Project 1

Cheryl Bowersox

Monday, April 09, 2018

# Project 1

## Part A

Create a forecast for how much is withdrawn from four ATM machines in May 2010 using given data.

The data was saved as a .csv file and loaded into R. It was evaluated for obvious outliers and missing values.

**Outliers and Missing values**

One value was found that appeared to be in error, on 2/9/2010 ATM 4 showed cash dispersed of over $10 Million. This is assumed to be in error, and was removed from the data set. When evaluating missing values, ATM3 appeared to have zero measurements from May 2009 through April 2010. It could be assumed from this information that ATM3 was not in use in 2009 and only began use at the end of April 2010.

Additionally, there are 3 missing values from ATM1, 2 from ATM2, and 14 with no ATM assigned.Since the data set is large these missing values are removed. All other zero values are assumed to be correct data points where no cash was withdrawn on that data.

suppressWarnings(library(fma))

## Loading required package: tseries  
## Loading required package: forecast  
## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric  
##   
## Loading required package: timeDate  
## This is forecast 7.1

suppressWarnings(library(fpp))

## Loading required package: expsmooth  
## Loading required package: lmtest

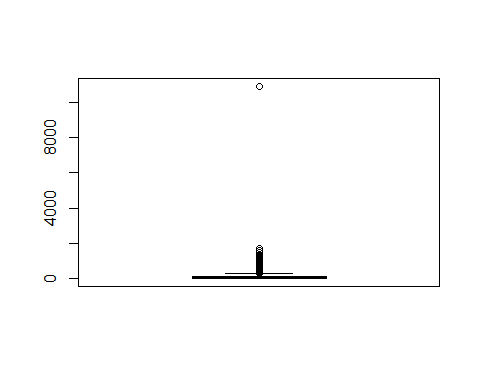
suppressWarnings(library(forecast))  
suppressWarnings(library(dplyr))

##   
## 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

suppressWarnings(library(xts))

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

suppressWarnings(library(ggplot2))  
  
#load data  
f <- "ATM624data.csv"  
atmdata <- read.csv(file = f,stringsAsFactors= FALSE, strip.white = TRUE)  
boxplot(atmdata$Cash)



summary(atmdata)

## DATE ATM Cash   
## Length:1474 Length:1474 Min. : 0.0   
## Class :character Class :character 1st Qu.: 0.5   
## Mode :character Mode :character Median : 73.0   
## Mean : 155.6   
## 3rd Qu.: 114.0   
## Max. :10920.0   
## NA's :19

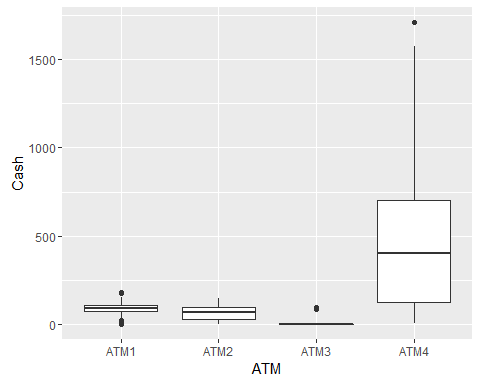
#check for NA  
atmNA <- atmdata[is.na(atmdata$Cash),]  
(atmNA%>%group\_by(ATM)%>%summarise(a=n()))

## Source: local data frame [3 x 2]  
##   
## ATM a  
## 1 14  
## 2 ATM1 3  
## 3 ATM2 2

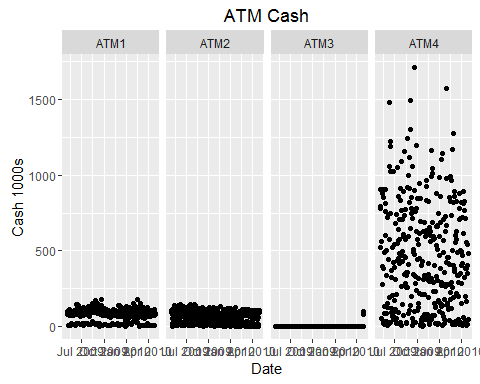
#new data set, no missing vaues and no max Cash  
atmdf<-atmdata[complete.cases(atmdata), ]  
x <- max(atmdf$Cash)  
  
atmdf<-atmdf%>%filter(Cash < x)  
  
atmdf$DATE <- as.Date(as.character(atmdf$DATE),format="%m/%d/%Y")

Reviewing the distribution of individual ATMS indicates that they have different models and should be forecast separately.

ggplot(aes(y = Cash, x = ATM), data = atmdf) + geom\_boxplot()



cashplot <- ggplot(atmdf, aes(DATE, Cash)) +  
 geom\_point() +  
 ggtitle("ATM Cash") +  
 xlab("Date") + ylab("Cash 1000s")  
(cashplot + facet\_grid(. ~ ATM))



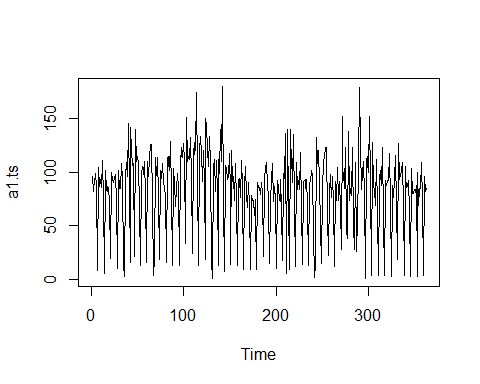
#convert to ts data ATM3 is a special case  
atm1 <- atmdf%>%filter(ATM == "ATM1")%>%select(DATE,Cash)  
atm2 <- atmdf%>%filter(ATM == "ATM2")%>%select(DATE,Cash)  
atm3 <- atmdf%>%filter(ATM == "ATM3")%>%select(DATE,Cash)  
atm4 <- atmdf%>%filter(ATM == "ATM4")%>%select(DATE,Cash)  
  
#min max ATM1  
min(atm1$DATE)

## [1] "2009-05-01"

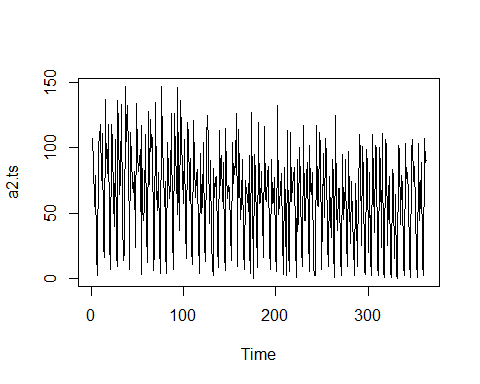
max(atm1$DATE)

## [1] "2010-04-30"

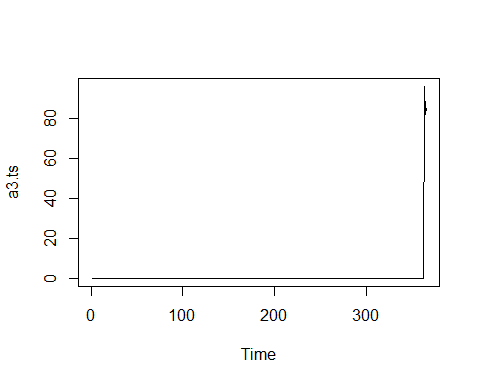
a1.ts <- as.ts(x = atm1[, -1], order.by = atm1$DATE)  
a2.ts <- as.ts(x = atm2[, -1], order.by = atm2$DATE)  
a3.ts <- as.ts(x = atm3[, -1], order.by = atm3$DATE)  
a4.ts <- as.ts(x = atm4[, -1], order.by = atm4$DATE)  
  
plot(a1.ts)



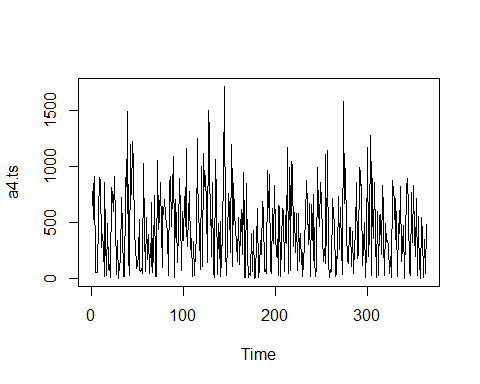
plot(a2.ts)



plot(a3.ts)



plot(a4.ts)



#Standard Deviation  
sd(a1.ts)

## [1] 36.65654

sd(a2.ts)

## [1] 38.89846

sd(a3.ts)

## [1] 7.944778

sd(a4.ts)

## [1] 351.3846

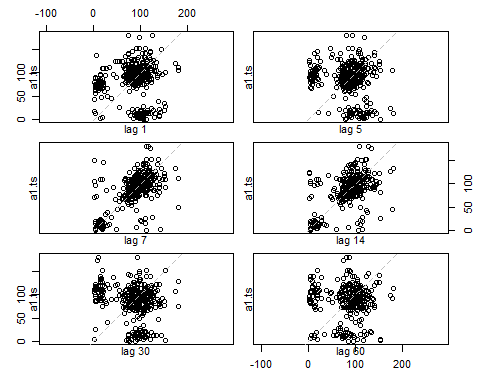
Plotting and examining the time series for each ATM shows that ATM 3 is special case in that it only has 3 observations, and ATM 4 also appears to be a unique case as it has a very large standard deviation, 351.3845996 and a transformation might be appropriate. Each ATM should be forecast separately.

