
library(xtable)
library(stats)
library(pastecs)
library(psych)

##
## Attaching package: 'psych'

## The following object is masked from 'package:plotrix':
##
##     rescale

## The following object is masked from 'package:DAAG':
##
##     cities

## The following object is masked from 'package:car':
##
##     logit
```

```r
library(TSA)

## Registered S3 methods overwritten by 'TSA':
##   method       from
##   fitted.Arima forecast
##   plot.Arima   forecast

##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar

library(timeSeries)

## Loading required package: timeDate

##
## Attaching package: 'timeDate'

## The following objects are masked from 'package:TSA':
##
##     kurtosis, skewness

## The following object is masked from 'package:xtable':
##
##     align

## The following objects are masked from 'package:tis':
##
##     dayOfWeek, dayOfYear, isHoliday

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:psych':
##
##     outlier

## The following objects are masked from 'package:tis':
##
##     description, interpNA

library(fUnitRoots)

## Loading required package: fBasics
```

```
##
## Attaching package: 'fBasics'

## The following object is masked from 'package:psych':
##
##     tr

## The following object is masked from 'package:astsa':
##
##     nyse

## The following object is masked from 'package:TTR':
##
##     volatility

## The following object is masked from 'package:car':
##
##     densityPlot

library(fBasics)
library(tseries)
library(timsac)
library(TTR)
library(fpp)

## Loading required package: fma

##
## Attaching package: 'fma'

## The following objects are masked from 'package:astsa':
##
##     chicken, sales

## The following objects are masked from 'package:DAAG':
##
##     milk, ozone

## The following objects are masked from 'package:MASS':
##
##     cement, housing, petrol

## Loading required package: expsmooth

## Loading required package: lmtest

## Loading required package: zoo

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

```
## The following object is masked from 'package:timeSeries':
##
##     time<-

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

##
## Attaching package: 'fpp'

## The following object is masked from 'package:astsa':
##
##     oil

library(RColorBrewer)
library(plotrix)
library(nlstools)

##
## 'nlstools' has been loaded.

## IMPORTANT NOTICE: Most nonlinear regression models and data set exam
ples

## related to predictive microbiolgy have been moved to the package 'nl
sMicrobio'

library(Metrics)

##
## Attaching package: 'Metrics'

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

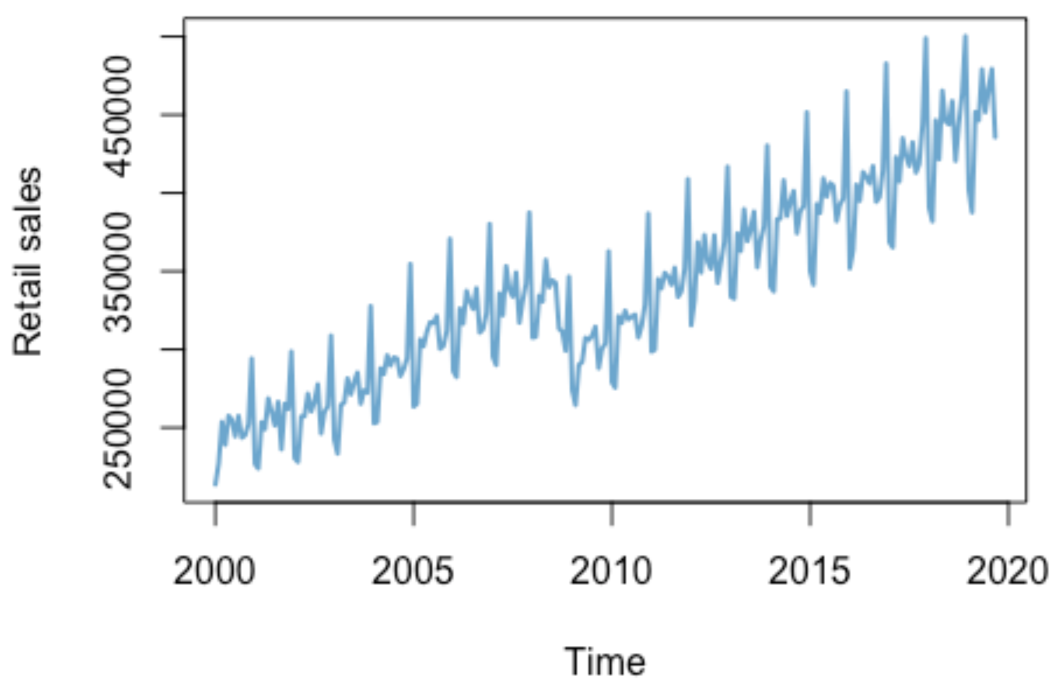
Read in the data

```
data1 <- read.csv("US Census Bureau Data.csv")
head(data1)

##        X retail book  food  auto furni electro health cloth jewelry
sport
## 1 Jan-00 213709 1505 22713 51413  3897    6276  11960  6683    1244
 1591
## 2 Feb-00 227087 1090 23493 58828  4072    6208  12151  7470    1979
 1681
## 3 Mar-00 253717 1031 25798 64815  4363    6364  12957  9283    1500
 2065
## 4 Apr-00 239051  975 25366 56524  4032    5749  12427  9420    1560
```

```
 1991
## 5 May-00 257581 1104 26125 62467  4256    5871  13237  9496    2126
 2187
## 6 Jun-00 255066 1122 26222 62396  4175    5833  12813  9156    1771
 2319

retail <- data1[, 2]
retail_ts <- ts(retail, start=2000, freq=12)
describe(retail_ts)

##    vars   n     mean       sd median  trimmed      mad    min    max
## X1    1 237 338579.8 64100.19 333789 335943.6 73317.54 213709 500260
##     range skew kurtosis      se
## X1 286551 0.31    -0.66 4163.75

t1 <- seq(2000, 2019.9, length=length(retail_ts))
```

Time-series plot

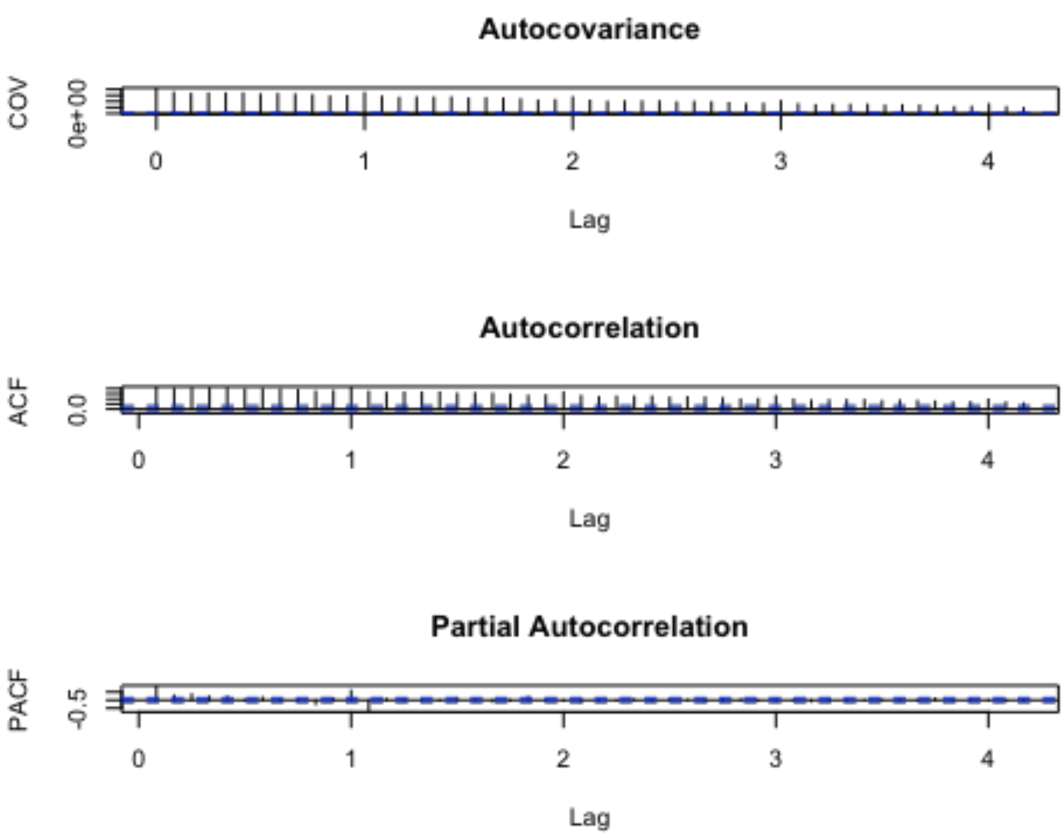
```
plot(retail_ts, ylab="Retail sales", xlab="Time", lwd=2, col='skyblue3')
```



We can see from the plot that it does not suggest that the data are covariance stationary. Covariance stationarity refers to a set of conditions on the underlying probability structure of a time series that has proven to be very especially valuable in this regard. A time series yt is said to be covariance stationary if it meets the following conditions: 1.Constant mean 2.Constant (and finite) variance 3.Stable autocovariance function. My data does not have these futures, so the plot are not covariance stationary.

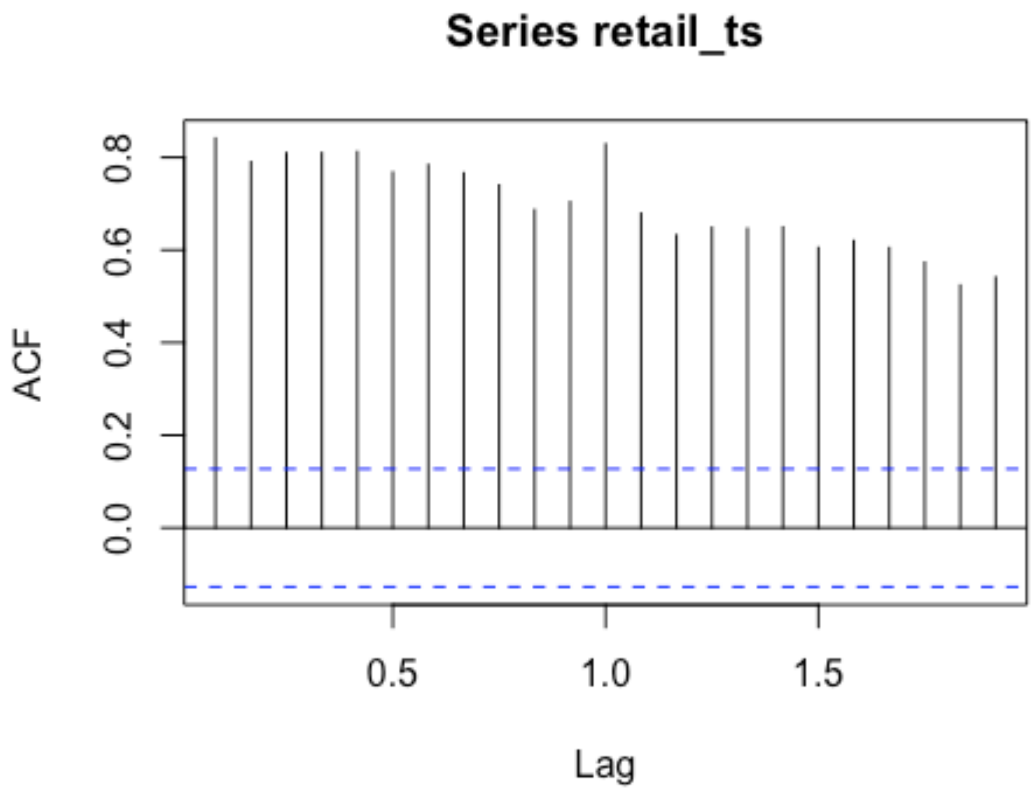
ACF & PACF

```
quartz()
par(mfrow=c(3,1))
acf(retail_ts, type = "covariance", main="Autocovariance", lag.max=50,
ylab="COV")
acf(retail_ts, type = "correlation", main="Autocorrelation", lag.max=50,
 ylab="ACF")
acf(retail_ts, type = "partial", main="Partial Autocorrelation", lag.ma
x=50, ylab="PACF")
```



From the ACF and PACF test, there is dependence on how retail sales have changed over time.

