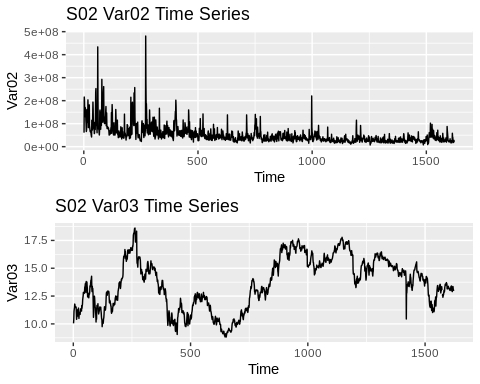
DATA624 Project #1

Jonathan Hernandez

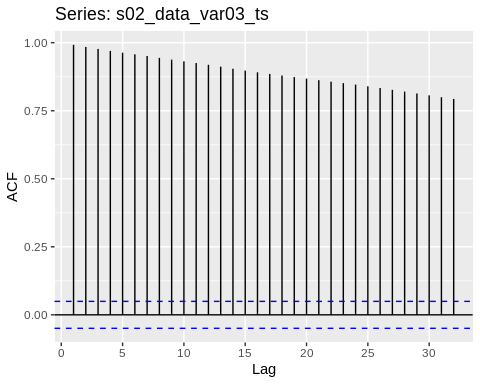
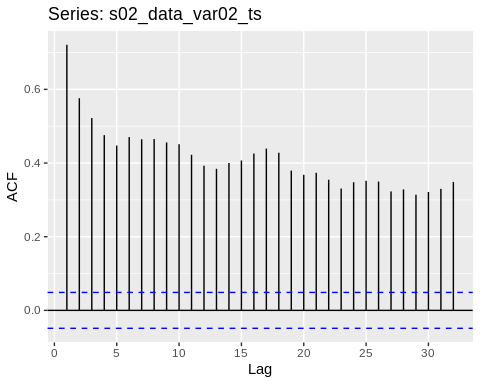
* First let’s acquire the data, extract the s02 group and the Var02 and Var03 features and convert them to a time series object for analysis.

## SeriesInd group Var02 Var03   
## Min. :40669 Length:1622 Min. : 7128800 Min. : 8.82   
## 1st Qu.:41253 Class :character 1st Qu.: 27880300 1st Qu.:11.82   
## Median :41846 Mode :character Median : 39767500 Median :13.76   
## Mean :41843 Mean : 50633098 Mean :13.68   
## 3rd Qu.:42430 3rd Qu.: 59050900 3rd Qu.:15.52   
## Max. :43021 Max. :480879500 Max. :38.28   
## NA's :4

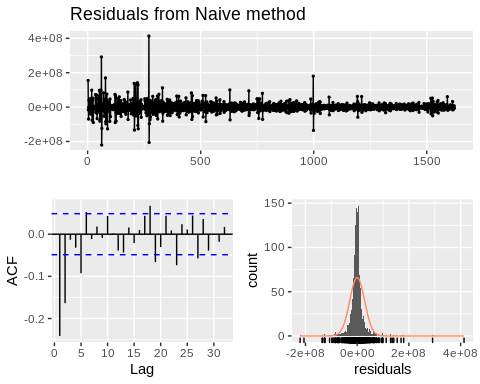
* Convert the variables to time series objects
* We see that there are several missing data in Var03. Let’s use R’s ImputeTS library and one of its functions na.interpolation and specify to replace the NA’s using the spline option which uses polynomial interpretation.
* Let’s remove the outlier from Var03 in the s02 group for better analysis
* With the data selected in question, let’s look at the time series of Var02 and Var03 using autoplot(). This will show us of the behavior of the data over time or the series Index



