ARIMA Time Series Implementation

This is an implementation of [ARIMA](https://www.rdocumentation.org/packages/forecast/versions/8.12/topics/Arima) model for forecasting future sales for a retailer. The data consists product sales information of united states. There are three product categories mentioned in the data set Office supplies, Technology and Furniture

# Importing required libraries  
library(tseries)

## Warning: package 'tseries' was built under R version 3.5.3

library(forecast)

## Warning: package 'forecast' was built under R version 3.5.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.5.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(xlsx)

## Warning: package 'xlsx' was built under R version 3.5.3

library(readxl)

## Warning: package 'readxl' was built under R version 3.5.3

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.3

# Assigning working directory  
setwd("F:/Mayur/ANALYTICS/Data scientist/Data scienc with R/Time series")  
  
# importing data  
df = read\_excel("Sample - Superstore.xls",sheet = "Orders")

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L2236 / R2236C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L5276 / R5276C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L8800 / R8800C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L9148 / R9148C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L9149 / R9149C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L9150 / R9150C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L9388 / R9388C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L9389 / R9389C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L9390 / R9390C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L9391 / R9391C12: '05408'

## Warning in read\_fun(path = enc2native(normalizePath(path)), sheet\_i =  
## sheet, : Coercing text to numeric in L9743 / R9743C12: '05408'

View(df)  
glimpse(df)

## Observations: 9,994  
## Variables: 21  
## $ `Row ID` <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,...  
## $ `Order ID` <chr> "CA-2016-152156", "CA-2016-152156", "CA-2016-1...  
## $ `Order Date` <dttm> 2016-11-08, 2016-11-08, 2016-06-12, 2015-10-1...  
## $ `Ship Date` <dttm> 2016-11-11, 2016-11-11, 2016-06-16, 2015-10-1...  
## $ `Ship Mode` <chr> "Second Class", "Second Class", "Second Class"...  
## $ `Customer ID` <chr> "CG-12520", "CG-12520", "DV-13045", "SO-20335"...  
## $ `Customer Name` <chr> "Claire Gute", "Claire Gute", "Darrin Van Huff...  
## $ Segment <chr> "Consumer", "Consumer", "Corporate", "Consumer...  
## $ Country <chr> "United States", "United States", "United Stat...  
## $ City <chr> "Henderson", "Henderson", "Los Angeles", "Fort...  
## $ State <chr> "Kentucky", "Kentucky", "California", "Florida...  
## $ `Postal Code` <dbl> 42420, 42420, 90036, 33311, 33311, 90032, 9003...  
## $ Region <chr> "South", "South", "West", "South", "South", "W...  
## $ `Product ID` <chr> "FUR-BO-10001798", "FUR-CH-10000454", "OFF-LA-...  
## $ Category <chr> "Furniture", "Furniture", "Office Supplies", "...  
## $ `Sub-Category` <chr> "Bookcases", "Chairs", "Labels", "Tables", "St...  
## $ `Product Name` <chr> "Bush Somerset Collection Bookcase", "Hon Delu...  
## $ Sales <dbl> 261.9600, 731.9400, 14.6200, 957.5775, 22.3680...  
## $ Quantity <dbl> 2, 3, 2, 5, 2, 7, 4, 6, 3, 5, 9, 4, 3, 3, 5, 3...  
## $ Discount <dbl> 0.00, 0.00, 0.00, 0.45, 0.20, 0.00, 0.00, 0.20...  
## $ Profit <dbl> 41.9136, 219.5820, 6.8714, -383.0310, 2.5164, ...

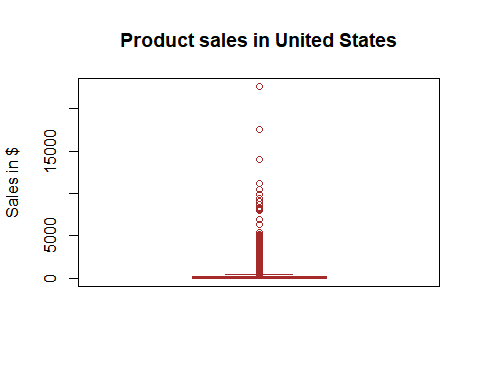
colSums(is.na(df)) # no missing values

## Row ID Order ID Order Date Ship Date Ship Mode   
## 0 0 0 0 0   
## Customer ID Customer Name Segment Country City   
## 0 0 0 0 0   
## State Postal Code Region Product ID Category   
## 0 0 0 0 0   
## Sub-Category Product Name Sales Quantity Discount   
## 0 0 0 0 0   
## Profit   
## 0

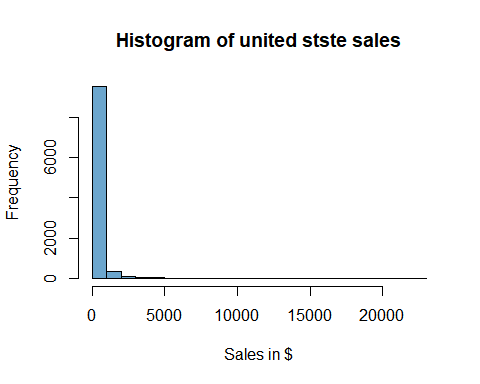
summary(df$Sales)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.444 17.280 54.490 229.858 209.940 22638.480

boxplot(df$Sales, main = "Product sales in United States", ylab="Sales in $",  
 color = "orange", border = "brown",  
 horizontal = FALSE, notch = TRUE)



hist(df$Sales,breaks = 20,col = "skyblue3",main = "Histogram of united stste sales",xlab = "Sales in $")



## observations with sales >=12000  
View(df %>% filter(Sales>=10000)) # techonology related products

length(unique(df$`Product Name`))

## [1] 1850

length(unique(df$`Product ID`))

## [1] 1862

# similar product id is assigned to different product names

length(unique(df$`Customer ID`))

## [1] 793

length(unique(df$`Customer Name`))

## [1] 793

product\_list = df %>% group\_by(`Product ID`,`Product Name`) %>%   
 summarise(count = n()) %>% arrange(`Product ID`)  
  
length(unique(product\_list$`Product Name`))

