Data624 Project 1, Part A

Andrew Goldberg

10/6/2017

library(dplyr)

##   
## 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(lubridate)

##   
## Attaching package: 'lubridate'

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

library(forecast)  
library(tseries)  
  
dat <- read.csv("https://raw.githubusercontent.com/aagoldberg/Work/master/ATM624Data.csv")  
summary(dat) #15 NA's, but 14 are from empty rows

## DATE ATM Cash   
## 1/1/2010 12:00:00 AM : 4 : 14 Min. : 0.00   
## 1/10/2010 12:00:00 AM: 4 ATM1:365 1st Qu.: 1.00   
## 1/11/2010 12:00:00 AM: 4 ATM2:365 Median : 73.00   
## 1/12/2010 12:00:00 AM: 4 ATM3:365 Mean : 58.53   
## 1/13/2010 12:00:00 AM: 4 ATM4:365 3rd Qu.: 96.50   
## 1/14/2010 12:00:00 AM: 4 Max. :1123.00   
## (Other) :1450 NA's :15

# Data Transformation

#split into invidual atm's  
atm1 <- dat %>% filter(ATM == 'ATM1')  
atm2 <- dat %>% filter(ATM == 'ATM2')  
atm3 <- dat %>% filter(ATM == 'ATM3')  
atm4 <- dat %>% filter(ATM == 'ATM4')  
  
  
atm1TS <- ts(atm1$Cash, start = decimal\_date(as.Date("2009-05-01")), frequency = 7)  
atm2TS <- ts(atm2$Cash, start = decimal\_date(as.Date("2009-05-01")), frequency = 7)  
atm3TS <- ts(atm3$Cash, start = decimal\_date(as.Date("2009-05-01")), frequency = 7)  
atm4TS <- ts(atm4$Cash, start = decimal\_date(as.Date("2009-05-01")), frequency = 7)  
  
summary(atm1TS)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 73.0 91.0 84.1 108.0 180.0

summary(atm2TS) #has a null

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 25.00 66.50 62.46 93.00 147.00 1

summary(atm3TS)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.7206 0.0000 96.0000

summary(atm4TS)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 73.00 91.00 86.84 108.00 1123.00

#impute into atm2  
library(forecast)  
library(imputeTS)

##   
## Attaching package: 'imputeTS'

## The following object is masked from 'package:tseries':  
##   
## na.remove

atm2TS <- na.interpolation((atm2TS))  
  
any(atm1TS == 0)

## [1] FALSE

any(atm2TS == 0) #has zero

## [1] TRUE

any(atm3TS == 0) #has zero

## [1] TRUE

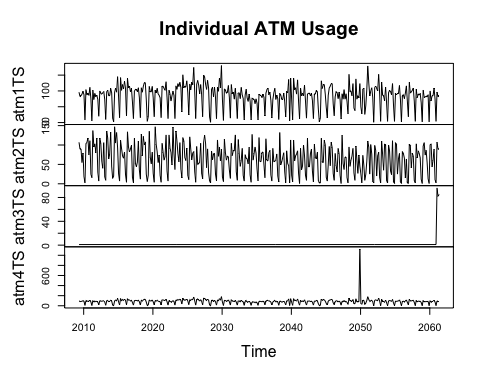
any(atm4TS == 0)

## [1] FALSE

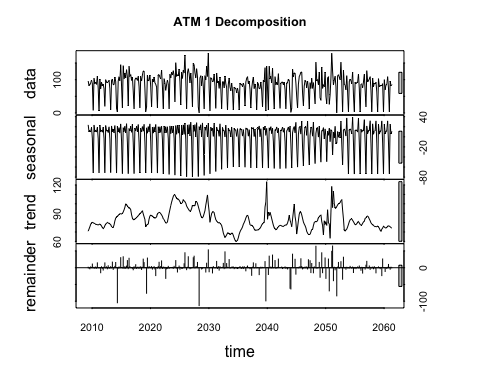
#replace zeros (not entirely sure what to do with 0's, so setting it to 1, which is the smallest integer...)  
atm2TS[atm2TS == 0] <- 1  
atm3TS[atm3TS == 0] <- 1

# Data Exploration

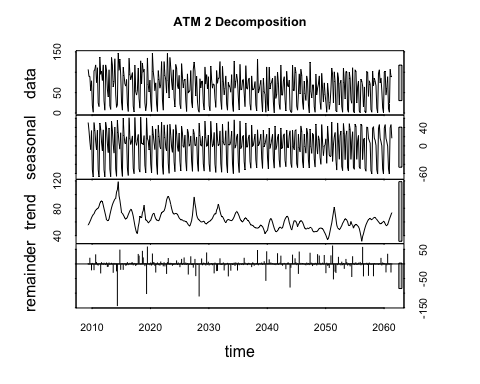
#plot of each ATM usage  
splitAtm <- cbind(atm1TS, atm2TS, atm3TS, atm4TS)  
plot(splitAtm, main = "Individual ATM Usage", ylab = "hundreds of dollars")



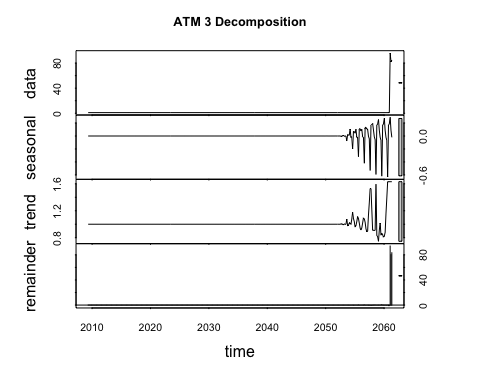
#Atm 3 appears to be new, with data for only the last few periods. I will explore using more sophisticated techniques here, but my hunch is that a basic naive forecast may be most appropriate.   
  
#Atm 1 and 2 saw the most usage, with clear weekly, and potentially seasonal or monthly, trends. Due to the seasonality component and daily data, STL should be the best decomposition technique here, although the more sophisticated X-12-Arima method may also be appropriate, but the book doesn't go into details, and I unfortunately don't have the time to explore this further.   
  
#Atm 4 saw much smaller deposits, with one large spike towards the end of the year. Unlikely that there is a robust seasonality component here, although the model will need to handle the outlier data.   
  
  
#STL decomposition  
  
#ATM 1  
atm1Decomp <- stl(atm1TS, t.window=7, s.window=7, robust=TRUE)   
plot(atm1Decomp, main = "ATM 1 Decomposition")



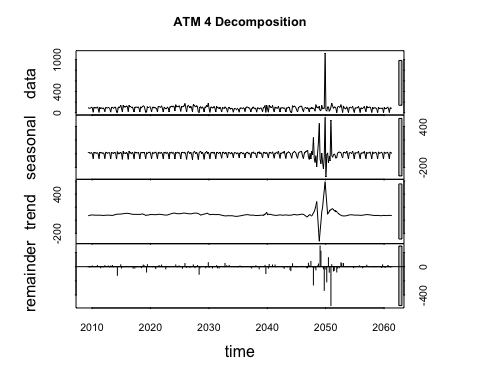
#Clear seasonal pattern -- appears to be less usage in the winter, large downward spikes on weekly trends, perhaps closed on some days. The season trend appears to be changing proportional to the level, so a mutiplicative method is likely to be the best fit.   
  
#ATM 2  
atm2Decomp <- stl(atm2TS, t.window=7, s.window=4, robust=TRUE)   
plot(atm2Decomp, main = "ATM 2 Decomposition")



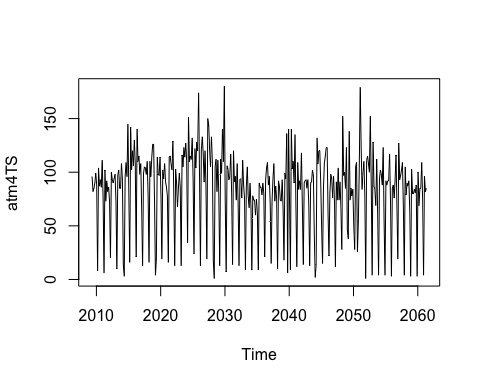
#Clear seasonal pattern -- appears to be slightly less usage in the winter, both high and low weekly trend spikes. The seasonal trend appears to be changing more proportionally with the level here, again pointing to a multiplicative method.   
  
#ATM 3  
atm3Decomp <- stl(atm3TS, t.window=7, s.window=7, robust=TRUE)   
plot(atm3Decomp, main = "ATM 3 Decomposition")



#Likely brand new ATM -- no usage until very end of data  
  
#ATM 4  
atm4Decomp <- stl(atm4TS, t.window=7, s.window=4, robust=TRUE)   
plot(atm4Decomp, main = "ATM 4 Decomposition")

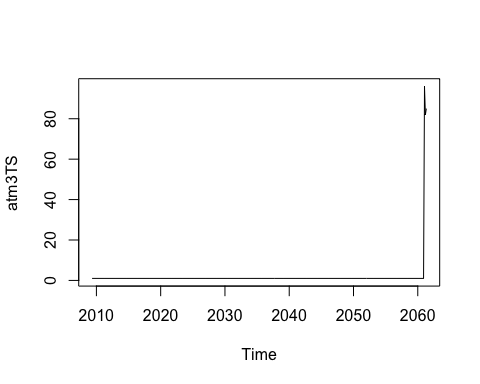


#Lower levels of usage with one large spike -- possibly associated with winter holidays. While there are some seasonality changes, I'm not sure if its in proportion to the level, let us look!  
  