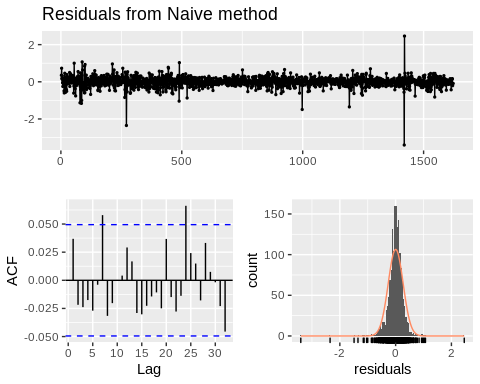
* By using the option “spline” in imputing data in our time series models, we see that it seems to do the best job in replacing NA values. Using splines estimates NA values using a polynomial interpolation. It helps us in the Var02 feature as it follows a downward trend.
* Making ACF plots of each time series:



* Now with out data cleaned/imputed, let’s do some forecasting using various techniques such as using ARIMA models, Naive forecasting, STL/ETS and Holt methods. I will use training/test sets preferably a 70/30 training/test set for each model. (training data will be from indexes 1 to , the test set the rest)
* Let’s start with fitting a naive model for both time series.

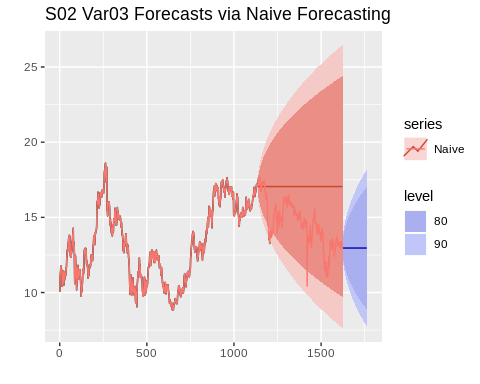
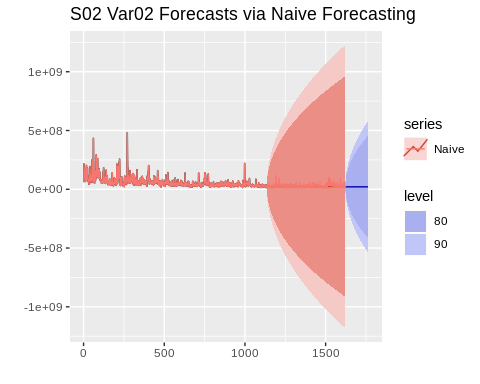


##   
## Ljung-Box test  
##   
## data: Residuals from Naive method  
## Q\* = 162.36, df = 10, p-value < 2.2e-16  
##   
## Model df: 0. Total lags used: 10



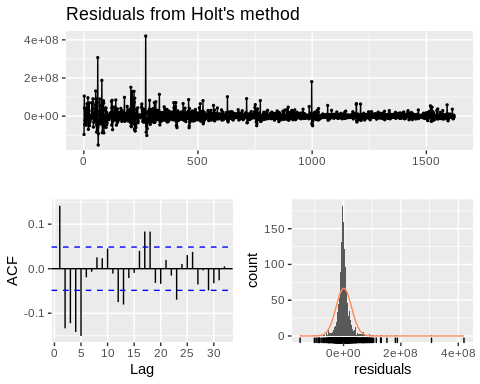
##   
## Ljung-Box test  
##   
## data: Residuals from Naive method  
## Q\* = 14.24, df = 10, p-value = 0.1623  
##   
## Model df: 0. Total lags used: 10

* Plotting the forecasts of both variables using the naive method

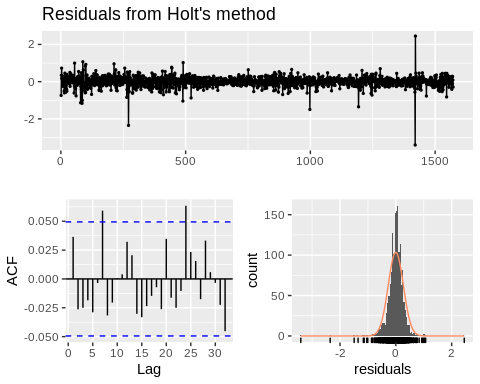


* Next is to use exponential smoothing and methods such as Holt’s methond and Holt-Winters Seasonal Method. It seems that Var02 as mentioned earlier is following a downward trend and looks to have so seasonality so Holt’s method may be useful. Var03 as what appears to be a seasonal trend.
* Using Holt’s method

