Time series on SFRPC

sanjeeva

November 18, 2017

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



### Clear the Global Environment

rm(list=ls(all=TRUE))

### Library Call

library(forecast)  
library(lubridate)

## Warning: package 'lubridate' was built under R version 3.4.2

##   
## Attaching package: 'lubridate'

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

library(DataCombine)

## Warning: package 'DataCombine' was built under R version 3.4.2

library(imputeTS)

## Warning: package 'imputeTS' was built under R version 3.4.2

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.2

##   
## Attaching package: 'dplyr'

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

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

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

library(TTR)

## Warning: package 'TTR' was built under R version 3.4.2

library(graphics)  
library(data.table)

## Warning: package 'data.table' was built under R version 3.4.2

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

## The following object is masked from 'package:DataCombine':  
##   
## shift

## The following objects are masked from 'package:lubridate':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday,  
## week, yday, year

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:lubridate':  
##   
## here

library(zoo)

## Warning: package 'zoo' was built under R version 3.4.2

##   
## Attaching package: 'zoo'

## The following object is masked from 'package:imputeTS':  
##   
## na.locf

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

* To read RData type file use readRDS function

getwd()

## [1] "C:/Users/hp/Desktop/PHD Hackthon"

df=read.csv("Train.csv",header = T)  
sum(is.na(df))

## [1] 13

## Basic data View

head(df,30)

## Year Month ProductCategory Sales.In.ThousandDollars.  
## 1 2009 1 WomenClothing 1755  
## 2 2009 1 MenClothing 524  
## 3 2009 1 OtherClothing 936  
## 4 2009 2 WomenClothing 1729  
## 5 2009 2 MenClothing 496  
## 6 2009 2 OtherClothing 859  
## 7 2009 3 WomenClothing 2256  
## 8 2009 3 MenClothing 542  
## 9 2009 3 OtherClothing 921  
## 10 2009 4 WomenClothing 2662  
## 11 2009 4 MenClothing 669  
## 12 2009 4 OtherClothing 914  
## 13 2009 5 WomenClothing 2732  
## 14 2009 5 MenClothing 650  
## 15 2009 5 OtherClothing 989  
## 16 2009 6 WomenClothing 2220  
## 17 2009 6 MenClothing 607  
## 18 2009 6 OtherClothing 932  
## 19 2009 7 WomenClothing 2164  
## 20 2009 7 MenClothing 575  
## 21 2009 7 OtherClothing 901  
## 22 2009 8 WomenClothing 2371  
## 23 2009 8 MenClothing 551  
## 24 2009 8 OtherClothing 865  
## 25 2009 9 WomenClothing 2421  
## 26 2009 9 MenClothing 579  
## 27 2009 9 OtherClothing 819  
## 28 2009 10 WomenClothing 2579  
## 29 2009 10 MenClothing 610  
## 30 2009 10 OtherClothing 914

str(df)

## 'data.frame': 252 obs. of 4 variables:  
## $ Year : int 2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 ...  
## $ Month : int 1 1 1 2 2 2 3 3 3 4 ...  
## $ ProductCategory : Factor w/ 3 levels "MenClothing",..: 3 1 2 3 1 2 3 1 2 3 ...  
## $ Sales.In.ThousandDollars.: int 1755 524 936 1729 496 859 2256 542 921 2662 ...

summary(df)

## Year Month ProductCategory  
## Min. :2009 Min. : 1.00 MenClothing :84   
## 1st Qu.:2010 1st Qu.: 3.75 OtherClothing:84   
## Median :2012 Median : 6.50 WomenClothing:84   
## Mean :2012 Mean : 6.50   
## 3rd Qu.:2014 3rd Qu.: 9.25   
## Max. :2015 Max. :12.00   
##   
## Sales.In.ThousandDollars.  
## Min. : 471   
## 1st Qu.: 714   
## Median :1136   
## Mean :1747   
## 3rd Qu.:2804   
## Max. :5874   
## NA's :13

sum(is.na(df))

## [1] 13

## filling the NA values using KNN imputation

library(DMwR)

## Loading required package: lattice

## Loading required package: grid

##   
## Attaching package: 'DMwR'

## The following object is masked from 'package:plyr':  
##   
## join

Train\_Data1 <- centralImputation(df) #KNN Imputation  
sum(is.na(Train\_Data1))

## [1] 0

str(Train\_Data1)

## 'data.frame': 252 obs. of 4 variables:  
## $ Year : int 2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 ...  
## $ Month : int 1 1 1 2 2 2 3 3 3 4 ...  
## $ ProductCategory : Factor w/ 3 levels "MenClothing",..: 3 1 2 3 1 2 3 1 2 3 ...  
## $ Sales.In.ThousandDollars.: int 1755 524 936 1729 496 859 2256 542 921 2662 ...

class(Train\_Data1)

## [1] "data.frame"

Train\_Data1$ProductCategory <- as.character(Train\_Data1$ProductCategory)

### use the aggregator function to get the Date format

head(Train\_Data1)

## Year Month ProductCategory Sales.In.ThousandDollars.  
## 1 2009 1 WomenClothing 1755  
## 2 2009 1 MenClothing 524  
## 3 2009 1 OtherClothing 936  
## 4 2009 2 WomenClothing 1729  
## 5 2009 2 MenClothing 496  
## 6 2009 2 OtherClothing 859

Train\_data2<-transform(Train\_Data1, Date = as.Date(paste(Year, Month, 1, sep = "-")))  
  
str(Train\_data2)

## 'data.frame': 252 obs. of 5 variables:  
## $ Year : int 2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 ...  
## $ Month : int 1 1 1 2 2 2 3 3 3 4 ...  
## $ ProductCategory : chr "WomenClothing" "MenClothing" "OtherClothing" "WomenClothing" ...  
## $ Sales.In.ThousandDollars.: int 1755 524 936 1729 496 859 2256 542 921 2662 ...  
## $ Date : Date, format: "2009-01-01" "2009-01-01" ...

