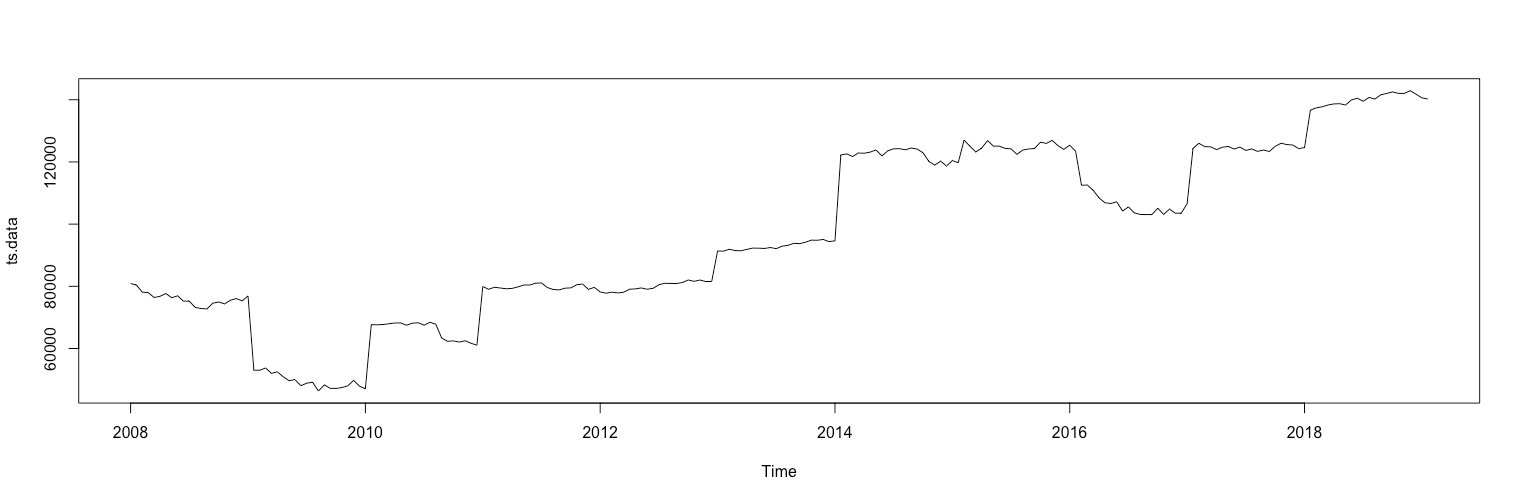
The January Effect in the Last Decade

The January Effect was first proposed by Sidney B. Wachtel in 1942 using data from 1897 to 1914 and later by Michael Rozeff and William R. Kinney Jr. in 1976. They hypothesised that January is the best time of the year to purchase stocks. They believed that the reason for this was that investors had more money from their end of the year Christmas bonus, which they want to use to invest. This is also because their data showed that in December, stocks were sold off for tax purposes. Third, there was a psychological aspect involved. The idea of starting the new year out by being smarter with your money and buying stocks could also have been seen as a possible cause. The data they examined was the stocks in the New York Stock Exchange from 1904 until 1974. The study concluded that the average return for stocks in the month of January is five times that of any other month. They also noted that the effect was more prominent in smaller stocks.