## Warning in ets(x, "AAN", alpha = alpha, beta = beta, phi = phi, damped =  
## damped, : Missing values encountered. Using longest contiguous portion of  
## time series

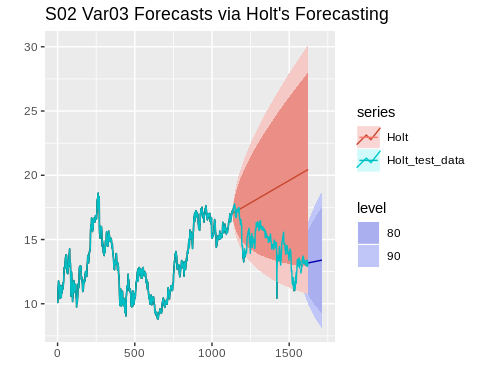
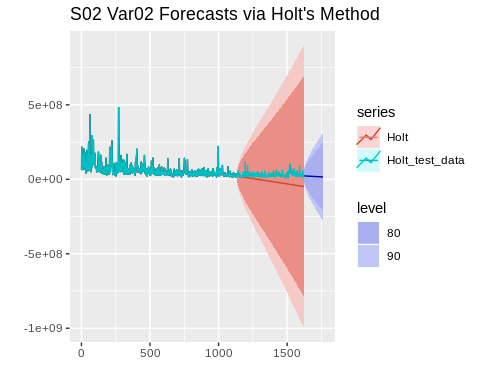


##   
## Ljung-Box test  
##   
## data: Residuals from Holt's method  
## Q\* = 162.04, df = 6, p-value < 2.2e-16  
##   
## Model df: 4. Total lags used: 10

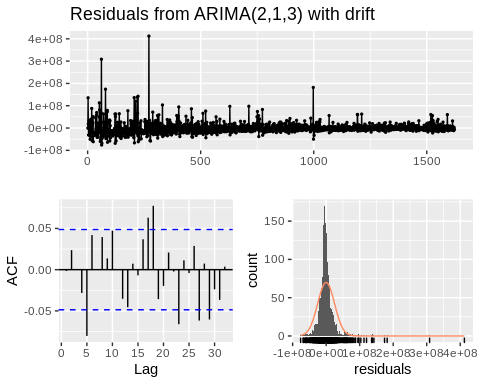


##   
## Ljung-Box test  
##   
## data: Residuals from Holt's method  
## Q\* = 13.838, df = 6, p-value = 0.0315  
##   
## Model df: 4. Total lags used: 10

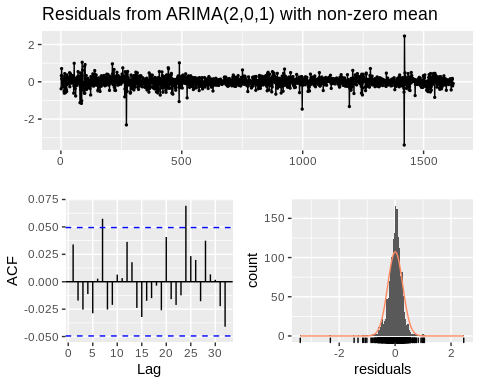
* Forecasts using Holt’s method:



* Using now ARIMA (auto.arima) models:

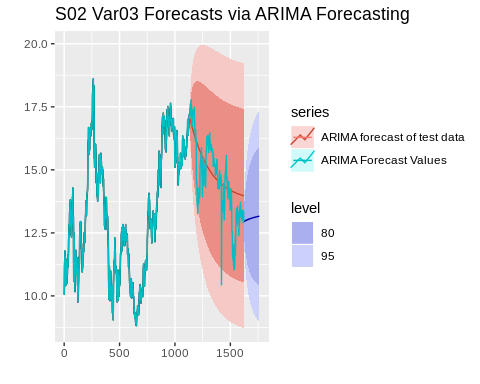
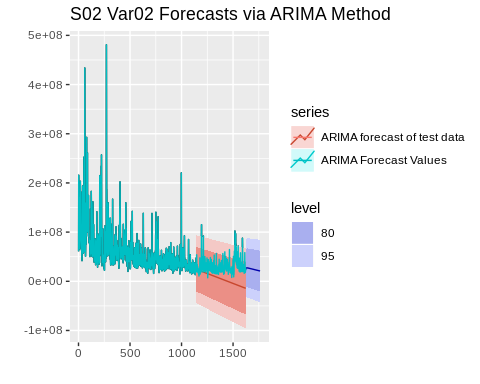