head(Train\_data2)

## Year Month ProductCategory Sales.In.ThousandDollars. Date  
## 1 2009 1 WomenClothing 1755 2009-01-01  
## 2 2009 1 MenClothing 524 2009-01-01  
## 3 2009 1 OtherClothing 936 2009-01-01  
## 4 2009 2 WomenClothing 1729 2009-02-01  
## 5 2009 2 MenClothing 496 2009-02-01  
## 6 2009 2 OtherClothing 859 2009-02-01

women\_data<-subset(Train\_Data1,Train\_Data1$ProductCategory=='WomenClothing')  
  
Men\_data<-subset(Train\_Data1,Train\_Data1$ProductCategory=='MenClothing')  
  
other\_data<-subset(Train\_Data1,Train\_Data1$ProductCategory=='OtherClothing')  
  
  
head(women\_data)

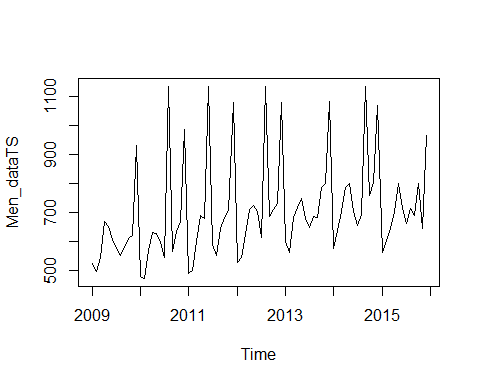
## Year Month ProductCategory Sales.In.ThousandDollars.  
## 1 2009 1 WomenClothing 1755  
## 4 2009 2 WomenClothing 1729  
## 7 2009 3 WomenClothing 2256  
## 10 2009 4 WomenClothing 2662  
## 13 2009 5 WomenClothing 2732  
## 16 2009 6 WomenClothing 2220

# Covert above dataframes in Timeseries dataframes--MEN DATA

Men\_dataTS<-ts(Men\_data[,4],start = c(2009,1),frequency = 12)  
Men\_dataTS

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2009 524 496 542 669 650 607 575 551 579 610 620 930  
## 2010 476 471 568 630 627 598 544 1136 563 634 669 988  
## 2011 487 497 599 690 677 1136 587 549 643 682 707 1081  
## 2012 525 545 623 711 725 703 613 1136 687 710 731 1080  
## 2013 599 560 682 718 749 678 648 687 681 785 798 1085  
## 2014 573 636 702 785 801 702 655 692 1136 757 803 1070  
## 2015 560 602 645 701 800 716 661 712 690 800 643 967

plot(Men\_dataTS)

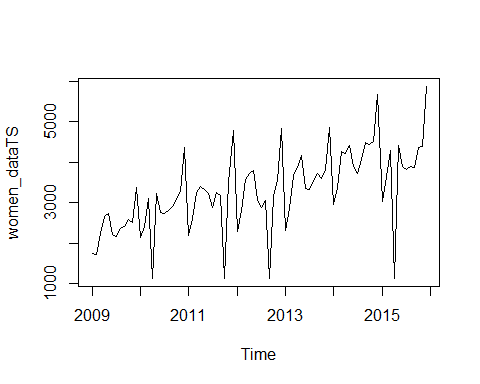


# Covert above dataframes in Timeseries dataframes--Women data

women\_dataTS<-ts(women\_data[,4],start = c(2009,1),frequency = 12)  
women\_dataTS

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2009 1755 1729 2256 2662 2732 2220 2164 2371 2421 2579 2521 3390  
## 2010 2142 2413 3102 1136 3221 2756 2749 2807 2919 3071 3276 4360  
## 2011 2193 2620 3266 3414 3340 3232 2896 3245 3182 1136 3499 4775  
## 2012 2287 2800 3566 3724 3800 3086 2877 3059 1136 3168 3558 4824  
## 2013 2320 2872 3687 3883 4170 3360 3320 3539 3720 3590 3834 4865  
## 2014 2961 3381 4268 4223 4421 3905 3733 4067 4481 4434 4525 5664  
## 2015 3041 3646 4294 1136 4413 3899 3817 3897 3881 4372 4401 5874

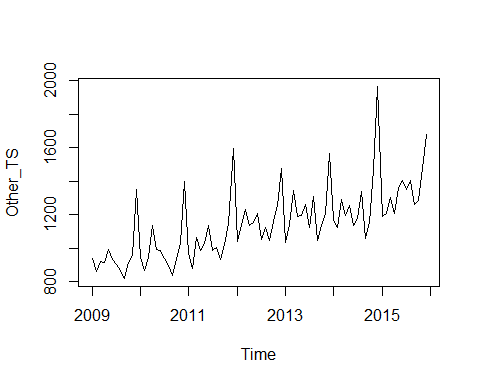
plot(women\_dataTS)

 ###Covert above dataframes in Timeseries dataframes--Others data

Other\_TS<-ts(other\_data[,4],start = c(2009,1),frequency = 12)  
Other\_TS

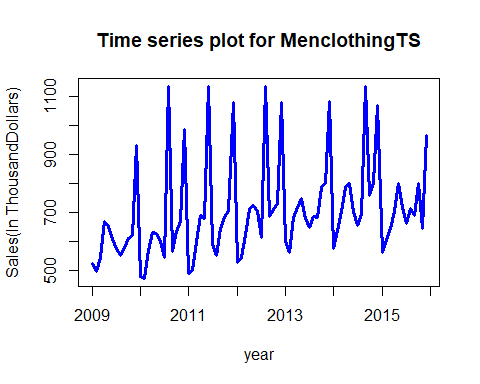
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 2009 936 859 921 914 989 932 901 865 819 914 955 1351  
## 2010 945 869 941 1136 991 984 937 897 839 930 1035 1398  
## 2011 968 878 1064 985 1027 1136 985 1001 930 1020 1147 1596  
## 2012 1039 1136 1229 1136 1155 1205 1049 1124 1045 1158 1252 1475  
## 2013 1036 1144 1346 1188 1194 1261 1126 1311 1043 1136 1209 1566  
## 2014 1172 1123 1293 1196 1257 1136 1175 1337 1056 1153 1468 1967  
## 2015 1190 1210 1300 1205 1363 1402 1349 1405 1261 1283 1478 1680

plot(Other\_TS)

