Time Series Modeling

#### Joshua M. Meli

#### Student ID: 000893883

# Part I: Research Question

Are we able to predict future revenue based on past revenue over time?

The goal in answering this data is to better prepare the organization for future profitability and benchmarking. Since a stated strategy of the organization is to increase revenue by increasing customer retention rates, the company can measure their success by using historical revenue related time series data to forecast future revenue values, including accounting for fluctuations due to seasonality or cyclic events.

# Part II: Method Justification & Assumptions

This research question will be addressed using time series modeling. Time series modeling seeks to understand data collected over elapsed time. An assumption of time series modeling is that the data exhibits stationarity. Stationarity means that the time series data reverts to a constant mean, does not increase in variance over time, and is without periodic fluctuations. Periodic fluctuations may be due to seasonality such as higher commerce during holiday seasons or home sales in the summer months. Stable variance over time is known as homoscedasticity, meaning the measurable distance between observations does not notably increase or decrease over time. Finally, a data series which reverts to a constant mean is one which does not show any trend.

R will be used for this analysis. R is open source software that was specifically made for statistical analysis. Using R, we can ingest the data set, and leveraging an extensive library of data manipulation packages, perform the market basket analysis steps. More information can be found on the R project website (<https://www.r-project.org/>).

The dplyr package will be used for data preparation and manipulation within R. More information for this package can be found on the tidyverse website (<https://tidyverse.org/>).

Model construction will be done using the arima function in the stats package. More information on the stats package can be found on the R documentation website (<https://www.rdocumentation.org/packages/stats/versions/3.6.2>).

Forecasting will be done by leveraging the forecast package. More information on the forecast package can be found on the CRAN R Project website (<https://cran.r-project.org/>).

# Part III: Data Preparation

First, the data set provided must be prepared for time series analysis. The provided data set is inspected for NA values, of which none are found. Next, the data set is inspected for duplicated values, of which none are found. Finally, the data set is inspected for gaps in observations, of which none are found. The data set provided requires no data cleaning for time series analysis. However, it does require conversion into a time series data type. For this, the ts function is applied and the frequency set to 365 as the data is daily data.

Next, the time series data is inspected by plotting a line graph. Visual confirmation is done to ensure no gaps exist in the time series. The visualization confirms an upward trend in the data with no mean reversion, thereby confirming the data the absence of stationarity. Having now visually inspected the data, the cleaned version of the time series to be used in analysis is written to a separate file.

Finally, the data is split into training and testing sets. The first 80% of the data is used for training the time series model while the final 20% of the data is used for testing the time series model.

#Prep  
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

df<-read.csv("c:/users/shua/documents/Advanced Data Analytics\_D213/teleco\_time\_series.csv", stringsAsFactors = F, header = T)  
  
sapply(df, function(x) sum(is.na(x)))

## Day Revenue   
## 0 0

which(duplicated(df))

## integer(0)

which(duplicated(df$Revenue))

## integer(0)

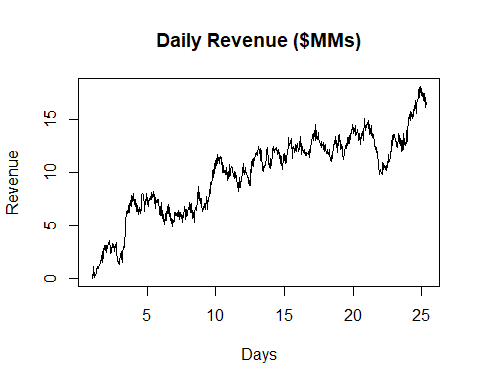
max(df$Day)==length(df$Day)

## [1] TRUE

#Convert to TS  
df.ts<-ts(df[,2], frequency=30)  
is.ts(df.ts)

## [1] TRUE

#Inspect  
plot(df.ts, xlab="Days", ylab="Revenue", main="Daily Revenue ($MMs)")



#Write  
write.csv(df.ts, "c:/users/shua/documents/Advanced Data Analytics\_D213/cleaned\_time\_series.csv", col.names = F, row.names = F)

## Warning in write.csv(df.ts, "c:/users/shua/documents/Advanced Data  
## Analytics\_D213/cleaned\_time\_series.csv", : attempt to set 'col.names' ignored

trainLength<-round(length(df.ts)\*.8)  
testLength<-length(df.ts) - trainLength  
df.ts.train<-df.ts[1:trainLength]  
df.ts.test<-df.ts[trainLength+1:length(df.ts)]

# Part IV Analysis

The first step taken in the analysis portion is to use the decompose function from the stats package on the time series data. This separates components of the time series data for easier identification. First, decomposing the time series data shows a strong upward trend. Second, seasonality appears to be observed when the data is decomposed. This requires further investigated using an auto-correlation function.