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,3) with drift  
## Q\* = 22.105, df = 4, p-value = 0.000191  
##   
## Model df: 6. Total lags used: 10



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,0,1) with non-zero mean  
## Q\* = 12.867, df = 6, p-value = 0.0452  
##   
## Model df: 4. Total lags used: 10

* Forecasts using auto.arima:



* We see that the ARIMA model used for Var02 shows good prediction for the confidence intervals and shows the downward trend. Using the auto.arima for Var03 doesn’t show the best results from looking at the p-value of the residuals plot.
* Let’s now compute some metrics of our models such as RMSE and MAPE and use the lowest value to predict the next 140 steps/points into the future.
* ARIMA evaluation

## ME RMSE MAE MPE MAPE MASE  
## Training set -73499.73 29019629 16666384 -10.71932 27.75076 0.9008189  
## Test set 25100784.96 31308017 25670259 77.18083 81.07837 1.3874788  
## ACF1 Theil's U  
## Training set 0.00434312 NA  
## Test set 0.76407392 2.431332

## ME RMSE MAE MPE MAPE  
## Training set 0.002870463 0.2573655 0.1827263 -0.01600712 1.430800  
## Test set -0.300049551 1.0769791 0.8348528 -2.71580192 6.065755  
## MASE ACF1 Theil's U  
## Training set 0.9959223 -0.005227703 NA  
## Test set 4.5502405 0.961327707 3.789606

* Holt evaluation

## ME RMSE MAE MPE MAPE MASE  
## Training set 1266142 31296498 17645535 -7.290419 27.88251 0.9537421  
## Test set 44179003 51119153 44254911 148.153794 148.73057 2.3919802  
## ACF1 Theil's U  
## Training set 0.1388371 NA  
## Test set 0.8727241 4.248005

## ME RMSE MAE MPE MAPE  
## Training set -0.001308986 0.2598169 0.1833154 -0.03338462 1.437822  
## Test set -4.160878963 4.7598242 4.1896830 -30.40420479 30.568227  
## MASE ACF1 Theil's U  
## Training set 0.9991329 0.1016951 NA  
## Test set 22.8352411 0.9849096 16.55006

* Naive evaluation

## ME RMSE MAE MPE MAPE MASE  
## Training set -33097.35 33190471 18501370 -6.866333 29.20479 1.0000000  
## Test set 7162454.21 15217194 9847560 11.053855 28.14745 0.5322611  
## ACF1 Theil's U  
## Training set -0.2400945 NA  
## Test set 0.5444130 1.060782

## ME RMSE MAE MPE MAPE  
## Training set 0.006172839 0.2594527 0.1834744 0.02577544 1.437301  
## Test set -2.462044739 2.8681128 2.5076416 -18.10709273 18.367871  
## MASE ACF1 Theil's U  
## Training set 1.00000 0.1045425 NA  
## Test set 13.66753 0.9756480 10.05445

* Looks like in regards to the MAPE, the ARIMA model works for the Var03 model and that for Var02, the ARIMA model works best when letting auto.arima do all the work.
* Finally, save our predictions 140 steps ahead (h=140)

# Appendix

library(readxl)  
library(dplyr)  
library(ggplot2)  
library(ggfortify)  
library(GGally)  
library(gridExtra)  
library(forecast)  
library(imputeTS)  
library(tidyverse)  
  
s02\_data <- read\_xls("Set for Class.xls", n\_max = 9732)  
# For my group, I requested to look at only the Series = 's02'. Per assignment,  
# you forecast Var02 and Var03 for S02  
  
