Data 624: Project 1

Group 2: Alice Friedman, Kayleah Griffen, Josh Iden, Michael Ippolito

6/11/2023

Table of Contents

## Introduction

This report is intended for colleagues from a variety of backgrounds and contains both technical and non-technical explanations of the work conducted. The objective of this project was to perform the appropriate analysis in order to forecast two variables (of five provided) each from six different time series sets. We were provided a spreadsheet that contains 1622 periods of every variable in every set and were expected to forecast 140 periods. The sets are labeled S01, S02, S03, S04, S05 and S06 and each contains variables labeled V01, V02, V03, V05, and V07. Different variables are required to be forecast depending on the set, specified below:

S01 – Forecast Var01, Var02 S02 – Forecast Var02, Var03 S03 – Forecast Var05, Var07 S04 – Forecast Var01, Var02 S05 – Forecast Var02, Var03 S06 – Forecast Var05, Var07

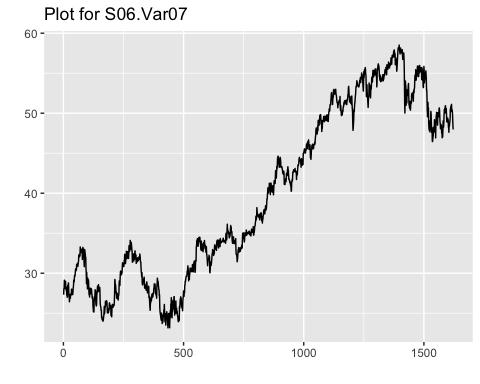
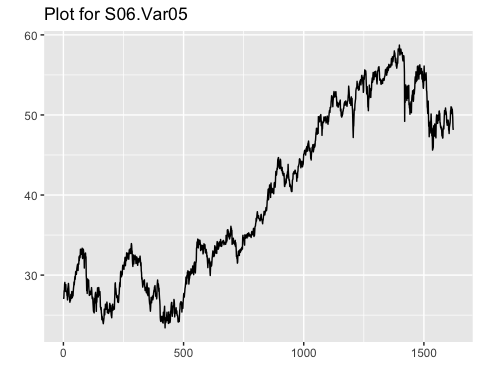
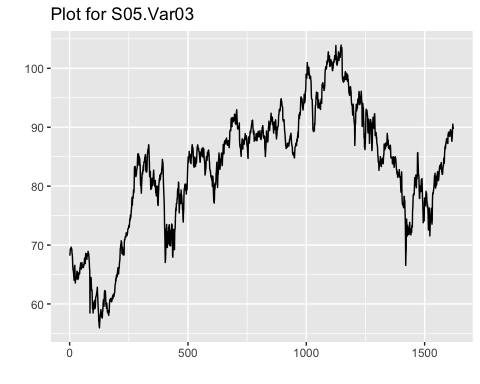
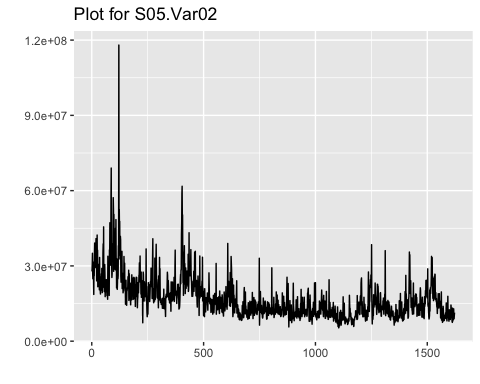
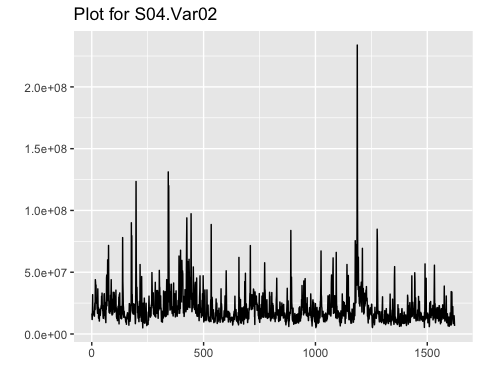
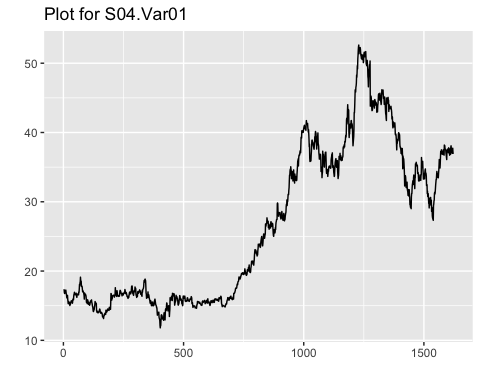
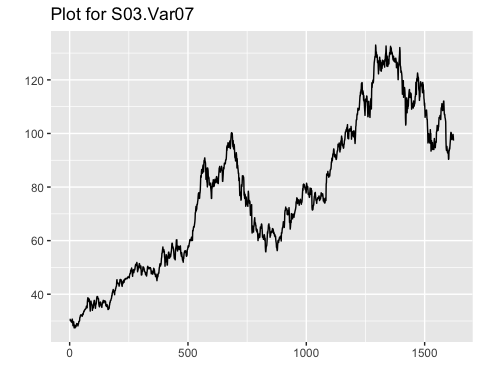
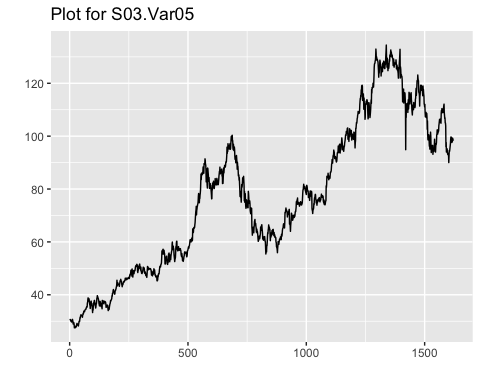
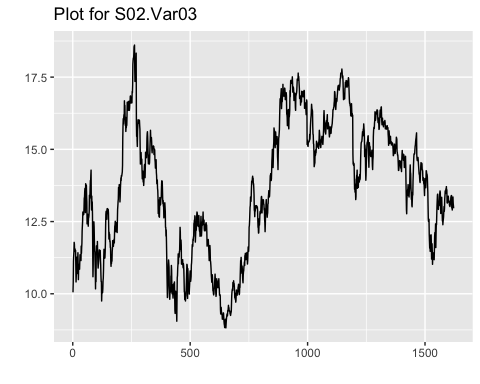
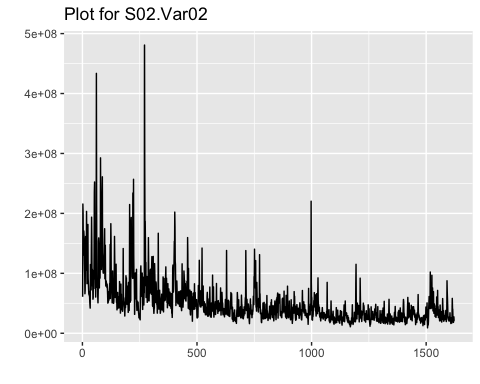
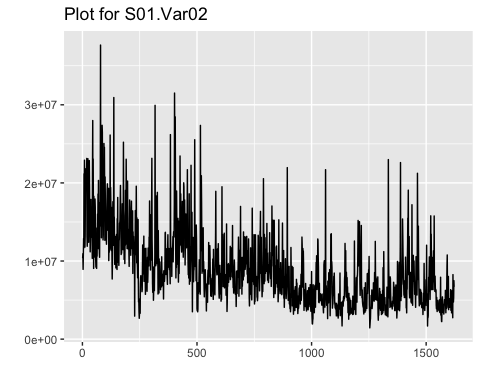
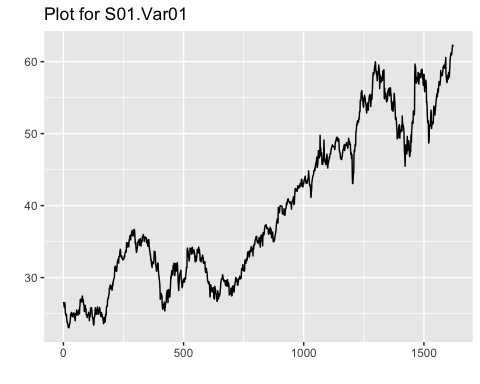
The goal of this report is to forecast 140 future time periods for each of the above variables, minimizing mean absolute percent error (MAPE). Results will be attached in an Excel file with each category in a separate tab.