#Fixing outlier in ATM 4 with modeling  
out4.pos <- which(atm4TS == 1123)  
out4.atm4TS <- atm4TS[1:out4.pos]  
out4.train.end <- time(out4.atm4TS)[length(out4.atm4TS)-1]  
out4.train <- window(out4.atm4TS,end=out4.train.end)  
out4.fit <- Arima(out4.train)  
out4.fc <- forecast(out4.fit,h=1)  
atm4TS[285] <- as.integer(out4.fc$mean)  
plot(atm4TS)



# ATM 3: Naive Model

plot(atm3TS)



#Since this atm appears to be new or only used for 3 periods, there isn't enough data for a legitimate model, so a naive forecast seems appropriate.   
  
atm3TS.active <- atm3TS[363:365]  
atm3Fit <- naive(atm3TS.active, 31)

# STL Decomposition

# ATM 1

getrmseSTL <- function(x,h,t,s)  
{  
 train.end <- time(x)[length(x)-h]  
 test.start <- time(x)[length(x)-h+1]  
 train <- window(x,end=train.end)  
 test <- window(x,start=test.start)  
 fit <- stl(train, t.window=t, s.window= s, robust=TRUE)  
 fcast <- forecast(fit, method="naive")  
 return(accuracy(fcast,test)[2,"RMSE"])  
}  
getrmseSTL(atm1TS, 30, 7, "periodic") #73.76

## [1] 72.49103

getrmseSTL(atm1TS, 30, 4, "periodic") #72.67

## [1] 72.66963

getrmseSTL(atm1TS, 30, 12, "periodic") #73.52

## [1] 73.52537

getrmseSTL(atm1TS, 30, 4, 7) #31.99 #winner

## [1] 31.99633

getrmseSTL(atm1TS, 30, 7, 4) #37.13

## [1] 37.13348

# ATM 2

getrmseSTL(atm2TS, 30, 7, "periodic") #100.42

## [1] 100.4199

getrmseSTL(atm2TS, 30, 4, "periodic") #98.46

## [1] 98.45713

getrmseSTL(atm2TS, 30, 4, 7) #27.11

## [1] 27.10985

getrmseSTL(atm2TS, 30, 7, 4) #18.28 #winner

## [1] 18.27538

# ATM 4

getrmseSTL(atm4TS, 30, 7, "periodic") #77.78

## [1] 72.78164

getrmseSTL(atm4TS, 30, 4, 7) #30.81 #winner

## [1] 30.81808

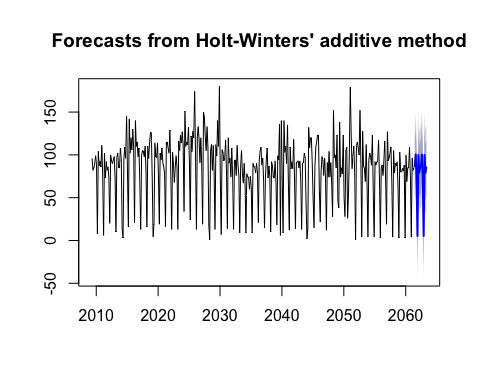
getrmseSTL(atm4TS, 30, 7, 4) #37.25

## [1] 37.24521

# Exponential Smoothing: Holt-Winters seasonal

# ATM 1

hwATM1 <- window(atm1TS)  
#ATM1  
StSp1 <- ets(hwATM1, model="ZZZ")  
#state space model selection suggests an ANA model (additive errors, no trend, additive seasonality)  
hwA1 <- hw(hwATM1,seasonal="additive")  
hwM1 <- hw(hwATM1,seasonal="multiplicative")  
hwDM1 <- hw(hwATM1,seasonal="multiplicative", Damped = TRUE)  
plot(hwA1)



str(hwA1)

## List of 10  
## $ model :List of 19  
## ..$ loglik : num -2233  
## ..$ aic : num 4490  
## ..$ bic : num 4537  
## ..$ aicc : num 4491  
## ..$ mse : num 565  
## ..$ amse : num 568  
## ..$ fit :List of 4  
## .. ..$ value : num 565  
## .. ..$ par : num [1:11] 0.02187 0.0001 0.33163 79.00568 -0.00626 ...  
## .. ..$ fail : int 0  
## .. ..$ fncount: int 836  
## ..$ residuals : Time-Series [1:365] from 2009 to 2061: 9.39 -10.51 -13.06 9.1 6.62 ...  
## ..$ fitted : Time-Series [1:365] from 2009 to 2061: 86.6 92.5 98.1 80.9 92.4 ...  
## ..$ states : Time-Series [1:366, 1:9] from 2009 to 2061: 79 79.2 79 78.7 78.9 ...  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : NULL  
## .. .. ..$ : chr [1:9] "l" "b" "s1" "s2" ...  
## ..$ par : Named num [1:11] 0.02187 0.0001 0.33163 79.00568 -0.00626 ...  
## .. ..- attr(\*, "names")= chr [1:11] "alpha" "beta" "gamma" "l" ...  
## ..$ m : num 7  
## ..$ method : chr "Holt-Winters' additive method"  
## ..$ series : chr "x"  
## ..$ components: chr [1:4] "A" "A" "A" "FALSE"  
## ..$ call : language hw(y = hwATM1, seasonal = "additive")  
## ..$ initstate : Named num [1:9] 79.00568 -0.00626 -53.50273 -2.2649 13.51309 ...  
## .. ..- attr(\*, "names")= chr [1:9] "l" "b" "s1" "s2" ...  
## ..$ sigma2 : num 565  
## ..$ x : Time-Series [1:365] from 2009 to 2061: 96 82 85 90 99 88 8 104 87 93 ...  
## ..- attr(\*, "class")= chr "ets"  
## $ mean : Time-Series [1:14] from 2061 to 2063: 86.7 100.4 72.9 5.3 99.8 ...  
## $ level : num [1:2] 80 95  
## $ x : Time-Series [1:365] from 2009 to 2061: 96 82 85 90 99 88 8 104 87 93 ...  
## $ upper : Time-Series [1:14, 1:2] from 2061 to 2063: 117.2 130.9 103.4 35.8 130.3 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : NULL  
## .. ..$ : chr [1:2] "80%" "95%"  
## $ lower : Time-Series [1:14, 1:2] from 2061 to 2063: 56.3 70 42.4 -25.2 69.4 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : NULL  
## .. ..$ : chr [1:2] "80%" "95%"  
## $ fitted : Time-Series [1:365] from 2009 to 2061: 86.6 92.5 98.1 80.9 92.4 ...  
## $ method : chr "Holt-Winters' additive method"  
## $ series : chr "hwATM1"  
## $ residuals: Time-Series [1:365] from 2009 to 2061: 9.39 -10.51 -13.06 9.1 6.62 ...  
## - attr(\*, "class")= chr "forecast"

#ATM Exponential Smoothing model results:  
A1fit <- hwA1$model[2:6]  
M1fit <- hwM1$model[2:6]  
DM1fit <- hwDM1$model[2:6]  
SS1fit <- StSp1[2:6]  
ATM1expfit <- rbind.data.frame(Additive = A1fit, Multiplicative = M1fit, Damped\_Multi = DM1fit, StateSpace = SS1fit)  
ATM1expfit <- ATM1expfit %>% mutate(rmse = sqrt(mse))  
row.names(ATM1expfit) <- c("Additive", "Mutliplicative", "DampedMultiplicative", "StateSpace")  
ATM1expfit

## aic bic aicc mse amse rmse  
## Additive 4490.277 4537.076 4491.164 564.8038 568.1681 23.76560  
## Mutliplicative 4601.507 4648.306 4602.393 567.0492 569.1033 23.81279  
## DampedMultiplicative 4601.507 4648.306 4602.393 567.0492 569.1033 23.81279  
## StateSpace 4484.301 4523.300 4484.923 561.7546 565.0351 23.70136

getrmseHW <- function(x,h,model)  
{  
 train.end <- time(x)[length(x)-h]  
 #print(train.end)  
 test.start <- time(x)[length(x)-h+1]  
 #print(test.start)  
 train <- window(x,end=train.end)  
 #print(train)  
 test <- window(x,start=test.start)  
 #print(test)  
 if(model == "additive"){fit5 <- hw(train,seasonal = "additive", h=h)}  
 if(model == "multiplicative"){fit5 <- hw(train, seasonal="multiplicative", h=h)}  
 if(model == "damped"){fit5 <- hw(train,seasonal="multiplicative", Damped = TRUE, h=h)}  
 if(model == "statespace"){  
 fitets <- ets(train, model="ZZZ")  
 fit5 <- forecast(fitets, h=h)}  
 #print(str(fit5))  
 #fc <- forecast(fit,h=h)  
 return(accuracy(fit5,test)[2,"RMSE"])  
}  
getrmseHW(hwATM1, 30, "additive") #11.75