**Model for ATM1**

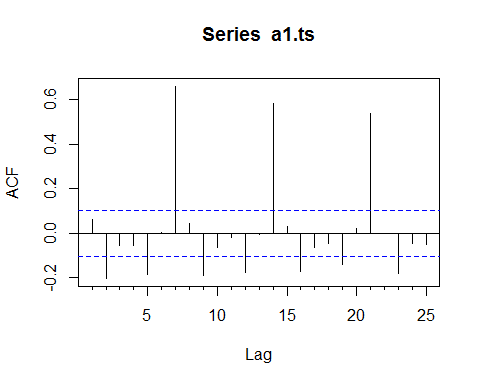
ATM1 has 362 data points, the first 292 points were used to train the models, with the remaining 70 (10 weeks) were used to evaluate the model. There does not appear to be a long-term or seasonal pattern to this ATM usage. The appears to be a strong weekly component when evaluating for different lag cycles.

Three models were selected to compare for forecasting ATM1 data: a Naive model, adjusted, a seasonally adjusted additive decomposition and an auto -AIRM model.

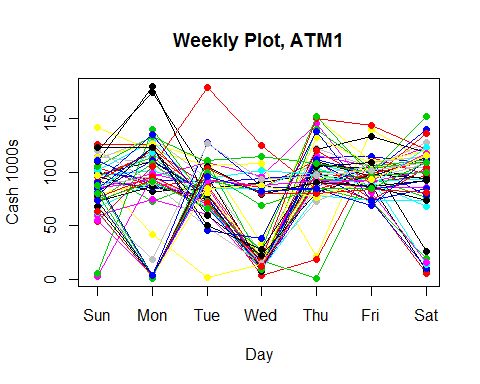
# checking ACF  
   
#Model for ATM1  
  
#testing - for weekly,biweekly lag  
lagdays <- c(1,5, 7, 14, 30,60)  
lag.plot(a1.ts, lags = 6, set.lags = lagdays, do.lines = FALSE)



Acf(a1.ts, plot=T)



#seasonal adjustment  
a1.ts <- ts(a1.ts,frequency=7)  
traina1 <- window(a1.ts,start = 1,end=42)  
testa1 <- window(a1.ts, start=43)  
  
seasonplot(a1.ts,ylab="Cash 1000s", xlab="Day", main="Weekly Plot, ATM1", col=1:20, pch=19)

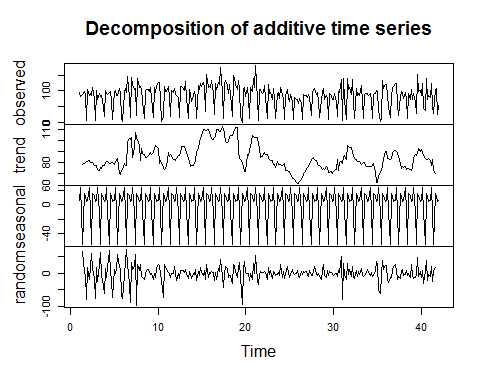


#Models  
#Naive  
fit1a1 <- naive(traina1, h=70)  
fcst1a1 <-forecast(fit1a1, h=70)  
accuracy(fit1a1, testa1)

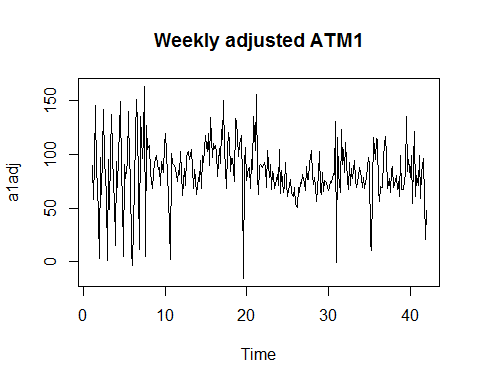
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.1463415 49.89739 37.24042 -86.69824 121.3876 1.953436  
## Test set 26.2031250 43.63503 39.98438 -209.70988 277.2967 2.097370  
## ACF1 Theil's U  
## Training set -0.35723260 NA  
## Test set -0.06680284 0.509645

Using Holt-Winters additive decomposition to adjust for the weekly trend results in

a1cad <- decompose(traina1)  
plot(a1cad)



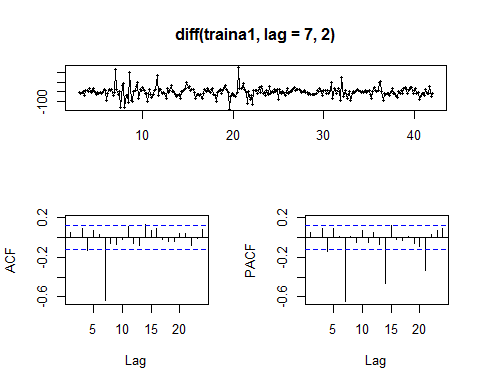
a1adj<- traina1-a1cad$seasonal  
plot(a1adj, main="Weekly adjusted ATM1")



#Holt-Winters seasonal method - additive, seasons are consistent  
fit2a1 <- hw(traina1,seasonal="additive", h=70)  
fcst2a1 <-forecast(fit2a1,h=70)

For the 3rd model select, the auto.arima algorithm was allowed to select the best ARIMA model. The best model selected was ARIMA(1,0,0)(2,0,0)[7] This model helps

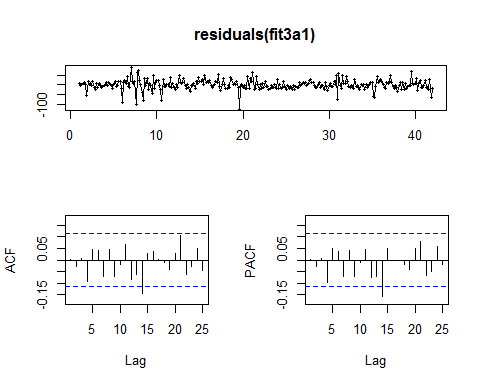
#ARIMA Model - machine selected  
#take the seasonal difference  
tsdisplay(diff(traina1,lag =7,2))



(fit3a1 <- auto.arima(traina1))

## Series: traina1   
## ARIMA(1,0,0)(2,0,0)[7] with non-zero mean   
##   
## Coefficients:  
## ar1 sar1 sar2 intercept  
## 0.1065 0.4449 0.3387 82.6682  
## s.e. 0.0589 0.0550 0.0561 6.9387  
##   
## sigma^2 estimated as 657.5: log likelihood=-1343.94  
## AIC=2697.88 AICc=2698.09 BIC=2716.19

tsdisplay(residuals(fit3a1))



Box.test(residuals(fit3a1), lag=7, fitdf=4, type="Ljung")

##   
## Box-Ljung test  
##   
## data: residuals(fit3a1)  
## X-squared = 5.5802, df = 3, p-value = 0.1339

#f2<-auto.arima(traina1, stepwise=FALSE, approximation=FALSE)  
  
fcst3a1=forecast(fit3a1, h=70)

Comparing the accuracy of the 3 models shows the, ARMIA model provides the best accuracy

round(accuracy(fcst1a1, testa1),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set -0.15 49.90 37.24 -86.70 121.39 1.95 -0.36 NA  
## Test set 26.20 43.64 39.98 -209.71 277.30 2.10 -0.07 0.51