## Data Preparation

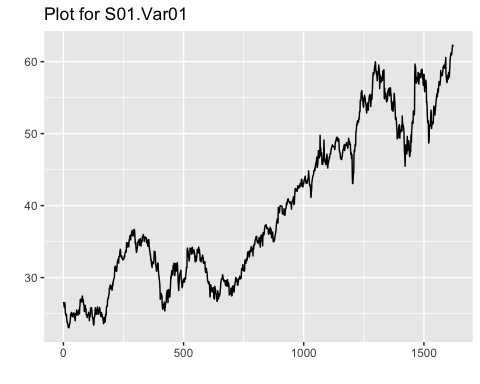
The data was provided as an excel (.xls) file. The columns provided were the series index, the category, and then a column for each variable. To conduct the data analysis and forecasting the open source software R was used. In order to begin processing the data, the data was read into R from github (where the provided data file was stored) and stored in a format in R called a dataframe. Below is a preview of the data to get an idea of the format.

| SeriesInd | category | Var01 | Var02 | Var03 | Var05 | Var07 |
| --- | --- | --- | --- | --- | --- | --- |
| 40,669 | S03 | 30.64286 | 123,432,400 | 30.34000 | 30.49000 | 30.57286 |
| 40,669 | S02 | 10.28000 | 60,855,800 | 10.05000 | 10.17000 | 10.28000 |
| 40,669 | S01 | 26.61000 | 10,369,300 | 25.89000 | 26.20000 | 26.01000 |
| 40,669 | S06 | 27.48000 | 39,335,700 | 26.82000 | 27.02000 | 27.32000 |
| 40,669 | S05 | 69.26000 | 27,809,100 | 68.19000 | 68.72000 | 69.15000 |
| 40,669 | S04 | 17.20000 | 16,587,400 | 16.88000 | 16.94000 | 17.10000 |
| 40,670 | S03 | 30.79857 | 150,476,200 | 30.46428 | 30.65714 | 30.62571 |
| 40,670 | S02 | 11.24000 | 215,620,200 | 10.40000 | 10.45000 | 10.96000 |
| 40,670 | S01 | 26.30000 | 10,943,800 | 25.70000 | 25.95000 | 25.86000 |
| 40,670 | S06 | 28.24000 | 55,416,000 | 27.24000 | 27.27000 | 28.07000 |

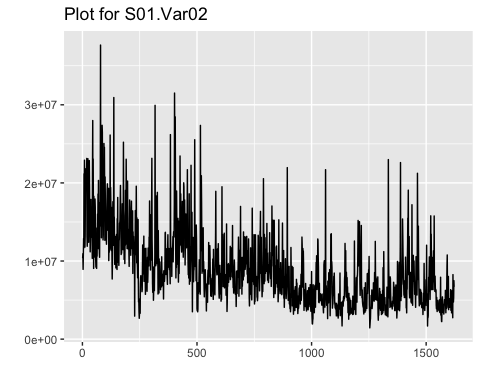
myts is now a list of clean time series objects – all the ones we need to forecast for



