Time Series Assignment

G. Holt Williams

6/15/2020

# Time Series Assignment

# Introduction

This assignment is designed to give an understanding of time series analysis through decomposing seasonal time series data, subtracting seasonality from data, and exploring the issue of correlation between successive values in a time series (lag values). The project is broken into two sections, first looking at decomposing time series data and removing those effects from the data for analysis, followed by the use of the autoARIMA function to explore forecasting of the time series into future time periods.

## Load Libraries and Dataset

library(tidyverse)  
library(psych)  
library(ggthemes)  
library(gridExtra)  
library(ggfortify)  
library(forecast)  
library(tsdl)  
library(zoo)

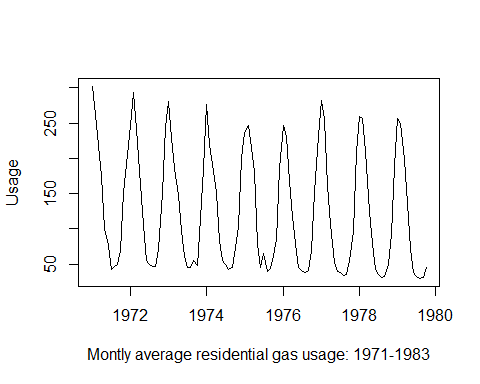
The dataset chosen for this assignment is from the “tsdl” libraries utilities dataset, of which the “Monthly average residential gas usage in Iowa” from January 1971 to August 1983 was chosen (Hyndman 2018).

Utilities <- subset(tsdl,"Utilities") #  
Utilities[1]

## Time Series Data Library: 1 Utilities time series with frequency 12   
##   
## Frequency  
## Subject 12  
## Utilities 1

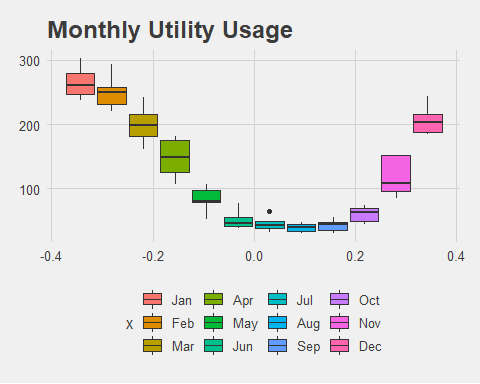
df<-Utilities[[1]]

plot(df, ylab="Usage", xlab="Montly average residential gas usage: 1971-1983")



The plot clearly shows the seasonality with a similar shaped graph for each year, which is explored further below.

y<-as.data.frame(df)  
x<-as.yearmon(time(df))  
x<-format(x,"%b")  
df2<-cbind(x,y)  
colnames(df2)<-c("x","y")  
df2$x<-factor(df2$x, levels=c("Jan","Feb","Mar",  
 "Apr","May","Jun",  
 "Jul","Aug","Sep",  
 "Oct","Nov","Dec"))  
df2%>%group\_by(x)%>%ggplot(aes(y,fill=x))+geom\_boxplot()+coord\_flip()+theme\_fivethirtyeight()+ggtitle("Monthly Utility Usage")



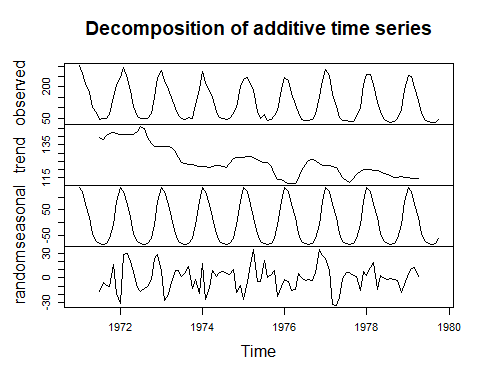
Above is the seasonality trend visualized by moth. Each boxplot corresponds to the range of values for that month over the 1971 to 1983 period. As expected, summer months are very low while winter months have the highest values as consumers are heating their homes. The summer levels are not zero for several possible reasons, one may be that consumers utilize the gas for cooking.

# Time Series Decomposition

## Decomposition and Insights

Here the data set is decomposed into its three components: trend, seasonal, and randomness, which are plotted below.

df\_decomp<-decompose(df)  
plot(df\_decomp)

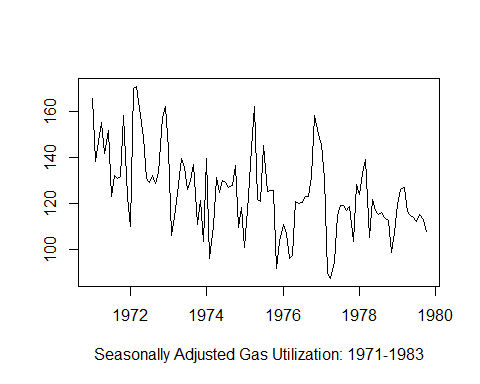


The first plot above is the original dataset, followed by the estimated trend, estimated seasonal component, and estimated randomness or irregular component. Looking at the trend component, one can see the overall decrease in gas usage across the time period.. The seasonal trend looks to have the smallest usage in the summer, with the largest usage in the winter months as would be expected.