# Extract only seriesid, group, var02 and var03  
s02\_data <- s02\_data %>% filter(group == "S02") %>%  
 select("SeriesInd", "group", "Var02", "Var03")  
  
summary(s02\_data)  
# Type Conversions. Change var02/03 to be time series  
s02\_data\_var02\_ts <- ts(s02\_data$Var02, start = 1, end = 1622,  
 frequency=1)  
s02\_data\_var03\_ts <- ts(s02\_data$Var03, start = 1, end = 1622,  
 frequency=1)  
s02\_data\_var03\_ts <- na.interpolation(s02\_data\_var03\_ts, option = "spline")  
# remove the large outlier the value == 38.28 per the summary  
idx\_outlier\_var03 <- which.max(s02\_data\_var03\_ts)  
s02\_data\_var03\_ts[idx\_outlier\_var03] <- NA  
# time series plot of var02  
var02\_plot <- autoplot(s02\_data\_var02\_ts) +  
 ggtitle("S02 Var02 Time Series") +  
 ylab("Var02")  
  
# time series plot of var03  
var03\_plot <- autoplot(s02\_data\_var03\_ts) +   
 ggtitle("S02 Var03 Time Series") +  
 ylab("Var03")  
  
# 2x1 plot arrangement  
grid.arrange(var02\_plot, var03\_plot)  
ggAcf(s02\_data\_var02\_ts)  
ggAcf(s02\_data\_var03\_ts)  
# naive forecasts and predict 140 steps ahead with 80% confidence invterval  
  
var02\_window\_training <- window(s02\_data\_var02\_ts, start=1, end=floor(1622\*0.7))  
var02\_window\_test <- window(s02\_data\_var02\_ts, start=floor(1622\*0.7))  
  
var03\_window\_training <- window(s02\_data\_var03\_ts, start=1, end=floor(1622\*0.7))  
var03\_window\_test <- window(s02\_data\_var03\_ts, start=floor(1622\*0.7))  
  
# train a naive forecast using training data  
s02\_var02\_naive\_test\_train <- naive(var02\_window\_training,  
 h = length(var02\_window\_test), level = c(80, 90))  
s02\_var03\_naive\_test\_train <- naive(var03\_window\_training,  
 h = length(var03\_window\_test), level = c(80, 90))  
  
# forecasts using naive method using the test windows/values  
s02\_var02\_naive\_test\_fit <- naive(s02\_data\_var02\_ts, h = 140, level = c(80, 90))  
s02\_var03\_naive\_test\_fit <- naive(s02\_data\_var03\_ts, h = 140, level = c(80, 90))  
  
# forecast values using forecast()  
  
checkresiduals(s02\_var02\_naive\_test\_fit)  
checkresiduals(s02\_var03\_naive\_test\_fit)  
# var02 plot  
autoplot(s02\_var02\_naive\_test\_fit) +  
 autolayer(s02\_var02\_naive\_test\_train, series="Naive") +  
 autolayer(s02\_data\_var02\_ts, series="Naive") +  
 ggtitle("S02 Var02 Forecasts via Naive Forecasting") +  
 xlab("") + ylab("")  
  
# var02 plot  
autoplot(s02\_var03\_naive\_test\_fit) +  
 autolayer(s02\_var03\_naive\_test\_train, series="Naive") +  
 autolayer(s02\_data\_var03\_ts, series="Naive") +  
 ggtitle("S02 Var03 Forecasts via Naive Forecasting") +  
 xlab("") + ylab("")  
# make holt predictions using the training data  
s02\_var02\_holt\_test\_train <- holt(var02\_window\_training,   
 h = length(var02\_window\_test), level=c(80,90))  
s02\_var03\_holt\_test\_train <- holt(var03\_window\_training,   
 h = length(var03\_window\_test), level=c(80,90))  
  