## [1] 1850

length(unique(product\_list$`Product ID`))

## [1] 1862

dim(product\_list)

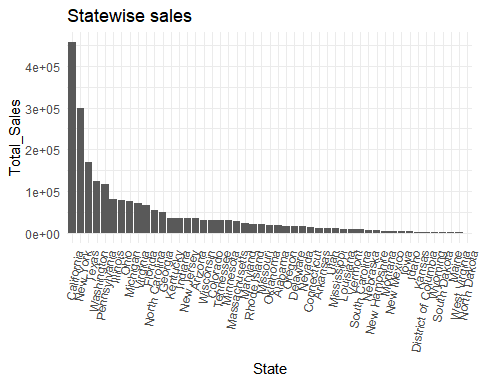
## [1] 1894 3

product\_id = product\_list %>% group\_by(`Product ID`) %>% mutate(count = row\_number())  
  
## finding out the prducts with similar product id  
similar\_id = product\_id %>% filter(count >= 2)  
  
View(df %>% filter(`Product ID` %in% similar\_id$`Product ID`  
 ) %>%   
 arrange(`Product ID`))  
  
View(df %>% filter(`Product Name`%in% similar\_id$`Product Name`) %>%   
 arrange(`Product ID`))  
  
# keeping the records though it has similar product id  
# the records are less than 4% and sales value will not have major impact with inclusion of these records

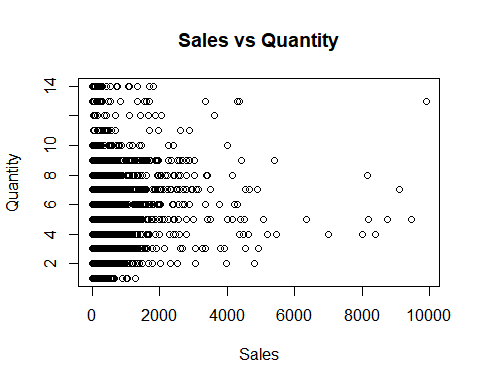
# total countries in the data set  
table(df$Country)

##   
## United States   
## 9989

# only united states  
  
# total states in the data set  
# visualising top sales contributors  
df %>% group\_by(State) %>% summarise(Total\_Sales = sum(Sales)) %>%   
 ggplot() + geom\_bar(aes(reorder(State,-Total\_Sales),Total\_Sales),stat = "identity")+  
 theme\_minimal() +   
 theme(axis.text.x = element\_text(angle = 80, hjust = 1)) +   
 labs(title = 'Statewise sales',x = 'State')



# California and New York are the top contributors  
  
View(df %>% group\_by(State) %>% summarise(N = n(),  
 Sales\_Max = max(Sales),  
 Sales\_min = min(Sales),  
 Sales\_Total = sum(Sales),  
 Sales\_mean = mean(Sales))  
 %>% arrange(desc(Sales\_Total)))  
  
View(df %>% group\_by(State,Category) %>% summarise(N = n(),  
 Sales\_Max = max(Sales),  
 Sales\_min = min(Sales),  
 Sales\_Total = sum(Sales),  
 Sales\_mean = mean(Sales))  
 %>% arrange(State,desc(Sales\_Total)))  
  
# from the top 5 sales contributors only washington has high sales contribution by furniture category  
# all other top 4 sales contributord are having technology as highest sales contributor  
  
# relation between quantity and sales  
plot(df$Sales,df$Quantity,xlab = "Sales",ylab = "Quantity",main = "Sales vs Quantity")



## which category has maximum quantity sold  
df %>% group\_by(Category) %>% summarise(total\_quantity = sum(Quantity),  
 total\_sales = sum(Sales)) %>% arrange(desc(total\_quantity))

## # A tibble: 3 x 3  
## Category total\_quantity total\_sales  
## <chr> <dbl> <dbl>  
## 1 Office Supplies 22906 719047.  
## 2 Furniture 8028 742000.  
## 3 Technology 6917 760316.

# Office supplies are having high quantity numbers where as technology sales amount is higher  
# may be technology items are more expensive

# pulling out year and month out of order date variable  
library(lubridate)

## Warning: package 'lubridate' was built under R version 3.5.3

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

summary(df$`Order Date`)

## Min. 1st Qu. Median   
## "2014-01-03 00:00:00" "2015-05-23 00:00:00" "2016-06-26 00:00:00"   
## Mean 3rd Qu. Max.   
## "2016-04-29 22:09:51" "2017-05-14 00:00:00" "2017-12-30 00:00:00"

df = df %>% mutate(month\_year = format(as.Date(`Order Date`),"%Y-%m"))

# Exploring sub category vaiable  
df %>% group\_by(`Sub-Category`) %>% summarise(Freq = n(),  
 Qty = sum(Quantity)) %>%   
 arrange(desc(Qty))

## # A tibble: 17 x 3  
## `Sub-Category` Freq Qty  
## <chr> <int> <dbl>  
## 1 Binders 1523 5974  
## 2 Paper 1370 5178  
## 3 Furnishings 957 3563  
## 4 Phones 889 3289  
## 5 Storage 846 3158  
## 6 Art 796 3000  
## 7 Accessories 775 2976  
## 8 Chairs 617 2356  
## 9 Appliances 466 1729  
## 10 Labels 364 1400  
## 11 Tables 319 1241  
## 12 Fasteners 217 914  
## 13 Envelopes 254 906  
## 14 Bookcases 228 868  
## 15 Supplies 190 647  
## 16 Machines 114 434  
## 17 Copiers 64 218

# Papers and Binders are the top 2 items in terms of quantity sold  
df %>% group\_by(`Sub-Category`) %>% summarise(Freq = n(),  
 Qty = sum(Quantity),  
 Total\_Sale = sum(Sales)) %>%   
 arrange(desc(Total\_Sale))

## # A tibble: 17 x 4  
## `Sub-Category` Freq Qty Total\_Sale  
## <chr> <int> <dbl> <dbl>  
## 1 Phones 889 3289 330007.  
## 2 Chairs 617 2356 328449.  
## 3 Storage 846 3158 223844.  
## 4 Tables 319 1241 206966.  
## 5 Binders 1523 5974 203413.  
## 6 Accessories 775 2976 167380.  
## 7 Machines 114 434 166600.  
## 8 Bookcases 228 868 114880.  
## 9 Appliances 466 1729 107532.  
## 10 Copiers 64 218 96328.  
## 11 Furnishings 957 3563 91705.  
## 12 Paper 1370 5178 78479.  
## 13 Supplies 190 647 46674.  
## 14 Art 796 3000 27119.  
## 15 Envelopes 254 906 16476.  
## 16 Labels 364 1400 12486.  
## 17 Fasteners 217 914 3024.