```
#White noise test
acf_val=acf(retail_ts)$acf
```




```
Box.test(acf_val, type = “Ljung-Box”)

##
##  Box-Ljung test
##
## data:  acf_val
## X-squared = 15.032, df = 1, p-value = 0.0001057

Box.test(acf_val, type = “Box-Pierce”)

##
##  Box-Pierce test
##
## data:  acf_val
## X-squared = 13.228, df = 1, p-value = 0.0002758

#MA(q) Model Fitting
ma1=arma(retail_ts, order=c(0,1))
summary(ma1)
```

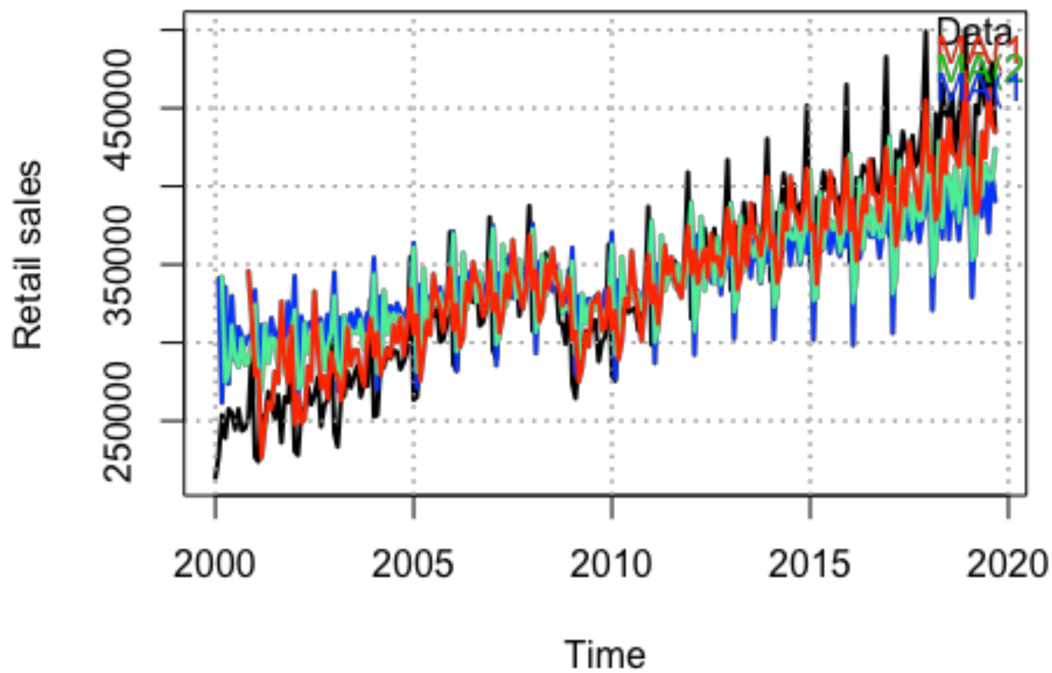
```
##
## Call:
## arma(x = retail_ts, order = c(0, 1))
##
## Model:
## ARMA(0,1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -113996  -31991   -5276   33731  112809
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ma1       6.972e-01   3.794e-02    18.38   <2e-16 ***
## intercept 3.411e+05   5.853e+03    58.28   <2e-16 ***
## ---
## Signif. Codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 2.107e+09,  Conditional Sum-of-Squares = 495496
524532,  AIC = 5764.65
```

```r
ma2=arma(retail_ts, order=c(0,2))
summary(ma2)
```

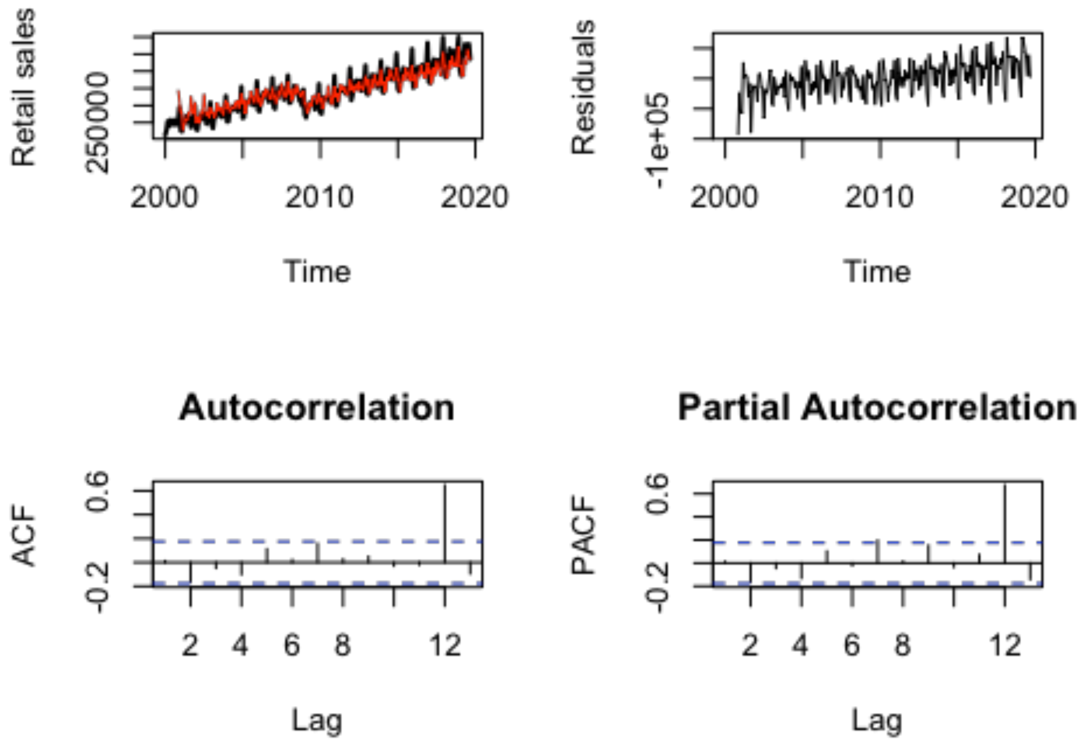
```
##
## Call:
## arma(x = retail_ts, order = c(0, 2))
##
## Model:
## ARMA(0,2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -97571 -27257  -4772  29695  94425
##
## Coefficient(s):
##            Estimate  Std. Error  t value Pr(>|t|)
## ma1       7.604e-01   6.162e-02   12.340  < 2e-16 ***
## ma2       3.988e-01   4.857e-02    8.211 2.22e-16 ***
## intercept 3.419e+05   8.645e+03   39.547  < 2e-16 ***
## ---
## Signif. Codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 1.73e+09,  Conditional Sum-of-Squares = 4050240
95746,  AIC = 5719.88
```

```r
ma3=arma(retail_ts, order=c(0,10))
summary(ma3)
```

```
##
## Call:
## arma(x = retail_ts, order = c(0, 10))
##
## Model:
## ARMA(0,10)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -93519.6 -18225.6    944.2  19206.5  68929.6
##
## Coefficient(s):
##             Estimate  Std. Error  t value Pr(>|t|)
## ma1        3.046e-01   5.443e-02    5.596 2.19e-08 ***
## ma2        6.383e-01   6.504e-02    9.814  < 2e-16 ***
## ma3        3.501e-01   7.548e-02    4.639 3.51e-06 ***
## ma4        6.412e-01   8.227e-02    7.794 6.44e-15 ***
## ma5        3.920e-01   7.896e-02    4.965 6.88e-07 ***
## ma6        4.142e-01   6.972e-02    5.940 2.85e-09 ***
## ma7        3.573e-01   6.178e-02    5.785 7.27e-09 ***
## ma8        3.709e-01   6.637e-02    5.588 2.30e-08 ***
## ma9        2.884e-01   8.510e-02    3.390    7e-04 ***
## ma10      -3.375e-01   6.467e-02   -5.218 1.80e-07 ***
## intercept  3.457e+05   1.101e+04   31.402  < 2e-16 ***
## ---
## Signif. Codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Fit:
## sigma^2 estimated as 745468928,  Conditional Sum-of-Squares = 168495
171921,  AIC = 5536.37

#All the MA(q) fits
quartz()
plot(retail_ts, xlab='Time', ylab="Retail sales", lwd=2)
grid()
lines(ma1$fitted.values, col="blue", lwd=2)
lines(ma2$fitted.values, col="seagreen2", lwd=2)
lines(ma3$fitted.values, col="red", lwd=2)
legend("topright", legend=c("Data","MA(10)","MA(2)","MA(1)"), text.col=
1:4, bty="n")
```