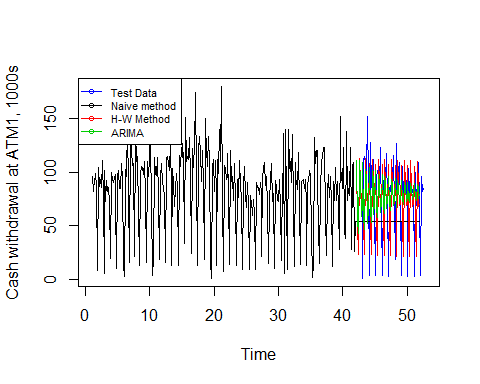
round(accuracy(fcst2a1, testa1),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set -0.55 25.75 17.66 -90.33 107.88 0.93 0.12 NA  
## Test set 5.98 54.59 42.89 -477.68 526.29 2.25 -0.17 0.54

round(accuracy(fcst3a1, testa1),2)

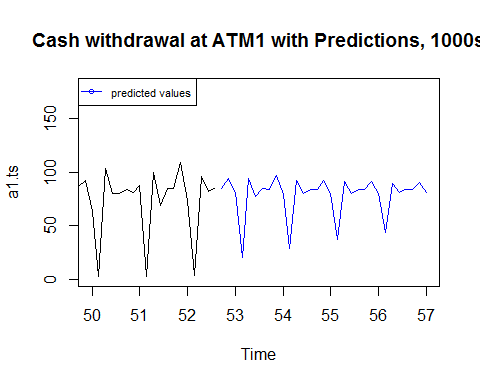
## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set 0.14 25.46 17.75 -92.45 108.04 0.93 0.00 NA  
## Test set 1.10 44.63 31.55 -450.40 480.59 1.65 -0.16 0.43

plot(traina1,ylab="Cash withdrawal at ATM1, 1000s", xlim=c(1,53))  
lines(testa1, col=4)  
lines(fcst1a1$mean, col=1)  
lines(fcst2a1$mean, col=2)  
lines(fcst3a1$mean, col=3)  
legend("topleft", lty=1, col=c(4,1,2,3),  
 legend=c("Test Data", "Naive method","H-W Method","ARIMA"),pch=1,cex=.69)

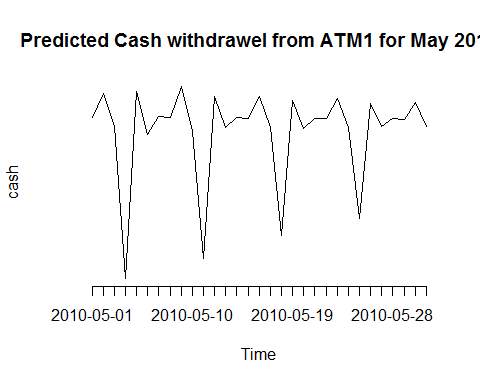


Refitting the ARIMA model to calculate for all data and create predictions for the next 30 days (May 2010) give the following model:

refit1 <- Arima( model=fit3a1, a1.ts)  
preda1 <- predict(refit1, n.ahead=31)  
  
plot(a1.ts,main="Cash withdrawal at ATM1 with Predictions, 1000s", xlim=c(50,57))  
lines(preda1$pred, col=4)  
legend("topleft", lty=1, col=c(4),  
 legend=c("predicted values"),pch=1,cex=.69)



#Just predictions  
d <- seq(atm1[362,1]+1,atm1[362,1]+31,by="days")  
p<- ts(preda1$pred,frequency=1)  
plot(p, axes=F, main="Predicted Cash withdrawel from ATM1 for May 2010", ylab="cash")  
axis(1,at=1:31, labels =d)



#Summary Stats  
summary(as.vector(p))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 20.69 80.26 83.80 78.69 89.90 96.56

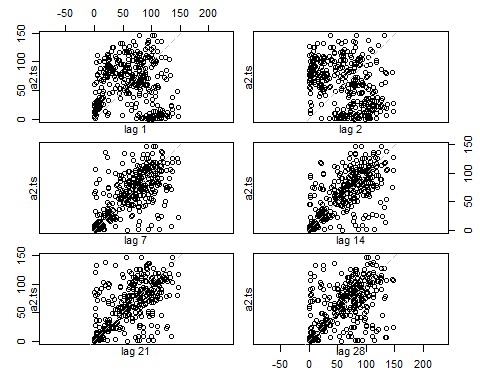
dfPred<-data.frame(Date = d)  
dfPred$ATM1<- preda1$pred

**Model for ATM2**

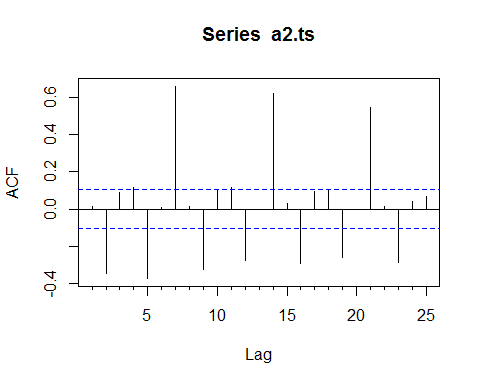
ATM2 has 363 data points, the first 294 points were used to train the models, with the remaining 69 (~10 weeks) were used to evaluate the model. There does not appear to be a long-term or seasonal pattern to this ATM usage. Like ATM1, the appears to be a strong weekly component when evaluating for different lag cycles at 7, 14, and 21 days.

Because of the similarity to ATM1 data set, three similar models were evaluated.

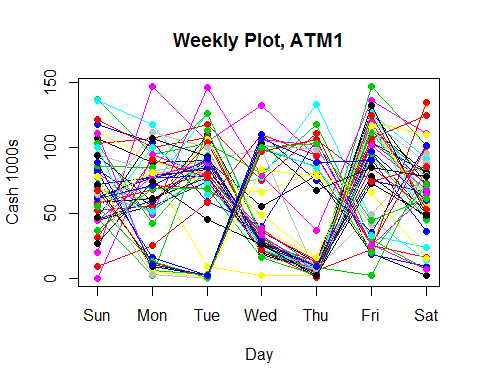
# checking ACF  
  
#testing - for weekly,biweekly lag  
lagdays <- c(1,2, 7, 14, 21,28)  
lag.plot(a2.ts, lags = 6, set.lags = lagdays, do.lines = FALSE)



Acf(a2.ts, plot=T)



#Model for ATM2  
a2.ts <- ts(a2.ts,frequency=7)  
traina2 <- window(a2.ts,start = 1,end=42)  
testa2 <- window(a2.ts, start=43)  
  
seasonplot(a2.ts,ylab="Cash 1000s", xlab="Day", main="Weekly Plot, ATM1", col=1:20, pch=19)

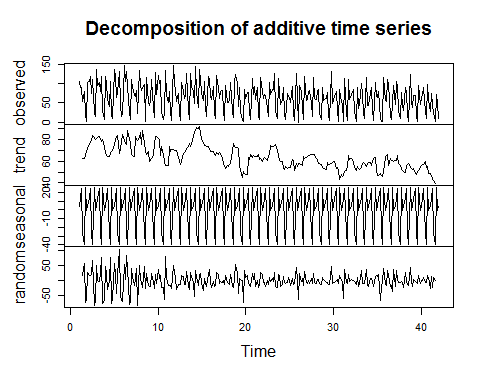


#Models  
#Naive  
fit1a2 <- snaive(traina2, h=70)  
fcst1a2 <-forecast(fit1a2, h=70)  
accuracy(fit1a2, testa2)

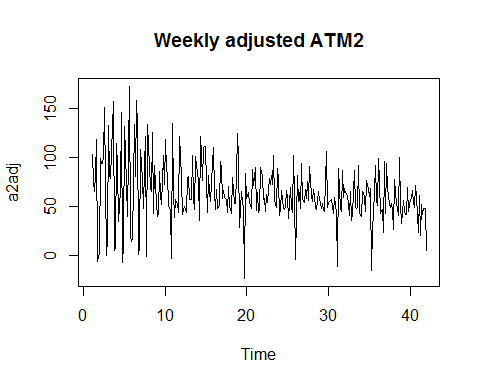
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -0.5836299 32.16417 21.92883 -Inf Inf 1.000000 -0.01225779  
## Test set 16.9843750 62.02129 57.10938 -Inf Inf 2.604306 -0.07667560  
## Theil's U  
## Training set NA  
## Test set NaN