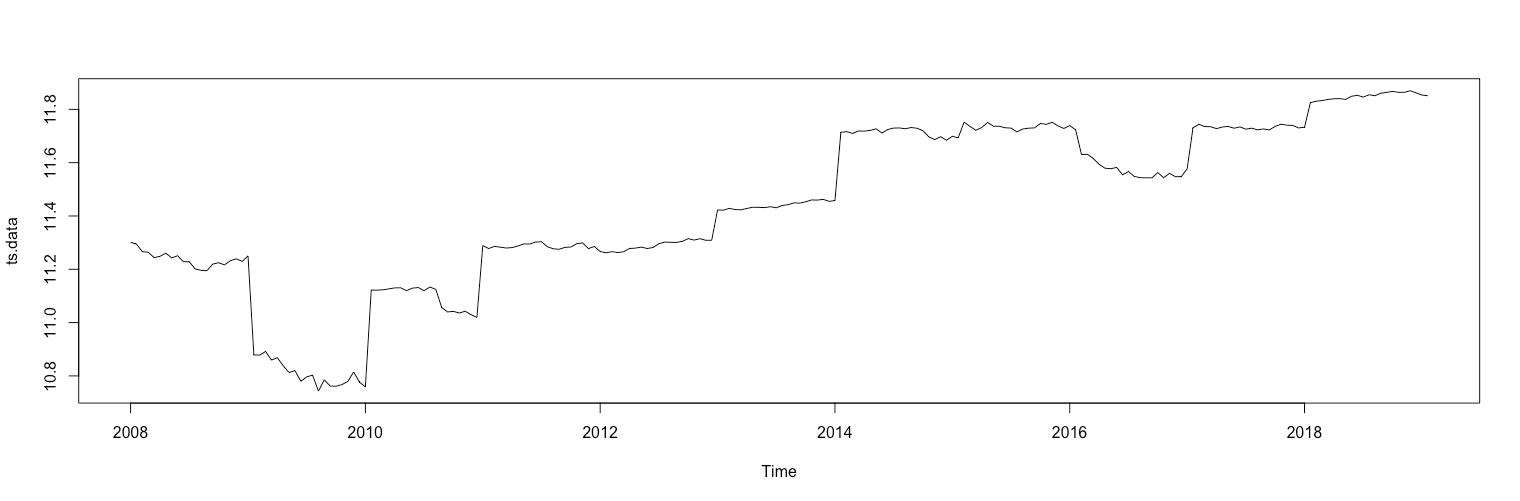
However, studies are now showing that the January Effect is no longer present in the market. Data has shown that in the past 11 years January has only performed positively 5 times. Reasons that experts consider for this shift that, now more well known, it is accounted for in trading algorithms. Other studies conclude that the January Effect is still present, however, the effect has become so diminished that its effect is minimal.

The data is the daily closing price of stocks throughout the month of January for the past 10 years. For each day, we are using the sum of the total closing amount for each of the stocks. We are using 3,000 stocks whose data were made available from Quandl.com.

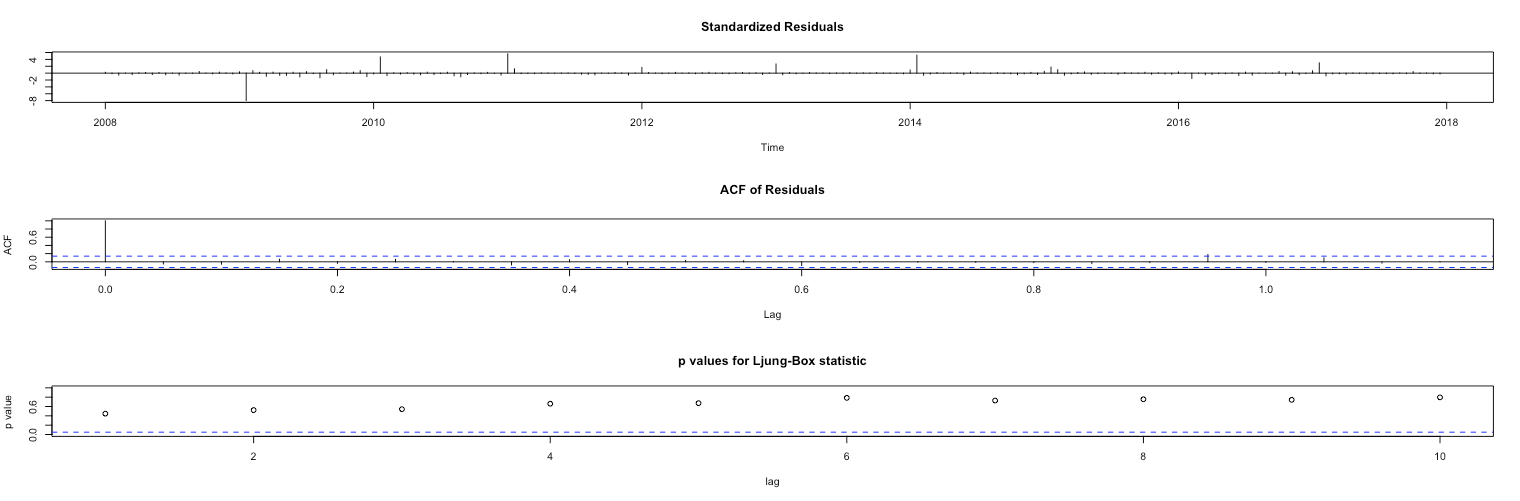
We first have to convert the raw data into a time series. We can observe that it looks like there may be some seasonality. One thing we do notice is that the closing prices from year to year a rarely similar.