```
#Examine the best fit MA(q) model
quartz()
par(mfrow=c(2,2))
plot(retail_ts, xlab='Time', ylab="Retail sales", lwd=2)
lines(ma3$fitted.values, col="red", lwd=1, lty=1)
plot(ma3$residuals, ylab="Residuals")
acf(ma3$residuals[12:136], type = "correlation", main="Autocorrelation",
lag.max=13,ylab="ACF")
acf(ma3$residuals[12:136], type = "partial",main="Partial Autocorrelati
on",lag.max=13, ylab="PACF")
```



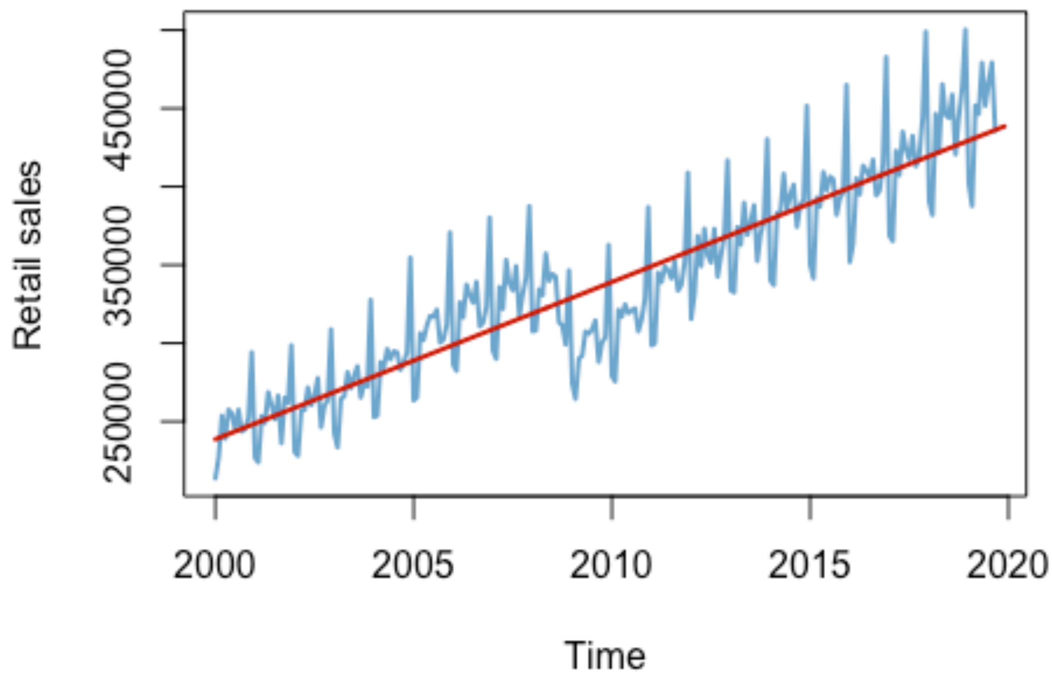
We can see a significant improvement from looking at the ACF and PACF plots, the residuals now look consistent with noise, suggesting we accounted for most of the dynamics left after detrending and seasonally adjusting the data.

Trend fitting

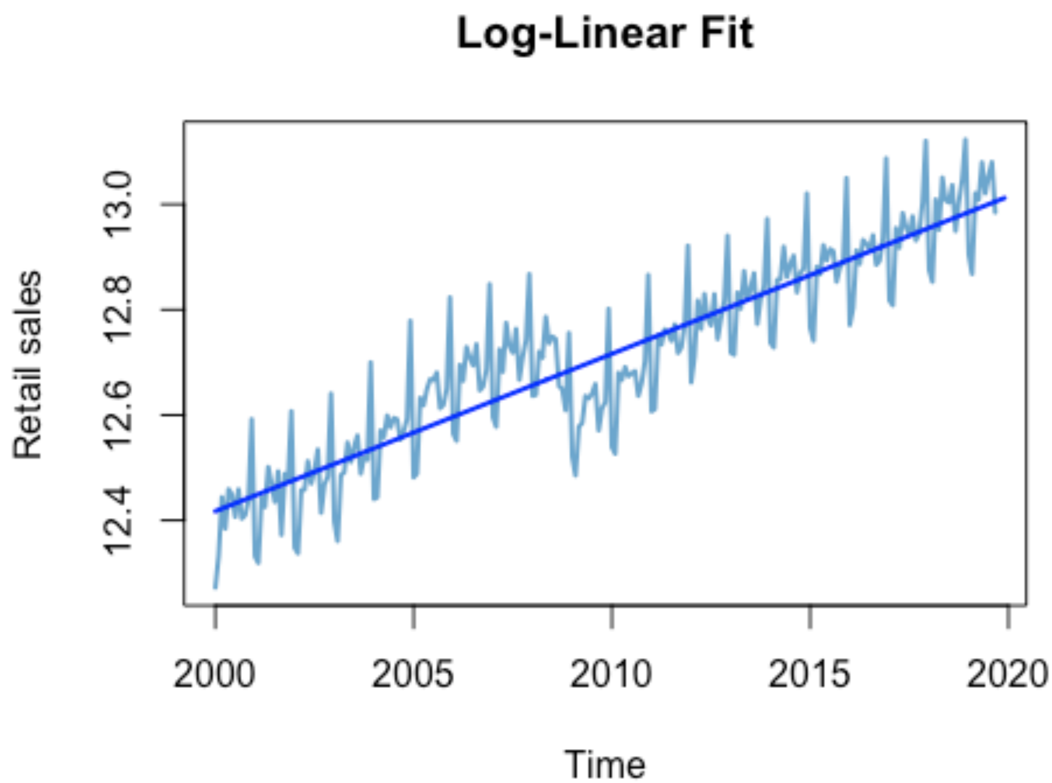
```r
#Mode1: Linear Fit model
m1 <- lm(retail_ts ~ t1)

#Mode2: Log-Linear Fit model
m2 <- lm(log(retail_ts) ~ t1)

#Mode3: Log-quadratic model
lretail <- log(retail_ts)
m3 <- lm(lretail ~ t1+t1^2)
```

Figures

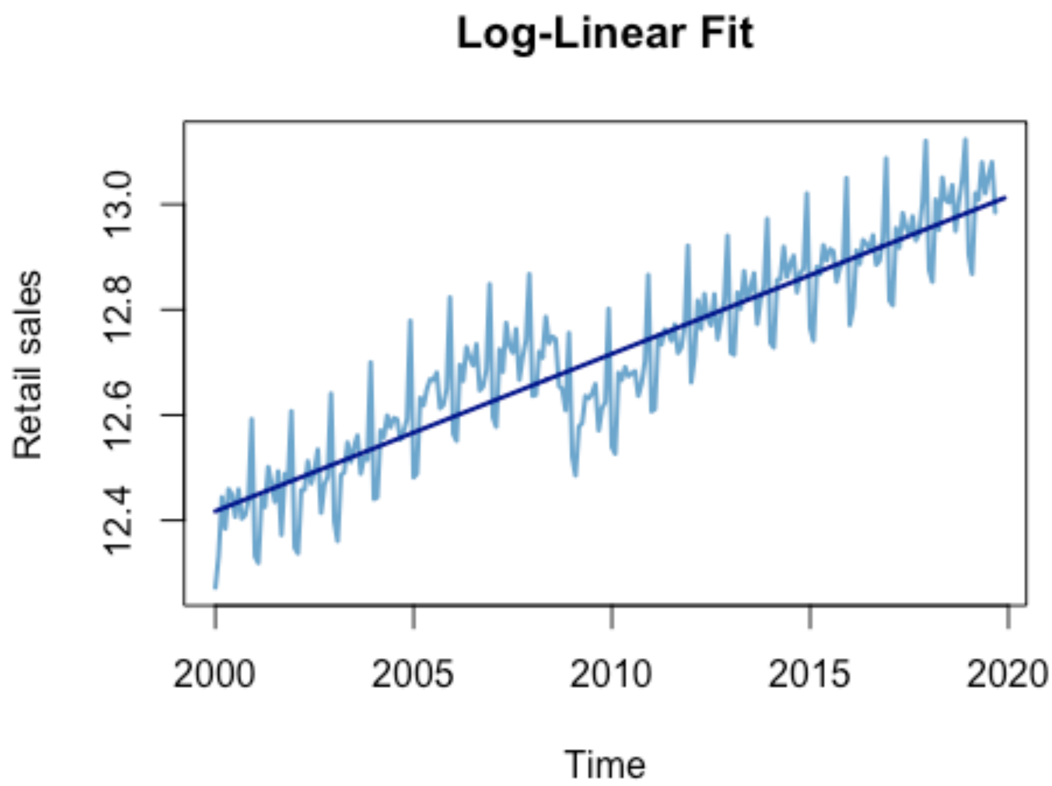
```
plot(retail_ts, ylab="Retail sales", xlab="Time", lwd=2, col='skyblue3')
lines(t1, m1$fit, col="red3", lwd=2)
```




```
plot(lretail, ylab="Retail sales", xlab="Time", lwd=2, col='skyblue3',
main="Log-Linear Fit")
lines(t1, m2$fit, col="blue", lwd=2)
```



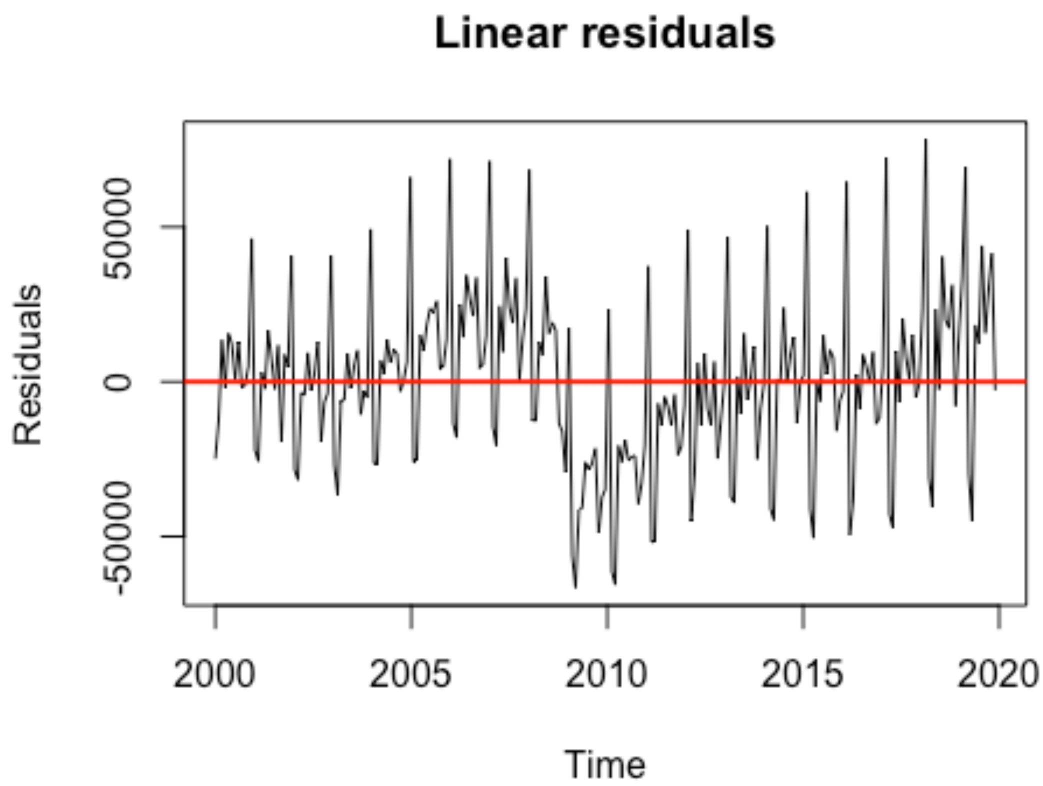