## [1] 11.75422

getrmseHW(hwATM1, 30, "multiplicative") #11.31

## [1] 11.31314

getrmseHW(hwATM1, 30, "damped") #11.31

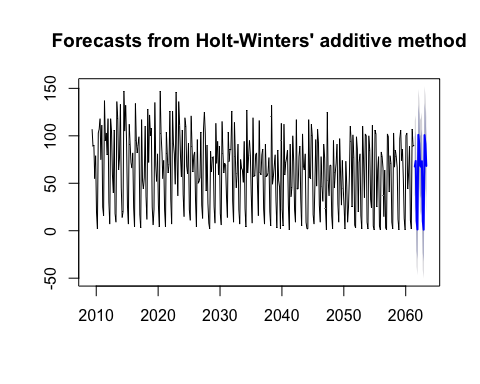
## [1] 11.31314

getrmseHW(hwATM1, 30, "statespace") #11.62

## [1] 11.61841

# ATM 2

hwATM2 <- window(atm2TS)  
#ATM2  
StSp2 <- ets(hwATM2, model="ZZZ")  
#state space model selection suggests an ANA model (additive errors, no trend, additive seasonality)  
hwATM2 <- window(atm2TS)  
hwA2 <- hw(hwATM2,seasonal="additive")  
hwM2 <- hw(hwATM2,seasonal="multiplicative")  
hwDM2 <- hw(hwATM2,seasonal="multiplicative", Damped = TRUE)  
plot(hwA2)



#ATM Exponential Smoothing model results:  
A2fit <- hwA2$model[2:6]  
M2fit <- hwM2$model[2:6]  
DM2fit <- hwDM2$model[2:6]  
SS2fit <- StSp2[2:6]  
ATM2expfit <- rbind.data.frame(A2fit, M2fit, DM2fit, SS2fit)  
ATM2expfit <- ATM2expfit %>% mutate(rmse = sqrt(mse))  
row.names(ATM2expfit) <- c("Additive", "Mutliplicative", "DampedMultiplicative", "StateSpace")  
ATM2expfit

## aic bic aicc mse amse rmse  
## Additive 4514.407 4561.205 4515.293 603.4038 604.8871 24.56428  
## Mutliplicative 5181.068 5227.867 5181.955 617.8005 617.7919 24.85559  
## DampedMultiplicative 5181.068 5227.867 5181.955 617.8005 617.7919 24.85559  
## StateSpace 4506.943 4545.942 4507.564 597.7041 599.1805 24.44799

getrmseHW(hwATM2, 30, "additive") #20.62

## [1] 20.61878

getrmseHW(hwATM2, 30, "multiplicative") #16.87

## [1] 16.86995

getrmseHW(hwATM2, 30, "damped") #16.87

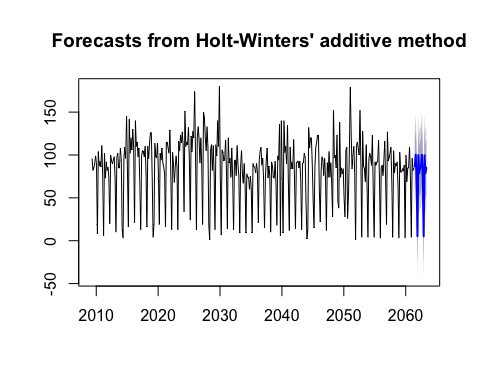
## [1] 16.86995

getrmseHW(hwATM2, 30, "statespace") #19.43

## [1] 19.43855

# ATM 4

#ATM4  
hwATM4 <- window(atm4TS)  
StSp4 <- ets(hwATM4, model="ZZZ")  
#state space model selection suggests an ANA model (additive errors, no trend, additive seasonality)  
hwA4 <- hw(hwATM4,seasonal="additive")  
hwM4 <- hw(hwATM4,seasonal="multiplicative")  
hwDM4 <- hw(hwATM4,seasonal="multiplicative", Damped = TRUE)  
plot(hwA4)



#ATM4 Exponential Smoothing model results:  
A4fit <- hwA2$model[2:6]  
M4fit <- hwM2$model[2:6]  
DM4fit <- hwDM2$model[2:6]  
SS4fit <- StSp2[2:6]  
ATM4expfit <- rbind.data.frame(A4fit, M4fit, DM4fit, SS4fit)  
ATM4expfit <- ATM4expfit %>% mutate(rmse = sqrt(mse))  
row.names(ATM4expfit) <- c("Additive", "Mutliplicative", "DampedMultiplicative", "StateSpace")  
ATM4expfit

## aic bic aicc mse amse rmse  
## Additive 4514.407 4561.205 4515.293 603.4038 604.8871 24.56428  
## Mutliplicative 5181.068 5227.867 5181.955 617.8005 617.7919 24.85559  
## DampedMultiplicative 5181.068 5227.867 5181.955 617.8005 617.7919 24.85559  
## StateSpace 4506.943 4545.942 4507.564 597.7041 599.1805 24.44799

getrmseHW(hwATM4, 30, "additive") #50.27

## [1] 11.66098

getrmseHW(hwATM4, 30, "multiplicative") #47.56

## [1] 10.89819

getrmseHW(hwATM4, 30, "damped") #10.89

## [1] 10.89819

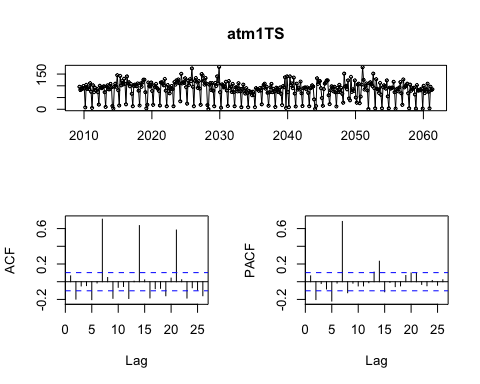
getrmseHW(hwATM4, 30, "statespace") #11.52

## [1] 11.5192

## Arima Modeling

# ATM1

#weekly seasonality  
atm1TS <- ts(atm1$Cash, start = decimal\_date(as.Date("2009-05-01")), frequency = 7)  
  
#Stationarity:  
tsdisplay(atm1TS)



#It's clear that there's still seasonal and monthly seasonality in the data. Because these trends are so clear, and the data appears stationary, it's not clear that any differencing is necessary.  
  
#All tests suggest the data is already stationary:  
ndiffs(atm1TS)

## [1] 0

adf.test(atm1TS, alternative = "stationary")

## Warning in adf.test(atm1TS, alternative = "stationary"): p-value smaller  
## than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: atm1TS  
## Dickey-Fuller = -4.4866, Lag order = 7, p-value = 0.01  
## alternative hypothesis: stationary

kpss.test(atm1TS)

##   
## KPSS Test for Level Stationarity  
##   
## data: atm1TS  
## KPSS Level = 0.42927, Truncation lag parameter = 4, p-value =  
## 0.06454

#Still, its clear there is still both monthly and seasonal seasonality within the data. Since the seasonal trends on the ACF appear to be exponentially decaying or sinusoidal, I suspect we may have variations of the ARIMA(p,d,0) (or in this case ARIMA(p,0,0)) models, with much of the weight falling on the seasonal component.   
  
auto.arima(atm1TS)$aicc #3374.79 aicc, so auto arima confirms this, but let us play with a few variations to confer.

## [1] 3374.786

Arima(atm1TS, order=c(1,0,0), seasonal=c(2,0,1))$aicc #3354.14 aicc: lower!

## [1] 3354.142

Arima(atm1TS, order=c(2,0,0), seasonal=c(2,0,1))$aicc #3364.94 aicc: very slightly higher

## [1] 3354.939

Arima(atm1TS, order=c(2,1,0), seasonal=c(2,0,1))$aicc #3431.17 aicc: even slightly lower

## [1] 3431.174

Arima(atm1TS, order=c(3,1,0), seasonal=c(2,0,1))$aicc #3410.45 aicc: lower still

## [1] 3410.449

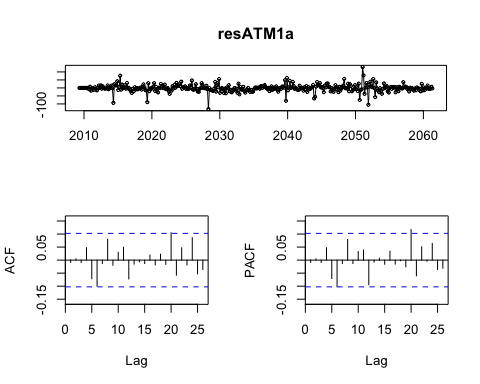
Arima(atm1TS, order=c(3,1,1), seasonal=c(2,0,1))$aicc #3348.25 aicc: still lower

## [1] 3348.252

Arima(atm1TS, order=c(3,1,1), seasonal=c(2,1,1))$aicc #3291.15 aicc: appears to be the lowest I can manually find