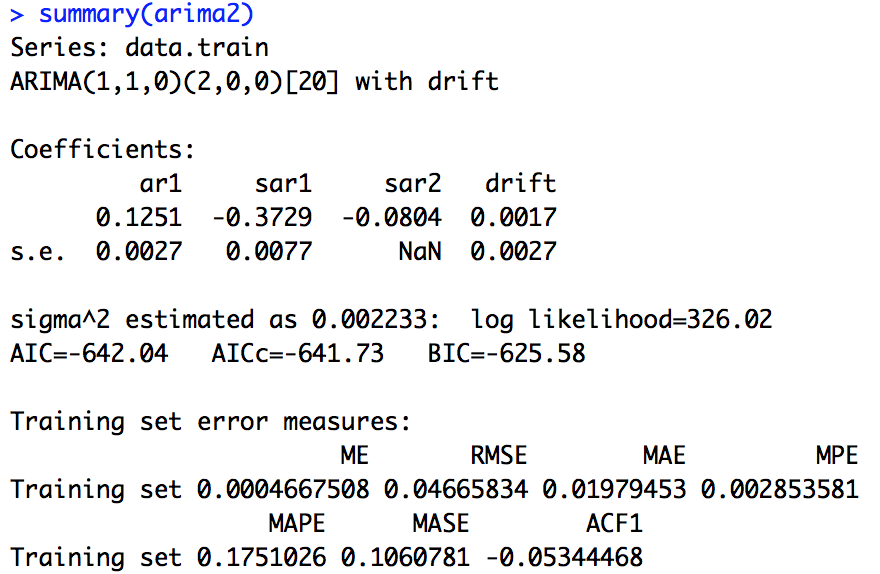
After some initial analysis on the data, a log transformation was performed as it gave more favorable results and lower MAPE values for our potential models.



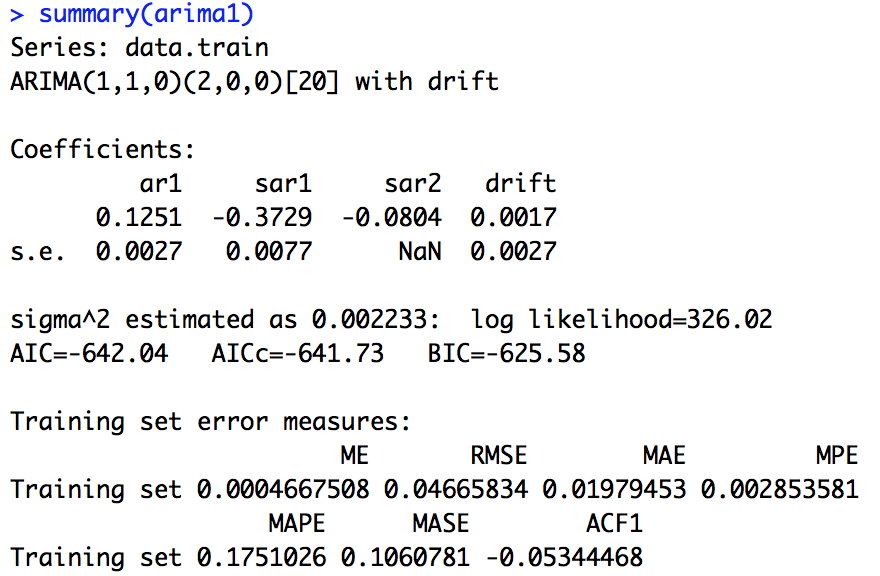
The data has been trained on daily stock data over the month of January from 2008 until 2017. The training set was then tested on data from January 2018. The first test is to see if we should use the Akaike’s Information Criterion (AIC) or the Bayesian Information Criterion (BIC) as a gauge to test our potential models. For both models a ‘kpss’ test is used. Both the AIC and BIC models call for an ARIMA (1,1,0)(2,0,0) model. However, the model with the BIC calculation has a superior score, though not by much. The various residuals look fine, do not tell much.