```
plot(lretail, ylab="Retail sales", xlab="Time", lwd=2, col='skyblue3',
main="Log-Linear Fit")
lines(t1, m3$fit, col="darkblue", lwd=2)
```



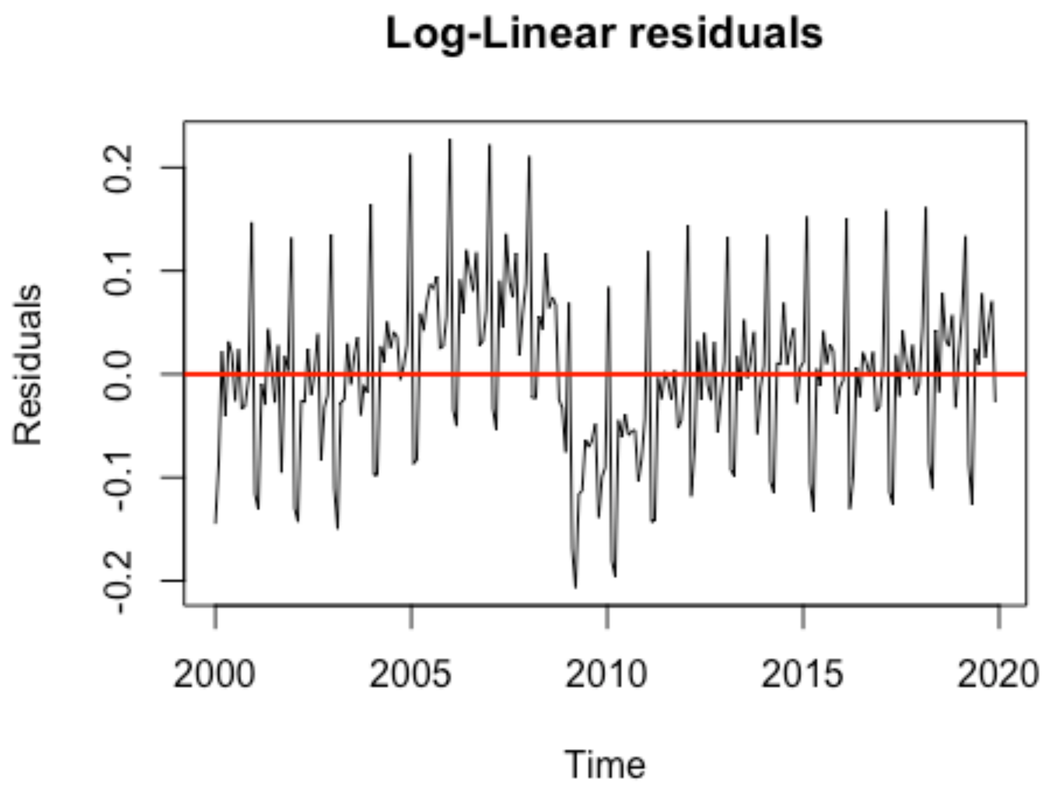
From the plot, we can see that the three models seems to be similar with each other.

Residuals & Fitted values

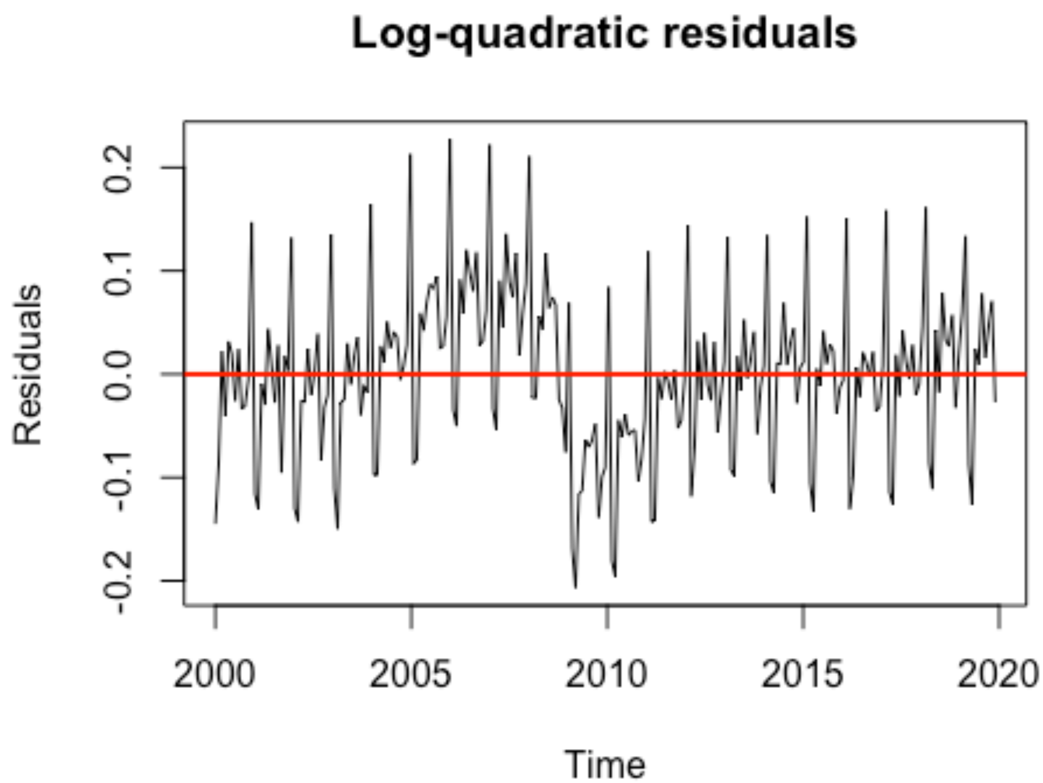
```
#Mode1: Linear Fit model
plot(t1, m1$res, ylab="Residuals", type='l', xlab="Time", main="Linear
residuals")
abline(h=0, col="red", lwd=2)
```




```
#Mode2: Log-Linear Fit model
plot(t1, m2$res, ylab="Residuals", type='l', xlab="Time", main="Log-Lin
ear residuals")
abline(h=0, col="red", lwd=2)
```



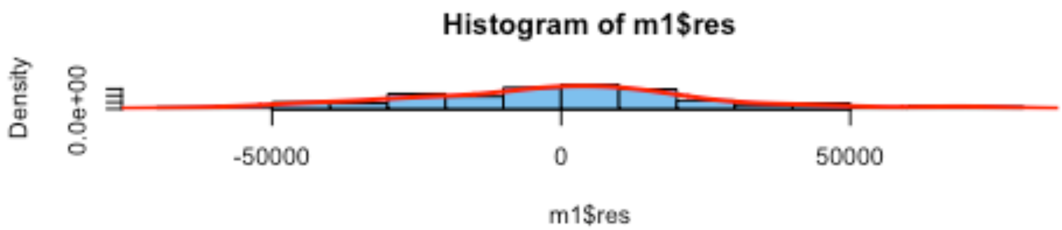

```
#Mode3: Log-quadratic model
plot(t1, m3$res, ylab="Residuals", type='l', xlab="Time", main="Log-qua
dratic residuals")
abline(h=0, col="red", lwd=2)
```

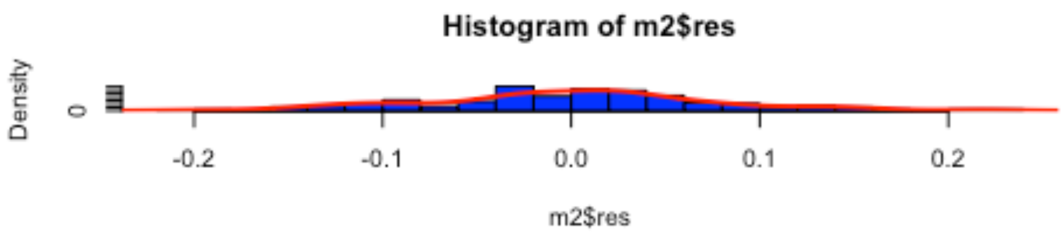


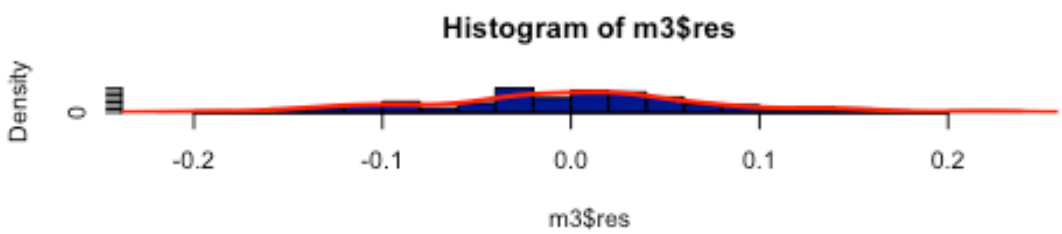
I observed that the residuals are showing the periodic characteristics, and the residuals around 2008 year are much larger and unstable when compared with other years. This may be because of the financial crisis at that time.

Histogram of residuals

```
par(mfrow=c(3,1))
hist(m1$res, breaks = "FD", col = "skyblue2", freq = FALSE, ylab = "Den
sity")
lines(density(m1$res), lwd = 2, col = "red")

hist(m2$res, breaks = "FD", col = "blue", freq = FALSE, ylab = "Density
")
lines(density(m2$res), lwd = 2, col = "red")

hist(m3$res, breaks = "FD", col = "darkblue", freq = FALSE, ylab = "Den
sity")
lines(density(m3$res), lwd = 2, col = "red")
```







I observed that the values of the density for the residuals are different from each model, but all the residuals from the six models are close to the normal distribution.

Diagnostic statistics

```
m1 <- lm(retail_ts ~ t1)
summary(m1)

##
## Call:
## lm(formula = retail_ts ~ t1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66492 -18047    222  14894  78231
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.985e+07  6.155e+05  -32.25   <2e-16 ***
## t1           1.004e+04  3.062e+02   32.80   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27200 on 235 degrees of freedom
## Multiple R-squared:  0.8207, Adjusted R-squared:   0.82
## F-statistic:  1076 on 1 and 235 DF,  p-value: < 2.2e-16
```

```
m2 <- lm(log(retail_ts) ~ t1)
summary(m2)
```

```
##
## Call:
## lm(formula = log(retail_ts) ~ t1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.206508 -0.042960  0.004453  0.042304  0.227002
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -47.351054   1.786846  -26.50   <2e-16 ***
## t1            0.029884   0.000889   33.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07895 on 235 degrees of freedom
## Multiple R-squared:  0.8278, Adjusted R-squared:  0.8271
## F-statistic:  1130 on 1 and 235 DF,  p-value: < 2.2e-16
```

```
m3 <- lm(lretail ~ t1+t1^2)
summary(m3)
```

```
##
## Call:
## lm(formula = lretail ~ t1 + t1^2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.206508 -0.042960  0.004453  0.042304  0.227002
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -47.351054   1.786846  -26.50   <2e-16 ***
## t1            0.029884   0.000889   33.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07895 on 235 degrees of freedom
## Multiple R-squared:  0.8278, Adjusted R-squared:  0.8271
## F-statistic:  1130 on 1 and 235 DF,  p-value: < 2.2e-16
```

R squared of the linear model is 0.8207, R squared of the log-linear model is 0.8278, R squared of the Log-quadratic model is 0.8278. It suggests that these three models are all good fit models. And the model 2 and model 3 are better than model 1. P-values are all small but far bigger than the t value, which suggests that the estimators are statistically significant.

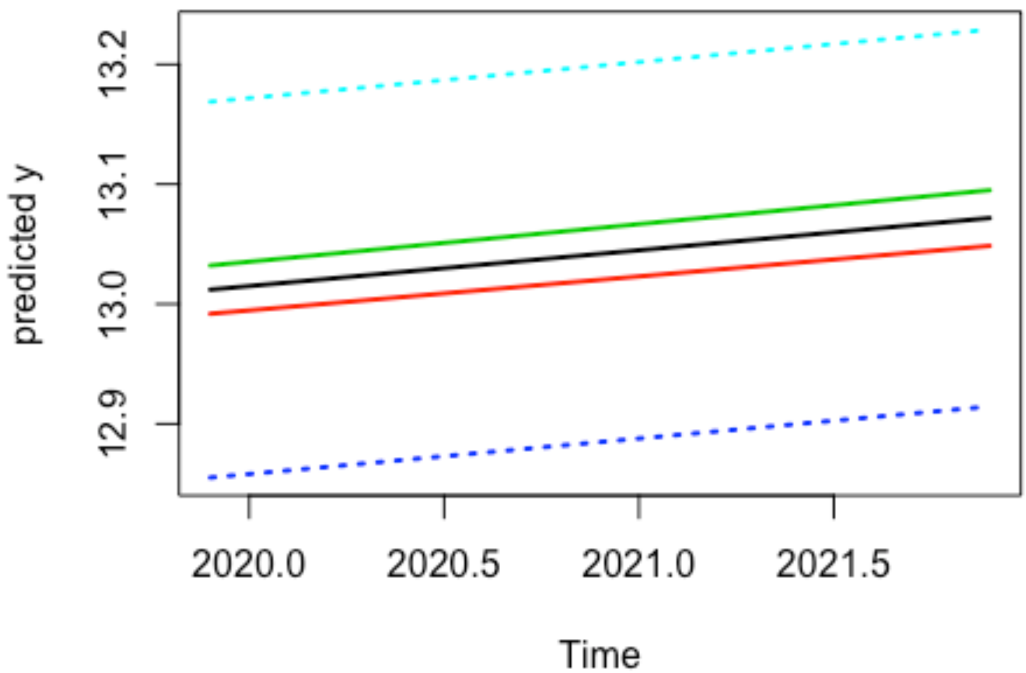
AIC & BIC