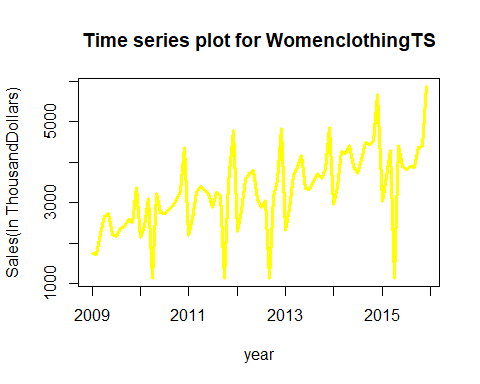


### Vizualize the time series Data for different product Categories

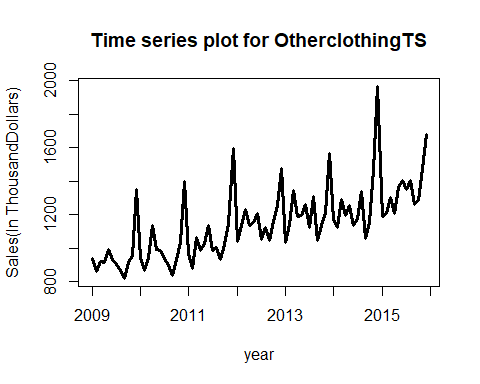
plot(Men\_dataTS,  
 type="l",  
 lwd=3,  
 col="blue",  
 xlab="year",  
 ylab="Sales(In ThousandDollars)",  
 main="Time series plot for MenclothingTS")



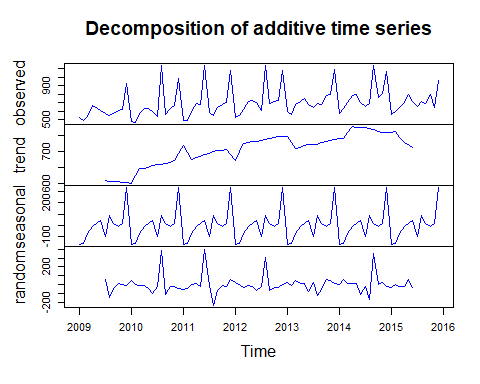
plot(women\_dataTS,  
 type="l",  
 lwd=3,  
 col="yellow",  
 xlab="year",  
 ylab="Sales(In ThousandDollars)",  
 main="Time series plot for WomenclothingTS")



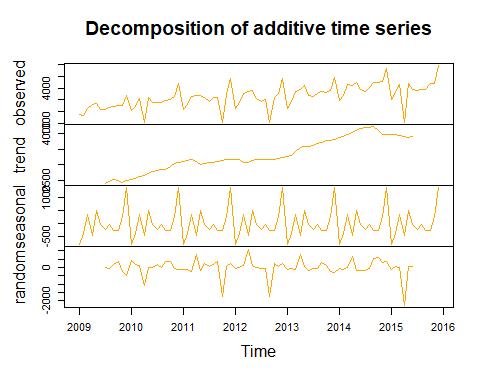
plot(Other\_TS,  
 type="l",  
 lwd=3,  
 col="black",  
 xlab="year",  
 ylab="Sales(In ThousandDollars)",  
 main="Time series plot for OtherclothingTS")

 ### Decomposed Time Series #Decompose will provide us with the info on seasonality,trend and randomness

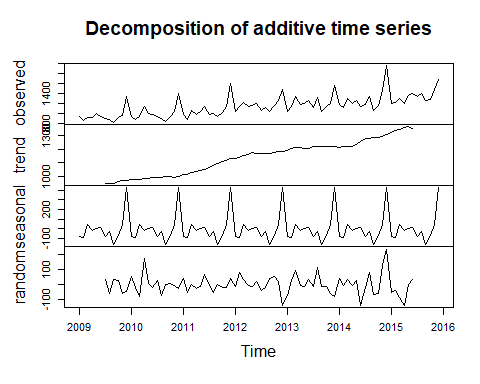
Men\_decomposed=decompose(Men\_dataTS)  
plot(Men\_decomposed,col="BLUE")



Women\_decomposed=decompose(women\_dataTS)  
plot(Women\_decomposed,col="ORANGE")



Other\_decomposed=decompose(Other\_TS)  
plot(Other\_decomposed,col="BLACK")



### Modelling the time series using simple moving averages

* Time series Price has trend
* Modelling the time series behaviour by simple moving averages

fitsma <- SMA(Men\_dataTS,n=2)

### Define the metric MAPE

smaMape <- mean(abs((Men\_dataTS[2:length(Men\_dataTS)]-fitsma[2:length(Men\_dataTS)])/Men\_dataTS[2:length(Men\_dataTS)]))  
smaMape

## [1] 0.1055211

### Weighted Moving Averages

fitwma<- WMA(Men\_dataTS,n=2,1:2)  
wmaMape\_men <- mean(abs((Men\_dataTS[2:length(Men\_dataTS)]-fitwma[2:length(Men\_dataTS)])/Men\_dataTS[2:length(Men\_dataTS)]))  
wmaMape\_men

## [1] 0.07034743

fitwma<- WMA(women\_dataTS,n=2,1:2)  
wmaMape\_women <- mean(abs((women\_dataTS[2:length(women\_dataTS)]-fitwma[2:length(women\_dataTS)])/women\_dataTS[2:length(women\_dataTS)]))  
wmaMape\_women