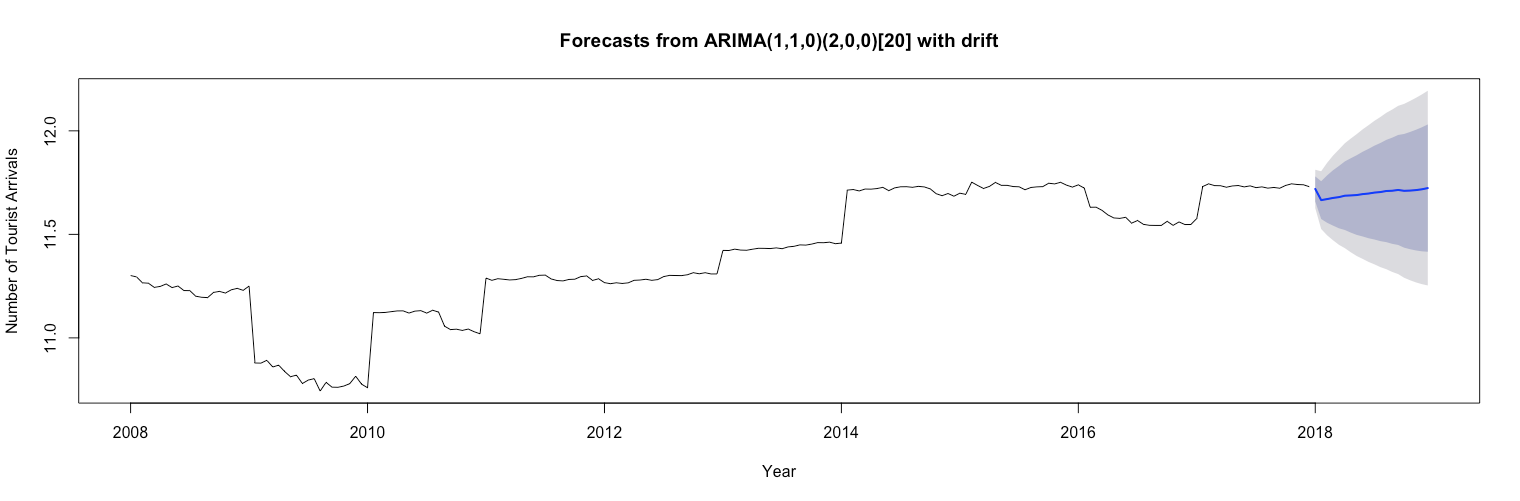
Looking at the summary of the ARIMA models, the coefficients are the same. Both potential models have a MAPE score of 0.175. This is a very good score and much lower than before the log transformation.



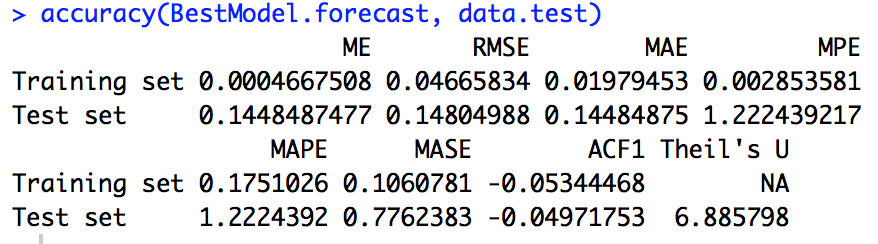
We now will look at the different test that we can perform for our model. Our options are the ‘kpss’, ‘adf’, and ‘pp’ tests. We will keep with the ‘bic’ as our information criterion. They are all similar, but we will use the ‘kpss’ test.