```
AIC(m1,m2,m3)

##    df       AIC
## m1  3 5516.5304
## m2  3 -526.8708
## m3  3 -526.8708

BIC(m1,m2,m3)

##    df       BIC
## m1  3 5526.9346
## m2  3 -516.4666
## m3  3 -516.4666
```

From the AIC and BIC tests, we can tell that the model 2 and model 3 are better than the model 1 since their test values are smaller. Model 2 and model 3 have same AIC and BIC values, so either model is great, and I plan to choose the model 3 as my preffered model.

Trend forecasting

```
m3 <- lm(lretail ~ t1+t1^2)
tn=data.frame(t1=seq(2019.9, 2021.9, length=24))
pred.m3=predict(m3, tn, se.fit = TRUE)
pred.plim.m3=predict(m3, tn, n.ahead=60, level = 0.95, interval="predic
tion")
pred.clim.m3=predict(m3, tn, n.ahead=60, level = 0.95, interval="confid
ence")
matplot(tn$t1, cbind(pred.clim.m3, pred.plim.m3[,-1]),
        lty=c(1,1,1,3,3), type="l", lwd=2, ylab="predicted y",xlab="Tim
e")
```

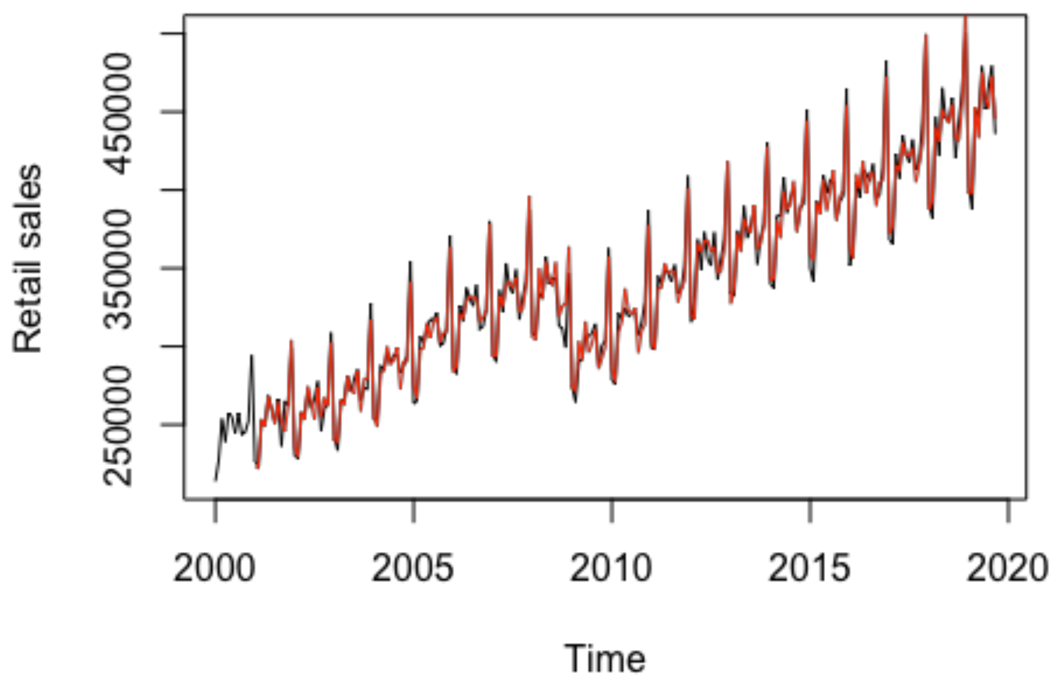


I predict the US monthly retail sales for the next two years as the plot shows. We can see the prediction interval for different confidence levels in this plot. It shows clearly that the US monthly retail sales will increases in the next two years.

```
#Holt-Winters Filter
HoltWinters(retail_ts)

## Holt-Winters exponential smoothing with trend and additive seasonal
component.
##
## Call:
## HoltWinters(x = retail_ts)
##
## Smoothing parameters:
##  alpha: 0.5076688
##  beta : 0.004069181
##  gamma: 0.859555
##
## Coefficients:
##             [,1]
## a   454127.3190
```

```
## b       792.2952
## s1     1151.3695
## s2    15686.6413
## s3    61513.0423
## s4   -40749.3157
## s5   -49871.7785
## s6    13358.8935
## s7     -290.1613
## s8    29995.4536
## s9     6298.4882
## s10   11897.3994
## s11   20602.1364
## s12  -17726.4437
```

```
quartz()
plot(retail_ts, xlab="Time", ylab="Retail sales")
lines(HoltWinters(retail_ts)$fitted[,1], col="red")
```

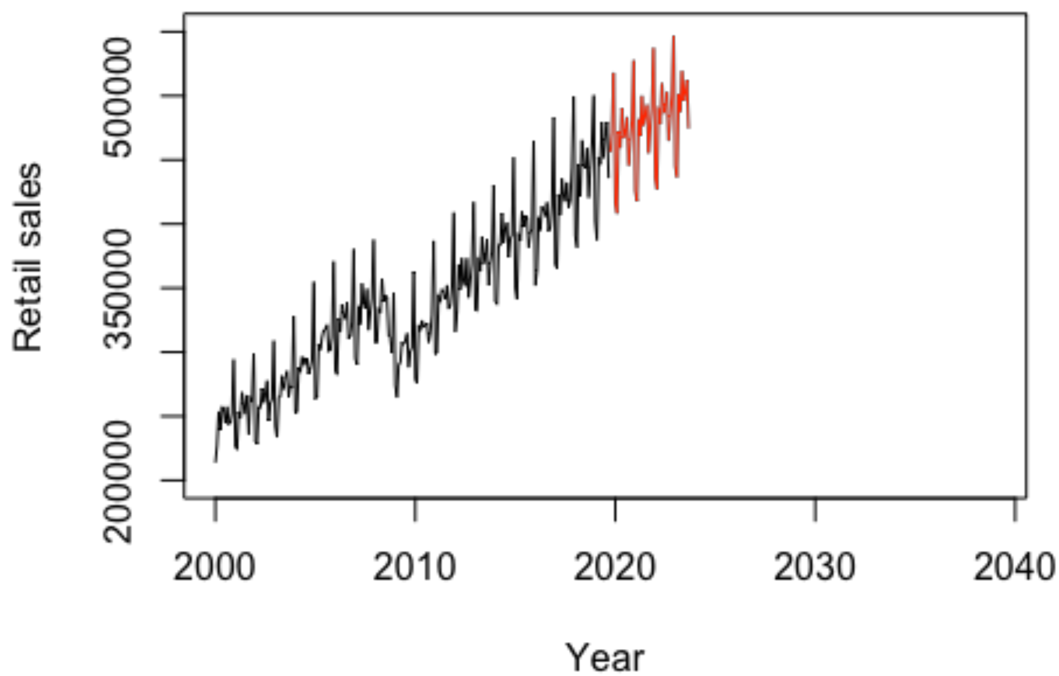



```
#Holt-Winters Prediction
retail_ts.hw <- HoltWinters(retail_ts)
predict(retail_ts.hw, n.ahead=12)
```

```
##            Jan      Feb      Mar      Apr      May      Jun      Jul
## 2019
```

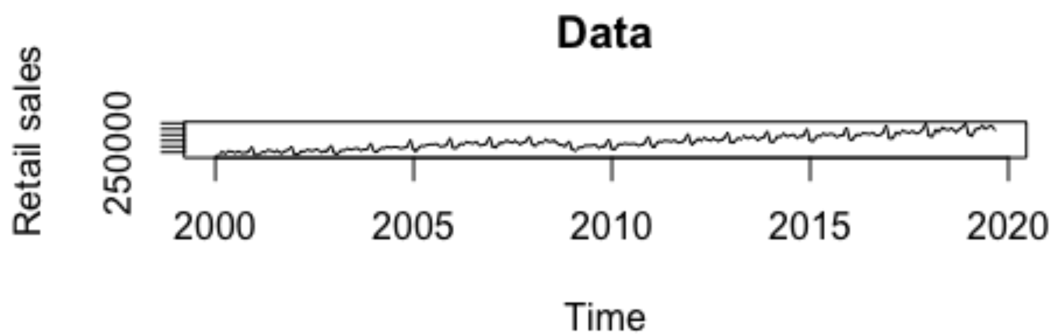
```
## 2020 416547.2 408217.0 472240.0 459383.2 490461.1 467556.5 473947.7
##           Aug      Sep      Oct      Nov      Dec
## 2019                     456071.0 471398.6 518017.2
## 2020 483444.7 445908.4
```

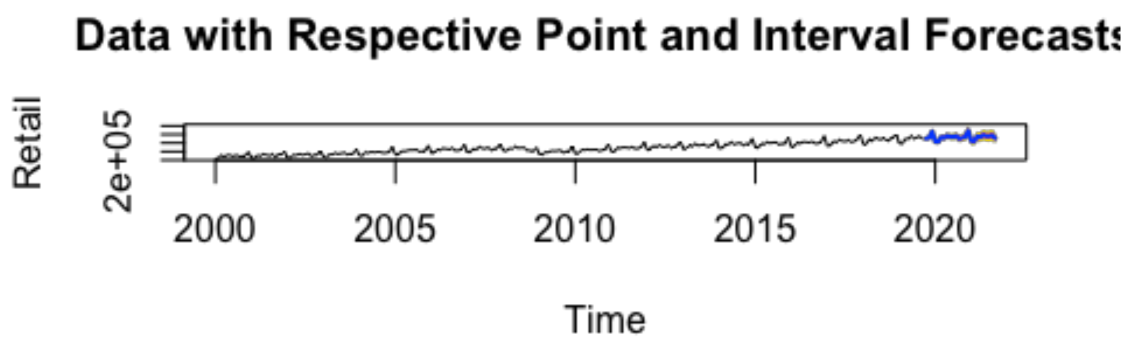
```
quartz()
plot(retail_ts, xlim=c(2000,2039), xlab="Year", ylab="Retail sales", yl
im=c(200000,550000))
lines(predict(retail_ts.hw, n.ahead=48), col=2)
```



Trend Forecasting