## [1] 3291.152

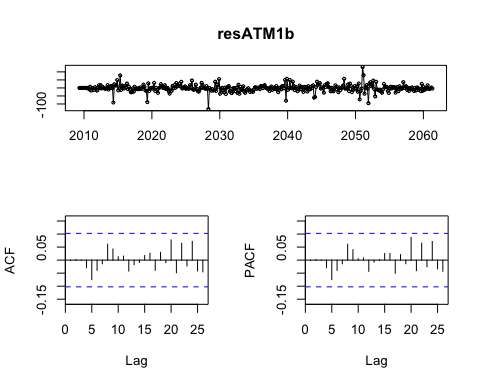
fitATM1a <- Arima(atm1TS, order=c(3,1,1), seasonal=c(2,1,1))  
resATM1a <- residuals(fitATM1a)  
tsdisplay(resATM1a)



Box.test(resATM1a, lag=26, fitdf=7, type="Ljung") #passes box-ljung test

##   
## Box-Ljung test  
##   
## data: resATM1a  
## X-squared = 24.272, df = 19, p-value = 0.1859

#we have nearly identical spikes on both graphs just touching the boundaries, so potentially need additional non-seasonal terms.   
fitATM1b <- Arima(atm1TS, order=c(5,1,3), seasonal=c(2,1,1)) #3290.65   
resATM1b <- residuals(fitATM1b)  
tsdisplay(resATM1b) #no significant spikes here



Box.test(resATM1b, lag=26, fitdf=10, type="Ljung") #passes box-ljung test

##   
## Box-Ljung test  
##   
## data: resATM1b  
## X-squared = 16.033, df = 16, p-value = 0.4507

#The full auto-arima took too long to process...  
  
#Compare fitted models using a test set consiting of the last month of data. So we'll be fitting the model from may '09 to march '10, and forcast the atm usage in April '10.   
  
getrmse <- function(x,h,...)  
{  
 train.end <- time(x)[length(x)-h]  
 test.start <- time(x)[length(x)-h+1]  
 train <- window(x,end=train.end)  
 test <- window(x,start=test.start)  
 fit <- Arima(train,...)  
 fc <- forecast(fit,h=h)  
 return(accuracy(fc,test)[2,"RMSE"])  
}  
  
getrmse(atm1TS,h=30,order=c(5,1,3),seasonal=c(2,1,1),lambda=0)

## [1] 9.836925

getrmse(atm1TS,h=30,order=c(3,1,1),seasonal=c(2,1,1),lambda=0)

## [1] 9.838622

getrmse(atm1TS,h=30,order=c(3,1,1),seasonal=c(2,0,1),lambda=0)

## [1] 8.937545

getrmse(atm1TS,h=30,order=c(2,1,0),seasonal=c(2,0,1),lambda=0)

## [1] 104.7979

getrmse(atm1TS,h=30,order=c(2,0,0),seasonal=c(2,0,1),lambda=0)

## [1] 9.27225

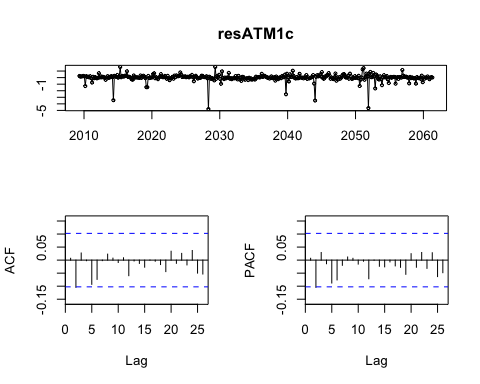
getrmse(atm1TS,h=30,order=c(1,0,0),seasonal=c(2,0,1),lambda=0) #lowest rmse (8.54)

## [1] 8.537476

getrmse(atm1TS,h=30,order=c(1,0,0),seasonal=c(2,0,0),lambda=0)

## [1] 12.23042

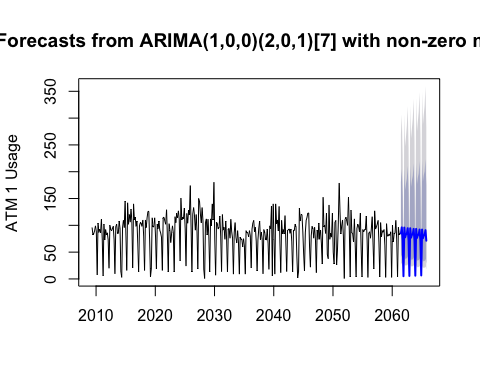
#check the residuals of the winning numbers...  
fitATM1c <- Arima(atm1TS, order=c(1,0,0), seasonal=c(2,0,1), lambda=0)  
resATM1c <- residuals(fitATM1c)  
tsdisplay(resATM1c) #only one spike closely approaches the boundary



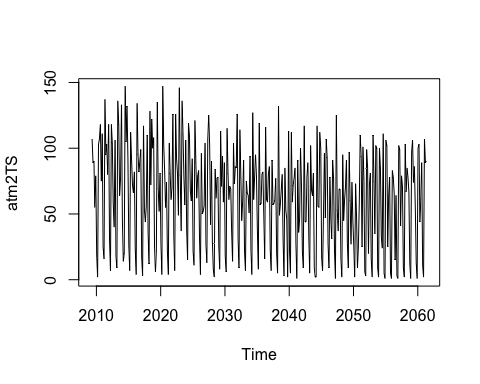
Box.test(resATM1c, lag=26, fitdf=5, type="Ljung") #passes box-ljung test, and I'm comfortable with it's slightly higher aicc, but less complicated coefficients

##   
## Box-Ljung test  
##   
## data: resATM1c  
## X-squared = 15.977, df = 21, p-value = 0.7709

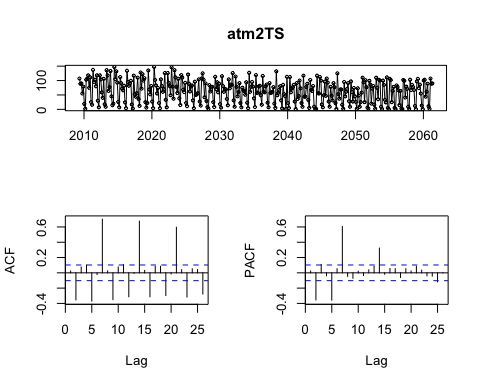
#forecasting:  
forecastATM1c <- forecast(fitATM1c, h=31)  
plot(forecastATM1c, ylab="ATM 1 Usage")

 #ATM 2

plot(atm2TS)



#Stationarity:  
tsdisplay(atm2TS)



#It's clear that there's still seasonal and monthly seasonality in the data. There also appears to be a downward trend which may need differencing.  
  
ndiffs(atm2TS) #suggests 1 difference

## [1] 1

adf.test(atm2TS, alternative = "stationary") #says is stationary

## Warning in adf.test(atm2TS, alternative = "stationary"): p-value smaller  
## than printed p-value

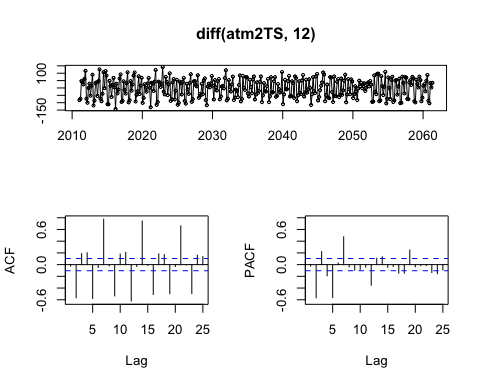
##   
## Augmented Dickey-Fuller Test  
##   
## data: atm2TS  
## Dickey-Fuller = -6.0215, Lag order = 7, p-value = 0.01  
## alternative hypothesis: stationary

kpss.test(atm2TS) #not stationary

## Warning in kpss.test(atm2TS): p-value smaller than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: atm2TS  
## KPSS Level = 1.6419, Truncation lag parameter = 4, p-value = 0.01

tsdisplay(diff(atm2TS,12))



#there's both seasonal and monthly seasonality in the data, in senusoidal patterns. I suspect at least ar(1) and a differencing for the base model, and ar(2) for the seasonal.  
  
auto.arima(atm2TS)

## Series: atm2TS   
## ARIMA(3,1,0)(2,0,0)[7]   
##   
## Coefficients:  
## ar1 ar2 ar3 sar1 sar2  
## -0.7013 -0.5292 -0.3279 0.4362 0.3975  
## s.e. 0.0496 0.0554 0.0497 0.0481 0.0487  
##   
## sigma^2 estimated as 770.1: log likelihood=-1727.83  
## AIC=3467.66 AICc=3467.9 BIC=3491.05

#Since the seasonal trends on the ACF appear to be exponentially decaying or sinusoidal, I suspect we may have variations of the ARIMA(p,d,0) (or in this case ARIMA(p,0,0)) models, with much of the weight falling on the seasonal component.   
  
auto.arima(atm2TS)$aicc #3467.9 aicc, auto arima suggests 2 more base ar components as well for an ARIMA(3,1,0)(2,0,0)