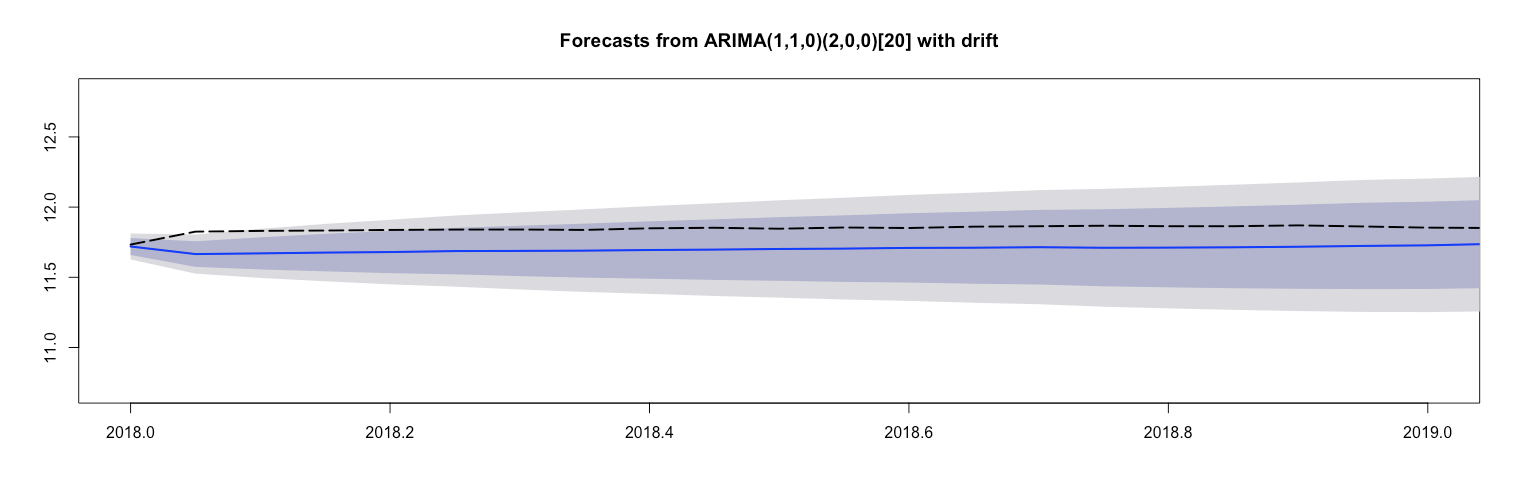


However, according to the Box-Ljung test we have a p-value >0.5. This means that our model has failed the autocorrelation test and autocorrelation exists. Therefore, the model is a GARCH model. However, it is normally distributed according to the Jarque Bera Test. Here is the plot of our best model with the forecast.

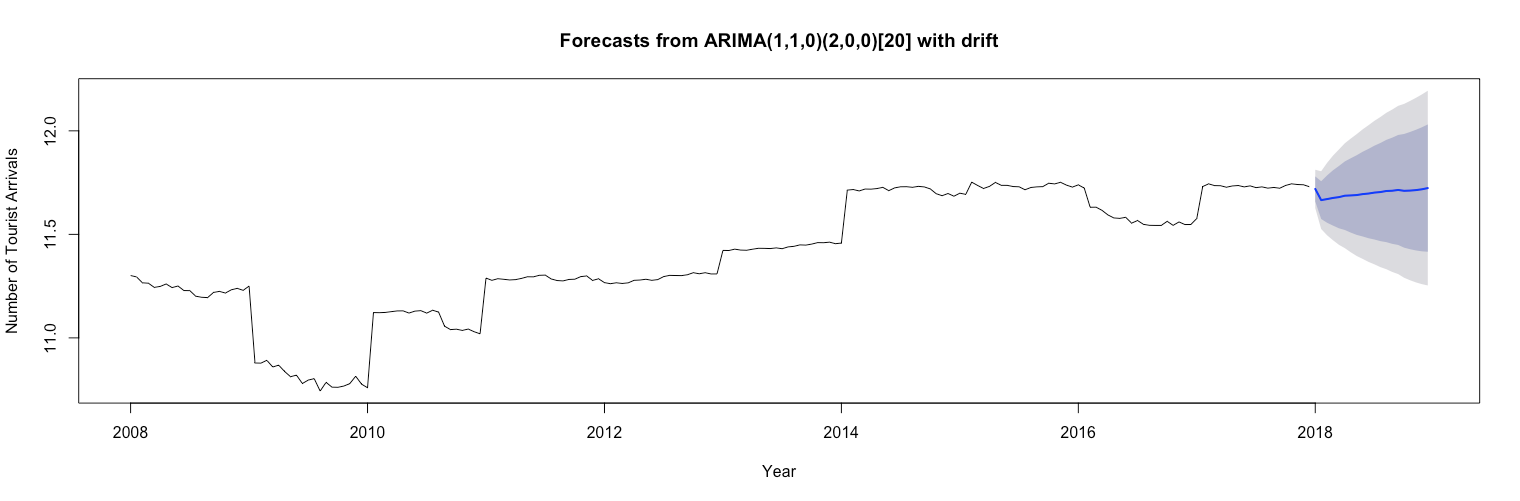


When forecasted, we can see that our model is not exactly accurate. Our forecast actually underperforms how well this year’s January effect is doing.





We can see though that the forecast is within our confidence interval when forecasting a January ahead. We attempted manual plotting and this is the best that we came up with. Therefore, our final model



However, this is consistent with literature. Many economists believe that the January effect is no longer an effective measure. In fact many believe that if it was, it ended as computational power increased in the nineties or early 2000s. One reason that it is believed that the January Effect ended is no longer relevant is because trader’s algorithms have already corrected for it. Here is a chart showing how much of a change occurred in January in the past 10 years.

|  |  |
| --- | --- |
| Year | % Change |
| 2008 | -5% |
| 209 | -12% |
| 2010 | -10% |
| 2011 | -1% |
| 2012 | +6% |
| 2013 | +5% |
| 2014 | -2% |
| 2015 | -3% |
| 2016 | -6% |
| 2017 | +0.2% |
| 2018 | +2% |

Appendix

library(Quandl)

Quandl.api\_key('zZr9ixixy3s7xbCpZTLP')

# memory check

rm(list = ls())

gc()

par(mfrow=c(1,1))

#Set the working directory...remeber to reset/change directory as needed...

setwd("/Users/Phillip/Documents/SMU/BusinessIntelligence/JanuaryEffect")

#read in the data from Quandl...