```
quartz()
par(mfrow=c(2,1))
plot(retail_ts, main="Data", xlab="Time", ylab="Retail sales")
plot(forecast(retail_ts), main="Data with Respective Point and Interval
 Forecasts",
     xlab="Time", ylab="Retail", shadecols="oldstyle")
```





Model with a full set of seasonal dummies

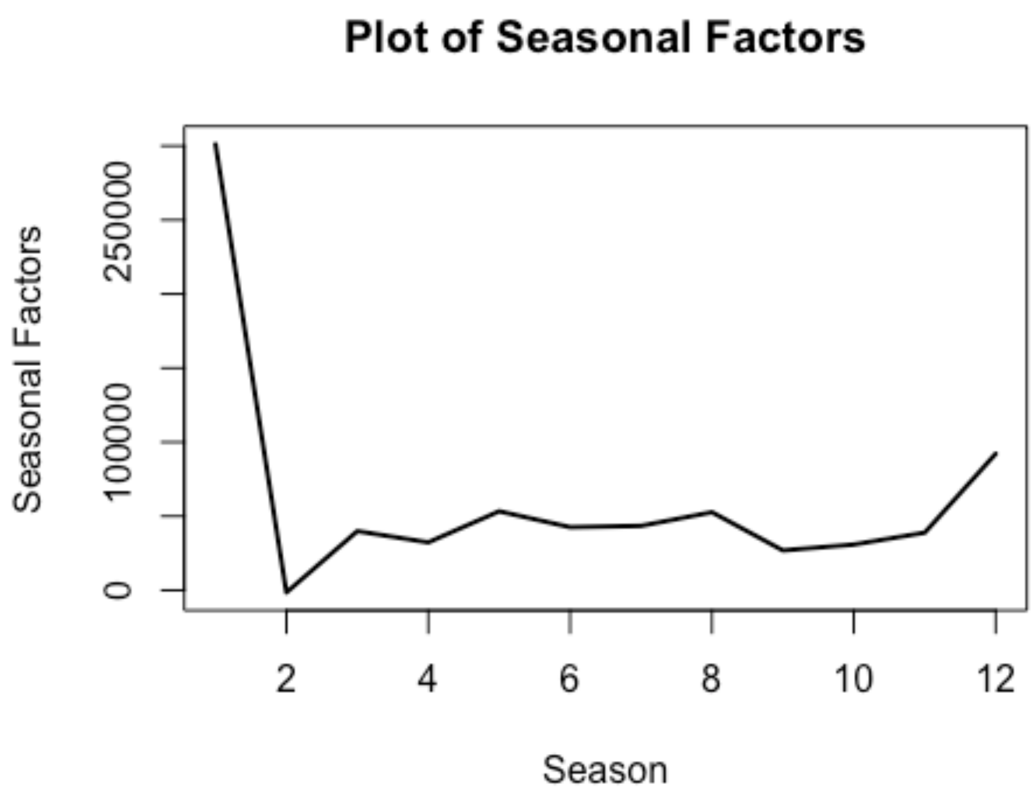
```
m4 <- tslm(retail_ts ~ season)
summary(m4)

##
## Call:
## tslm(formula = retail_ts ~ season)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -100184  -45759   -5874   50223  125464
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   301158      13661  22.046  < 2e-16 ***
## season2        -1493      19319  -0.077  0.93847
## season3        39906      19319   2.066  0.04001 *
## season4        32307      19319   1.672  0.09586 .
## season5        53311      19319   2.760  0.00627 **
## season6        42584      19319   2.204  0.02852 *
## season7        43471      19319   2.250  0.02541 *
## season8        52674      19319   2.727  0.00691 **
```

```
## season9         26886      19319   1.392  0.16539
## season10        30852      19572   1.576  0.11635
## season11        38879      19572   1.986  0.04819 *
## season12        92164      19572   4.709 4.35e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61090 on 225 degrees of freedom
## Multiple R-squared:  0.134,  Adjusted R-squared:  0.09165
## F-statistic: 3.165 on 11 and 225 DF,  p-value: 0.0005253
```

We can see that the p-values of most seasons are very small which suggest statistically significance. Therefore, we can say that there are seasonal factors in this model.

Plot seasonal factors

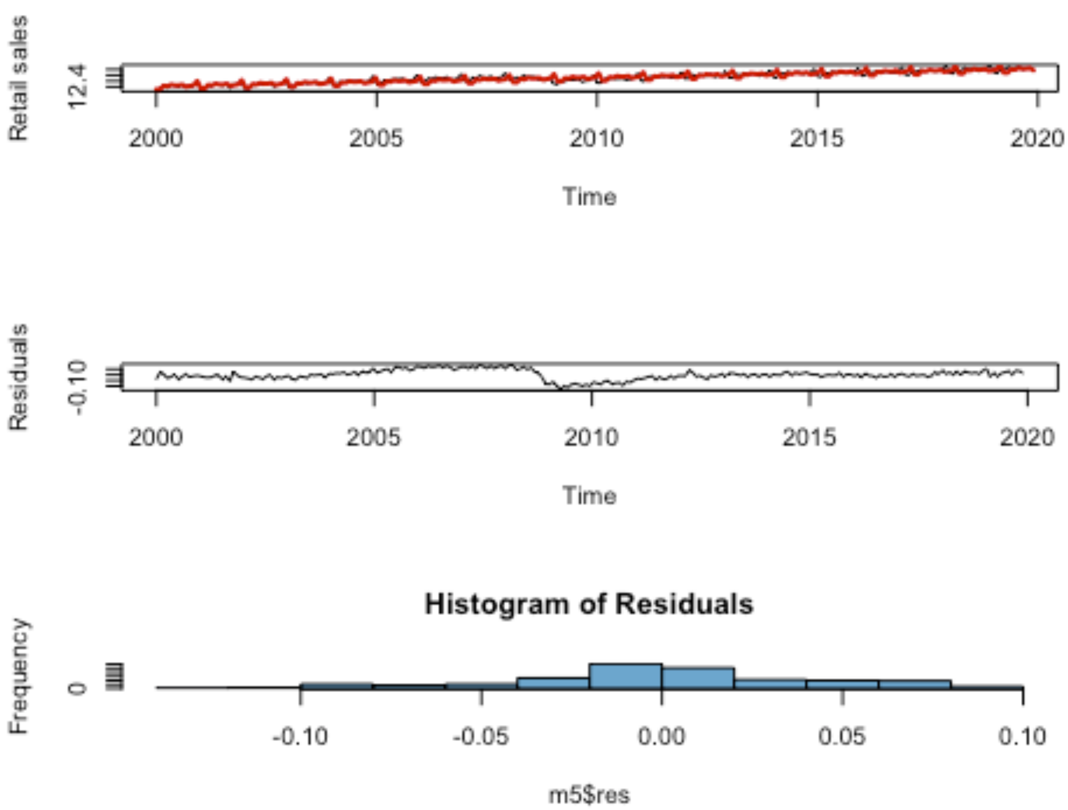
```
plot(m4$coef,type='l',ylab='Seasonal Factors', xlab="Season",lwd=2, mai
n="Plot of Seasonal Factors")
```



We can see that the seasonal factors are extremely significant when approaching Christmas and New Years Day. It fits well with the fact that people will buy much more things during these holidays. The seasonal factors drop very low after these holidays.

Full model

```
m5 <- tslm(lretail ~ t1+t1^2+season)
par(mfrow=c(3,1))
plot(lretail, xlab="Time", ylab="Retail sales")
lines(t1,m5$fit,col="red3",lwd=2)
plot(t1, m5$res, ylab="Residuals", type='l', xlab="Time")
hist(m5$res, main="Histogram of Residuals", col="skyblue3")
```



Summary statistics and the error metrics of full model

```
S(m5)

## Call: tslm(formula = lretail ~ t1 + t1^2 + season)
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.707e+01  9.893e-01 -47.581  < 2e-16 ***
## t1           2.969e-02  4.922e-04  60.320  < 2e-16 ***
## season2     -6.791e-03  1.382e-02  -0.491    0.624
## season3      1.190e-01  1.382e-02   8.609 1.32e-15 ***
## season4      9.478e-02  1.382e-02   6.860 6.65e-11 ***
## season5      1.525e-01  1.382e-02  11.037  < 2e-16 ***
## season6      1.206e-01  1.382e-02   8.726 6.11e-16 ***
## season7      1.197e-01  1.382e-02   8.663 9.22e-16 ***
## season8      1.441e-01  1.382e-02  10.427  < 2e-16 ***
## season9      6.527e-02  1.382e-02   4.723 4.10e-06 ***
## season10     9.214e-02  1.400e-02   6.582 3.24e-10 ***
## season11     1.116e-01  1.400e-02   7.976 7.74e-14 ***
## season12     2.573e-01  1.400e-02  18.380  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.04369 on 224 degrees of freedom
## Multiple R-squared: 0.9497
## F-statistic: 352.8 on 12 and 224 DF,  p-value: < 2.2e-16
##     AIC     BIC
## -796.69 -748.13
```

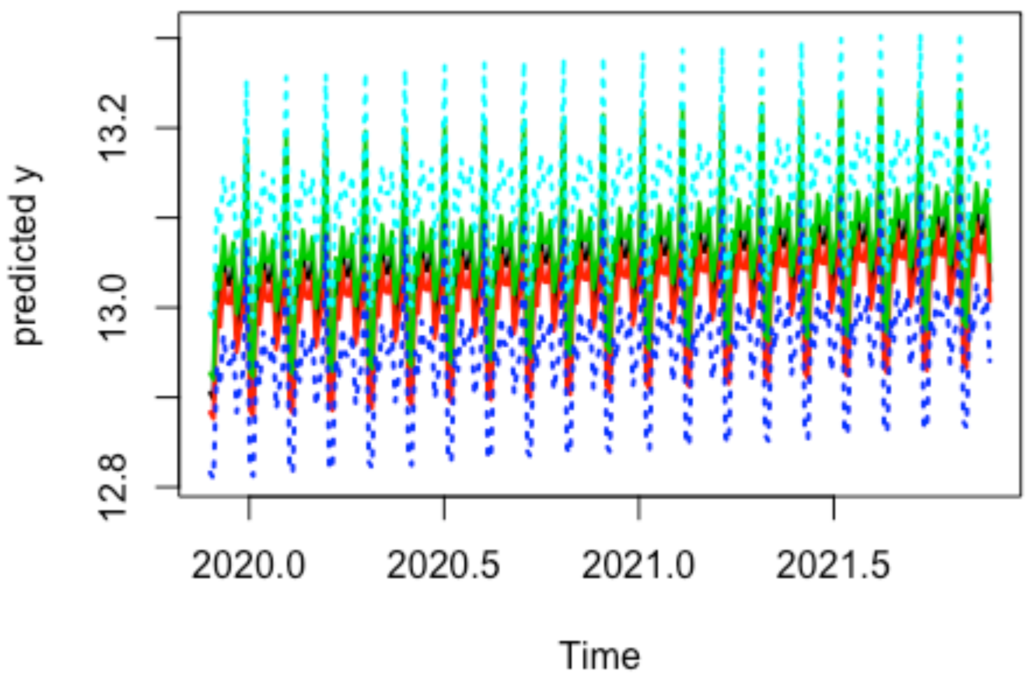
In this full model, we can see that except for season 2(Feb), all other months have strong significant seasonal effects on the retail sales. In details, people tend to buy more goods in April, September and October. And they will buy fewer goods during other months.