## [1] 3467.898

Arima(atm2TS, order=c(1,1,0), seasonal=c(2,0,0))$aicc #3546 aicc: my guess was higher

## [1] 3546.972

Arima(atm2TS, order=c(4,1,0), seasonal=c(2,0,0))$aicc #3467 aicc: slightly lower

## [1] 3467.575

Arima(atm2TS, order=c(5,1,0), seasonal=c(2,0,0))$aicc #3462 aicc

## [1] 3461.364

Arima(atm2TS, order=c(4,1,1), seasonal=c(2,0,0))$aicc #3395 aicc

## [1] 3395.711

Arima(atm2TS, order=c(4,1,1), seasonal=c(3,0,0))$aicc #3391 aicc

## [1] 3391.128

Arima(atm2TS, order=c(4,1,1), seasonal=c(3,0,1))$aicc #3381 aicc

## [1] 3381.526

Arima(atm2TS, order=c(4,1,1), seasonal=c(3,1,1))$aicc #3320 aicc

## [1] 3320.373

Arima(atm2TS, order=c(4,1,1), seasonal=c(3,2,1))$aicc #3315 aicc

## [1] 3315.13

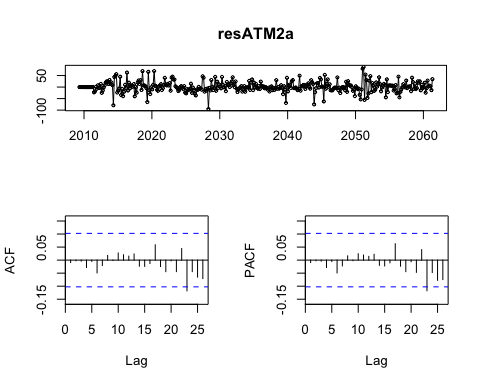
Arima(atm2TS, order=c(4,1,1), seasonal=c(4,2,1))$aicc #3311 aicc

## [1] 3311.629

Arima(atm2TS, order=c(5,1,3), seasonal=c(4,2,1))$aicc #3303 aicc:appears to be the lowest I can manually find

## [1] 3301.385

fitATM2a <- Arima(atm2TS, order=c(5,1,3), seasonal=c(4,2,1))  
resATM2a <- residuals(fitATM2a)  
tsdisplay(resATM2a) #significant spikes on both graphs at 23, but I've tried adding more components to the model with little to no improvement



Box.test(resATM2a, lag=26, fitdf=13, type="Ljung") #this passes the box-ljung test

##   
## Box-Ljung test  
##   
## data: resATM2a  
## X-squared = 16.457, df = 13, p-value = 0.2253

#Compare fitted models using a test set consiting of the last month of data. So we'll be fitting the model from may '09 to march '10, and forcast the atm usage in April '10.   
  
getrmse(atm2TS,h=30,order=c(5,1,3), seasonal=c(4,1,1),lambda=0) #25.25

## [1] 25.24575

getrmse(atm2TS,h=30,order=c(4,1,1), seasonal=c(4,2,1),lambda=0)

## [1] 30.01589

getrmse(atm2TS,h=30,order=c(4,1,1), seasonal=c(3,2,1),lambda=0) #28.45

## [1] 28.44755

getrmse(atm2TS,h=30,order=c(4,1,1), seasonal=c(3,1,1),lambda=0) #25.67

## [1] 25.66263

getrmse(atm2TS,h=30,order=c(4,1,1), seasonal=c(3,0,1),lambda=0) #25.87

## [1] 25.87606

getrmse(atm2TS,h=30,order=c(4,1,1), seasonal=c(3,0,0),lambda=0) #26.43

## [1] 26.43095

getrmse(atm2TS,h=30,order=c(4,1,1), seasonal=c(2,0,0),lambda=0) #27.42

## [1] 27.42026

getrmse(atm2TS,h=30,order=c(5,1,0), seasonal=c(2,0,0),lambda=0)

## [1] 41.5973

getrmse(atm2TS,h=30,order=c(4,1,0), seasonal=c(2,0,0),lambda=0)

## [1] 42.93765

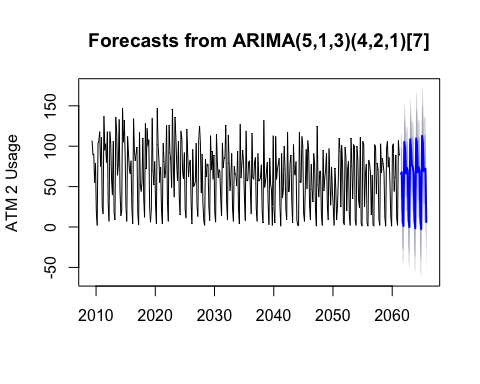
getrmse(atm2TS,h=30,order=c(1,1,0), seasonal=c(2,0,0),lambda=0) #26.13

## [1] 26.13035

getrmse(atm2TS,h=30,order=c(3,1,0), seasonal=c(2,0,0),lambda=0)

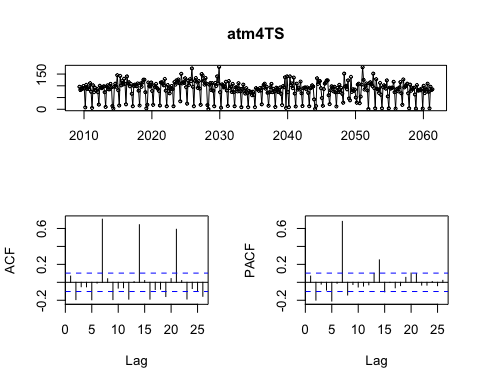
## [1] 41.72233

#Arima(5,1,3)(4,1,1) is best on both RMSE and AICc  
  
#forecasting:  
forecastATM2a <- forecast(fitATM2a, h=31)  
plot(forecastATM2a, ylab="ATM 2 Usage")



# ATM4

#Stationarity:  
tsdisplay(atm4TS)



#It's clear that there's still seasonal and monthly seasonality in the data. There also appears to be some cyclical behavior; tests suggest it doesn't need differencing:  
  
ndiffs(atm4TS) #suggests no differencing

## [1] 0

adf.test(atm4TS, alternative = "stationary") #says is stationary

## Warning in adf.test(atm4TS, alternative = "stationary"): p-value smaller  
## than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: atm4TS  
## Dickey-Fuller = -4.5587, Lag order = 7, p-value = 0.01  
## alternative hypothesis: stationary

kpss.test(atm4TS) #mostly stationary

##   
## KPSS Test for Level Stationarity  
##   
## data: atm4TS  
## KPSS Level = 0.45059, Truncation lag parameter = 4, p-value =  
## 0.05535

#Appears to be 2 seasonal patterns amounting to sAR(2), and likely one non-seasonal for ARIMA(1,0,0)(2,0,0). Basic Arima agrees with me.   
auto.arima(atm4TS)

## Series: atm4TS   
## ARIMA(1,0,0)(2,0,0)[7] with non-zero mean   
##   
## Coefficients:  
## ar1 sar1 sar2 mean  
## 0.1717 0.498 0.3114 82.7087  
## s.e. 0.0517 0.049 0.0495 7.1590  
##   
## sigma^2 estimated as 577.5: log likelihood=-1679.67  
## AIC=3369.35 AICc=3369.51 BIC=3388.85

#Manuel double checking:  
auto.arima(atm4TS)$aicc #3369 aicc

## [1] 3369.514

Arima(atm4TS, order=c(1,1,0), seasonal=c(2,0,0))$aicc #3494 aicc

## [1] 3494.026

Arima(atm4TS, order=c(2,0,0), seasonal=c(2,0,0))$aicc #3370 aicc

## [1] 3370.851

Arima(atm4TS, order=c(1,0,1), seasonal=c(2,0,0))$aicc #3370 aicc

## [1] 3370.396

Arima(atm4TS, order=c(1,0,0), seasonal=c(3,0,0))$aicc #3363 aicc: slightly lower than auto arima

## [1] 3363.484

Arima(atm4TS, order=c(1,0,0), seasonal=c(4,0,0))$aicc #3356 aicc: slightly lower

## [1] 3356.601

Arima(atm4TS, order=c(1,0,0), seasonal=c(5,0,0))$aicc #3353 aicc: marginally lower

## [1] 3353.315

Arima(atm4TS, order=c(1,0,0), seasonal=c(5,1,0))$aicc #3290 aicc: notably still lower

## [1] 3290.124

Arima(atm4TS, order=c(1,0,0), seasonal=c(5,2,0))$aicc #3331 aicc

## [1] 3331.673

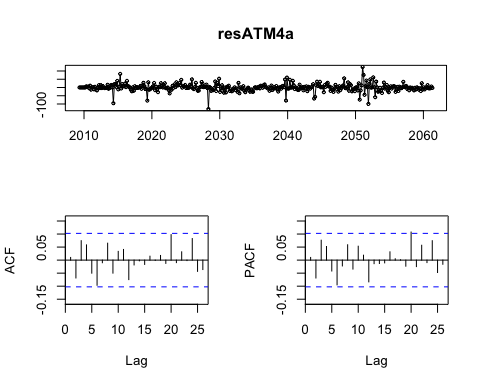
Arima(atm4TS, order=c(1,0,0), seasonal=c(5,1,1))$aicc #3288 aicc: another marginal improvement