# sales is dominated by phones and chairs sub category  
# checking the sales percentage share of each category  
  
Sales\_Share = df %>% group\_by(Category) %>%   
 summarise(Sales = sum(Sales))  
  
Sales\_Share %>% mutate(Share\_Per = paste0(round(Sales/sum(Sales)\*100,2),"%")) %>%   
 arrange(desc(Share\_Per))

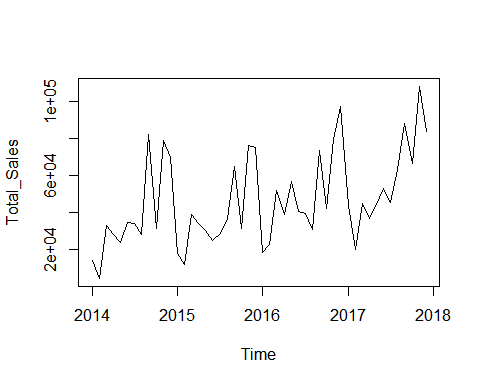
## # A tibble: 3 x 3  
## Category Sales Share\_Per  
## <chr> <dbl> <chr>   
## 1 Technology 760316. 34.23%   
## 2 Furniture 742000. 33.4%   
## 3 Office Supplies 719047. 32.37%

# all three categories seems to have quiet equal share

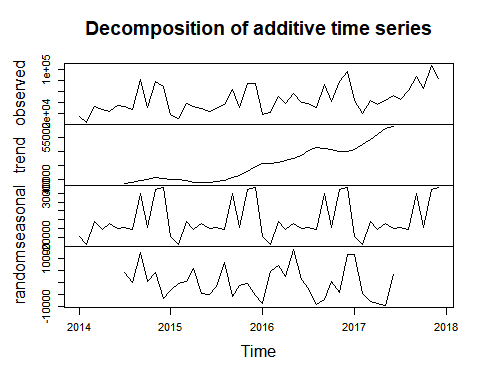
# Buidling time series model  
# forecasting at united states level  
ts\_data = df %>%group\_by(month\_year) %>% summarise(Total\_Sales = sum(Sales)) %>%   
 arrange(month\_year) %>% select(Total\_Sales)  
  
# converting data into ts class  
ts\_df = ts(ts\_data,frequency = 12,start = c(2014,1))  
end(ts\_df)

## [1] 2017 12

plot.ts(ts\_df)



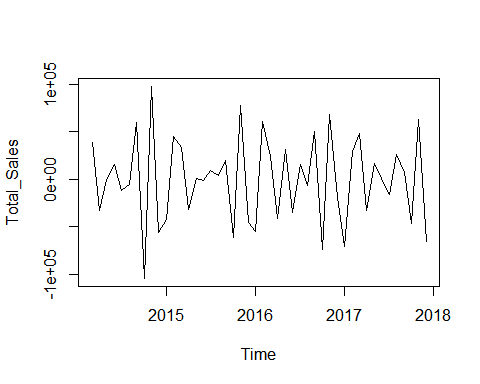
plot(decompose(ts\_df))



# statistical check for stationarity  
adf.test(ts\_df) #p-value is 0.07

##   
## Augmented Dickey-Fuller Test  
##   
## data: ts\_df  
## Dickey-Fuller = -3.3304, Lag order = 3, p-value = 0.07796  
## alternative hypothesis: stationary

# differencing time series  
ts\_diff = diff(ts\_df,differences = 2)  
plot.ts(ts\_diff)

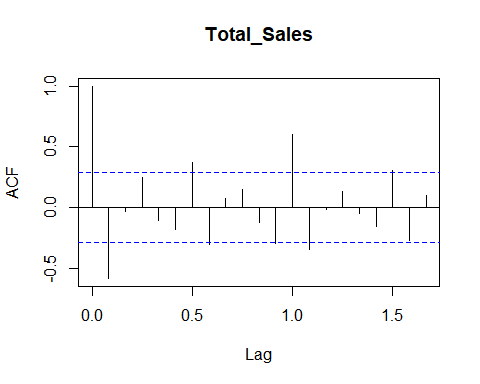


adf.test(ts\_diff)

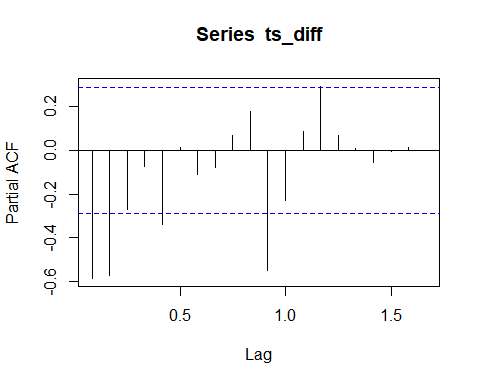
## Warning in adf.test(ts\_diff): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: ts\_diff  
## Dickey-Fuller = -5.2651, Lag order = 3, p-value = 0.01  
## alternative hypothesis: stationary