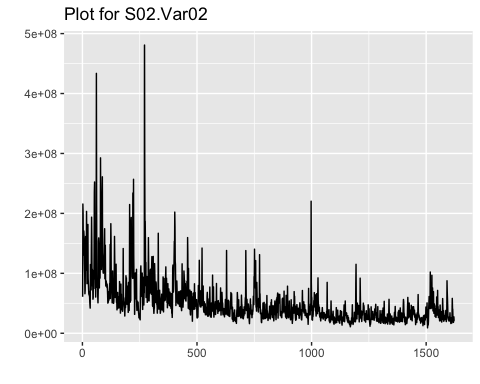
## [[1]]



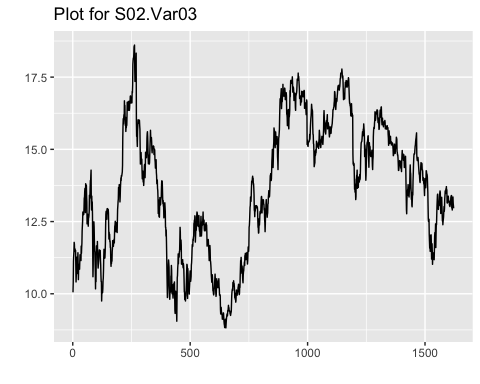
##   
## [[2]]



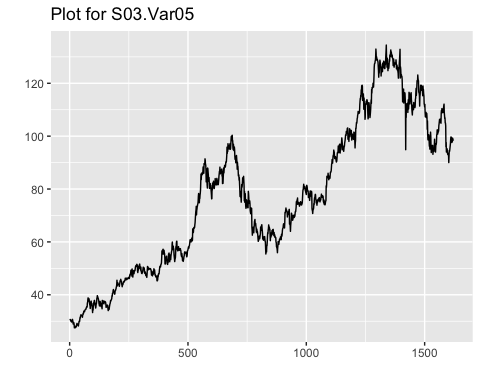
##   
## [[3]]



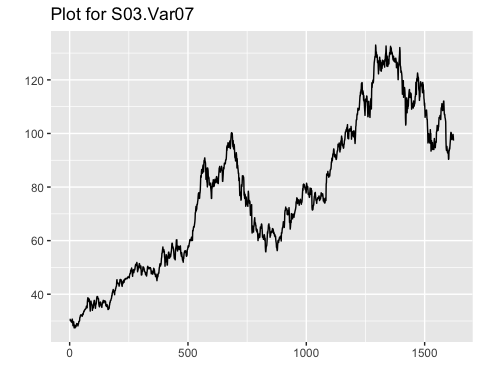
##   
## [[4]]



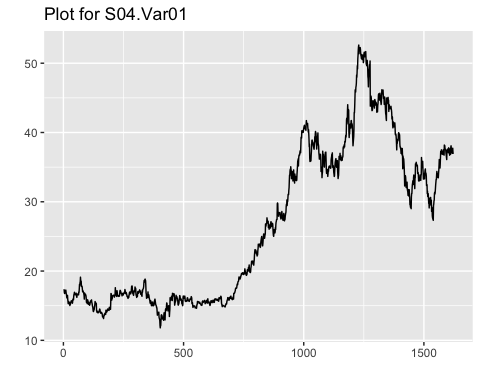
##   
## [[5]]



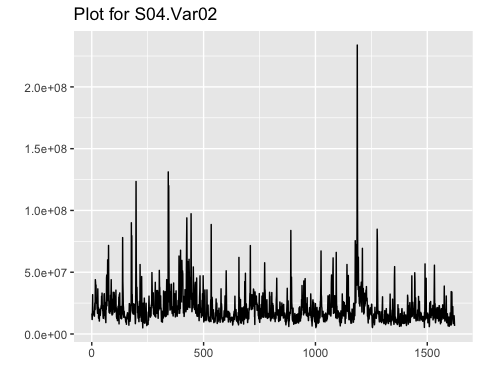
##   
## [[6]]



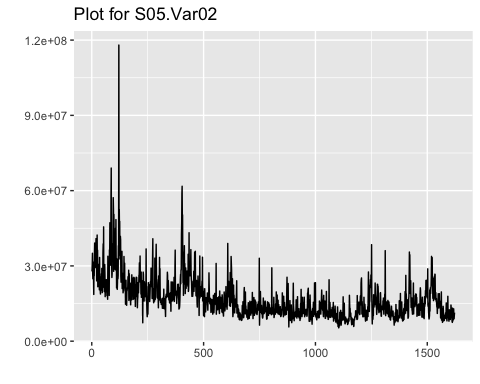
##   
## [[7]]