## [1] 0.09326361

fitwma<- WMA(Other\_TS,n=2,1:2)  
wmaMape\_other <- mean(abs((Other\_TS[2:length(Other\_TS)]-fitwma[2:length(Other\_TS)])/Other\_TS[2:length(Other\_TS)]))  
wmaMape\_other

## [1] 0.04186116

### Exponential Moving Averages

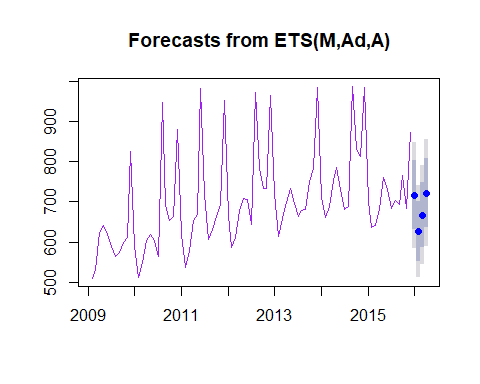
fitEma <- EMA(Men\_dataTS, n = 2)  
emaMape <- mean(abs((Men\_dataTS[2:length(Men\_dataTS)]-fitEma[2:length(Men\_dataTS)])/Men\_dataTS[2:length(Men\_dataTS)]))  
emaMape

## [1] 0.06518856

pred<-forecast(fitEma,h=4)

## Warning in ets(object, lambda = lambda, allow.multiplicative.trend =  
## allow.multiplicative.trend, : Missing values encountered. Using longest  
## contiguous portion of time series

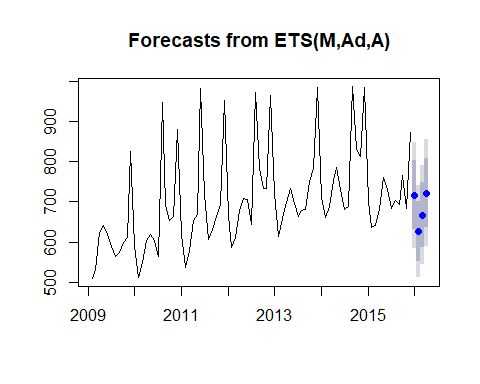
plot(pred)  
lines(fitEma,col="purple")



fitEma <- EMA(women\_dataTS, n = 2)  
emaMape <- mean(abs((Men\_dataTS[2:length(women\_dataTS)]-fitEma[2:length(women\_dataTS)])/women\_dataTS[2:length(women\_dataTS)]))  
emaMape

## [1] 0.79503

plot(pred)



fitEma <- EMA(Other\_TS, n = 2)  
emaMape <- mean(abs((Other\_TS[2:length(Other\_TS)]-fitEma[2:length(Other\_TS)])/Other\_TS[2:length(Other\_TS)]))  
emaMape

## [1] 0.03736012

plot(pred)

## HoltWinters model with trend and Seasonality--MEN

sales\_men <-  
HoltWinters(Men\_dataTS, beta=FALSE, gamma=FALSE, seasonal="multiplicative")  
head(sales\_men$fitted)

## xhat level  
## Feb 2009 524.0000 524.0000  
## Mar 2009 521.5182 521.5182  
## Apr 2009 523.3336 523.3336  
## May 2009 536.2447 536.2447  
## Jun 2009 546.3274 546.3274  
## Jul 2009 551.7051 551.7051

\*Since you are building the models on weekly data, you will get 52 seasonal components. If you are reading the monthly data, you will get 12 seasonal components

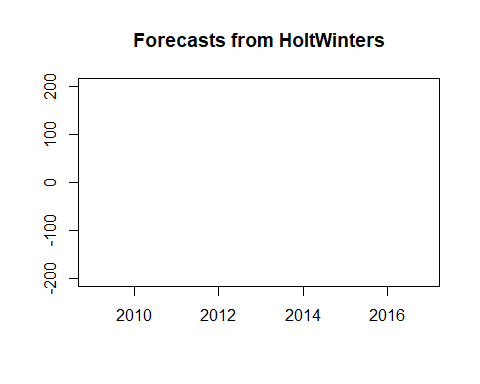
### Prediction on the Train

holtforecastTrain <- data.frame(sales\_men$fitted)  
holtforecastTrainpredictions <- holtforecastTrain$xhat  
head(holtforecastTrainpredictions)

## [1] 524.0000 521.5182 523.3336 536.2447 546.3274 551.7051

### Prediction on test data

priceforecast<-forecast(sales\_men,h = 12)  
plot(priceforecast,ylim = c(-200,200))

 ##holt winters for women

sales\_women <-  
HoltWinters(women\_dataTS, beta=FALSE, gamma=FALSE, seasonal="multiplicative")  
head(sales\_women$fitted)

## xhat level  
## Feb 2009 1755.000 1755.000  
## Mar 2009 1750.806 1750.806  
## Apr 2009 1832.297 1832.297  
## May 2009 1966.135 1966.135  
## Jun 2009 2089.674 2089.674  
## Jul 2009 2110.697 2110.697

\*Since you are building the models on weekly data, you will get 52 seasonal components. If you are reading the monthly data, you will get 12 seasonal components

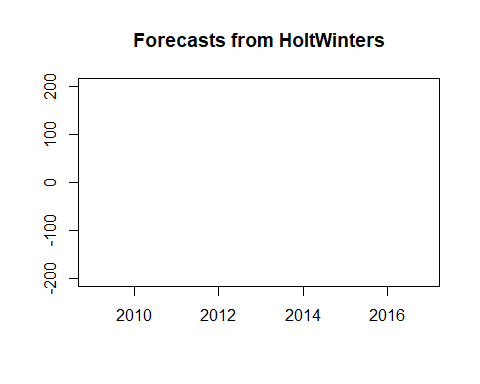
### Prediction on the Train

holtforecastTrain\_women <- data.frame(sales\_women$fitted)  
holtforecastTrainpredictions\_women <- holtforecastTrain\_women$xhat  
head(holtforecastTrainpredictions\_women)