# time series looks stationary  
  
# selecting p,d,q  
acf(ts\_diff, lag.max = 20)



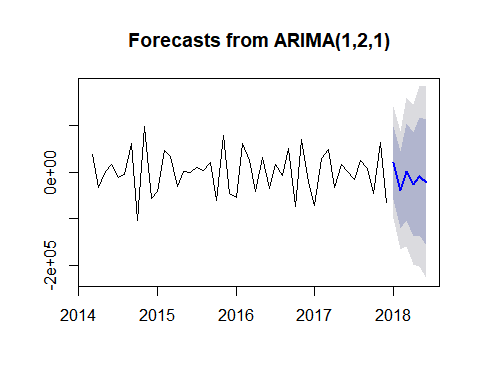
pacf(ts\_diff, lag.max = 20)



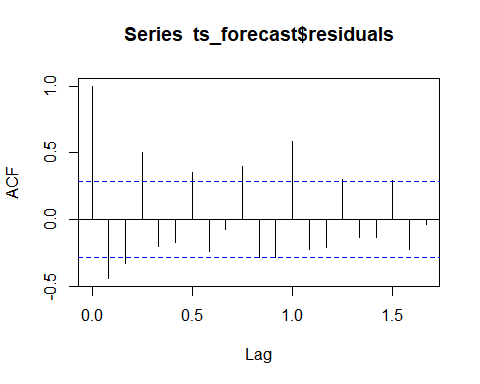
# selecting p as 1 and q as 0  
ts\_model = arima(ts\_diff,order = c(1,2,1))  
ts\_model

##   
## Call:  
## arima(x = ts\_diff, order = c(1, 2, 1))  
##   
## Coefficients:  
## ar1 ma1  
## -0.6761 -1.0000  
## s.e. 0.1125 0.0571  
##   
## sigma^2 estimated as 3.613e+09: log likelihood = -549.32, aic = 1104.64

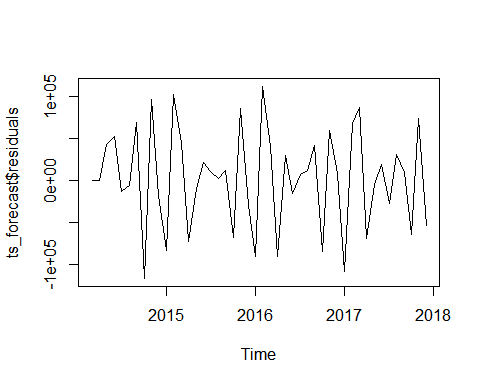
# forecasting for next two years  
ts\_forecast = forecast:::forecast.Arima(ts\_model,h=6)  
plot(ts\_forecast)



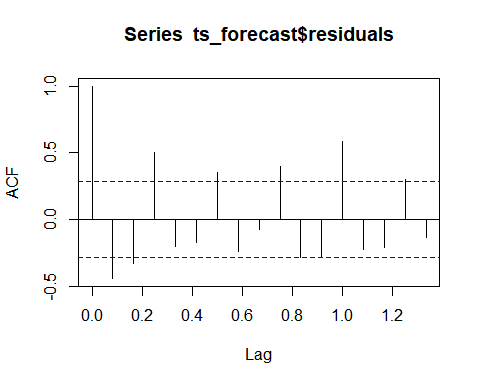
# checking residuals  
acf(ts\_forecast$residuals, lag.max = 20)



plot.ts(ts\_forecast$residuals) # residuals looks random



acf(ts\_forecast$residuals)



Box.test(ts\_forecast$residuals,lag = 20,type = 'Ljung-Box')

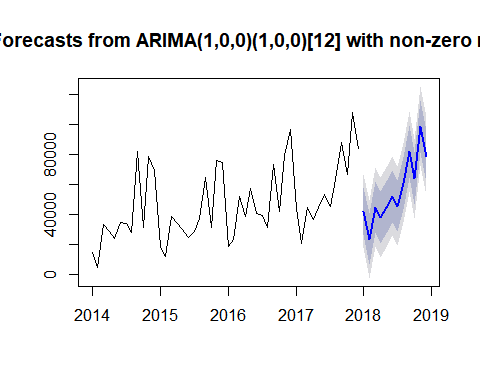
##   
## Box-Ljung test  
##   
## data: ts\_forecast$residuals  
## X-squared = 110.87, df = 20, p-value = 1.366e-14

# p value is < 0.05  
# residuals are not independent  
# forecasting needs improvement  
# model is not performing well

# using auto.arima function  
ts\_auto\_arima = auto.arima(ts\_df,seasonal = TRUE,stationary = TRUE)  
ts\_auto\_arima

## Series: ts\_df   
## ARIMA(1,0,0)(1,0,0)[12] with non-zero mean   
##   
## Coefficients:  
## ar1 sar1 mean  
## 0.4011 0.8529 47439.420  
## s.e. 0.1308 0.0582 9940.418  
##   
## sigma^2 estimated as 154065998: log likelihood=-526.92  
## AIC=1061.84 AICc=1062.77 BIC=1069.32

auto\_arima\_forecast = forecast:::forecast.Arima(ts\_auto\_arima,h=12)  
plot(auto\_arima\_forecast)



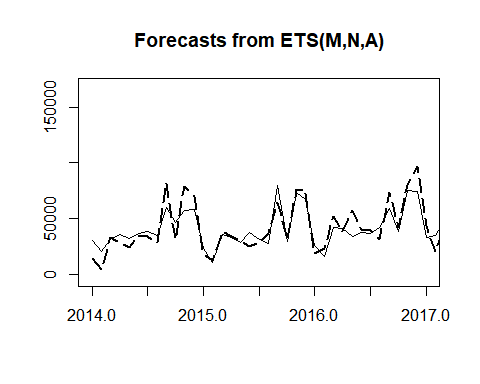
auto\_arima\_forecast

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2018 42122.40 26215.358 58029.45 17794.677 66450.13  
## Feb 2018 23346.14 6207.008 40485.28 -2865.902 49558.19  
## Mar 2018 44870.34 27541.129 62199.55 18367.598 71373.08  
## Apr 2018 37974.99 20615.388 55334.59 11425.770 64524.21  
## May 2018 44667.47 27302.985 62031.96 18110.781 71224.16  
## Jun 2018 52142.10 34776.829 69507.38 25584.208 78700.00  
## Jul 2018 45574.47 28209.071 62939.87 19016.384 72132.56  
## Aug 2018 60810.63 43445.209 78176.05 34252.511 87368.75  
## Sep 2018 81919.33 64553.904 99284.75 55361.204 108477.45  
## Oct 2018 63761.71 46396.287 81127.14 37203.587 90319.84  
## Nov 2018 99048.46 81683.036 116413.88 72490.336 125606.58  
## Dec 2018 78477.26 61111.831 95842.68 51919.131 105035.38