WikiCodes <- read.csv('WIKI-datasets-codes.csv')

WikiCodes <- subset(WikiCodes, select = -2)

WikiCodes$WIKI.AAPL <- gsub("WIKI/", "", WikiCodes$WIKI.AAPL)

# mydata <- Quandl.datatable("WIKI/PRICES", ticker='ABC', qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g1 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1:100,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g2 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[101:200,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g3 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[201:300,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g4 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[301:400,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g5 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[401:500,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

#Let's combine into one dataframe for simplicity...

dfCombined <- rbind(g1,g2,g3,g4,g5)

nrow(dfCombined)

rowCount <- nrow(g1) + nrow(g2) + nrow(g3) + nrow(g4) + nrow(g5) #to compare and make sure we have not dropped any rows...

rowCount

identical(nrow(dfCombined), rowCount) #Are row counts same? True if yes...

rm(g1, g2, g3, g4, g5)

gc()

g6 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[501:600,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g7 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[601:700,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g8 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[701:800,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g9 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[801:900,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g10 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[901:1000,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

#Let's combine into one dataframe for simplicity...

dfCombined <- rbind(dfCombined, g6,g7,g8,g9,g10)

nrow(dfCombined)

rowCount <- nrow(g6) + nrow(g7) + nrow(g8) + nrow(g9) + nrow(g10) #to compare and make sure we have not dropped any rows...

rowCount

identical(nrow(dfCombined), rowCount) #Are row counts same? True if yes...

rm(g6, g7, g8, g9, g10)

gc()

g11 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1001:1100,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g12 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1101:1200,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g13 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1201:1300,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g14 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1301:1400,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g15 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1401:1500,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

#Let's combine into one dataframe for simplicity...

dfCombined <- rbind(dfCombined, g11,g12,g13,g14,g15)

nrow(dfCombined)

rowCount <- nrow(g11) + nrow(g12) + nrow(g13) + nrow(g14) + nrow(g15) #to compare and make sure we have not dropped any rows...

rowCount

identical(nrow(dfCombined), rowCount) #Are row counts same? True if yes...

rm(g11, g12, g13, g14, g15)

gc()

g16 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1501:1600,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g17 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1601:1700,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g18 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1701:1800,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g19 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1801:1900,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g20 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[1901:2000,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

#Let's combine into one dataframe for simplicity...

dfCombined <- rbind(dfCombined, g16,g17,g18,g19,g20)

nrow(dfCombined)

rowCount <- nrow(g16) + nrow(g17) + nrow(g18) + nrow(g19) + nrow(g20) #to compare and make sure we have not dropped any rows...

rowCount

identical(nrow(dfCombined), rowCount) #Are row counts same? True if yes...

rm(g16, g17, g18, g19, g20)

gc()

g21 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2001:2100,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g22 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2101:2200,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g23 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2201:2300,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g24 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2301:2400,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g25 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2401:2500,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

#Let's combine into one dataframe for simplicity...

dfCombined <- rbind(dfCombined, g21,g22,g23,g24,g25)

nrow(dfCombined)

rowCount <- nrow(g21) + nrow(g22) + nrow(g23) + nrow(g24) + nrow(g25) #to compare and make sure we have not dropped any rows...

rowCount

identical(nrow(dfCombined), rowCount) #Are row counts same? True if yes...

rm(g21, g22, g23, g24, g25)

gc()

g26 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2501:2600,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g27 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2601:2700,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g28 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2701:2800,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g29 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2801:2900,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

g30 <- Quandl.datatable("WIKI/PRICES", ticker=WikiCodes[2901:3000,], qopts.columns=c("ticker", "date", "close"), paginate = TRUE)

#Let's combine into one dataframe for simplicity...

dfCombined <- rbind(dfCombined, g26,g27,g28,g29,g30)

nrow(dfCombined)

rowCount <- nrow(dfComined) + nrow(g26) + nrow(g27) + nrow(g28) + nrow(g29) + nrow(g30) #to compare and make sure we have not dropped any rows...

rowCount

identical(nrow(dfCombined), rowCount) #Are row counts same? True if yes...

rm(g26, g27, g28, g29, g30)

gc()

# write csv

write.csv(dfCombined, "wikiprices.csv") #For fun and learning, let's dump to csv file...

gc()

#OK, let's get a feel for what we are dealing with...