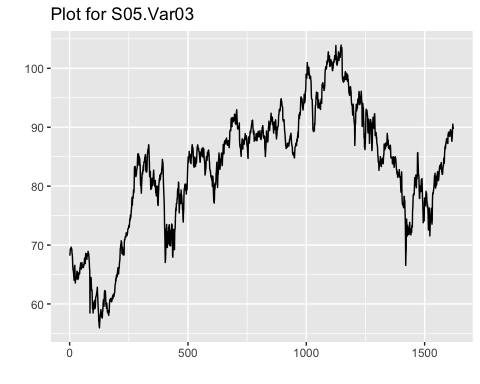
##   
## [[8]]



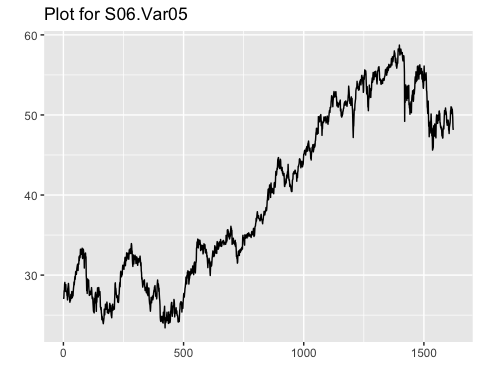
##   
## [[9]]



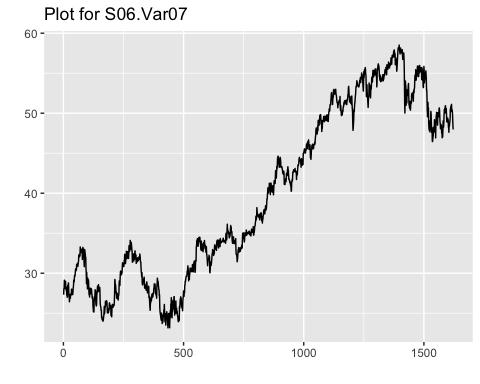
##   
## [[10]]