## Seasonally Adjust

To seasonally adjust the dataset, the seasonality information gained from the decomposition is subtracted from the original dataset.

df\_seasadj<- df-df\_decomp$seasonal  
plot(df\_seasadj, ylab="",xlab="Seasonally Adjusted Gas Utilization: 1971-1983")



What remains is the trend and randomness of the dataset. Looking at the previous decomposition plot, one can see the overall plot here follows a combination of the trend and randomness plots above.

# ARIMA Modelling

## Differencing Time Series

To properly utilize an ARIMA model it is important to use a stationary time series. This means the mean of the data is constant over time, the variance does not increase over time, and that seasonality effects are small. (SOURCE) One way to accomplish this is with differencing as shown below. First, it is important to determine how many differences are required which can be accomplished with the nsdiffs and ndiffs functions. nsdiffs determines seasonal differencing while ndiffs estimates the number of first differences, which in this case is zero.

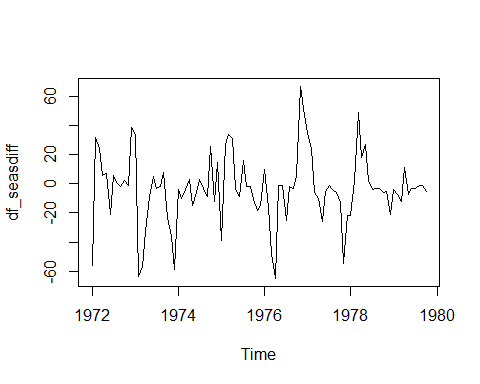
ndiffs(df)

## [1] 0

nsdiffs(df)

## [1] 1

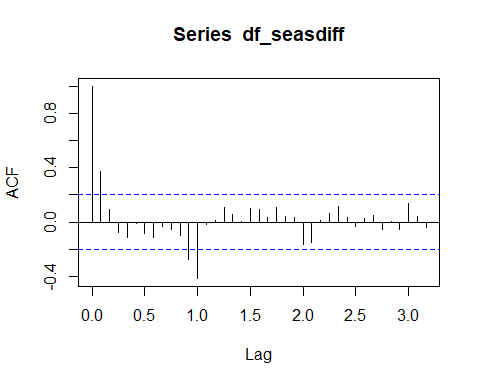
df\_seasdiff <- diff(df, lag=frequency(df), differences=1) ## Lag = 12 to use last years value  
plot(df\_seasdiff)



## ARIMA Modelling and Correlation Exploration

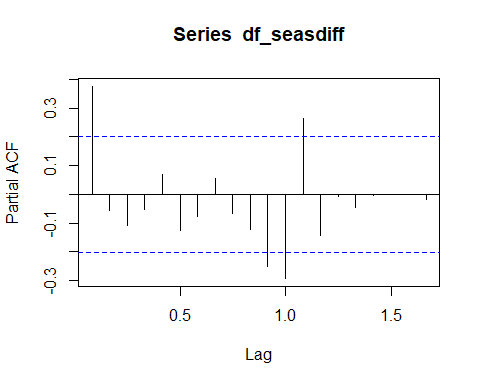
After processing the data above, the data is mostly stationary and ARIMA modeling can be applied. To visualize the autocorrelations of the different lag values, the acf function is used.

acf(df\_seasdiff,lag.max=38)



The above graph shows how there is autocorrelation with the previous period, and also the 12th period, which corresponds to the previous year’s value which is to be expected. Larger seasonal changes due to global warming or other casuses are unlikely to show up in this small dataset, and as such the above results are expected.

pacf(df\_seasdiff,lag.max=20)



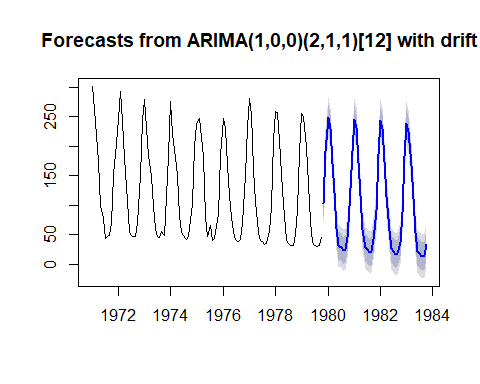
The pacf shows the partial correlogram which shows a similar pattern. Through analyzing the above information one can determine the type of ARIMA model to utilize. An alternative method is to use the auto.arima function which runs multiple arima models and then outputs the optimal model for the user based on the resulting AIC, AICc, or BIC values.

df\_autoArima<-auto.arima(df)  
df\_autoArima

## Series: df   
## ARIMA(1,0,0)(2,1,1)[12] with drift   
##   
## Coefficients:  
## ar1 sar1 sar2 sma1 drift  
## 0.4204 -0.1900 -0.2513 -0.6711 -0.2985  
## s.e. 0.0964 0.2431 0.1931 0.3289 0.0829  
##   
## sigma^2 estimated as 267.7: log likelihood=-400.72  
## AIC=813.44 AICc=814.41 BIC=828.7

The auto ARIMA function suggests utilizing a (1,0,0)(2,1,1) model and has the above outputs. Next, the ARIMA model is used to predict the next 48 periods, or four years of usage.