Using Holt-Winters additive decomposition to adjust for the weekly trend results in

a2cad <- decompose(traina2)  
plot(a2cad)



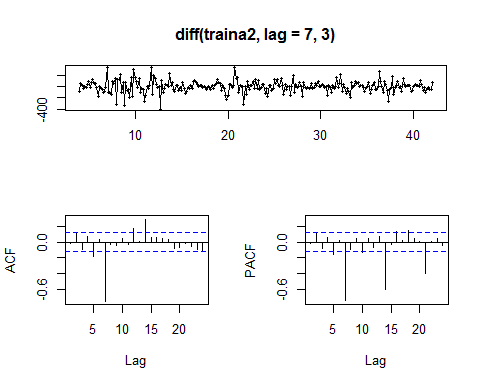
a2adj<- traina2-a2cad$seasonal  
plot(a2adj, main="Weekly adjusted ATM2")



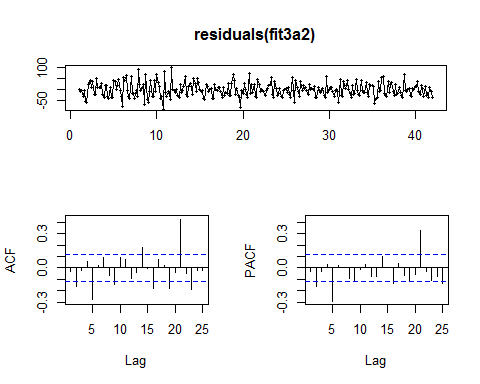
#Holt-Winters seasonal method - additive, seasons are consistent  
fit2a2 <- hw(traina2,seasonal="additive", h=70)  
fcst2a2 <-forecast(fit2a2,h=70)

For the 3rd model select, the auto.arima algorithm was allowed to select the best ARIMA model. The best model selected was ARIMA(1,0,0)(2,0,0)[7] This model helps

#ARIMA Model - machine selected  
#take the seasonal difference  
tsdisplay(diff(traina2,lag =7,3))



fit3a2 <- auto.arima(traina2, allowdrift=F)  
tsdisplay(residuals(fit3a2))



Box.test(residuals(fit3a2), lag=7, fitdf=4, type="Ljung")

##   
## Box-Ljung test  
##   
## data: residuals(fit3a2)  
## X-squared = 34.4316, df = 3, p-value = 1.606e-07

#f2<-auto.arima(traina1, stepwise=FALSE, approximation=FALSE)  
  
fcst3a2=forecast(fit3a2, h=70)

Comparing the accuracy of the 3 models the ARIMA model appears to be a better fit, as the MASE is significantly lower than the other models, and at the RMSE is comparable to the seasonal naive method.Like the forecast for ATM 1, the ARIMA model does appear to be narrowing down the further out it predicting, possibly providing a conservative forecast. This could be problematic if it routinely under-forecasting the withdrawals expected.

round(accuracy(fcst1a2, testa2),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set -0.58 32.16 21.93 -Inf Inf 1.0 -0.01 NA  
## Test set 16.98 62.02 57.11 -Inf Inf 2.6 -0.08 NaN

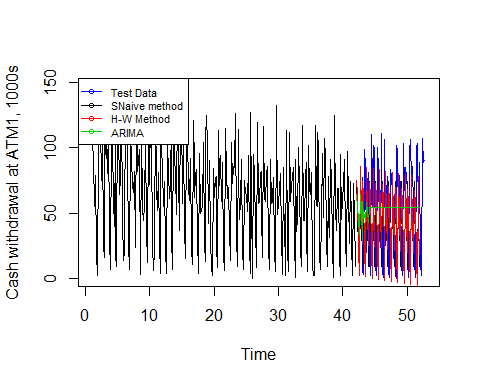
round(accuracy(fcst2a2, testa2),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set -0.25 27.75 19.07 -Inf Inf 0.87 -0.01 NA  
## Test set 14.14 64.39 59.77 -Inf Inf 2.73 -0.04 NaN

round(accuracy(fcst3a2, testa2),2)

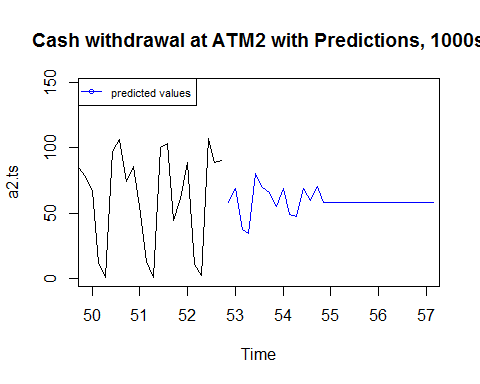
## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set -2.63 31.04 24.71 -Inf Inf 1.13 -0.03 NA  
## Test set 2.36 39.59 36.45 -Inf Inf 1.66 0.01 NaN

plot(traina2,ylab="Cash withdrawal at ATM1, 1000s", xlim=c(1,53))  
lines(testa2, col=4)  
lines(fcst1a2$mean, col=1)  
lines(fcst2a2$mean, col=2)  
lines(fcst3a2$mean, col=3)  
legend("topleft", lty=1, col=c(4,1,2,3),  
 legend=c("Test Data", "SNaive method","H-W Method","ARIMA"),pch=1,cex=.69)

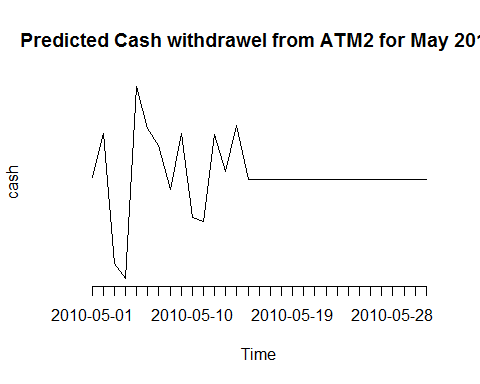


Refitting the ARIMA model to calculate for all data and create predictions for the next 30 days (May 2010) give the following model:

refit2 <- Arima(model=fit3a2, a2.ts)  
preda2 <- predict(refit2, n.ahead=31)  
  
  
plot(a2.ts,main="Cash withdrawal at ATM2 with Predictions, 1000s", xlim=c(50,57))  
lines(preda2$pred, col=4)  
legend("topleft", lty=1, col=c(4),  
 legend=c("predicted values"),pch=1,cex=.69)



#Just predictions  
d <- seq(atm2[363,1]+1,atm2[363,1]+31,by="days")  
p<- ts(preda2$pred,frequency=1)  
plot(p, axes=F, main="Predicted Cash withdrawel from ATM2 for May 2010", ylab="cash")  
axis(1,at=1:31, labels =d)



#Summary Stats  
summary(as.vector(p))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 34.05 57.75 57.75 58.57 58.94 80.03

dfPred$ATM2 <-preda2$pred

**Model for ATM3**

ATM 3 is a special case, in that there is not enough historical data to model the future needs for this ATM.  
There are two possible approaches, the first use the current mean for the data that is available to predict May 2010, the second is to use one of the other ATM models to forecast.  
The advantage of using the mean for the existing data is that it is simple and does not create any false accuracy, The disadvantage to doing this is that it does not take into account the weekly nature of the ATM withdrawals seen in other machines.

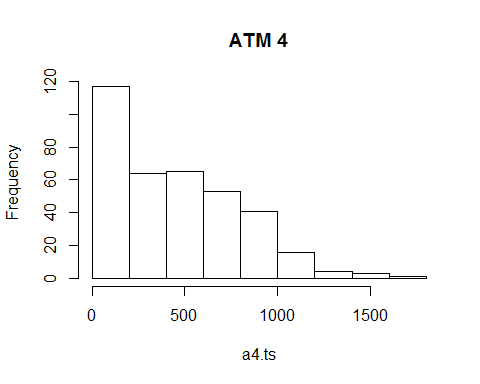
a3.ts <- window(a3.ts, start=363)  
t1<- as.vector(tail(a1.ts,3))  
t2<- as.vector(tail(a2.ts,3))  
t3<- as.vector(tail(a3.ts,3))  
# forcing the model to match ATM1  
  
  
  
dfPred$ATM3 <- preda1$pred

To create the forecast for this ATM machine, I compared the only 3 data points we have for this ATM with the other two similar ATMs, and used the ATM that most closely matched to forecast future values. These data points happen to exactly match the data for ATM1, so the same model is used to calculate forecaster values for May 2010.