## [1] 3288.977

Arima(atm4TS, order=c(1,0,0), seasonal=c(5,1,2))$aicc #3291 aicc: worse

## [1] 3291.033

fitATM4a <- Arima(atm4TS, order=c(1,0,0), seasonal=c(5,1,2))  
resATM4a <- residuals(fitATM4a)  
tsdisplay(resATM4a) #one slightly significant spike in the pacf, but I didn't find improvements to the parameters after further testing



Box.test(resATM4a, lag=26, fitdf=13, type="Ljung") #passes the box-ljung test

##   
## Box-Ljung test  
##   
## data: resATM4a  
## X-squared = 23.819, df = 13, p-value = 0.03284

#Compare fitted models using a test set consiting of the last month of data. So we'll be fitting the model from may '09 to march '10, and forcast the atm usage in April '10.   
  
getrmse(atm2TS,h=30,order=c(1,1,0), seasonal=c(2,0,0),lambda=0) #26.1

## [1] 26.13035

getrmse(atm2TS,h=30,order=c(2,0,0), seasonal=c(2,0,0),lambda=0)

## [1] 27.41837

getrmse(atm2TS,h=30,order=c(1,0,1), seasonal=c(2,0,0),lambda=0)

## [1] 27.47209

getrmse(atm2TS,h=30,order=c(1,0,0), seasonal=c(3,0,0),lambda=0) #26.46

## [1] 26.46497

getrmse(atm2TS,h=30,order=c(1,0,0), seasonal=c(4,0,0),lambda=0) #26.17

## [1] 26.17454

getrmse(atm2TS,h=30,order=c(1,0,0), seasonal=c(5,0,0),lambda=0, method="CSS")

## [1] 27.15686

getrmse(atm2TS,h=30,order=c(1,0,0), seasonal=c(5,1,0),lambda=0, method="CSS") #23.72

## [1] 23.72505

getrmse(atm2TS,h=30,order=c(1,0,0), seasonal=c(5,2,0),lambda=0, method="CSS")

## [1] 75.77177

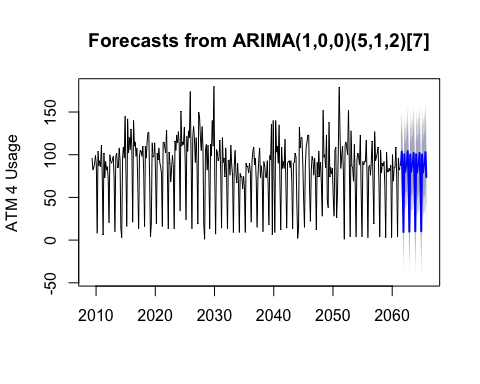
getrmse(atm2TS,h=30,order=c(1,0,0), seasonal=c(5,1,1),lambda=0, method="CSS") #23.79

## [1] 23.79254

getrmse(atm2TS,h=30,order=c(1,0,0), seasonal=c(5,1,2),lambda=0) #24.15

## [1] 24.1542

#Arima(1,0,0)(5,1,0) is best on RMSE and second best on AICc, and is the simpler of the two  
  
#forecasting:  
forecastATM4a <- forecast(fitATM4a, h=31)  
plot(forecastATM4a, ylab="ATM 4 Usage")

 ###Model Selection and Forecast: STL vs Holt-Winters vs Seasonal Arima

# ATM 1

#STL:  
getrmseSTL(atm1TS, 30, 4, 7) #31.99

## [1] 31.99633

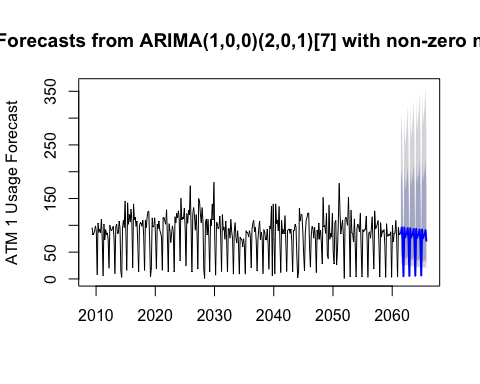
#Holt-Winters  
getrmseHW(hwATM1, 30, "multiplicative") #11.31

## [1] 11.31314

#Arima:  
getrmse(atm1TS,h=30,order=c(1,0,0),seasonal=c(2,0,1),lambda=0) #lowest rmse of 8.54

## [1] 8.537476

#Will choose seasonal arima, which had the lowest RMSE:  
forecastATM1c <- forecast(fitATM1c, h=31)  
plot(forecastATM1c, ylab="ATM 1 Usage Forecast")



# ATM 2

#STL:  
getrmseSTL(atm2TS, 30, 7, 4) #18.28

## [1] 18.27538

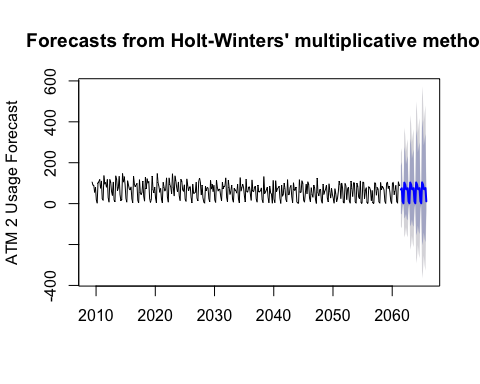
#Holt-Winters  
getrmseHW(hwATM2, 30, "damped") #16.87 #lowest RMSE

## [1] 16.86995

#Arima:  
getrmse(atm2TS,h=30,order=c(5,1,3), seasonal=c(4,1,1),lambda=0) #25.25

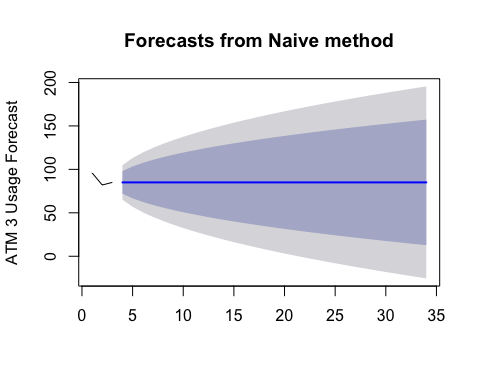
## [1] 25.24575

#Will choose Holt-Winters multiplicative damped, which had the lowest RMSE:  
forecastATM2hw <- hw(hwATM2,seasonal="multiplicative", Damped = TRUE, h = 31)  
plot(forecastATM2hw, ylab="ATM 2 Usage Forecast")



# ATM 3

atm3Fit <- naive(atm3TS.active, 31)  
forecastATM3n <- forecast(atm3Fit, 31)  
plot(forecastATM3n, ylab="ATM 3 Usage Forecast")



# ATM 4

#STL:  
getrmseSTL(atm4TS, 30, 4, 7) #30.81

## [1] 30.81808

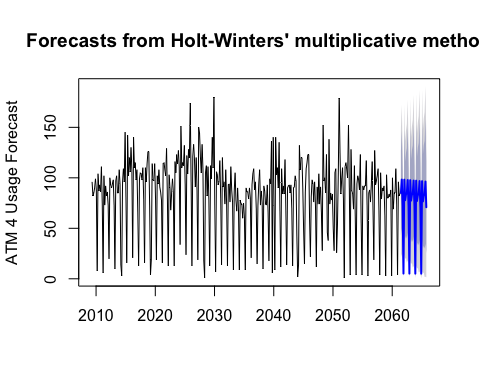
#Holt-Winters  
getrmseHW(hwATM4, 30, "damped") #10.89

## [1] 10.89819

#Arima:  
getrmse(atm2TS,h=30,order=c(1,0,0), seasonal=c(5,1,0),lambda=0, method="CSS") #23.72

## [1] 23.72505

#Will choose Holt-Winters multiplicative damped, which had the lowest RMSE:  
forecastATM4hw <- hw(hwATM4,seasonal="multiplicative", Damped = TRUE, h = 31)  
plot(forecastATM4hw, ylab="ATM 4 Usage Forecast")

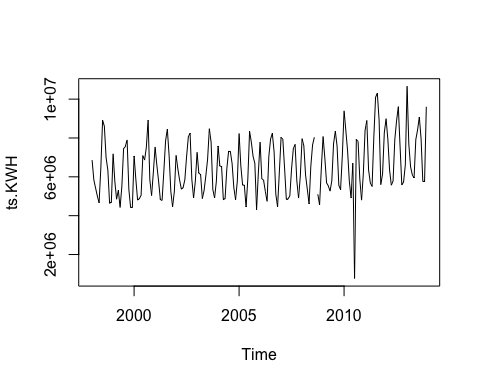


write.csv(forecastATM4hw, file="atmfcast.csv")