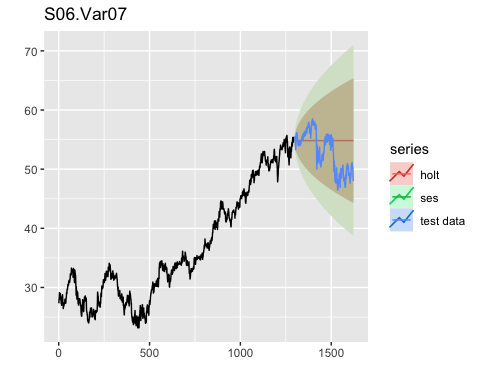
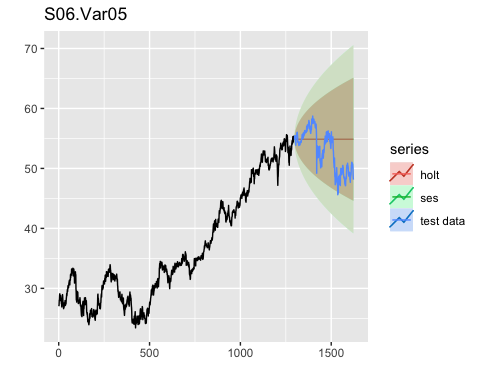
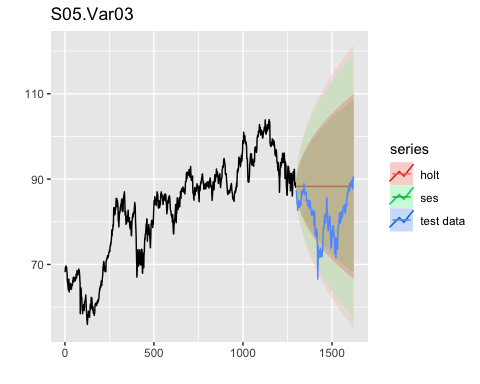
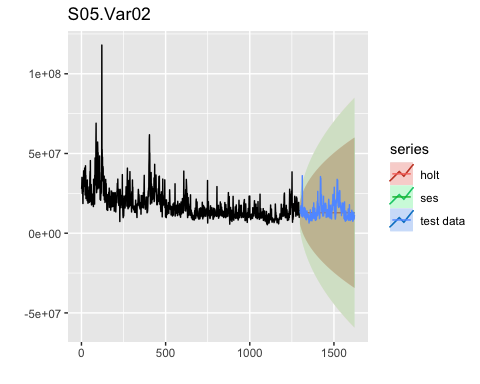
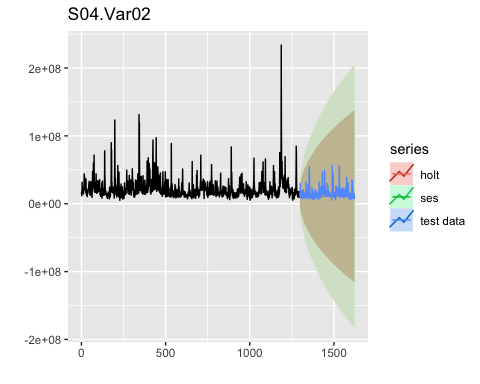
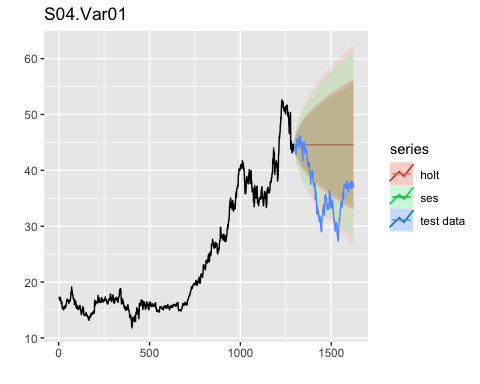
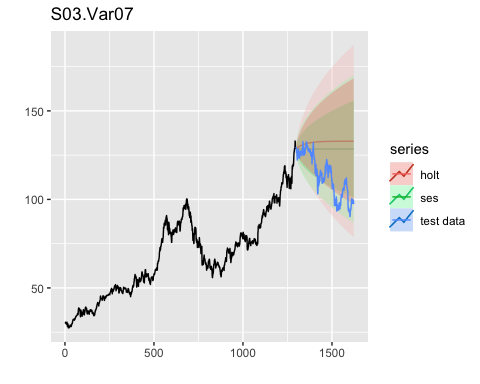
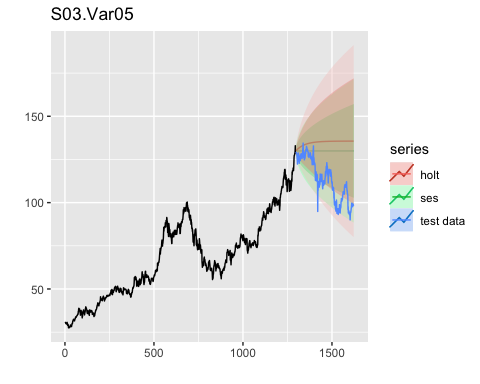
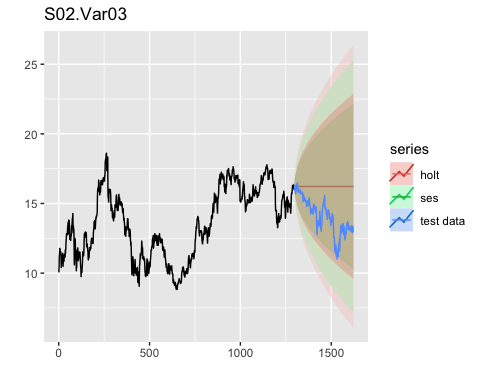
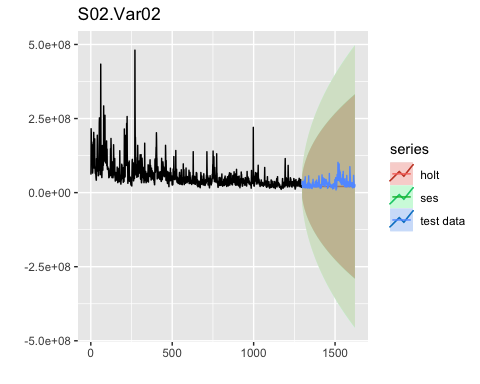
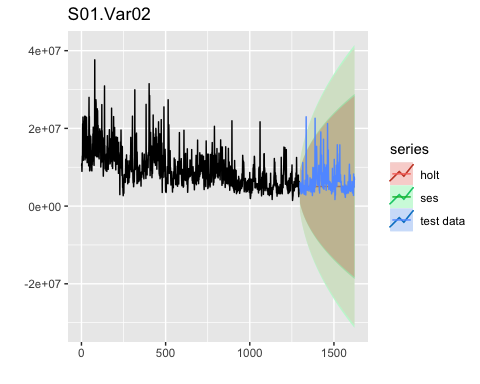
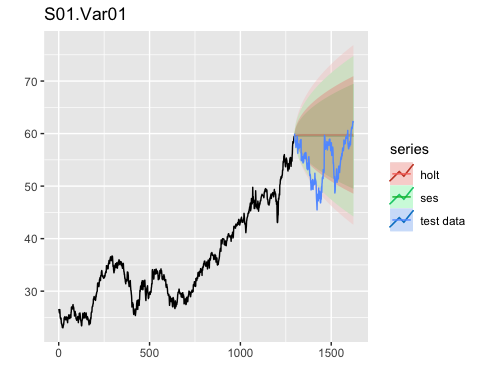
##   
## [[11]]