```r
mape(lretail, m5$fitted.values)

## [1] 0.002518091

mse(lretail, m5$fitted.values)

## [1] 0.001804325

bias(lretail, m5$fitted.values)

## [1] 7.494715e-18

mae(lretail, m5$fitted.values)

## [1] 0.03191987
```

MAPE: The caculated value of MAPE is 0.0025, which is close to zero. MSE: The caculated value of MSE is 0.0018, which is close to zero. BIAS: The caculated value of BIAS is so small and is close to zero. MAE: The caculated value of MAE is 0.03 which is very small. Therefore, my full model m5 is great.

Seasonality Forecasting

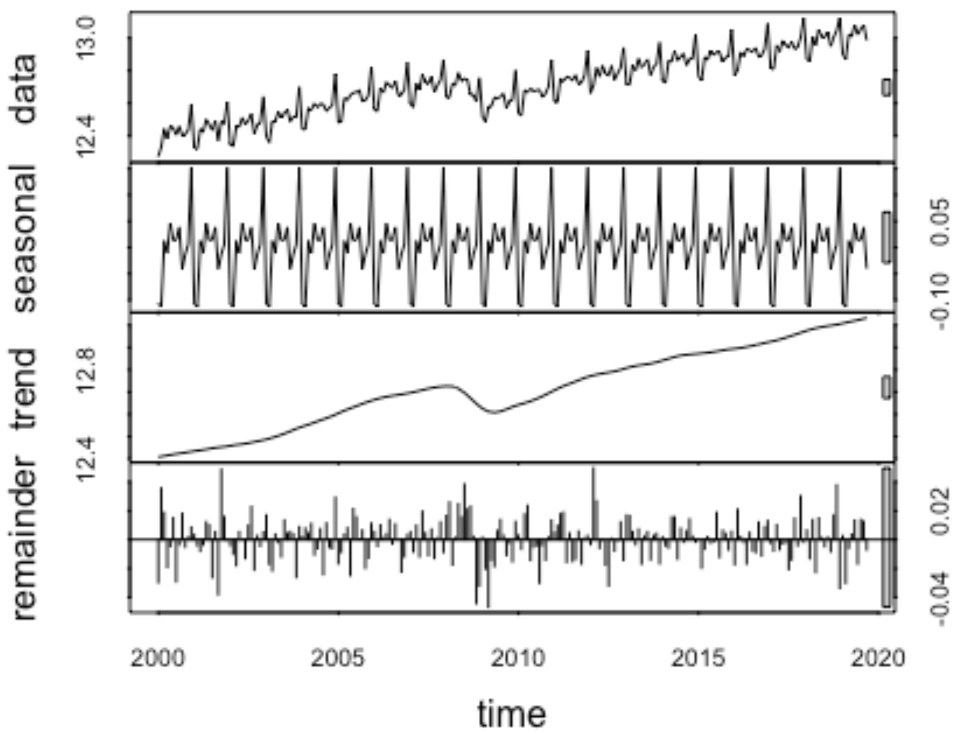
```
m5 <- tslm(lretail ~ t1+t1^2+season)
tn=data.frame(t1=seq(2019.9, 2021.9, length=length(lretail)))
pred.m5=predict(m5, tn, se.fit = TRUE)
pred.plim.m5=predict(m5, tn, n.ahead=60, level = 0.95, interval="predic
tion")
pred.clim.m5=predict(m5, tn, n.ahead=60, level = 0.95, interval="confid
ence")
matplot(tn$t1, cbind(pred.clim.m5, pred.plim.m5[,-1]),
        lty=c(1,1,1,3,3), type="l", lwd=2, ylab="predicted y",xlab="Tim
e")
```



I predict that the next two years as the graph shows. We can see that the US retail sales will follow the seasonal regulation with an increasing trend.

STL

```
plot(stl(lretail,s.window="periodic"))
```



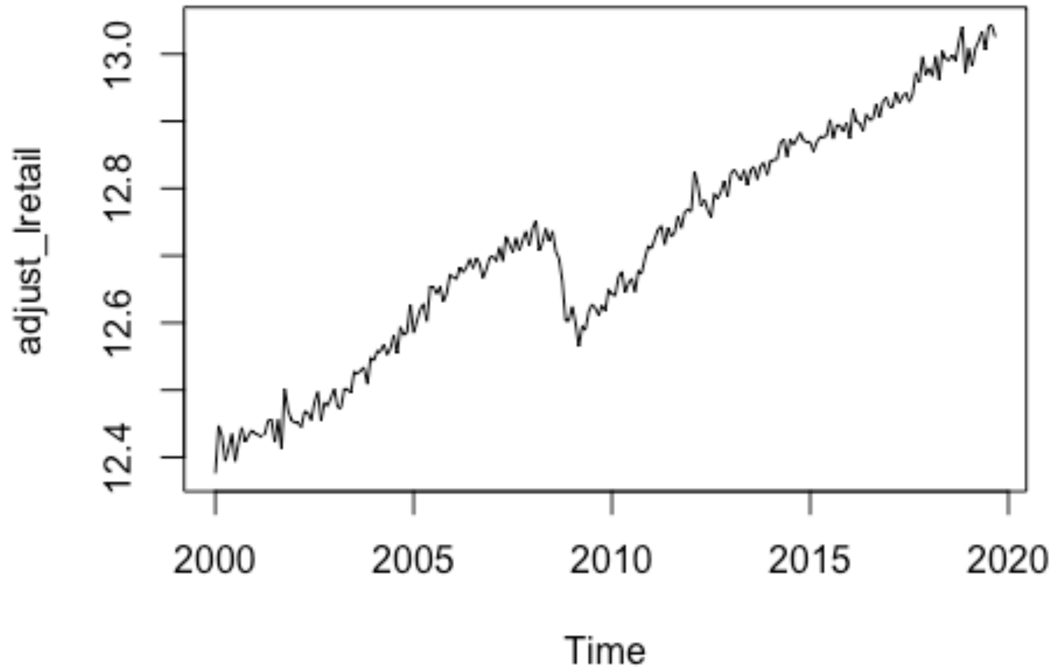

```
summary(forecast(lretail))

##
## Forecast method: ETS(A,Ad,A)
##
## Model Information:
## ETS(A,Ad,A)
##
## Call:
##  ets(y = object, lambda = lambda, biasadj = biasadj, allow.multiplic
ative.trend = allow.multiplicative.trend)
##
##   Smoothing parameters:
##     alpha = 0.4933
##     beta  = 0.0433
##     gamma = 1e-04
##     phi   = 0.9707
##
##   Initial states:
```

```
##     l = 12.3992
##     b = 0.0027
##     s = 0.1507 0.0054 -0.0133 -0.0417 0.0375 0.0134
##            0.0146 0.0468 -0.0105 0.0141 -0.1116 -0.1053
##
##   sigma:  0.02
##
##       AIC      AICc       BIC
## -539.2602 -536.1225 -476.8351
##
## Error measures:
##                       ME       RMSE       MAE        MPE     MAPE
## Training set 0.001369832 0.01929983 0.0151208 0.01063538 0.119041
##                   MASE      ACF1
## Training set 0.3345294 -0.095441
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Oct 2019       13.02239 12.99672 13.04806 12.98313 13.06165
## Nov 2019       13.04341 13.01429 13.07253 12.99888 13.08794
## Dec 2019       13.19101 13.15835 13.22366 13.14106 13.24095
## Jan 2020       12.93712 12.90084 12.97341 12.88163 12.99261
## Feb 2020       12.93297 12.89299 12.97296 12.87183 12.99412
## Mar 2020       13.06076 13.01701 13.10451 12.99386 13.12767
## Apr 2020       13.03816 12.99059 13.08573 12.96540 13.11091
## May 2020       13.09743 13.04599 13.14888 13.01876 13.17611
## Jun 2020       13.06705 13.01169 13.12242 12.98238 13.15173
## Jul 2020       13.06773 13.00840 13.12706 12.97699 13.15846
## Aug 2020       13.09354 13.03022 13.15687 12.99669 13.19039
## Sep 2020       13.01607 12.94872 13.08343 12.91306 13.11908
## Oct 2020       13.04615 12.97474 13.11756 12.93693 13.15536
## Nov 2020       13.06647 12.99098 13.14196 12.95101 13.18192
## Dec 2020       13.21339 13.13380 13.29298 13.09167 13.33511
## Jan 2021       12.95885 12.87515 13.04256 12.83084 13.08687
## Feb 2021       12.95407 12.86623 13.04191 12.81973 13.08840
## Mar 2021       13.08124 12.98926 13.17322 12.94057 13.22191
## Apr 2021       13.05803 12.96190 13.15416 12.91102 13.20505
## May 2021       13.11673 13.01645 13.21702 12.96336 13.27011
## Jun 2021       13.08579 12.98134 13.19023 12.92605 13.24552
## Jul 2021       13.08591 12.97730 13.19452 12.91981 13.25202
## Aug 2021       13.11120 12.99842 13.22397 12.93872 13.28367
## Sep 2021       13.03321 12.91627 13.15015 12.85437 13.21205
```

From the STL graph, we can see that the seasonal effect and long-term trend are independent. Therefore, I choose the additive model.

Seasonal adjusted plot

```
decompose_lretail <- decompose(lretail, "additive")
adjust_lretail <- lretail-decompose_lretail$seasonal
plot(adjust_lretail)
```

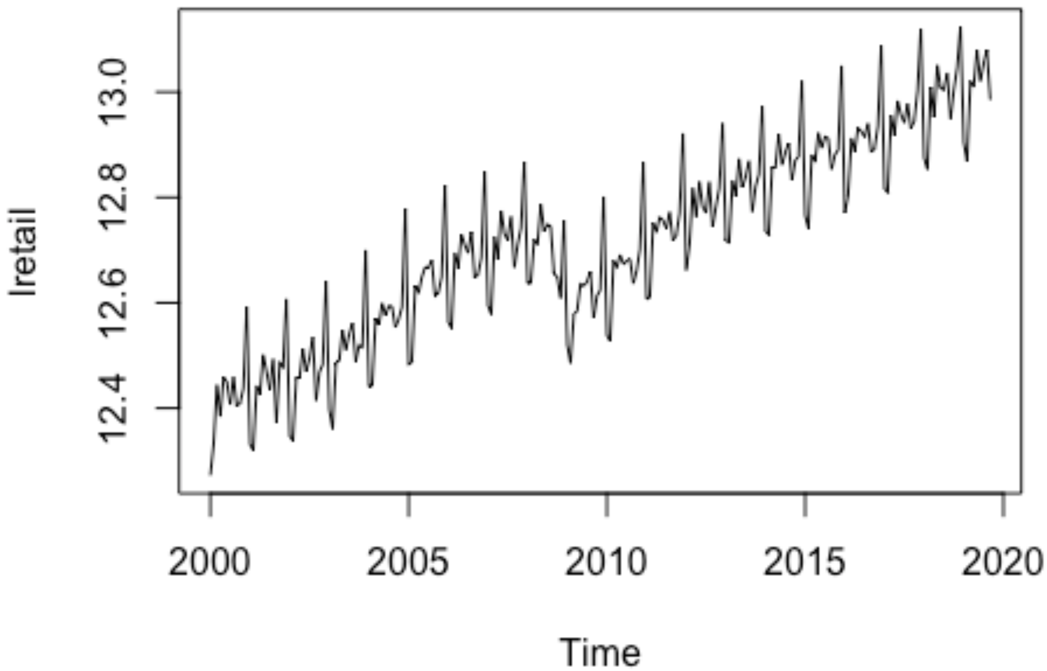


After the seasonal adjustment, I think my model is ok. From the graph, we can see an increasing trend of the data with seasonal trend.

Improve the model through ets