## [1] 1755.000 1750.806 1832.297 1966.135 2089.674 2110.697

### Prediction on test data

priceforecast\_women<-forecast(sales\_women,h = 12)  
plot(priceforecast\_women,ylim = c(-200,200))

 ##holt winters for Others

sales\_others <-  
HoltWinters(Other\_TS, beta=FALSE, gamma=FALSE, seasonal="multiplicative")  
head(sales\_others$fitted)

## xhat level  
## Feb 2009 936.0000 936.0000  
## Mar 2009 924.7041 924.7041  
## Apr 2009 924.1607 924.1607  
## May 2009 922.6701 922.6701  
## Jun 2009 932.4007 932.4007  
## Jul 2009 932.3419 932.3419

\*Since you are building the models on weekly data, you will get 52 seasonal components. If you are reading the monthly data, you will get 12 seasonal components

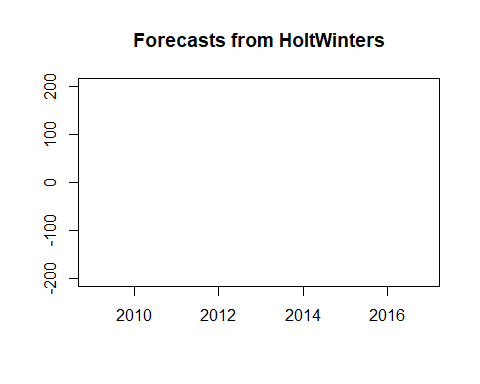
### Prediction on the Train

holtforecastTrain\_others <- data.frame(sales\_others$fitted)  
holtforecastTrainpredictions\_others<- holtforecastTrain\_others$xhat  
head(holtforecastTrainpredictions\_others)

## [1] 936.0000 924.7041 924.1607 922.6701 932.4007 932.3419

### Prediction on test data

priceforecast\_others<-forecast(sales\_others,h = 12)  
plot(priceforecast\_others,ylim = c(-200,200))

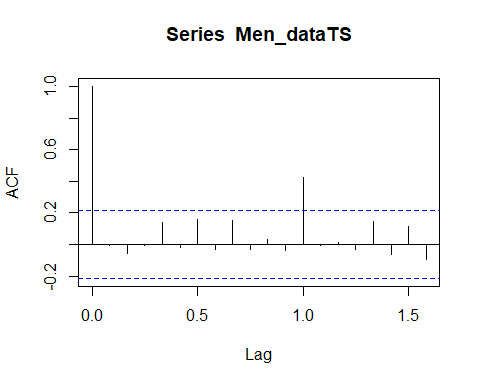


### Arima Models

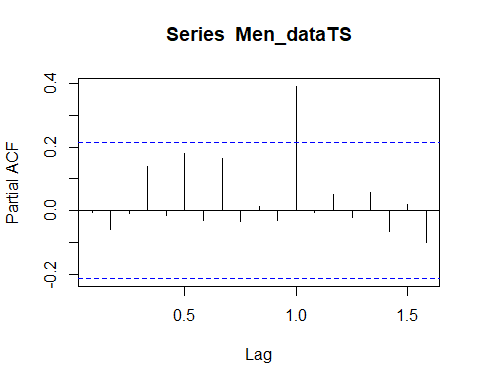
##men  
model1\_men <- arima(Men\_dataTS,c(0,0,0))  
model1\_men

##   
## Call:  
## arima(x = Men\_dataTS, order = c(0, 0, 0))  
##   
## Coefficients:  
## intercept  
## 700.7143  
## s.e. 17.9724  
##   
## sigma^2 estimated as 27132: log likelihood = -547.95, aic = 1099.89

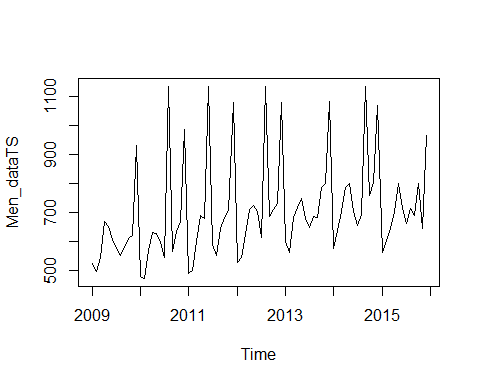
acf(Men\_dataTS)



pacf(Men\_dataTS)



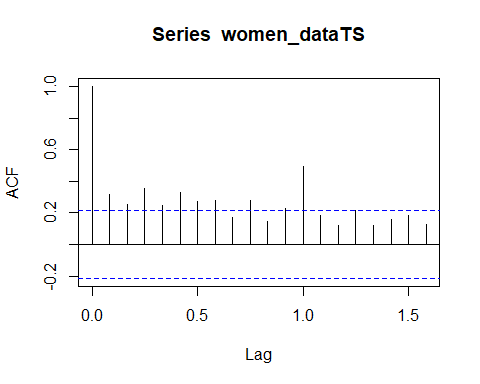
plot(Men\_dataTS)



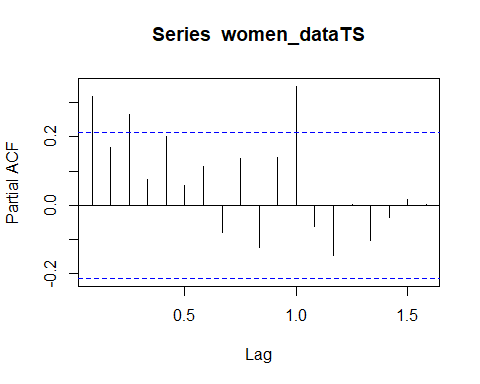
##women  
  
model1\_women <- arima(women\_dataTS,c(0,0,0))  
model1\_women

##   
## Call:  
## arima(x = women\_dataTS, order = c(0, 0, 0))  
##   
## Coefficients:  
## intercept  
## 3301.5357  
## s.e. 103.6372  
##   
## sigma^2 estimated as 902215: log likelihood = -695.12, aic = 1394.24