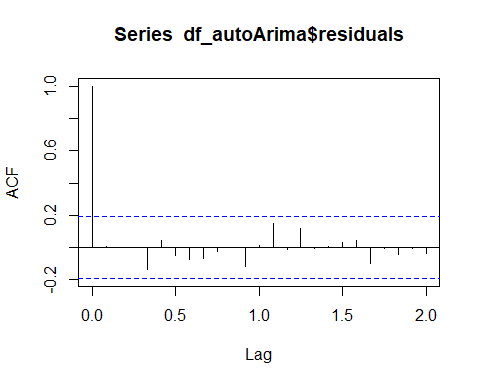
plot(forecast(df\_autoArima,h=48))



The above plots show how the model is predicting out over the next four years. The same shape of the data remains and there seems to be a slight decrease in usage over the predicted period. This corresponds to the early decomposition which showed the overall trend was showing lower utilization.

Next, the acf function is used to see if there are correlations with the predicted forecast errors. Also, a Ljung-Box test is performed.

acf(df\_autoArima$residuals, lag.max=24)

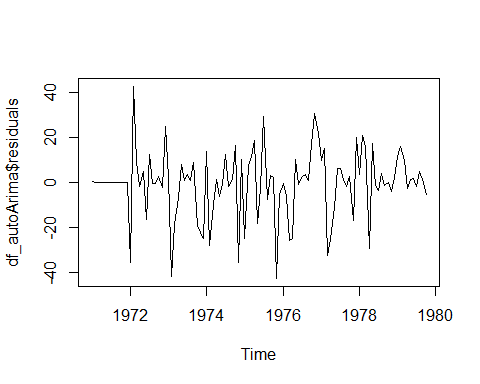


Box.test(df\_autoArima$residuals, lag=24, type="Ljung-Box")

##   
## Box-Ljung test  
##   
## data: df\_autoArima$residuals  
## X-squared = 11.8, df = 24, p-value = 0.9821

The resulting correlogram along with the p-value of the Ljung-Box test, leads one to not reject the null hypothesis that the autocorrelations of the forecast errors are zero.

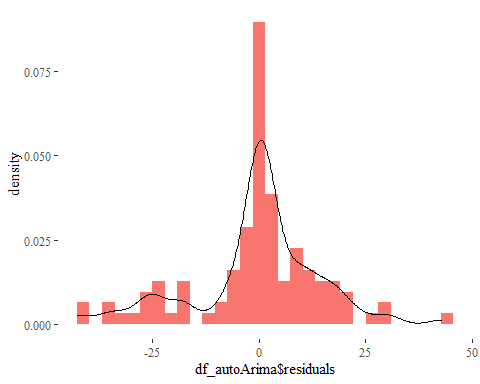
plot(df\_autoArima$residuals)



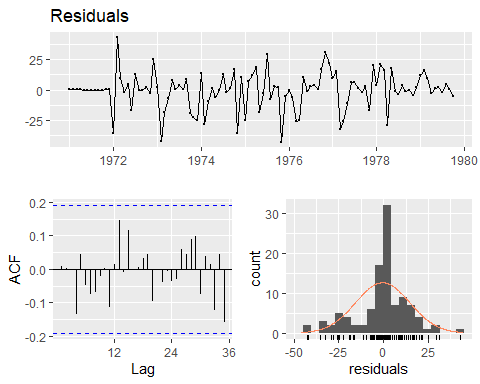
mean(df\_autoArima$residuals)

## [1] -0.5093544

df\_autoArima%>%  
 ggplot(aes(df\_autoArima$residuals))+  
 geom\_histogram(aes(y=..density.., fill='red'))+  
 geom\_density(aes())+  
 theme\_tufte()+theme(legend.position = "none")



checkresiduals(df\_autoArima$residuals)



The Residuals graph shows the residuals plotted along the time period and shows that the variance of the forecast errors is mostly equal over time, with a few spikes during different years. The Count chart with the overlaid histogram and density plot, show that the residuals are roughly normally distributed, but does have a few outliers which may correspond to the spikes in the previous chart. Those may be events of warmer or exceptionally cold winters that can occur but are not as intense as average years.

# References

Rob Hyndman and Yangzhuoran Yang (2018). tsdl: Time Series Data Library. v0.1.0. <https://pkg.yangzhuoranyang.com/tsdl/>