The acf auto-correlation function will show the degree to which observations in the time series are related to the preceding observations. Applying the auto-correlation function to the time series does not reveal a typical seasonal pattern or scalloped pattern associated with cyclic events. Instead, it shows a slow decrease in correlation over time indicating the trend of the data. In this case, the highest correlation is on the first lag, indicating that each time series observation is most correlated with the immediately preceding observation. To finalize the periodicity check, spectral density analysis is applied using the spec.ar function. This function converts the time series data from a time series domain to a frequency domain to better identify periodicities. The frequency would be the number of observations before a seasonal or periodic pattern repeats. When applying spectral density analysis by way of this function, lack of seasonality is again noted.

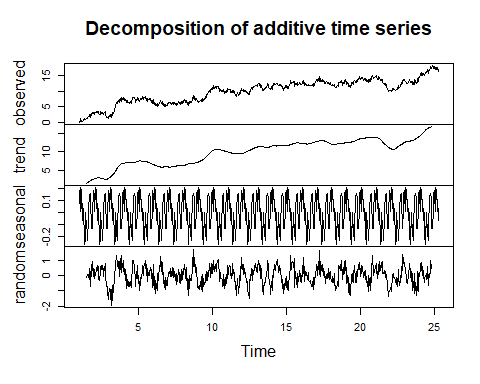
Next, the residuals of the time series are inspected by differencing the observations. R’s base diff function will difference the time series by returning the increase or decrease between observations. The assumption that the differenced values will have a mean of zero is confirmed. By decomposing the differenced time series the lack of trend in the residuals is also confirmed.

Next, the time series data must be fitted to an ARIMA model to be used in forecasting. Since the acf function showed the highest correlation for an observation was the immediately preceding observation followed by a tail off in correlation the p value for the ARIMA model order will be 1 to represent an auto regressive model, likely with a degree of differencing of 1 as well. However, multiple parameter values will be tested in order to select the model with the lowest AIC as a lower AIC indicates a better performing model. Fortunately the forecast package in R provides an auto arima function which not only selects the best fitting arima model for the time series but also will auto-set any seasonality parameters as well. More information on the forecast package in R can be found on the CRAN R Project website (<https://cran.r-project.org/>). The auto arima function confirmed an ARIMA model of 1,1,0 as the best fit having an AIC of 773.89.

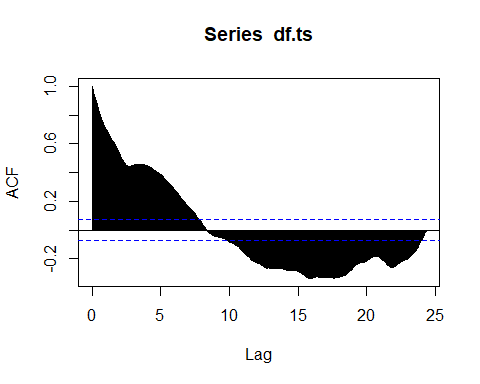
Finally, diagnostics are performed on the model. When plotting the model residuals it is evident that the residuals are white noise. The residuals plot does not show a trend or changing variance. Neither does it demonstrate periodicity. Additionally, by placing the residuals in a QQ plot it is evident that the residuals follow the diagonal line indicating normal distribution. When performing a Box Ljung test at various lag intervals, the p value of the residuals is consistently much greater than the .05 threshold confirming that the model is a good fit.

Forecasting can be accomplished using the forecast function with the arima model as an input. In this case, the model is forecast out to the number of testing observations. When plotting the forecast series we can view the forecast values along with a range of potential spread.

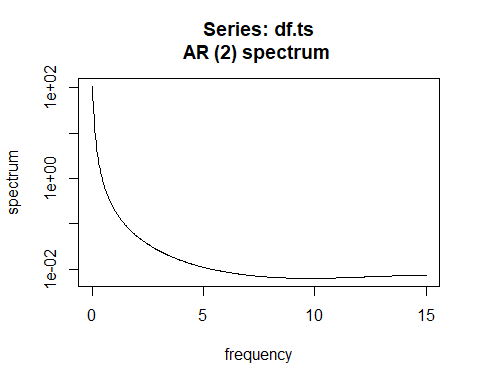
decomp<-decompose(df.ts)  
plot(decomp)



acf(df.ts, lag.max = 731)



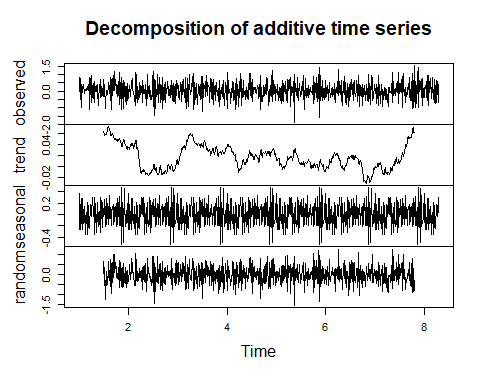
spec.ar(df.ts, log="yes")



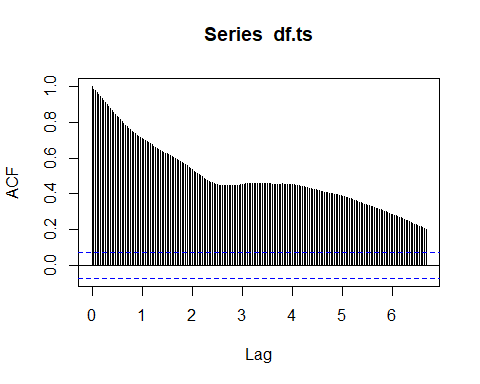
diff.df.ts<-ts(diff(df.ts), frequency = 100)  
round(mean(diff.df.ts))