# forecasts using naive method using the test windows/values  
s02\_var02\_holt\_test\_fit <- holt(s02\_data\_var02\_ts, h = 140, level=c(80,90))  
s02\_var03\_holt\_test\_fit <- holt(s02\_data\_var03\_ts, h = 140, level=c(80,90))  
  
checkresiduals(s02\_var02\_holt\_test\_fit)  
checkresiduals(s02\_var03\_holt\_test\_fit)  
# var02 plot  
autoplot(s02\_var02\_holt\_test\_fit) +  
 autolayer(s02\_var02\_holt\_test\_train, series="Holt") +  
 autolayer(s02\_data\_var02\_ts, series="Holt\_test\_data") +  
 ggtitle("S02 Var02 Forecasts via Holt's Method") +  
 xlab("") + ylab("")  
  
  
# var02 plot  
autoplot(s02\_var03\_holt\_test\_fit) +  
 autolayer(s02\_var03\_holt\_test\_train, series="Holt") +  
 autolayer(s02\_data\_var03\_ts, series="Holt\_test\_data") +  
 ggtitle("S02 Var03 Forecasts via Holt's Forecasting") +  
 xlab("") + ylab("")  
# train an arima model using the training data  
# var02 not seasonal but more of a trend  
s02\_var02\_arima\_train <- auto.arima(var02\_window\_training, seasonal = FALSE)  
s02\_var03\_arima\_train <- Arima(var03\_window\_training, order = c(2,0,1))  
  
# make forecasts of the training data using arima models  
s02\_var02\_arima\_fit <- forecast(s02\_var02\_arima\_train, h=length(var02\_window\_test))  
s02\_var03\_arima\_fit <- forecast(s02\_var03\_arima\_train, h=length(var03\_window\_test))  
  
# forecast on the test data for arima  
s02\_var02\_arima\_test <- auto.arima(s02\_data\_var02\_ts, seasonal = FALSE) %>%  
 forecast(h=140)  
s02\_var03\_arima\_test <- Arima(s02\_data\_var03\_ts, order = c(2,0,1), seasonal = FALSE) %>%  
 forecast(h=140)  
  
# stl decomposition  
  
checkresiduals(s02\_var02\_arima\_test)  
checkresiduals(s02\_var03\_arima\_test)  
  
# var02 plot  
autoplot(s02\_var02\_arima\_test) +  
 autolayer(s02\_var02\_arima\_fit, series="ARIMA forecast of test data") +  
 autolayer(s02\_data\_var02\_ts, series="ARIMA Forecast Values") +  
 ggtitle("S02 Var02 Forecasts via ARIMA Method") +  
 xlab("") + ylab("")  
  
  
# var02 plot  
autoplot(s02\_var03\_arima\_test) +  
 autolayer(s02\_var03\_arima\_fit, series="ARIMA forecast of test data") +  
 autolayer(s02\_data\_var03\_ts, series="ARIMA Forecast Values") +  
 ggtitle("S02 Var03 Forecasts via ARIMA Forecasting") +  
 xlab("") + ylab("")  
# ARIMA  
accuracy(s02\_var02\_arima\_fit, var02\_window\_test)  
accuracy(s02\_var03\_arima\_fit, var03\_window\_test)  
accuracy(forecast(s02\_var02\_holt\_test\_train,  
 h=length(var02\_window\_test)), var02\_window\_test)  
accuracy(forecast(s02\_var03\_holt\_test\_train,  
 h=length(var03\_window\_test)), var03\_window\_test)  
accuracy(forecast(s02\_var02\_naive\_test\_train,  
 h=length(var02\_window\_test)), var02\_window\_test)  
accuracy(forecast(s02\_var03\_naive\_test\_train,  
 h=length(var03\_window\_test)), var03\_window\_test)  
predictions\_var02 <- s02\_var02\_arima\_test$mean  
write.csv(round(predictions\_var02), "s02\_var02\_forecasts.csv")  
predictions\_var03 <- s02\_var03\_arima\_test$mean  
write.csv(round(predictions\_var03, digits = 3), "s02\_var03\_forecasts.csv")