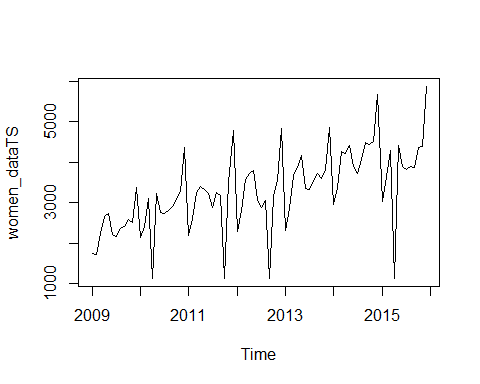
acf(women\_dataTS)



pacf(women\_dataTS)



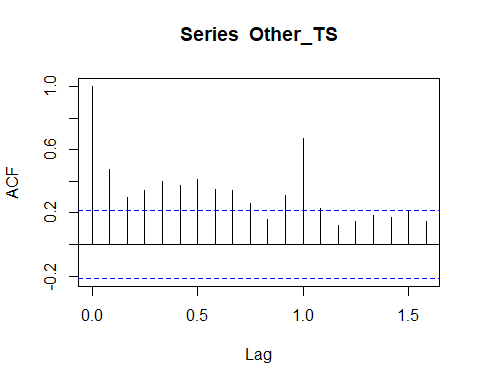
plot(women\_dataTS)



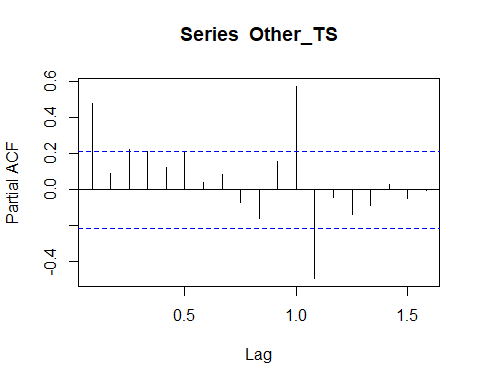
##others  
model1\_Other <- arima(Other\_TS,c(0,0,0))  
model1\_Other

##   
## Call:  
## arima(x = Other\_TS, order = c(0, 0, 0))  
##   
## Coefficients:  
## intercept  
## 1143.0595  
## s.e. 22.7925  
##   
## sigma^2 estimated as 43638: log likelihood = -567.91, aic = 1139.81

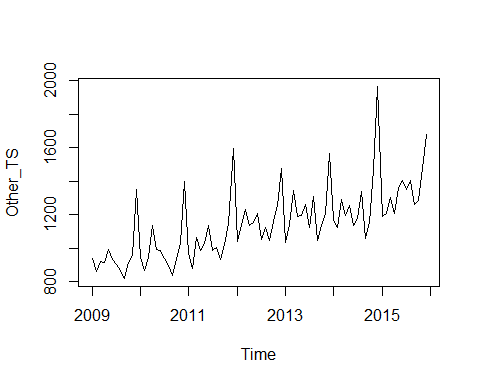
acf(Other\_TS)



pacf(Other\_TS)



plot(Other\_TS)



# AUTO ARIMA

Men\_AA <- auto.arima(Men\_dataTS,ic='aic')  
Men\_AA

## Series: Men\_dataTS   
## ARIMA(0,1,1)(2,0,0)[12]   
##   
## Coefficients:  
## ma1 sar1 sar2  
## -0.9503 0.1778 0.5761  
## s.e. 0.0316 0.0779 0.0902  
##   
## sigma^2 estimated as 14625: log likelihood=-521.11  
## AIC=1050.22 AICc=1050.73 BIC=1059.89

Women\_AA<- auto.arima(women\_dataTS,ic='aic')  
Women\_AA

## Series: women\_dataTS   
## ARIMA(2,1,1)(2,0,0)[12]   
##   
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

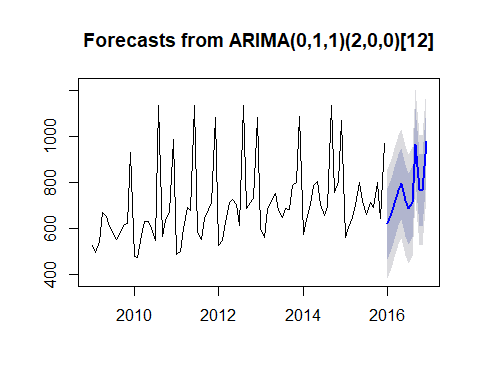
## ar1 ar2 ma1 sar1 sar2  
## -0.0073 -0.0055 -0.8729 0.4113 0.2113  
## s.e. NaN NaN 0.0962 NaN NaN  
##   
## sigma^2 estimated as 489671: log likelihood=-661.97  
## AIC=1335.94 AICc=1337.04 BIC=1350.45

Other\_AA <- auto.arima(Other\_TS,ic='aic')  
Other\_AA

## Series: Other\_TS   
## ARIMA(0,0,1)(0,1,1)[12] with drift   
##   
## Coefficients:  
## ma1 sma1 drift  
## 0.2057 -0.7231 5.6486  
## s.e. 0.1357 0.1564 0.4445  
##   
## sigma^2 estimated as 6232: log likelihood=-419.58  
## AIC=847.15 AICc=847.75 BIC=856.26

# MenAutoArima <- forecast.Arima(MenAutoArima, h=12)

MenTSforecasts <- forecast(Men\_AA, h=12)  
  
plot(MenTSforecasts)