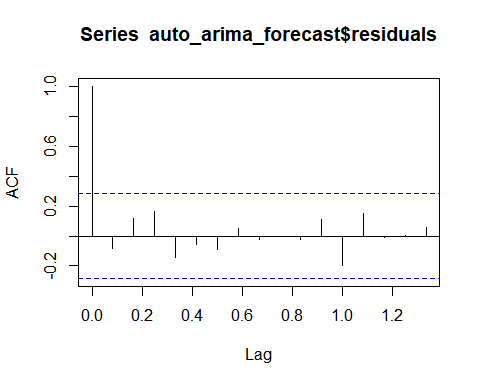
library(TSPred)

## Warning: package 'TSPred' was built under R version 3.5.3

# predicted vd actual graph  
plotarimapred(ts\_df,auto\_arima\_forecast$fitted,xlim = c(2014,2017))

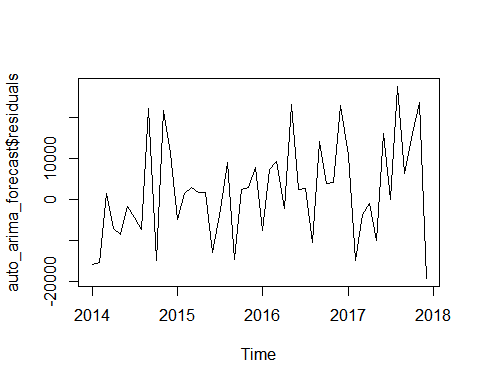


# checking residuals auto corelation  
acf(auto\_arima\_forecast$residuals)



# no auto corelation

# checking residuals  
plot(auto\_arima\_forecast$residuals)



# residuals are random  
  
Box.test(auto\_arima\_forecast$residuals,lag = 20,type = "Ljung-Box")

##   
## Box-Ljung test  
##   
## data: auto\_arima\_forecast$residuals  
## X-squared = 18.663, df = 20, p-value = 0.5438

# p value is 0.33 i.e. we can not reject null hypothesis  
# we can say residuals are independent  
# model is performing well

# mean squared error  
mean(auto\_arima\_forecast$residuals)

## [1] 2105.651

sqrt(mean(auto\_arima\_forecast$residuals))

## [1] 45.88737

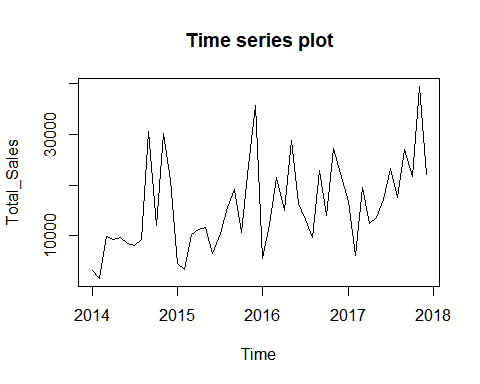
# checking mean absolute percentage error  
mean((abs((ts\_df-auto\_arima\_forecast$fitted)/ts\_df))\*100)

## [1] 28.95379

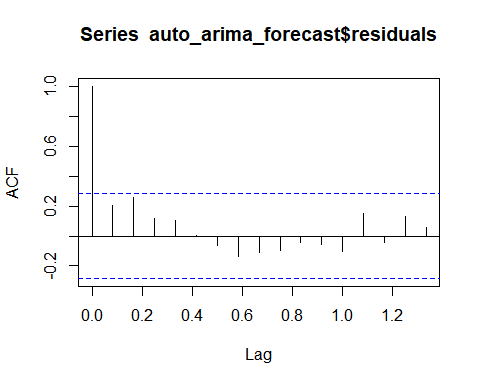
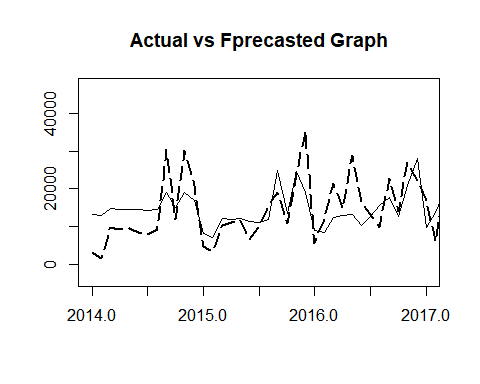
# the model is off by 29%  
# the model is performing well at (100-29) 71%

Creating forecast at a category level

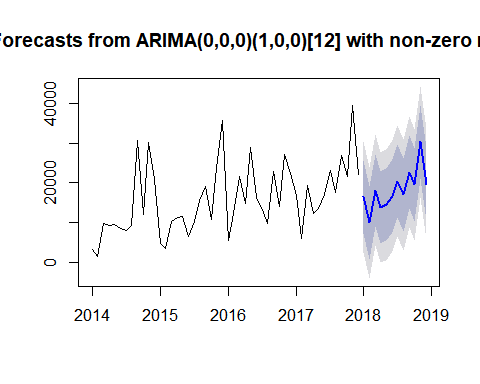
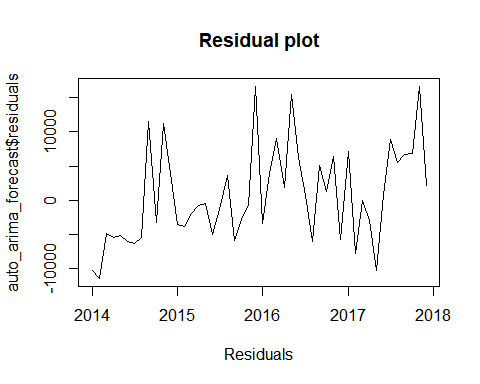
Category\_Forecast = function(data,Catg)  
{  
 ts\_data = data %>% filter(Category == Catg) %>%  
 group\_by(month\_year) %>% summarise(Total\_Sales = sum(Sales)) %>%  
 arrange(month\_year) %>% select(Total\_Sales)  
  