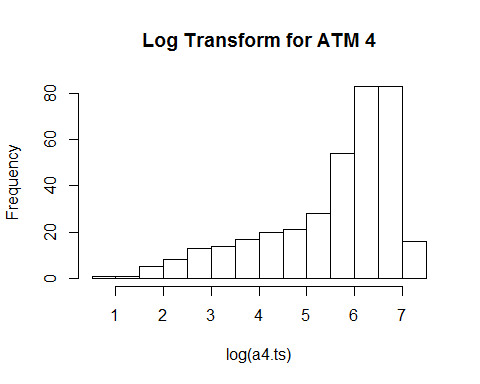
**Model for ATM4**

The data for ATM 4 presents very differently from that used for ATM 1 and 2. The standard deviation and mean are both much higher than the other forecasts, and the data is skewed. Because of the large standard deviation, a log transformation and a Box-Cox transformation with lambda = .15 were compared.

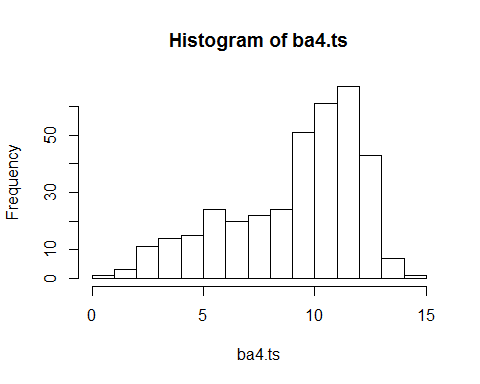
#ATM 4 transformed  
hist(a4.ts, main= "ATM 4")



hist(log(a4.ts),main="Log Transform for ATM 4")



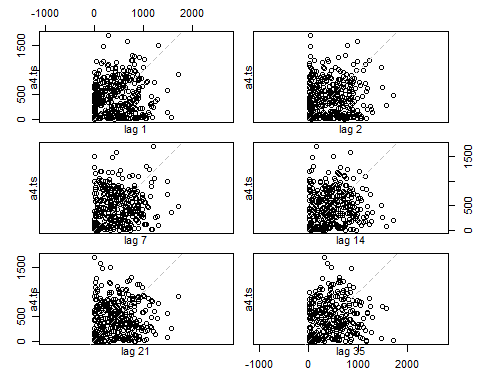
lambda <- BoxCox.lambda(a4.ts) # = 0.16  
  
ba4.ts <- BoxCox(a4.ts,lambda)  
hist(ba4.ts)



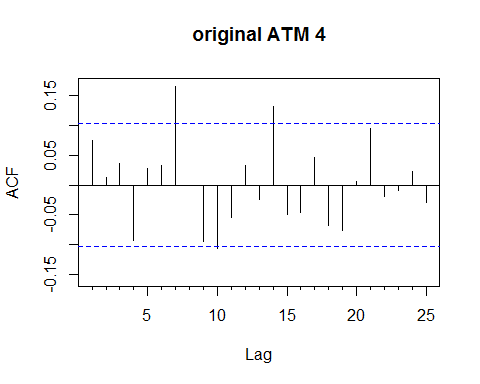
sd(ba4.ts)

## [1] 2.953783

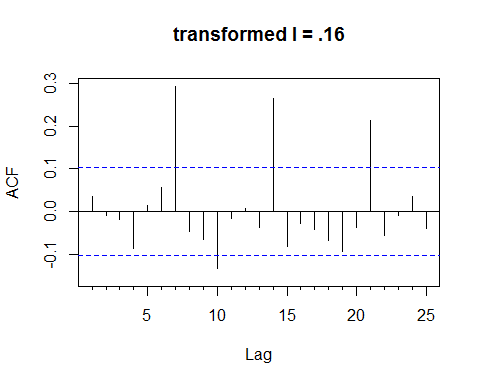
# checking ACF  
   
#Model for ATM1  
  
#testing - for weekly,biweekly lag  
lagdays <- c(1,2, 7, 14, 21,35)  
lag.plot(a4.ts, lags = 6, set.lags = lagdays, do.lines = FALSE)



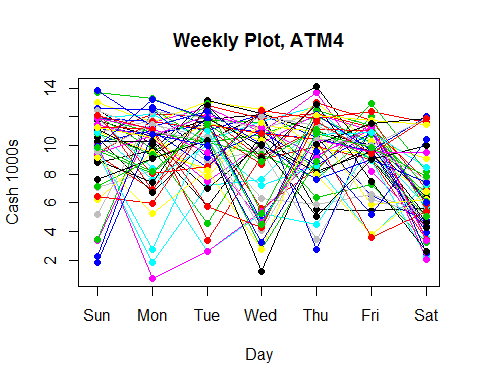
Acf(a4.ts, plot=T, main ="original ATM 4")



Acf(ba4.ts, plot=T,main="transformed l = .16")

 The weekly lag is more apparent when viewing the transformed data,and is still significant at 7, 15, and 21 days. There are 364 data points available in this data. As a points of reference, both the seasonal naive and the mean forecast methods were first used.

#seasonal adjustment  
ba4.ts <- ts(ba4.ts,frequency=7)  
traina4 <- window(ba4.ts,start = 1,end=42)  
testa4 <- window(ba4.ts, start=43)  
  
seasonplot(ba4.ts,ylab="Cash 1000s", xlab="Day", main="Weekly Plot, ATM4", col=1:20, pch=19)



#Models  
#Seasonal Naive  
fit1a4 <- snaive(traina4, h=70)  
fcst1a4 <-forecast(fit1a4, h=70)  
accuracy(fit1a4, testa4)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.00205610 3.504552 2.683945 -16.99765 43.13898 1.000000  
## Test set -0.07174008 4.190588 3.454700 -37.14497 67.78284 1.287172  
## ACF1 Theil's U  
## Training set 0.0273485 NA  
## Test set -0.3633359 0.6896384

#mean  
fit2a4 <- meanf(traina4, h=70)  
fcst2a4 <-forecast(fit2a4, h=70)  
accuracy(fit2a4, testa4)

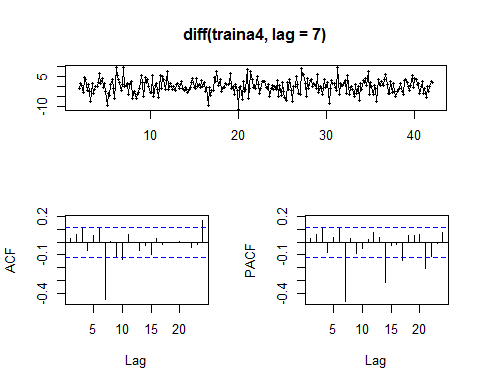
## ME RMSE MAE MPE MAPE MASE  
## Training set -7.031405e-16 2.941360 2.453103 -22.42095 43.41551 0.9139915  
## Test set -6.905972e-02 3.019377 2.372591 -31.03763 51.18513 0.8839939  
## ACF1 Theil's U  
## Training set 0.07301746 NA  
## Test set -0.11046928 0.3106543

Stl was used to decompose the data. The residuals of the model appear fairly uniform and the seasonal pattern is consistent.

fit3a4 <- stl(traina4, s.window = 15)  
fcst3a4 <- forecast(fit3a4, method = "naive", h=70)

AS a comparison to the other ATM machine models, the auto.arima algorithm was again used to select the best ARIMA model. The best model selected was ARIMA(0,0,0)(2,0,0)[7], which was quite similar to the one found by ATM1.