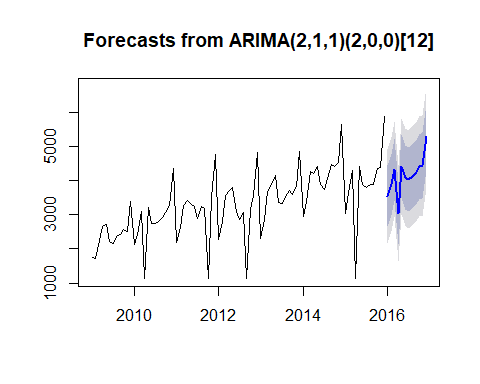


MenTSforecasts

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2016 618.1225 463.1331 773.1119 381.0867 855.1583  
## Feb 2016 661.8856 506.7047 817.0664 424.5570 899.2142  
## Mar 2016 707.5548 552.1828 862.9269 469.9338 945.1759  
## Apr 2016 765.3294 609.7664 920.8925 527.4163 1003.2425  
## May 2016 792.1469 636.3931 947.9006 553.9420 1030.3517  
## Jun 2016 720.1765 564.2322 876.1208 481.6803 958.6727  
## Jul 2016 683.3206 527.1861 839.4552 444.5334 922.1079  
## Aug 2016 713.7040 557.3794 870.0286 474.6261 952.7819  
## Sep 2016 965.5985 809.0841 1122.1130 726.2303 1204.9667  
## Oct 2016 766.7968 610.0927 923.5008 527.1386 1006.4549  
## Nov 2016 765.3893 608.4959 922.2826 525.4415 1005.3370  
## Dec 2016 976.8157 819.7331 1133.8982 736.5787 1217.0526

# WomenAutoArima <- forecast.Arima(WomenAutoArima, h=12)

WomenTSforecastsAutoArima <- forecast(Women\_AA, h=12)  
  
plot(WomenTSforecastsAutoArima)

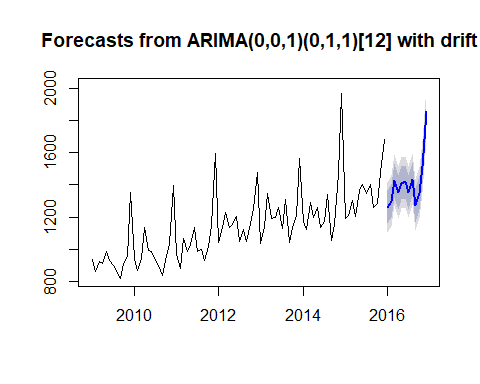


WomenTSforecastsAutoArima

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2016 3543.707 2646.922 4440.492 2172.193 4915.221  
## Feb 2016 3883.157 2979.959 4786.354 2501.836 5264.477  
## Mar 2016 4341.780 3432.119 5251.441 2950.574 5732.986  
## Apr 2016 3033.238 2116.636 3949.841 1631.415 4435.061  
## May 2016 4422.986 3499.498 5346.473 3010.633 5835.338  
## Jun 2016 4102.538 3172.219 5032.857 2679.737 5525.339  
## Jul 2016 4032.468 3095.367 4969.569 2599.296 5465.640  
## Aug 2016 4135.944 3192.110 5079.778 2692.474 5579.413  
## Sep 2016 4216.834 3266.315 5167.353 2763.140 5670.528  
## Oct 2016 4408.868 3451.710 5366.026 2945.021 5872.715  
## Nov 2016 4440.024 3476.273 5403.775 2966.094 5913.953  
## Dec 2016 5286.570 4316.272 6256.869 3802.626 6770.515

# OtherAutoArima <- forecast.Arima(OtherAutoArima, h=12)

OtherTSforecasts <- forecast(Other\_AA, h=12)  
#plot.forecast(OtherTSforecastsAutoArima)  
plot(OtherTSforecasts)



OtherTSforecasts$mean

## Jan Feb Mar Apr May Jun Jul  
## 2016 1261.637 1297.722 1427.493 1352.239 1410.917 1418.512 1352.975  
## Aug Sep Oct Nov Dec  
## 2016 1435.823 1271.779 1347.491 1513.139 1852.575

## AUTO ARIMA OUTPUT

final\_data\_forecast<-cbind(WomenTSforecastsAutoArima$mean,MenTSforecasts$mean,OtherTSforecasts$mean)  
  
  
  
###output file   
##write.csv(final\_data\_forecast, "E:/proofs/PHD/output\_45.csv")

## HOLTWINTERS OUTPUT

final\_data\_forecast\_holt1<-cbind(priceforecast\_women$mean,priceforecast$mean,priceforecast\_others$mean)  
  
  
  
###output file   
##write.csv(final\_data\_forecast\_holt1, "E:/proofs/PHD/output\_holt\_2.csv")

Men\_AA\_v1 <- auto.arima(Men\_dataTS,ic='aicc')  
  
Men\_AA\_v1

## Series: Men\_dataTS   
## ARIMA(0,1,1)(2,0,0)[12]   
##   
## Coefficients:  
## ma1 sar1 sar2  
## -0.9503 0.1778 0.5761  
## s.e. 0.0316 0.0779 0.0902  
##   
## sigma^2 estimated as 14625: log likelihood=-521.11  
## AIC=1050.22 AICc=1050.73 BIC=1059.89