## [1] 0

decomp.dif<-decompose(diff.df.ts)  
plot(decomp.dif)



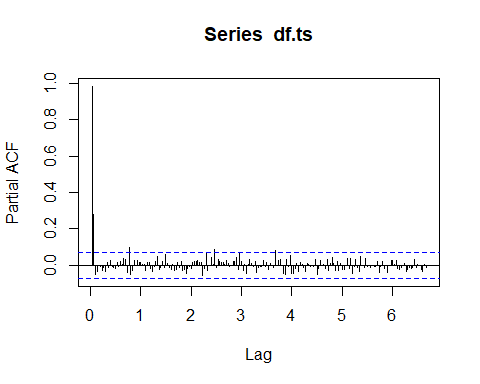
acf(df.ts, lag.max = 200)



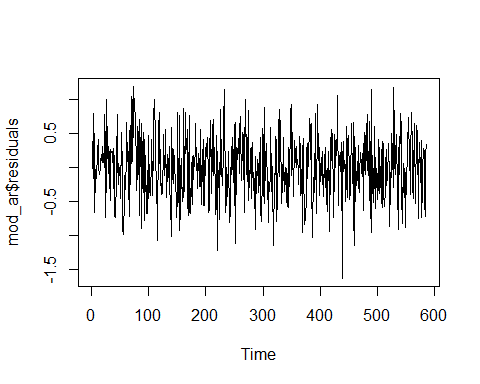
pacf(df.ts, lag.max=200)  
  
library(forecast)

## Warning: package 'forecast' was built under R version 4.2.1

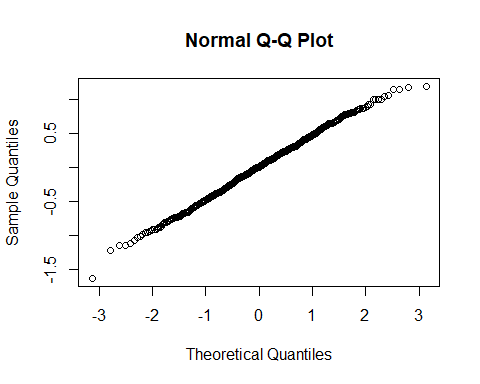
## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo



mod\_ar<-auto.arima(df.ts.train, D=1)  
  
  
plot.ts(mod\_ar$residuals)



qqnorm(mod\_ar$residuals)



Box.test(mod\_ar$residuals, lag=1, type="Ljung-Box")

##   
## Box-Ljung test  
##   
## data: mod\_ar$residuals  
## X-squared = 0.0031091, df = 1, p-value = 0.9555

Box.test(mod\_ar$residuals, lag=5, type="Ljung-Box")

##   
## Box-Ljung test  
##   
## data: mod\_ar$residuals  
## X-squared = 1.9825, df = 5, p-value = 0.8516

Box.test(mod\_ar$residuals, lag=10, type="Ljung-Box")

##   
## Box-Ljung test  
##   
## data: mod\_ar$residuals  
## X-squared = 2.4189, df = 10, p-value = 0.992

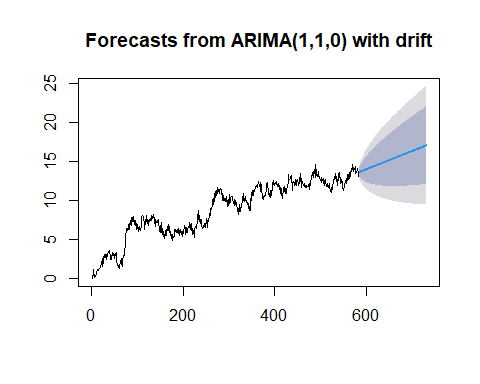
Box.test(mod\_ar$residuals, lag=20, type="Ljung-Box")

##   
## Box-Ljung test  
##   
## data: mod\_ar$residuals  
## X-squared = 13.242, df = 20, p-value = 0.8667

Box.test(mod\_ar$residuals, lag=40, type="Ljung-Box")

##   
## Box-Ljung test  
##   
## data: mod\_ar$residuals  
## X-squared = 36.501, df = 40, p-value = 0.6285

plot(forecast(mod\_ar, h=testLength))



# Part V: Summary

In this analysis an arima 1,1,0 model was selected for time series modeling based on the auto arima function recommendation. Having a time series with an interval of 1 (daily observations), the prediction length was determined based on an 80/20 train/test split of the time series observations. The model selection was evaluated automatically based on the model having the lowest AIC value. Based on this model selection and forecasting capabilities, it is recommended for the organization to implement use of this model to gauge future revenue expectations and benchmark performance against projections. In this way the organization can determine if their efforts to increase revenue by decreasing churn are being realized.