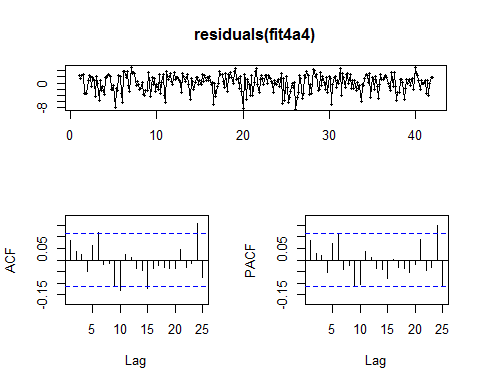
#ARIMA Model - machine selected  
#take the seasonal difference  
tsdisplay(diff(traina4,lag =7))



(fit4a4 <- auto.arima(traina4))

## Series: traina4   
## ARIMA(0,0,0)(2,0,0)[7] with non-zero mean   
##   
## Coefficients:  
## sar1 sar2 intercept  
## 0.2474 0.1543 9.1168  
## s.e. 0.0582 0.0586 0.2675  
##   
## sigma^2 estimated as 7.792: log likelihood=-703.28  
## AIC=1414.56 AICc=1414.7 BIC=1429.21

tsdisplay(residuals(fit4a4))



Box.test(residuals(fit3a1), lag=7, fitdf=4, type="Ljung")

##   
## Box-Ljung test  
##   
## data: residuals(fit3a1)  
## X-squared = 5.5802, df = 3, p-value = 0.1339

#f2<-auto.arima(traina1, stepwise=FALSE, approximation=FALSE)  
  
fcst4a4=forecast(fit4a4, h=70)

Comparing the accuracy of these four models again gives the ARIMA model as the most accurate, but its very close in accuracy measures to the mean model used in forecast2, which may indicate the simpler model may be the better one. This result makes sense when evaluating the graph, as the ARIMA model approaches the mean forecast value the further out the forecast runs.

round(accuracy(fcst1a4, testa4),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set 0.00 3.50 2.68 -17.00 43.14 1.00 0.03 NA  
## Test set -0.07 4.19 3.45 -37.14 67.78 1.29 -0.36 0.69

round(accuracy(fcst2a4, testa4),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set 0.00 2.94 2.45 -22.42 43.42 0.91 0.07 NA  
## Test set -0.07 3.02 2.37 -31.04 51.19 0.88 -0.11 0.31

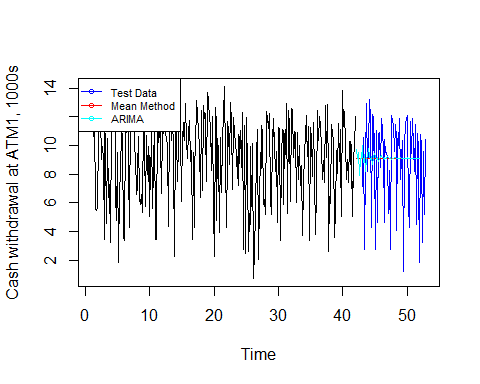
round(accuracy(fcst3a4, testa4),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set 0.00 3.18 2.46 -14.87 40.04 0.92 -0.47 NA  
## Test set -1.59 4.06 2.94 -57.06 68.81 1.09 0.08 0.37

round(accuracy(fcst4a4, testa4),2)

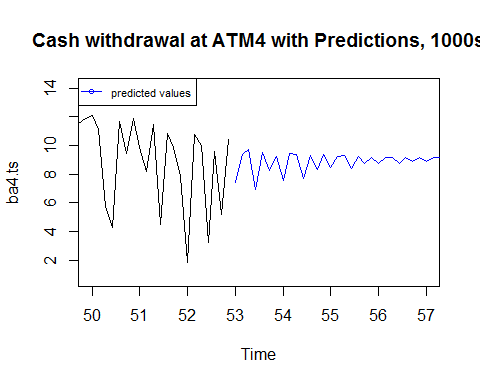
## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set 0.00 2.78 2.25 -20.57 39.95 0.84 0.08 NA  
## Test set -0.09 3.06 2.41 -31.54 51.84 0.90 -0.13 0.32

plot(traina4,ylab="Cash withdrawal at ATM1, 1000s", xlim=c(1,53))  
lines(testa4, col=4)  
#lines(fcst1a4$mean, col=1)  
lines(fcst2a4$mean, col=2)  
#lines(fcst3a4$mean, col=3)  
lines(fcst4a4$mean, col=5)  
legend("topleft", lty=1, col=c(4,2,5),  
 legend=c("Test Data", "Mean Method", "ARIMA"),pch=1,cex=.69)

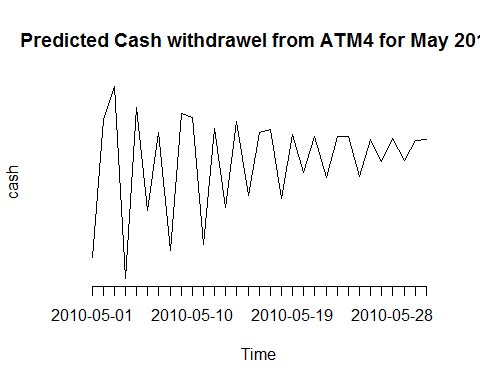


Creating a prediction for May 2010 for ATM 4 results in

refit <- Arima( model=fit4a4, ba4.ts)  
preda4<- predict(refit, n.ahead=31)  
  
plot(ba4.ts,main="Cash withdrawal at ATM4 with Predictions, 1000s", xlim=c(50,57))  
lines(preda4$pred, col=4)  
legend("topleft", lty=1, col=c(4),  
 legend=c("predicted values"),pch=1,cex=.69)



#Just predictions - Transformed  
d <- seq(atm4[364,1]+1,atm4[364,1]+31,by="days")  
  
p4<- ts(preda4$pred,frequency=1)  
p<-(lambda\*p4+1)^(1/lambda)  
plot(p, axes=F, main="Predicted Cash withdrawel from ATM4 for May 2010", ylab="cash")  
axis(1,at=1:31, labels =d)



#Summary Stats  
summary(as.vector(p))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 110.3 231.4 293.3 264.9 304.4 362.5

dfPred$ATM4 <- p

**Summary Forecast for May 2010 by ATM** Summary output file name: ATM Predictions May 2010 CB.csv These values can be used to forecast the cash withdrawal per ATM for the month of May, however they are less precise the further out the forecast is made. If the goal of the forecast is to ensure adequate cash available in the machines and a full month is required at a time, a better method may be to forecast using the upper bound of the weekly seasonal patterns to avoid under-forecasting.

###Summary and output to excel  
(summary(dfPred))

## Date ATM1 ATM2 ATM3   
## Min. :2010-05-01 Min. :20.69 Min. :34.05 Min. :20.69   
## 1st Qu.:2010-05-08 1st Qu.:80.26 1st Qu.:57.75 1st Qu.:80.26   
## Median :2010-05-16 Median :83.80 Median :57.75 Median :83.80   
## Mean :2010-05-16 Mean :78.69 Mean :58.57 Mean :78.69   
## 3rd Qu.:2010-05-23 3rd Qu.:89.90 3rd Qu.:58.94 3rd Qu.:89.90   
## Max. :2010-05-31 Max. :96.56 Max. :80.03 Max. :96.56   
## ATM4   
## Min. :110.3   
## 1st Qu.:231.4   
## Median :293.3   
## Mean :264.9   
## 3rd Qu.:304.4   
## Max. :362.5

write.csv(dfPred, "ATM Predictions May 2010 CB.csv")

## Part B

Using data provided for KWH usage for 1993 through 2013, model KWH usage for 2014.

**Outliers and Null values** The data set contains 192 monthly observations, with a minimum usage3 of 77K and max of 107K. There is a single 'NA' value and no blank values. The single null value was replaced with the mean value for the current year. The minimum value of 77K is an outlier, but does not appear to be an erroneous observation

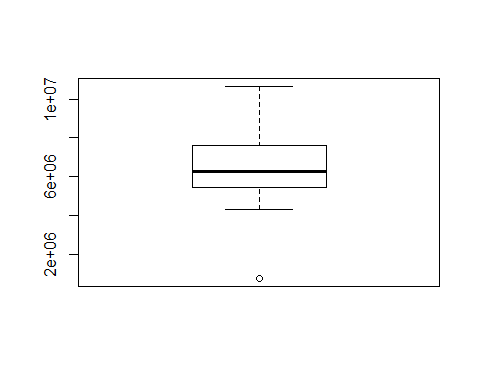
**Transformation**

The data has a large standard deviation and slightly skewed, but not dramatically so. It is appears to be approximately normally distributed and a transformation does not appear to appropriate.