 # converting data into ts class  
 ts\_df = ts(ts\_data,frequency = 12,start = c(2014,1))  
 end(ts\_df)  
  
 print(plot.ts(ts\_df,main="Time series plot"))   
 #plot(decompose(ts\_df))  
  
 # statistical check for stationarity  
 adf.test(ts\_df)   
   
 # checking with auto arima  
 ts\_auto\_arima = auto.arima(ts\_df,seasonal = TRUE,stationary = TRUE)  
  
 auto\_arima\_forecast = forecast:::forecast.Arima(ts\_auto\_arima,h=12)  
 #print(auto\_arima\_forecast)  
   
 # predicted vd actual graph  
 library(TSPred)  
 plotarimapred(ts\_df,auto\_arima\_forecast$fitted,xlim = c(2014,2017),  
 main = "Actual vs Fprecasted Graph")  
   
 # checking residuals auto corelation  
 acf(auto\_arima\_forecast$residuals)  
   
 print("checking stationarity of residuals")  
 print(adf.test(auto\_arima\_forecast$residuals))  
   
 # checking residuals  
 plot(auto\_arima\_forecast$residuals,main="Residual plot",xlab="Residuals")   
 # residuals are random  
   
 print(plot(auto\_arima\_forecast))  
   
 # mean squared error  
 mean(auto\_arima\_forecast$residuals)  
 sqrt(mean(auto\_arima\_forecast$residuals))  
   
 # checking mean absolute percentage error  
 print("Mean absolute percentage error")  
 print(mean((abs((ts\_df-auto\_arima\_forecast$fitted)/ts\_df))\*100))  
   
 print(Box.test(auto\_arima\_forecast$residuals,lag = 20,type = "Ljung-Box"))  
  
 forecasted\_values = as.data.frame(auto\_arima\_forecast)  
   
 return(forecasted\_values)  
}  
  
Technology\_forecast = Category\_Forecast(data = df,Catg = "Technology")



## NULL



## [1] "checking stationarity of residuals"  
##   
## Augmented Dickey-Fuller Test  
##   
## data: auto\_arima\_forecast$residuals  
## Dickey-Fuller = -2.6799, Lag order = 3, p-value = 0.3026  
## alternative hypothesis: stationary

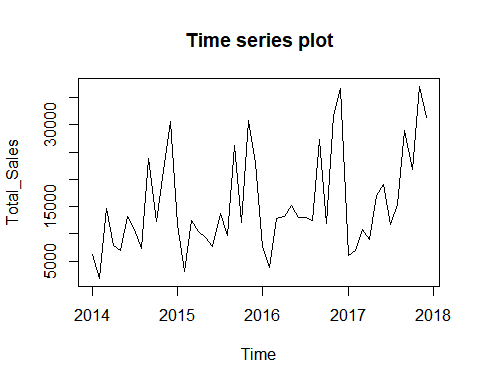


## $mean  
## Jan Feb Mar Apr May Jun Jul  
## 2018 16445.830 9910.238 18091.389 13790.537 14512.833 16646.089 20399.695  
## Aug Sep Oct Nov Dec  
## 2018 16986.763 22678.342 19450.901 30294.550 19651.873  
##   
## $lower  
## 80% 95%  
## Jan 2018 7338.2681 2517.0160  
## Feb 2018 802.6765 -4018.5756  
## Mar 2018 8983.8268 4162.5747  
## Apr 2018 4682.9754 -138.2767  
## May 2018 5405.2711 584.0190  
## Jun 2018 7538.5273 2717.2752  
## Jul 2018 11292.1334 6470.8813  
## Aug 2018 7879.2013 3057.9492  
## Sep 2018 13570.7796 8749.5275  
## Oct 2018 10343.3391 5522.0870  
## Nov 2018 21186.9879 16365.7359  
## Dec 2018 10544.3114 5723.0593  
##   
## $upper  
## 80% 95%  
## Jan 2018 25553.39 30374.64  
## Feb 2018 19017.80 23839.05  
## Mar 2018 27198.95 32020.20  
## Apr 2018 22898.10 27719.35  
## May 2018 23620.39 28441.65  
## Jun 2018 25753.65 30574.90  
## Jul 2018 29507.26 34328.51  
## Aug 2018 26094.33 30915.58  
## Sep 2018 31785.90 36607.16  
## Oct 2018 28558.46 33379.72  
## Nov 2018 39402.11 44223.36  
## Dec 2018 28759.44 33580.69  
##   
## [1] "Mean absolute percentage error"  
## [1] 58.10326  
##   
## Box-Ljung test  
##   
## data: auto\_arima\_forecast$residuals  
## X-squared = 21.791, df = 20, p-value = 0.3519

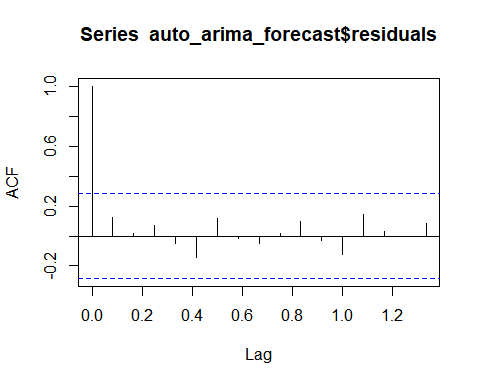
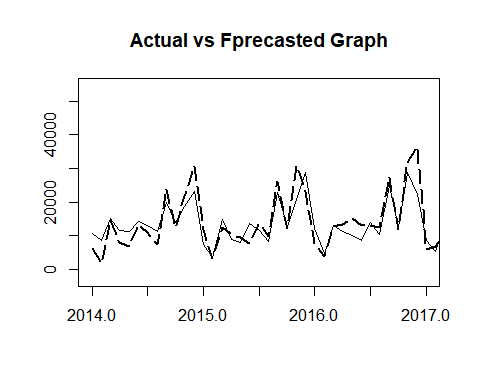
Technology\_forecast