```
plot(lretail,s.window="periodic")

## Warning in plot.window(xlim, ylim, log, ...): "s.window"不是图形参数

## Warning in title(main = main, xlab = xlab, ylab = ylab, ...): "s.win
dow"不
## 是图形参数

## Warning in axis(1, ...): "s.window"不是图形参数

## Warning in axis(2, ...): "s.window"不是图形参数
```

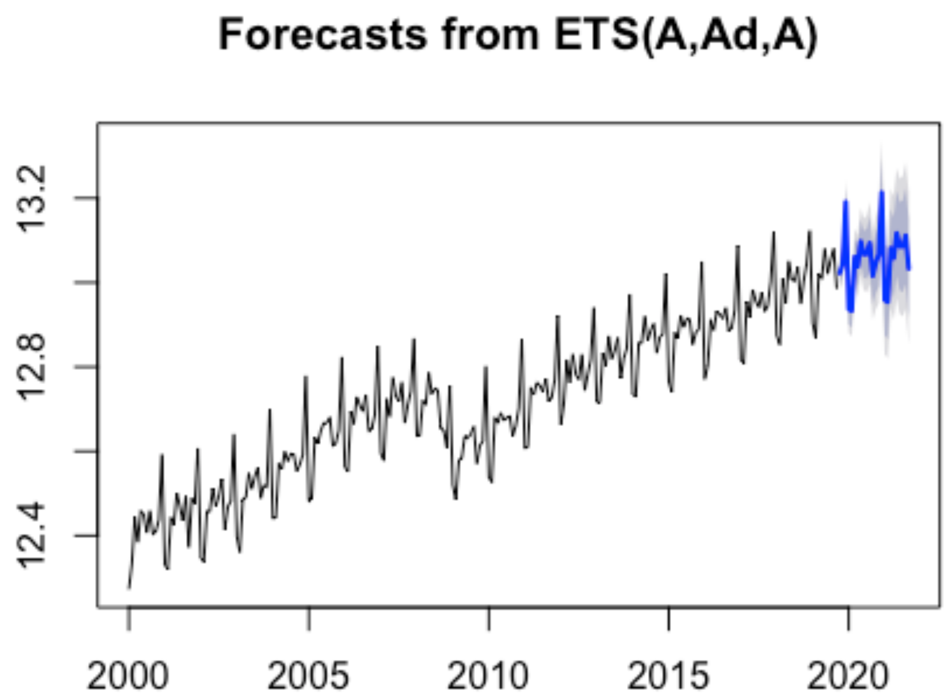
```
## Warning in box(...): "s.window"不是图形参数
```



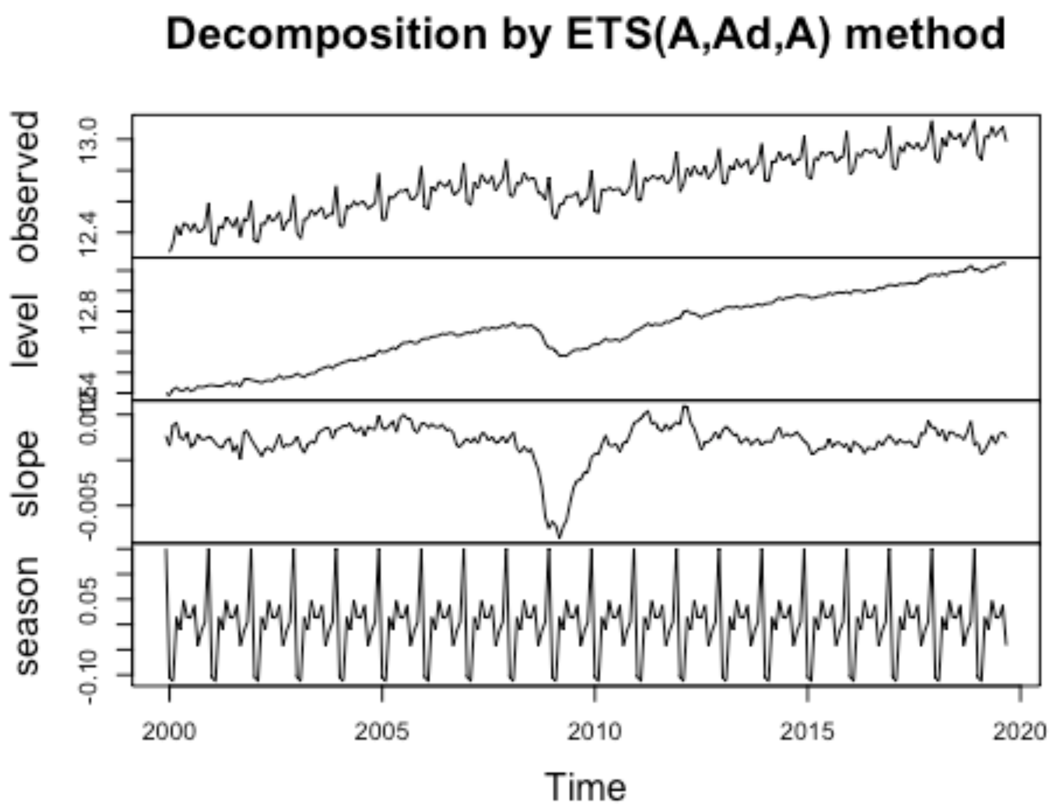

```
forecast(lretail)

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Oct 2019       13.02239 12.99672 13.04806 12.98313 13.06165
## Nov 2019       13.04341 13.01429 13.07253 12.99888 13.08794
## Dec 2019       13.19101 13.15835 13.22366 13.14106 13.24095
## Jan 2020       12.93712 12.90084 12.97341 12.88163 12.99261
## Feb 2020       12.93297 12.89299 12.97296 12.87183 12.99412
## Mar 2020       13.06076 13.01701 13.10451 12.99386 13.12767
## Apr 2020       13.03816 12.99059 13.08573 12.96540 13.11091
## May 2020       13.09743 13.04599 13.14888 13.01876 13.17611
## Jun 2020       13.06705 13.01169 13.12242 12.98238 13.15173
## Jul 2020       13.06773 13.00840 13.12706 12.97699 13.15846
## Aug 2020       13.09354 13.03022 13.15687 12.99669 13.19039
## Sep 2020       13.01607 12.94872 13.08343 12.91306 13.11908
## Oct 2020       13.04615 12.97474 13.11756 12.93693 13.15536
## Nov 2020       13.06647 12.99098 13.14196 12.95101 13.18192
## Dec 2020       13.21339 13.13380 13.29298 13.09167 13.33511
## Jan 2021       12.95885 12.87515 13.04256 12.83084 13.08687
## Feb 2021       12.95407 12.86623 13.04191 12.81973 13.08840
## Mar 2021       13.08124 12.98926 13.17322 12.94057 13.22191
```

```
## Apr 2021       13.05803 12.96190 13.15416 12.91102 13.20505
## May 2021       13.11673 13.01645 13.21702 12.96336 13.27011
## Jun 2021       13.08579 12.98134 13.19023 12.92605 13.24552
## Jul 2021       13.08591 12.97730 13.19452 12.91981 13.25202
## Aug 2021       13.11120 12.99842 13.22397 12.93872 13.28367
## Sep 2021       13.03321 12.91627 13.15015 12.85437 13.21205
```

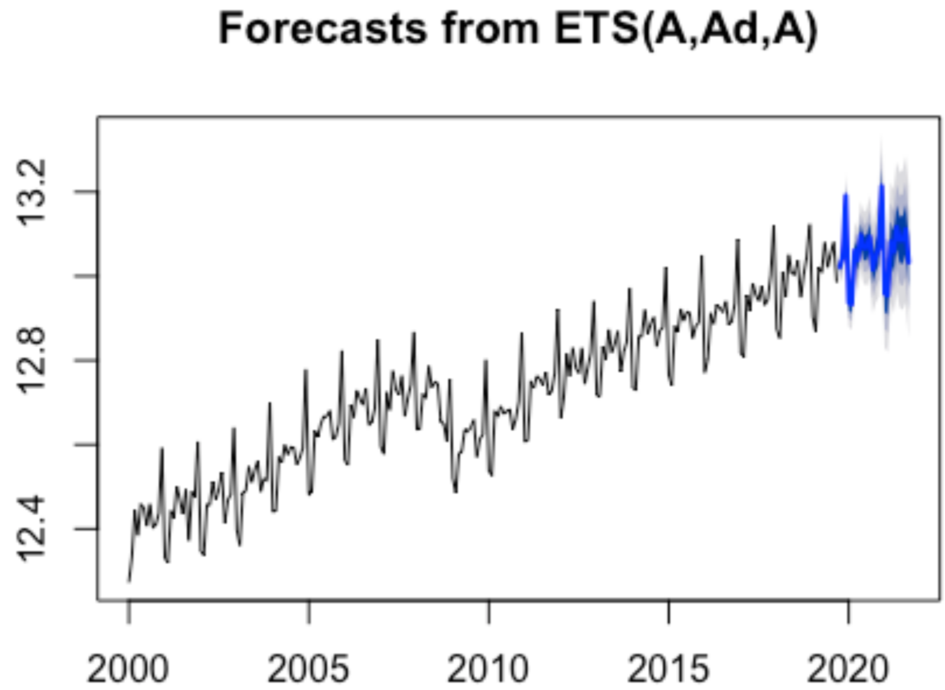
```
plot(forecast(lretail))
```




```
fit=ets(lretail)
plot(fit)
```




```r
plot(forecast(fit,level=c(50,80,95),h=24))
```



We can see that the seasonal effect is linked to time except for the year around 2008, it may be because of the financial crisis happened at that year.

Conclusion

From the analysis above, we can conclude that the US monthly retail sale has a trend that increases as the time goes by except for the time when economics crisis happens. Meantime, it has very significant seasonal effects. That is to say, people tend to buy more goods in specific months such as April, September and October, and fewer in other months. And the seasonal effects are stable as the time goes by.

Future work

I can try to analyze the specific time around 2008 to make the model more perfect by using the sectional functions. I believe that my model will become even better than this model in that way

Reference:

https://www.census.gov/retail/index.html