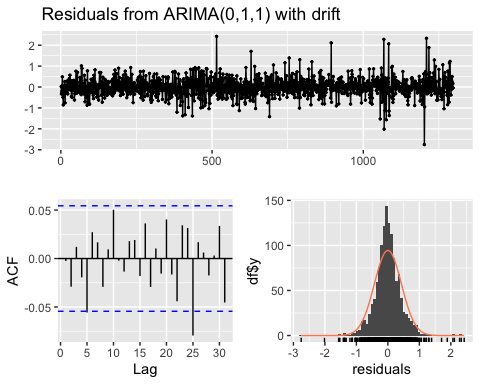


##   
## [[12]]

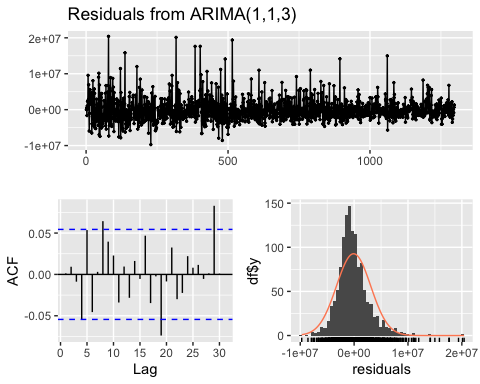


Using the function ndiffs we can see that each series requires 1 differencing to become stationary.