#What are the dimensions...

x<- dfCombined

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# BEGIN - Exploratory Data Analysis

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

dim(x) # How big is x?

x[1:6,] # Show the first 6 rows and columns.

#What is the day count...

dayCount = integer(7)

for (i in 1:7)

dayCount[i] <- sum(x[,"DayOfWeek"] == i)

dayCount

summary (dfCombined)

dim(dfCombined)

library(dplyr)

AEP\_ret <- diff( AEP$close) / lag( AEP$close, k = -1) \* 100

# Defining variables

Y <- AEP\_ret

d.Y <- diff(Y)

t <- AEP$date

# Descriptive statistics and plotting the data

summary(Y)

summary(d.Y)

plot(t,Y)

plot(d.Y)

#Test for stationarity

# Dickey-Fuller test for variable

adf.test(Y, alternative="stationary", k=0)

# Augmented Dickey-Fuller test - test for stationarity (Non-Seasonal)

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

# DF and ADF tests for differenced variable

adf.test(d.Y, k=0)

adf.test(d.Y)

data = c(J2008,J2009, J2010,J2011,J2012,J2013,J2014,J2015,J2016,J2017, J2018)

data<-log(data) #to log the data

#data<-1/data # inverse transofrmation

#data<-sqrt(data) # square root

#data<-sin(data)

ts.data<-ts(data, frequency = 20, start = c(2008,1))

plot(ts.data)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Split data into training and testing data set

data.train <- window(ts.data, start=c(2008,1), end=c(2017,20))

plot(data.train)

dim(as.matrix(data.train))

data.test <- window(ts.data, start=c(2018,1))

plot(data.test)

dim(as.matrix(data.test))

library(forecast)

# difference is using aic or bic. trsin using kpss

arima1 <- auto.arima(data.train, trace=TRUE, test="kpss", ic="aic")

arima2 <- auto.arima(data.train, trace=TRUE, test="kpss", ic="bic") #Note: BIC is usually better for larger samples...

summary(arima1)

confint(arima1)

# arima 1 recommends ARIMA(1,1,0)(1,0,2) : 3847.061

# arima 2 recommends ARIMA(1,1,0)(1,0,1) : 3857.68

#going with arima 2 ARIMA(1,1,0)(1,0,1) using bic

# minimal difference

tsdiag(arima1)

# standardied residuals - white noise

# ACF - 1 large positive then nothing else special

# Ljun-Box p values - 1st 2 high but then close to 0. Stays above 0.01 until lag 10

tsdiag(arima2)

# not much of a difference from arim1

summary(arima1)

summary(arima2)

# not much of a difference from arima1

#Note that the model is the same, but the differing values of aic vs bic...

#If we want to change the model creation, we can vary the criteria...

arima1 <- auto.arima(data.train, trace=TRUE, test="kpss", ic="bic")

arima2 <- auto.arima(data.train, trace=TRUE, test="adf", ic="bic") #Note: BIC is usually better for larger samples...

arima3 <- auto.arima(data.train, trace=TRUE, test="pp", ic="bic") #Note: BIC is usually better for larger samples...

#and let's check what the differences are...

summary(arima1)

summary(arima2)

summary(arima3)

#Lowest MAPE (and others) shows a bias towards kpss as the test to use...

BestModel <- auto.arima(data.train, seasonal = TRUE, trace=TRUE, test="kpss", ic="bic")

BestModel

summary(BestModel)

confint(BestModel)

tsdiag(BestModel)

BestModel<-arima1

#Residual Diagnostics

plot.ts(BestModel$residuals)

Box.test(BestModel$residuals, lag=20, type="Ljung-Box") #If p-val >0.05 then fail, meaning autocorrelation exists

acf(BestModel$residuals, lag.max = 24, main="ACF of the Model") #if spikes touch the bounds, this is also an indication of failing the autocorrelation test

#Box.test(BestModel$residuals^2, lag = 20, type = "Ljung-Box") #Test for GARCH effect, if pval >0.05 then have ARCH effect, and should consider a volatility model (i.e. GARCH, etc.)

library(tseries)

# p less than 0.5 normal dist

jarque.bera.test(BestModel$residuals) #if p-value >0.05 then the residuals have a normal distibution (not reject null Hypothesis of normality)