#import data  
f <- "ResidentialCustomerForecastLoad-624.csv"  
cldata <- read.csv(file = f,stringsAsFactors= FALSE, strip.white = TRUE)  
head(cldata)

## CaseSequence YYYY.MMM KWH  
## 1 733 1998-Jan 6862583  
## 2 734 1998-Feb 5838198  
## 3 735 1998-Mar 5420658  
## 4 736 1998-Apr 5010364  
## 5 737 1998-May 4665377  
## 6 738 1998-Jun 6467147

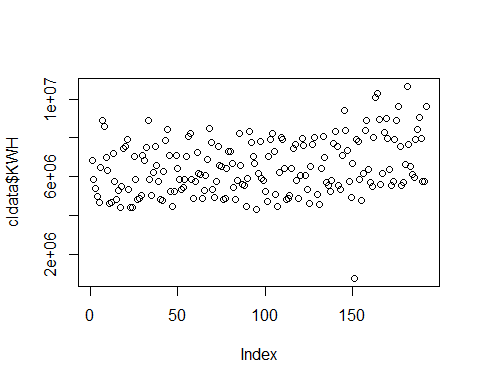
boxplot(cldata$KWH)



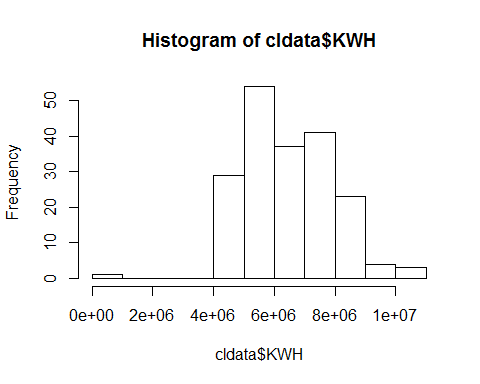
summary(cldata)

## CaseSequence YYYY.MMM KWH   
## Min. :733.0 Length:192 Min. : 770523   
## 1st Qu.:780.8 Class :character 1st Qu.: 5429912   
## Median :828.5 Mode :character Median : 6283324   
## Mean :828.5 Mean : 6502475   
## 3rd Qu.:876.2 3rd Qu.: 7620524   
## Max. :924.0 Max. :10655730   
## NA's :1

plot(cldata$KWH)



#check for NA  
clNA <- cldata[is.na(cldata$KWH),]  
#add year column  
cldata <- cldata%>%  
 mutate(YM= as.yearmon(as.Date(paste("01-",YYYY.MMM, sep = ""), format = "%d-%Y-%b")))%>%  
 mutate(Year =format(YM, "%Y"))  
  
m <- cldata%>%filter(!is.na(KWH))%>%group\_by(Year)%>%summarise(m<- mean(KWH))%>%filter(Year == "2008")  
#replace with mean for 2008  
cldata$KWH[is.na(cldata$KWH)] <- as.integer(m[1,2])  
  
hist(cldata$KWH)



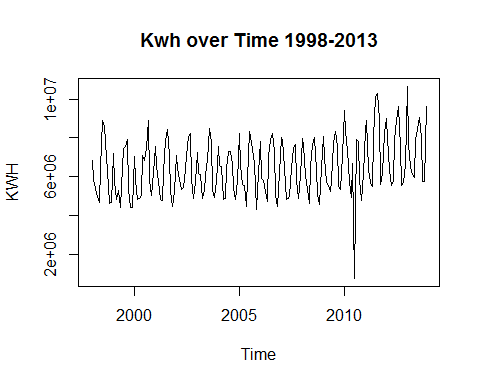
kurtosis(cldata$KWH)

## [1] 0.4729965  
## attr(,"method")  
## [1] "excess"

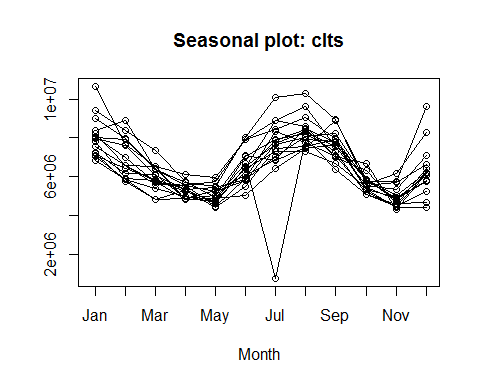
skewness(cldata$KWH)

## [1] 0.1703248  
## attr(,"method")  
## [1] "moment"

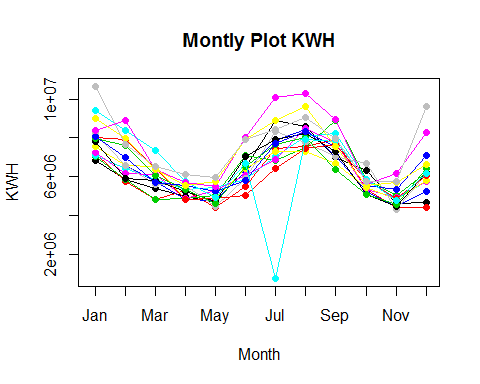
#transform to time series with frequency 12  
  
clts <- ts(cldata%>%select(KWH),start = c(1998,1), frequency = 12)  
plot(clts, main = "Kwh over Time 1998-2013")



seasonplot(clts)



seasonplot(clts,ylab="KWH", xlab="Month", main="Montly Plot KWH", col=1:20, pch=19)

 The seasonal plot shows an definitive seasonal trend, which makes intuitive sense as energy usage would fluctuate with the season.

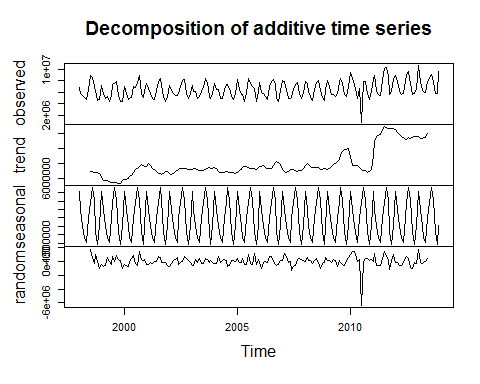
Evaluating the ACF and a decomposed view of the data shows there appears to be an overall increasing trend along with the seasonal competent, however the ACF plot shows has consistent pattern of lags, indicating the data be stationary when the seasonal trend is accounted for. The ACF test also indicates the data is stationary, with a very low p-value.

To create a model that accounts for this

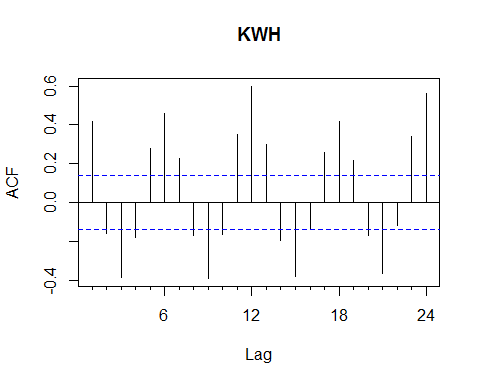
To model a full year of 2014 a few seasonally adjusted models may be appropriate. Three different models were selected to compare.  
A seasonal Naive model was used as a simple control model. A multiplicative STL decomposition was used as

#Eval Lags

plot(decompose(clts))



Acf(clts, plot=T)



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

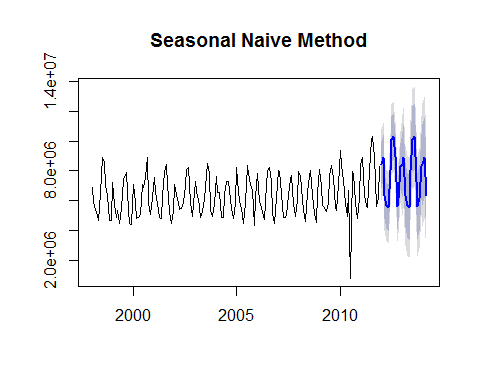
## Warning in adf.test(clts, alternative = "stationary"): p-value smaller than  
## printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: clts  
## Dickey-Fuller = -4.7775, Lag order = 5, p-value = 0.01  
## alternative hypothesis: stationary

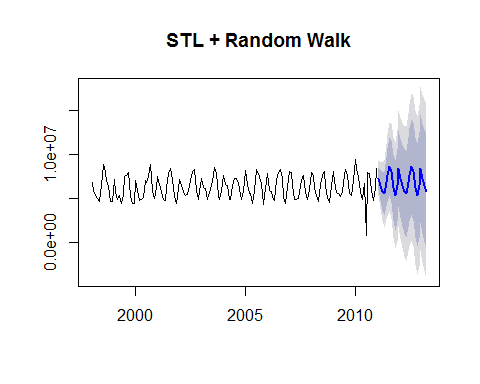
To find the appropriate model the data was broken into a training and test data sets, the training set accounting for 85% of the data from 1998-2011, and a test set the remaining data from 2012 - 2013 to evaluate each model's accuracy.