## Part 2: Residential Customer Forecast

# Import and Transfom

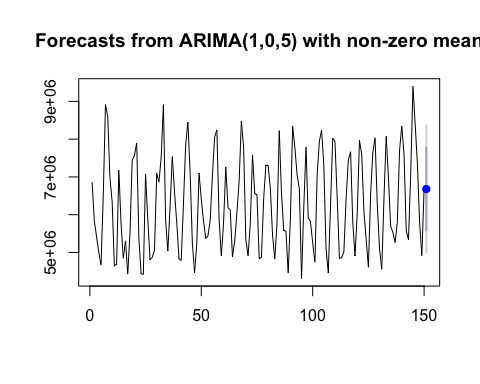
library(dplyr)  
library(lubridate)  
  
dat <- read.csv("https://raw.githubusercontent.com/aagoldberg/Work/master/ResidentialCustomerForecastLoad-624.csv")  
#Create time series  
ts.KWH <- ts(dat$KWH, start=c(1998, 1), end=c(2013,12), frequency = 12)  
  
#Basic data hygiene  
plot(ts.KWH) #looks like one clear outlier



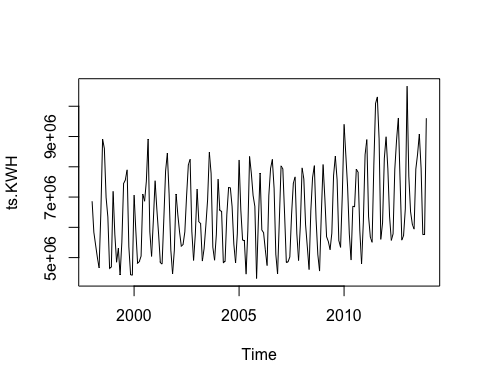
summary(ts.KWH) #and 1 NA

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 770523 5429912 6283324 6502475 7620524 10655730 1

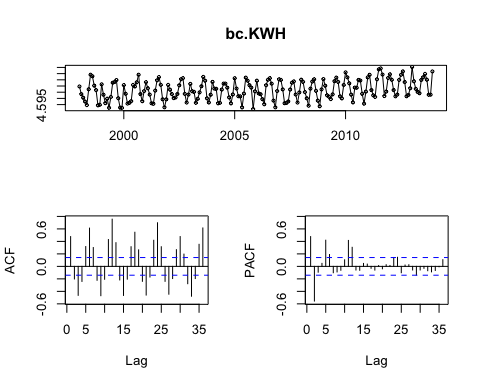
#Modeling NA  
NA.pos <- which(is.na(ts.KWH))  
upto.NA <- ts.KWH[1:NA.pos]  
NA.train.end <- time(upto.NA)[length(upto.NA)-1]  
NA.train <- window(upto.NA,end=NA.train.end)  
NA.fit <- Arima(NA.train)  
NA.fc <- forecast(NA.fit,h=1)  
ts.KWH[NA.pos] <- as.integer(NA.fc$mean)  
  
#Modeling outlier  
outlier.pos <- which(ts.KWH == 770523)  
upto.outlier <- ts.KWH[1:outlier.pos]  
outlier.train.end <- time(upto.outlier)[length(upto.outlier)-1]  
outlier.train <- window(upto.outlier,end=outlier.train.end)  
outlier.fit <- auto.arima(outlier.train, seasonal=TRUE)  
outlier.fc <- forecast(outlier.fit,h=1)  
plot(outlier.fc)



ts.KWH[outlier.pos] <- as.integer(outlier.fc$mean)  
  
#Plot with modeled NA and outlier  
plot(ts.KWH)



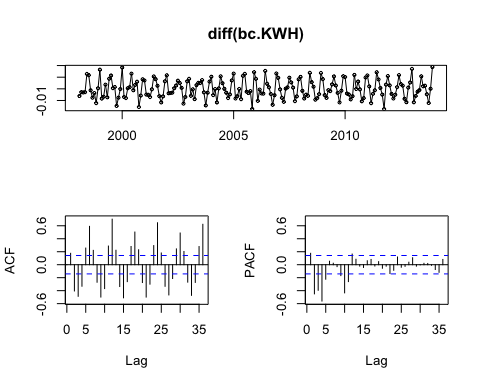
#Check if box-cox transform is helpful  
lambdaBC <- BoxCox.lambda(ts.KWH) #non trivial lambda  
bc.KWH <- BoxCox(ts.KWH, lambdaBC)  
  
#Examine new plots  
tsdisplay(bc.KWH)



ndiffs(bc.KWH) #says one difference

## [1] 1

tsdisplay(diff(bc.KWH)) #looks ok



kpss.test(diff(bc.KWH)) #passes, I believe, since it says "p-value greater than printed p-value"

## Warning in kpss.test(diff(bc.KWH)): p-value greater than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: diff(bc.KWH)  
## KPSS Level = 0.018391, Truncation lag parameter = 3, p-value = 0.1

adf.test(diff(bc.KWH)) #passes

## Warning in adf.test(diff(bc.KWH)): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: diff(bc.KWH)  
## Dickey-Fuller = -9.3126, Lag order = 5, p-value = 0.01  
## alternative hypothesis: stationary

# STL Decomposition

getrmseSTL <- function(x,h,t,s)  
{  
 train.end <- time(x)[length(x)-h]  
 test.start <- time(x)[length(x)-h+1]  
 train <- window(x,end=train.end)  
 test <- window(x,start=test.start)  
 fit <- stl(train, t.window=t, s.window= s, robust=TRUE)  
 fcast <- forecast(fit, method="naive")  
 return(accuracy(fcast,test)[2,"RMSE"])  
}  
getrmseSTL(ts.KWH, 12, 7, "periodic") #1133616

## [1] 1133616

getrmseSTL(ts.KWH, 12, 4, "periodic") #1163481

## [1] 1163481

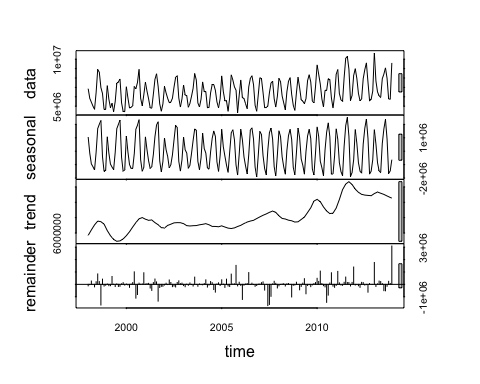
getrmseSTL(ts.KWH, 12, 4, 7) #1132799

## [1] 1132799

getrmseSTL(ts.KWH, 12, 12, 4) #1047176 #winner

## [1] 1047176

ts.KWH.stl <- stl(ts.KWH, t.window=12, s.window=4, robust=TRUE)  
plot(ts.KWH.stl)

 #STL Decomposition

getrmseSTL <- function(x,h,t,s)  
{  
 train.end <- time(x)[length(x)-h]  
 test.start <- time(x)[length(x)-h+1]  
 train <- window(x,end=train.end)  
 test <- window(x,start=test.start)  
 fit <- stl(train, t.window=t, s.window= s, robust=TRUE)  
 fcast <- forecast(fit, method="naive")  
 return(accuracy(fcast,test)[2,"RMSE"])  
}  
getrmseSTL(ts.KWH, 12, 7, "periodic") #1133616

## [1] 1133616

getrmseSTL(ts.KWH, 12, 4, "periodic") #1163481

## [1] 1163481

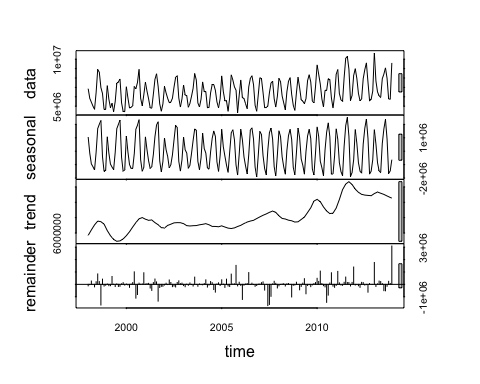
getrmseSTL(ts.KWH, 12, 4, 7) #1132799

## [1] 1132799

getrmseSTL(ts.KWH, 12, 12, 4) #1047176 #winner

## [1] 1047176

ts.KWH.stl <- stl(ts.KWH, t.window=12, s.window=4, robust=TRUE)  
plot(ts.KWH.stl)



# Holt-Winters Seasonal

hw.StSp <- ets(ts.KWH, model="ZZZ", lambda=lambdaBC)  
#state space model selection suggests an MNM model (additive errors, additive trend, additive seasonality)  
  
hw.KWH <- window(ts.KWH, t.window=12, s.window=4)  
  
hw.A <- hw(hw.KWH,seasonal="additive", lambda=lambdaBC)  
hw.M <- hw(hw.KWH,seasonal="multiplicative")  
hw.DM <- hw(hw.KWH,seasonal="multiplicative", Damped = TRUE)  
  