BestModel.forecast <- forecast(BestModel, h=20) #forecast 24 periods ahead

BestModel.forecast

plot(BestModel.forecast, xlab="Year", ylab="Number of Tourist Arrivals")

library(TSPred)

plotarimapred(data.test, BestModel, xlim=c(2018, 2019), range.percent = 0.05)

accuracy(BestModel.forecast, data.test)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Residual Diagnostics

plot.ts(arima1$residuals)

Box.test(arima1$residuals, lag=20, type="Ljung-Box") #If p-val <0.05 then fail, meaning autocorrelation exists

acf(arima1$residuals, lag.max = 24, main="ACF of the Model") #if spikes touch the bounds, this is also an indication of failing the autocorrelation test

Box.test(arima1$residuals^2, lag = 20, type = "Ljung-Box") #Test for GARCH effect, if pval <0.05 then have ARCH effect, and should consider a volatility model (i.e. GARCH, etc.)

library(tseries)

jarque.bera.test(arima1$residuals) #if p-value >0.05 then the residuals have a normal distibution (not reject null Hypothesis of normality)

#library(forecast)

#install.packages('forecast')

#install.packages('forecast', dependencies = TRUE)

#arima1.forecast <- forecast.Arima(arima1, h=24) #forecast 24 periods ahead

#arima1.forecast <- forecast::auto.arima(data.train)

arima1.forecast <- forecast(arima1, h=20) #forecast 24 periods ahead

arima1.forecast

plot(arima1.forecast, xlab="Year", ylab="Number of Tourist Arrivals")

library(TSPred)

plotarimapred(data.test, arima1, xlim=c(2011, 2013), range.percent = 0.05)

accuracy(arima1.forecast, data.test)

#Does not look particularly great, so let's start trying different models...

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Different Model Combinations....

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BestModel1 <- auto.arima(data.train, trace=TRUE, test="kpss", ic="aic")

BestModel2 <- auto.arima(data.train, trace=TRUE, test="kpss", ic="bic")

BestModel3 <- auto.arima(data.train, trace=TRUE, test="kpss", ic="aicc")

BestModel4 <- auto.arima(data.train, trace=TRUE, test="adf", ic="aic")

BestModel5 <- auto.arima(data.train, trace=TRUE, test="adf", ic="bic")

BestModel6 <- auto.arima(data.train, trace=TRUE, test="adf", ic="aicc")

BestModel7 <- auto.arima(data.train, trace=TRUE, test="pp", ic="aic")

BestModel8 <- auto.arima(data.train, trace=TRUE, test="pp", ic="bic")

BestModel9 <- auto.arima(data.train, trace=TRUE, test="pp", ic="aicc")

modList <- list(BestModel1, BestModel2, BestModel3, BestModel4, BestModel5, BestModel6, BestModel7, BestModel8, BestModel9)

modList

Amodel <- BestModel1

#setwd("D:/My Document Storage/2015 SMU/Econometrics/Week 8\_Special Topics/Plots\_forecast")

par(mfrow=c(1,1))

for (Amodel in modList){

i=0

arima1.forecast <- forecast(Amodel, h=24) #forecast 24 periods ahead

arima1.forecast

plot(arima1.forecast, xlab="Year", ylab="Number of Tourist Arrivals")

library(TSPred)

plotarimapred(data.test, Amodel, xlim=c(2011, 2013), range.percent = 0.05)

accuracy(arima1.forecast, data.test)

#plot as confirmation

pdf( paste0("Plot - ", Amodel,format(Sys.time(), "%a %b %d %H %M %S %Y"), " .pdf"),width=7,height=5)

plotarimapred(data.test, Amodel, xlim=c(2011, 2013), range.percent = 0.05)

accuracy(arima1.forecast, data.test)

dev.off()

}

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#Manual Plotting...

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*

plotarimapred(data.test, BestModel1, xlim=c(2011, 2013), range.percent = 0.05)

plotarimapred(data.test, BestModel2, xlim=c(2011, 2013), range.percent = 0.05)

plotarimapred(data.test, BestModel3, xlim=c(2011, 2013), range.percent = 0.05)

plotarimapred(data.test, BestModel4, xlim=c(2011, 2013), range.percent = 0.05)

plotarimapred(data.test, BestModel5, xlim=c(2011, 2013), range.percent = 0.05)

plotarimapred(data.test, BestModel6, xlim=c(2011, 2013), range.percent = 0.05)

plotarimapred(data.test, BestModel7, xlim=c(2011, 2013), range.percent = 0.05)

plotarimapred(data.test, BestModel8, xlim=c(2011, 2013), range.percent = 0.05)

plotarimapred(data.test, BestModel9, xlim=c(2011, 2013), range.percent = 0.05)