Women\_AA\_v1<- auto.arima(women\_dataTS,ic='aicc')  
Women\_AA\_v1

## Series: women\_dataTS   
## ARIMA(2,1,1)(2,0,0)[12]   
##   
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

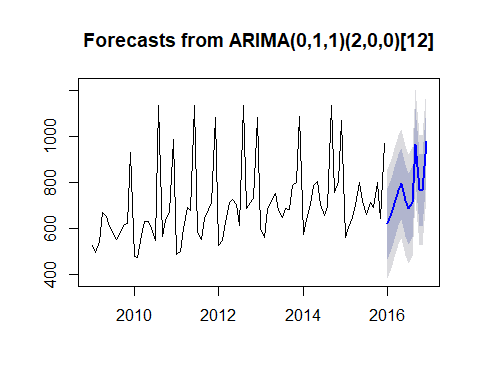
## ar1 ar2 ma1 sar1 sar2  
## -0.0073 -0.0055 -0.8729 0.4113 0.2113  
## s.e. NaN NaN 0.0962 NaN NaN  
##   
## sigma^2 estimated as 489671: log likelihood=-661.97  
## AIC=1335.94 AICc=1337.04 BIC=1350.45

Other\_AA\_v1 <- auto.arima(Other\_TS,ic='aicc')  
Other\_AA

## Series: Other\_TS   
## ARIMA(0,0,1)(0,1,1)[12] with drift   
##   
## Coefficients:  
## ma1 sma1 drift  
## 0.2057 -0.7231 5.6486  
## s.e. 0.1357 0.1564 0.4445  
##   
## sigma^2 estimated as 6232: log likelihood=-419.58  
## AIC=847.15 AICc=847.75 BIC=856.26

# MenAutoArima <- forecast.Arima(MenAutoArima, h=12)

MenTSforecasts\_v1 <- forecast(Men\_AA\_v1, h=12)  
  
plot(MenTSforecasts\_v1)

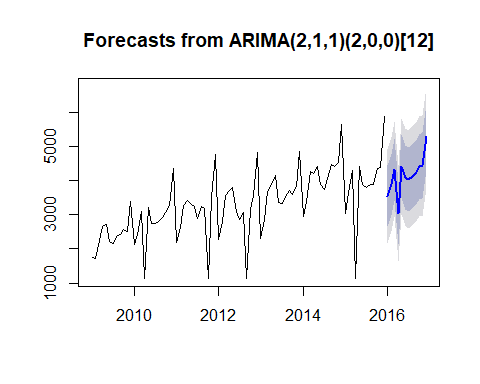


MenTSforecasts\_v1

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2016 618.1225 463.1331 773.1119 381.0867 855.1583  
## Feb 2016 661.8856 506.7047 817.0664 424.5570 899.2142  
## Mar 2016 707.5548 552.1828 862.9269 469.9338 945.1759  
## Apr 2016 765.3294 609.7664 920.8925 527.4163 1003.2425  
## May 2016 792.1469 636.3931 947.9006 553.9420 1030.3517  
## Jun 2016 720.1765 564.2322 876.1208 481.6803 958.6727  
## Jul 2016 683.3206 527.1861 839.4552 444.5334 922.1079  
## Aug 2016 713.7040 557.3794 870.0286 474.6261 952.7819  
## Sep 2016 965.5985 809.0841 1122.1130 726.2303 1204.9667  
## Oct 2016 766.7968 610.0927 923.5008 527.1386 1006.4549  
## Nov 2016 765.3893 608.4959 922.2826 525.4415 1005.3370  
## Dec 2016 976.8157 819.7331 1133.8982 736.5787 1217.0526

# WomenAutoArima <- forecast.Arima(WomenAutoArima, h=12)

WomenTSforecastsAutoArima\_v1 <- forecast(Women\_AA\_v1, h=12)  
  
plot(WomenTSforecastsAutoArima)

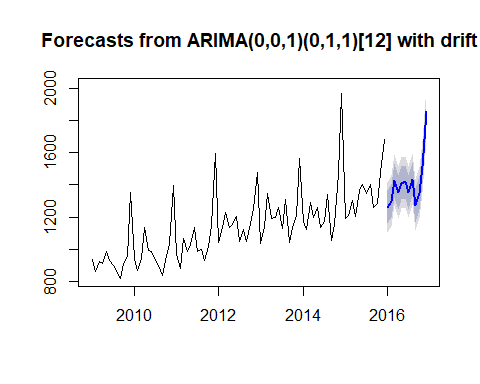


WomenTSforecastsAutoArima\_v1

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2016 3543.707 2646.922 4440.492 2172.193 4915.221  
## Feb 2016 3883.157 2979.959 4786.354 2501.836 5264.477  
## Mar 2016 4341.780 3432.119 5251.441 2950.574 5732.986  
## Apr 2016 3033.238 2116.636 3949.841 1631.415 4435.061  
## May 2016 4422.986 3499.498 5346.473 3010.633 5835.338  
## Jun 2016 4102.538 3172.219 5032.857 2679.737 5525.339  
## Jul 2016 4032.468 3095.367 4969.569 2599.296 5465.640  
## Aug 2016 4135.944 3192.110 5079.778 2692.474 5579.413  
## Sep 2016 4216.834 3266.315 5167.353 2763.140 5670.528  
## Oct 2016 4408.868 3451.710 5366.026 2945.021 5872.715  
## Nov 2016 4440.024 3476.273 5403.775 2966.094 5913.953  
## Dec 2016 5286.570 4316.272 6256.869 3802.626 6770.515

# OtherAutoArima <- forecast.Arima(OtherAutoArima, h=12)

OtherTSforecasts\_v1 <- forecast(Other\_AA\_v1, h=12)  
#plot.forecast(OtherTSforecastsAutoArima)  
plot(OtherTSforecasts)



OtherTSforecasts$mean

## Jan Feb Mar Apr May Jun Jul  
## 2016 1261.637 1297.722 1427.493 1352.239 1410.917 1418.512 1352.975  
## Aug Sep Oct Nov Dec  
## 2016 1435.823 1271.779 1347.491 1513.139 1852.575

## AUTO ARIMA OUTPUT

final\_data\_forecast<-cbind(WomenTSforecastsAutoArima\_v1$mean,MenTSforecasts\_v1$mean,OtherTSforecasts\_v1$mean)  
  
  
  
###output file   
##write.csv(final\_data\_forecast, "E:/proofs/PHD/output\_v2.csv")