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2018 16445.830 7338.2681 25553.39 2517.0160 30374.64  
## Feb 2018 9910.238 802.6765 19017.80 -4018.5756 23839.05  
## Mar 2018 18091.389 8983.8268 27198.95 4162.5747 32020.20  
## Apr 2018 13790.537 4682.9754 22898.10 -138.2767 27719.35  
## May 2018 14512.833 5405.2711 23620.39 584.0190 28441.65  
## Jun 2018 16646.089 7538.5273 25753.65 2717.2752 30574.90  
## Jul 2018 20399.695 11292.1334 29507.26 6470.8813 34328.51  
## Aug 2018 16986.763 7879.2013 26094.33 3057.9492 30915.58  
## Sep 2018 22678.342 13570.7796 31785.90 8749.5275 36607.16  
## Oct 2018 19450.901 10343.3391 28558.46 5522.0870 33379.72  
## Nov 2018 30294.550 21186.9879 39402.11 16365.7359 44223.36  
## Dec 2018 19651.873 10544.3114 28759.44 5723.0593 33580.69

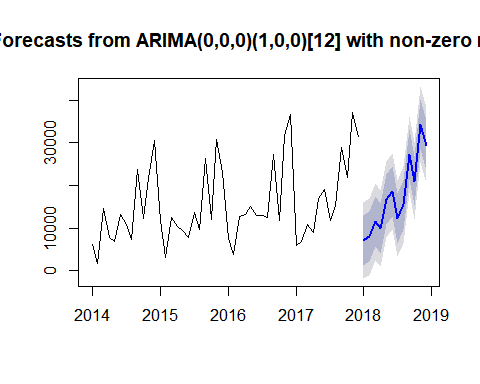
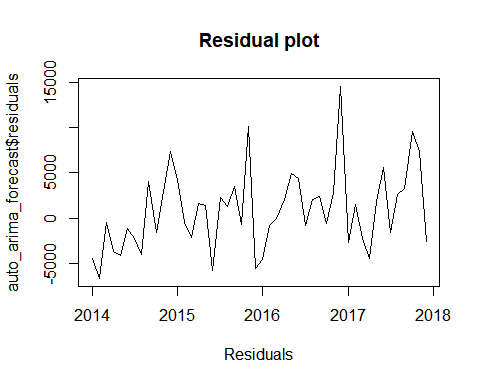
# forecasting for furniture product  
Furniture\_forecast = Category\_Forecast(data = df,Catg = "Furniture")



## NULL



## [1] "checking stationarity of residuals"  
##   
## Augmented Dickey-Fuller Test  
##   
## data: auto\_arima\_forecast$residuals  
## Dickey-Fuller = -3.7778, Lag order = 3, p-value = 0.02829  
## alternative hypothesis: stationary



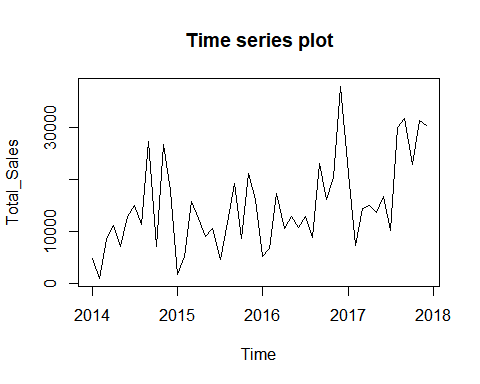
## $mean  
## Jan Feb Mar Apr May Jun Jul  
## 2018 7180.888 7968.165 11481.885 9887.372 16772.928 18562.485 12284.232  
## Aug Sep Oct Nov Dec  
## 2018 15450.467 27304.775 21071.392 34309.786 29380.721  
##   
## $lower  
## 80% 95%  
## Jan 2018 1334.485 -1760.4139  
## Feb 2018 2121.762 -973.1370  
## Mar 2018 5635.482 2540.5832  
## Apr 2018 4040.969 946.0698  
## May 2018 10926.525 7831.6260  
## Jun 2018 12716.082 9621.1835  
## Jul 2018 6437.829 3342.9301  
## Aug 2018 9604.064 6509.1656  
## Sep 2018 21458.372 18363.4729  
## Oct 2018 15224.989 12130.0904  
## Nov 2018 28463.383 25368.4846  
## Dec 2018 23534.318 20439.4187  
##   
## $upper  
## 80% 95%  
## Jan 2018 13027.29 16122.19  
## Feb 2018 13814.57 16909.47  
## Mar 2018 17328.29 20423.19  
## Apr 2018 15733.77 18828.67  
## May 2018 22619.33 25714.23  
## Jun 2018 24408.89 27503.79  
## Jul 2018 18130.63 21225.53  
## Aug 2018 21296.87 24391.77  
## Sep 2018 33151.18 36246.08  
## Oct 2018 26917.80 30012.69  
## Nov 2018 40156.19 43251.09  
## Dec 2018 35227.12 38322.02  
##   
## [1] "Mean absolute percentage error"  
## [1] 31.54591  
##   
## Box-Ljung test  
##   
## data: auto\_arima\_forecast$residuals  
## X-squared = 8.683, df = 20, p-value = 0.9863