#ATM Exponential Smoothing model results:  
A.fit <- hw.A$model[2:6]  
M.fit <- hw.M$model[2:6]  
DM.fit <- hw.DM$model[2:6]  
SS.fit <- hw.StSp[2:6]  
KWH.expfit <- rbind.data.frame(Additive = A.fit, Multiplicative = M.fit, Damped\_Multi = DM.fit, StateSpace = SS.fit)  
KWH.expfit <- KWH.expfit %>% mutate(rmse = sqrt(mse))  
row.names(KWH.expfit) <- c("Additive", "Mutliplicative", "DampedMultiplicative", "StateSpace")  
KWH.expfit #additive model looks best

## aic bic aicc mse  
## Additive -1135.541 -1080.164 -1132.024 1.178298e-05  
## Mutliplicative 6150.913 6206.290 6154.430 3.984254e+11  
## DampedMultiplicative 6150.913 6206.290 6154.430 3.984254e+11  
## StateSpace -1135.541 -1080.164 -1132.024 1.178298e-05  
## amse rmse  
## Additive 1.224021e-05 3.432634e-03  
## Mutliplicative 4.146821e+11 6.312095e+05  
## DampedMultiplicative 4.146821e+11 6.312095e+05  
## StateSpace 1.224021e-05 3.432634e-03

getrmseHW <- function(x,h,model)  
{  
 train.end <- time(x)[length(x)-h]  
 #print(train.end)  
 test.start <- time(x)[length(x)-h+1]  
 #print(test.start)  
 train <- window(x,end=train.end)  
 #print(train)  
 test <- window(x,start=test.start)  
 #print(test)  
 if(model == "additive"){fit5 <- hw(train,seasonal = "additive", h=h)}  
 if(model == "multiplicative"){fit5 <- hw(train, seasonal="multiplicative", h=h)}  
 if(model == "damped"){fit5 <- hw(train,seasonal="multiplicative", Damped = TRUE, h=h)}  
 if(model == "statespace"){  
 fitets <- ets(train, model="ZZZ",lambda=lambdaBC)  
 fit5 <- forecast(fitets, h=h)}  
 #print(str(fit5))  
 #fc <- forecast(fit,h=h)  
 return(accuracy(fit5,test)[2,"RMSE"])  
}  
getrmseHW(hw.KWH, 12, "additive") #1034801

## [1] 1034801

getrmseHW(hw.KWH, 12, "multiplicative") #947832

## [1] 947832

getrmseHW(ts.KWH, 12, "damped") #947832

## [1] 947832

getrmseHW(hw.KWH, 12, "statespace") #1049509

## [1] 1049509

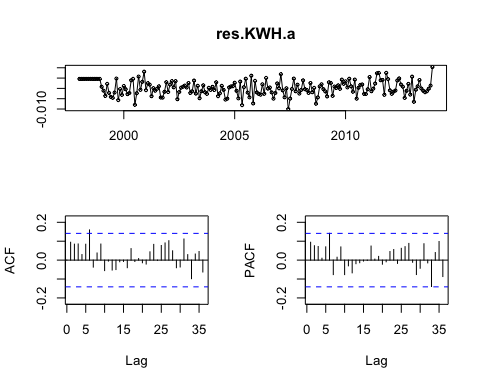
#The additive models with box-cox performs better on aic, aicc and rmse tests, yet the mutliplicative models marginally fit the data better (box-cox appears to also hurt here). I'll go with the additive, in concerns of overfitting.

# Seasonal Arima modeling

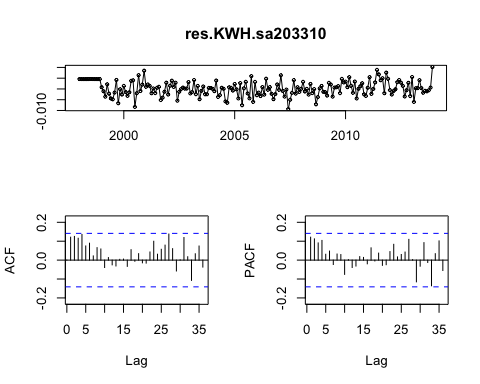
#Looks like seasonal and monthly seasonality (sAR(2)) with one seasonal base trend (sMA(1)) and 1-2 non-seasonal trends (MA(2))  
  
#Check auto-arima for baseline model  
saa.KWH <- auto.arima(ts.KWH, lambda = lambdaBC, stepwise=FALSE, approximation=FALSE) #auto-arima says (0,0,3)(2,1,0)  
saa.KWH$aicc #-1510 aicc

## [1] -1510.866

#Check if I can manually improve: all models have similar aicc  
sa103210 <- Arima(ts.KWH, order=c(1,0,3), seasonal=c(2,1,0), lambda = lambdaBC, include.drift = TRUE) #-1508 aicc:   
sa203210 <- Arima(ts.KWH, order=c(2,0,3),seasonal=c(2,1,0), lambda = lambdaBC, include.drift = TRUE) #-1510 aicc:   
sa203310 <- Arima(ts.KWH, order=c(2,0,3),seasonal=c(3,1,0), lambda = lambdaBC, include.drift = TRUE) #-1509 aicc:   
sa203410 <- Arima(ts.KWH, order=c(2,0,3),seasonal=c(4,1,0), lambda = lambdaBC, include.drift = TRUE) #-1504 aicc:   
  
res.KWH.a <- residuals(saa.KWH)  
tsdisplay(res.KWH.a) #both ACF and PACF have a slightly significant spike early on.



res.KWH.sa203310 <- residuals(sa203310)  
tsdisplay(res.KWH.sa203310) #spikes are no longer officially significant.



Box.test(res.KWH.sa203310, lag=36, fitdf=8, type="Ljung") #passes box-ljung test

##   
## Box-Ljung test  
##   
## data: res.KWH.sa203310  
## X-squared = 37.341, df = 28, p-value = 0.1115

#Now check RMSEs  
getrmse <- function(x,h,...)  
{  
 train.end <- time(x)[length(x)-h]  
 test.start <- time(x)[length(x)-h+1]  
 train <- window(x,end=train.end)  
 test <- window(x,start=test.start)  
 fit <- Arima(train,...)  
 fc <- forecast(fit,h=h)  
 return(accuracy(fc,test)[2,"RMSE"])  
}  
  
getrmse(ts.KWH,h=12,order=c(0,0,3),seasonal=c(2,1,0),lambda=lambdaBC, include.drift=TRUE) #948448

## [1] 948448.4

getrmse(ts.KWH,h=12,order=c(1,0,3),seasonal=c(2,1,0),lambda=lambdaBC, include.drift=TRUE) #945567

## [1] 945567

getrmse(ts.KWH,h=12,order=c(2,0,3),seasonal=c(2,1,0),lambda=lambdaBC, include.drift=TRUE) #933451

## [1] 933451.1

getrmse(ts.KWH,h=12,order=c(2,0,3),seasonal=c(3,1,0),lambda=lambdaBC, include.drift=TRUE) #931248

## [1] 931247.9

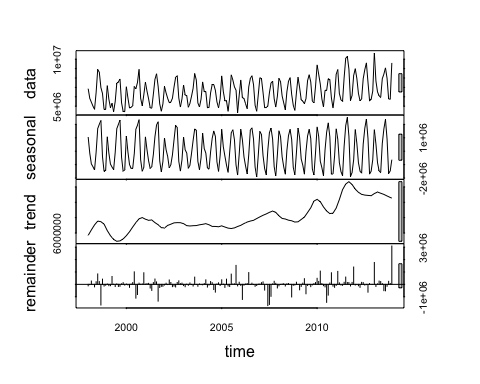
getrmse(ts.KWH,h=12,order=c(2,0,3),seasonal=c(4,1,0),lambda=lambdaBC, include.drift=TRUE) #919864 #best

## [1] 919864.4

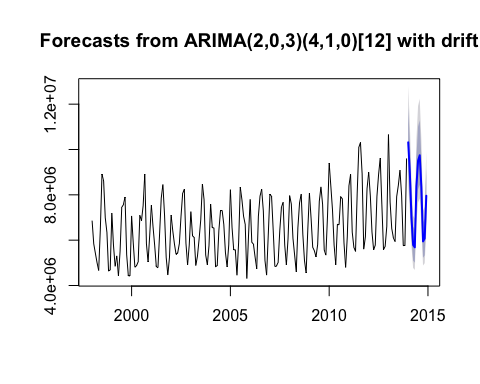
#while (203)(410) has the best rmse, I'm a little worried about overfitting, so going with (203)(310) with drift

# Model Choice and Forecasting

#STL:  
ts.KWH.stl <- stl(ts.KWH, t.window=12, s.window=4, robust=TRUE) #RMSE 1047176  
plot(ts.KWH.stl)



#Holt-Winters Seasonal (additive)  
hw.A <- hw(hw.KWH,seasonal="additive", lambda=lambdaBC) #RMSE 1034801  
  
#Seasonal Arima  
sa203410 <- Arima(ts.KWH, order=c(2,0,3),seasonal=c(4,1,0), lambda = lambdaBC, include.drift = TRUE) #RMSE 919864  
  
#Choosing Arima, which has the best RMSE  
hw.Arima.forecast <- forecast(sa203410, h=12)  
plot(hw.Arima.forecast)



write.csv(hw.Arima.forecast, file="KWHfcast.csv")