Four different models were selected to compare.  
- seasonal Naive model was used as a simple control model. - STL decomposition was used with random walk was used.  
- Holt-Winter seasonal additive method was used.  
Additive methods were used because the seasonal trend does not appear to be changing over time. - ARIMA model auto-selected

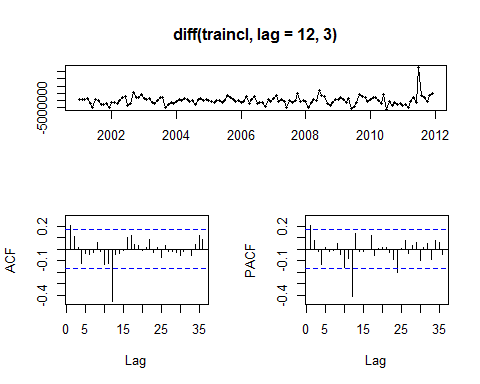
#train and test forecasts 12 years train, 4years test, using full   
  
traincl <- window(clts, start = c(1998,1), end = c(2011,12))  
testcl <- window(clts, start = c(2012, 1), end = c(2013,12))  
  
  
#Model1: Seasonal Naive  
fit1cl <- snaive(traincl, h=27)  
fcst1 <- forecast(fit1cl, h = 27)  
plot(fcst1, main = "Seasonal Naive Method")



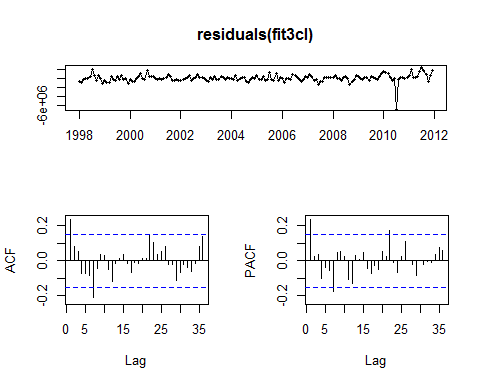
#reformat data for stl   
traincl2 <- ts(clts, start = 1998, end = 2011, frequency=12)  
testcl2 <- ts(clts,start = 2012, end = 2014, frequency=12)  
fit2cl <- stl(traincl2, s.window ="periodic")  
fcst2 <- forecast(fit2cl, method = "naive", h = 27)  
plot(fcst2, main = "STL + Random Walk")



#h-w additive  
fit3cl <- hw(traincl,seasonal="additive", h=27)  
fcst3 <- forecast(fit3cl, h=27)  
  
  
#Auto ARIMA modle  
  
tsdisplay(diff(traincl,lag =12,3))



fit4cl <- auto.arima(traincl)  
tsdisplay(residuals(fit3cl))



Box.test(residuals(fit3a1), lag=12, fitdf=9, type="Ljung")

##   
## Box-Ljung test  
##   
## data: residuals(fit3a1)  
## X-squared = 11.2617, df = 3, p-value = 0.01039

fcst4 <- forecast(fit4cl, h=27)  
  
  
  
# Summary of the models to compare accuracy  
  
 round(accuracy(fcst1, testcl),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set 114098.3 1213668.2 711294.9 -4.41 15.73 1.00 0.25 NA  
## Test set -272199.3 965567.1 742063.5 -3.81 9.30 1.04 -0.20 0.53

round(accuracy(fcst2, testcl2),2)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 9821.56 973319.2 586537.2 -5.34 14.22 0.94 -0.43  
## Test set -824803.22 1131270.6 998035.6 -15.45 17.44 1.59 0.40  
## Theil's U  
## Training set NA  
## Test set 0.83

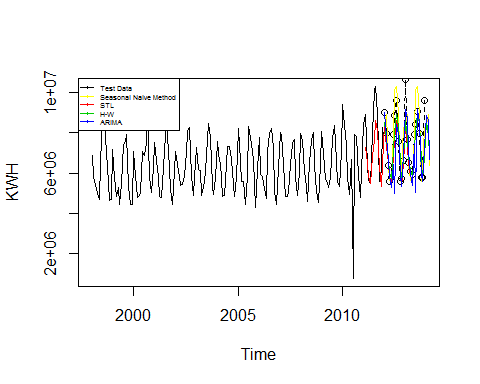
round(accuracy(fcst3, testcl),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set 23295.64 829174.1 512448.7 -5.32 12.50 0.72 0.24 NA  
## Test set 210136.47 871429.2 584496.3 1.41 7.34 0.82 0.10 0.54

round(accuracy(fcst4, testcl),2)

## ME RMSE MAE MPE MAPE MASE ACF1 Theil's U  
## Training set -10777.58 788532.3 479536.5 -5.33 11.80 0.67 0.01 NA  
## Test set 513921.29 1313888.0 842344.2 5.66 10.43 1.18 0.17 0.72

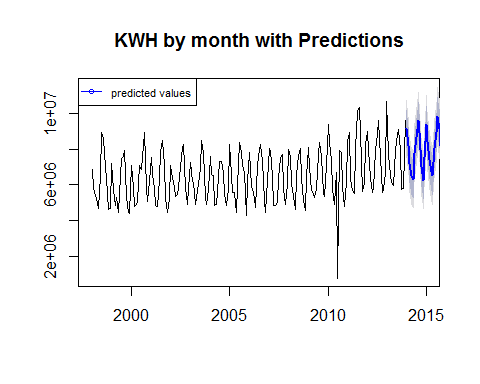
plot(traincl,ylab="KWH", xlim = c(1998,2014))  
lines(testcl, col=1, type="o", lty=2)  
lines(fcst1$mean, col=7)  
lines(fcst2$mean, col=2)  
lines(fcst3$mean, col=3)  
lines(fcst4$mean, col=4)  
legend("topleft", lty=1, col=c(1,7,2,3,4),  
 legend=c("Test Data", "Seasonal Naive Method", "STL", "H-W", "ARIMA"),pch=1,cex=.45)



Based on both the accuracy measures and the clarity of the model, the Holt-Winters additive method creates the best model.

Using this model to forecast 2014 gives

#refitting with all data  
fit3cl <- hw(clts,seasonal="additive")  
  
predictcl<- predict(fit3cl, n.ahead=12)  
  
  
plot(predictcl,main="KWH by month with Predictions", xlim=c(1998,2015))  
lines(predictcl$mean, col=4)  
legend("topleft", lty=1, col=c(4),  
 legend=c("predicted values"),pch=1,cex=.69)



#Just predictions data  
d <- as.yearmon(seq(as.Date("2014/1/1"), as.Date("2016/1/1"), by="mon"))[1:24]  
dfPredictCL <- data.frame(Date = d)  
dfPredictCL$Prediction <- as.vector(predictcl$mean)  
dfPredictCL$lower <- as.vector(predictcl$lower[,1])  
dfPredictCL$upper <-as.vector(predictcl$upper[,1])  
  
summary(dfPredictCL)

## Date Prediction lower upper   
## Min. :2014 Min. :6258152 Min. :5178312 Min. : 7337991   
## 1st Qu.:2014 1st Qu.:6937516 1st Qu.:5856074 1st Qu.: 8018958   
## Median :2015 Median :7768118 Median :6673148 Median : 8863087   
## Mean :2015 Mean :7872353 Mean :6786172 Mean : 8958534   
## 3rd Qu.:2015 3rd Qu.:8907589 3rd Qu.:7811574 3rd Qu.:10002168   
## Max. :2016 Max. :9806588 Max. :8700212 Max. :10912964

#export to location

\*\*Summary Output Summary output file name: KWH Predictions 2014 CB.csv These values can be used to forecast the KWH by month for 2014. A lower and upper interval at 80% confidence is provided. Depending on the purpose of the forecast, these bounds can create more conservative estimates. If the goal is to make sure there is enough energy available, and it it not easily adjusted when needed, then using the upper bound would provide a better forecast. If the goal is to review revenue based on KWH usage, using a lower level is more conservative as it will typically under-forecast the expected usage.

###Summary and output to excel  
  
write.csv(dfPredictCL, "KWH Predictions 2014 CB.csv")