Furniture\_forecast

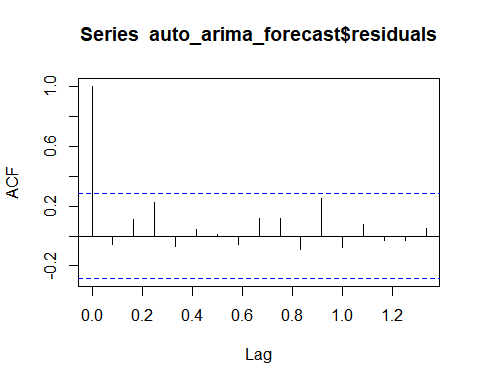
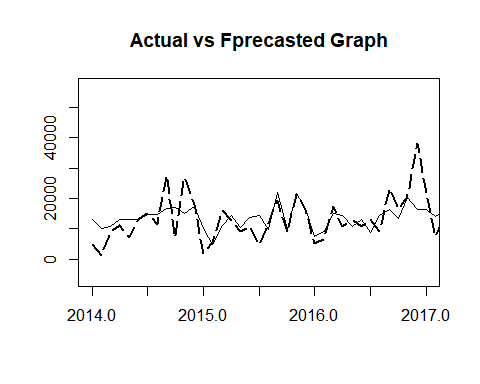
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2018 7180.888 1334.485 13027.29 -1760.4139 16122.19  
## Feb 2018 7968.165 2121.762 13814.57 -973.1370 16909.47  
## Mar 2018 11481.885 5635.482 17328.29 2540.5832 20423.19  
## Apr 2018 9887.372 4040.969 15733.77 946.0698 18828.67  
## May 2018 16772.928 10926.525 22619.33 7831.6260 25714.23  
## Jun 2018 18562.485 12716.082 24408.89 9621.1835 27503.79  
## Jul 2018 12284.232 6437.829 18130.63 3342.9301 21225.53  
## Aug 2018 15450.467 9604.064 21296.87 6509.1656 24391.77  
## Sep 2018 27304.775 21458.372 33151.18 18363.4729 36246.08  
## Oct 2018 21071.392 15224.989 26917.80 12130.0904 30012.69  
## Nov 2018 34309.786 28463.383 40156.19 25368.4846 43251.09  
## Dec 2018 29380.721 23534.318 35227.12 20439.4187 38322.02

Office supplies forecast

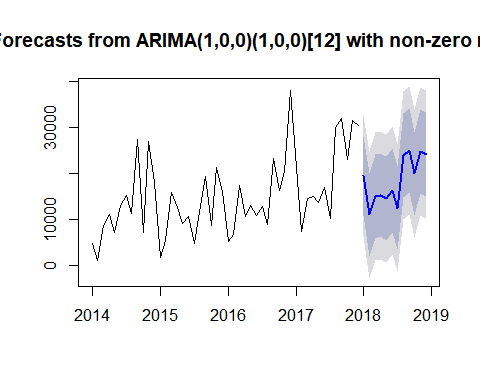
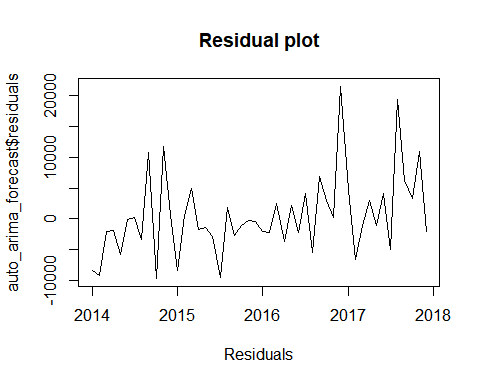
OS\_forecast = Category\_Forecast(df,"Office Supplies")



## NULL



## [1] "checking stationarity of residuals"  
##   
## Augmented Dickey-Fuller Test  
##   
## data: auto\_arima\_forecast$residuals  
## Dickey-Fuller = -3.4232, Lag order = 3, p-value = 0.06366  
## alternative hypothesis: stationary



## $mean  
## Jan Feb Mar Apr May Jun Jul  
## 2018 19468.12 11051.35 15056.17 15319.69 14536.12 16367.79 12510.86  
## Aug Sep Oct Nov Dec  
## 2018 23966.18 25027.34 19906.72 24782.49 24184.00  
##   
## $lower  
## 80% 95%  
## Jan 2018 10827.348 6253.199  
## Feb 2018 2009.235 -2777.371  
## Mar 2018 5976.833 1170.522  
## Apr 2018 6236.819 1428.640  
## May 2018 5452.914 644.557  
## Jun 2018 7284.550 2476.176  
## Jul 2018 3427.625 -1380.751  
## Aug 2018 14882.942 10074.566  
## Sep 2018 15944.097 11135.721  
## Oct 2018 10823.480 6015.104  
## Nov 2018 15699.254 10890.878  
## Dec 2018 15100.759 10292.383  
##   
## $upper  
## 80% 95%  
## Jan 2018 28108.90 32683.05  
## Feb 2018 20093.46 24880.07  
## Mar 2018 24135.51 28941.82  
## Apr 2018 24402.56 29210.73  
## May 2018 23619.32 28427.68  
## Jun 2018 25451.02 30259.40  
## Jul 2018 21594.10 26402.48  
## Aug 2018 33049.42 37857.79  
## Sep 2018 34110.57 38918.95  
## Oct 2018 28989.96 33798.33  
## Nov 2018 33865.73 38674.11  
## Dec 2018 33267.24 38075.61  
##   
## [1] "Mean absolute percentage error"  
## [1] 61.04858  
##   
## Box-Ljung test  
##   
## data: auto\_arima\_forecast$residuals  
## X-squared = 12.274, df = 20, p-value = 0.9063

OS\_forecast

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2018 19468.12 10827.348 28108.90 6253.199 32683.05  
## Feb 2018 11051.35 2009.235 20093.46 -2777.371 24880.07  
## Mar 2018 15056.17 5976.833 24135.51 1170.522 28941.82  
## Apr 2018 15319.69 6236.819 24402.56 1428.640 29210.73  
## May 2018 14536.12 5452.914 23619.32 644.557 28427.68  
## Jun 2018 16367.79 7284.550 25451.02 2476.176 30259.40  
## Jul 2018 12510.86 3427.625 21594.10 -1380.751 26402.48  
## Aug 2018 23966.18 14882.942 33049.42 10074.566 37857.79  
## Sep 2018 25027.34 15944.097 34110.57 11135.721 38918.95  
## Oct 2018 19906.72 10823.480 28989.96 6015.104 33798.33  
## Nov 2018 24782.49 15699.254 33865.73 10890.878 38674.11  
## Dec 2018 24184.00 15100.759 33267.24 10292.383 38075.61