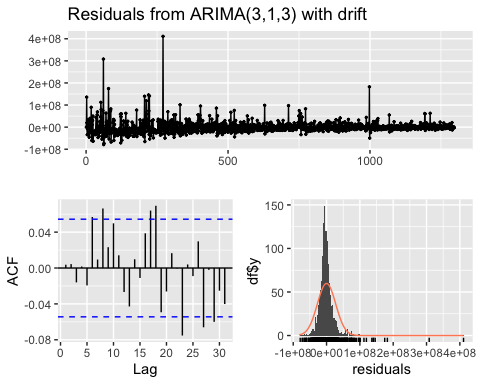




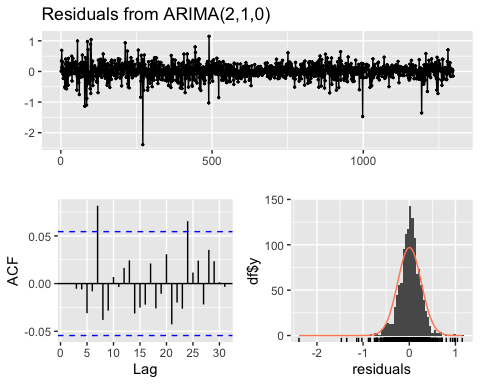
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1) with drift  
## Q\* = 11.711, df = 9, p-value = 0.2301  
##   
## Model df: 1. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1) with drift  
## Q\* = 11.711, df = 9, p-value = 0.2301



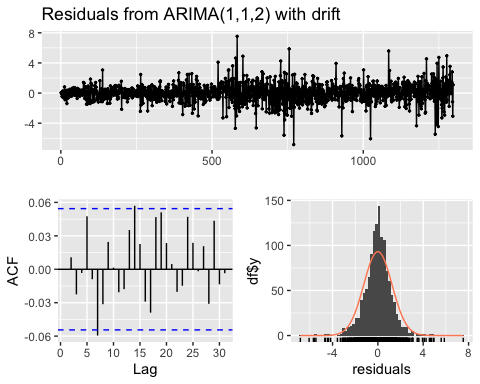
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 18.686, df = 6, p-value = 0.004728  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 18.686, df = 6, p-value = 0.004728



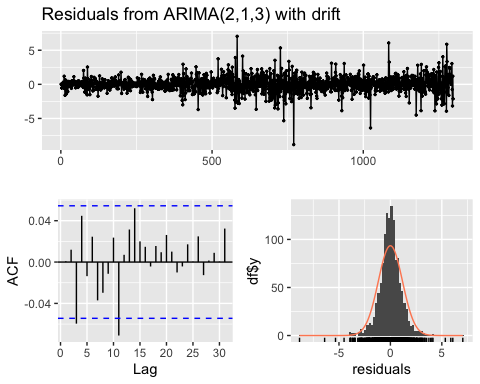
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,1,3) with drift  
## Q\* = 14.923, df = 4, p-value = 0.004863  
##   
## Model df: 6. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(3,1,3) with drift  
## Q\* = 14.923, df = 4, p-value = 0.004863



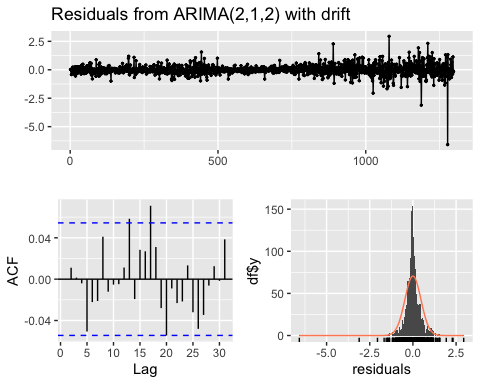
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,0)  
## Q\* = 13.102, df = 8, p-value = 0.1084  
##   
## Model df: 2. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,0)  
## Q\* = 13.102, df = 8, p-value = 0.1084



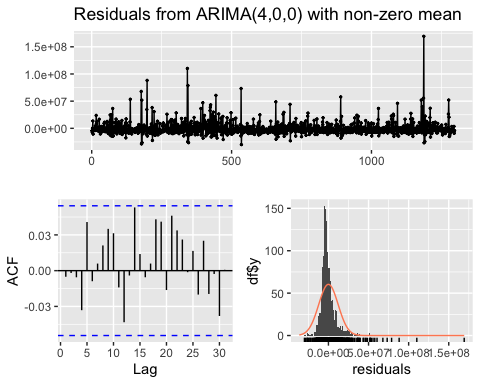
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,2) with drift  
## Q\* = 10.572, df = 7, p-value = 0.1584  
##   
## Model df: 3. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,2) with drift  
## Q\* = 10.572, df = 7, p-value = 0.1584



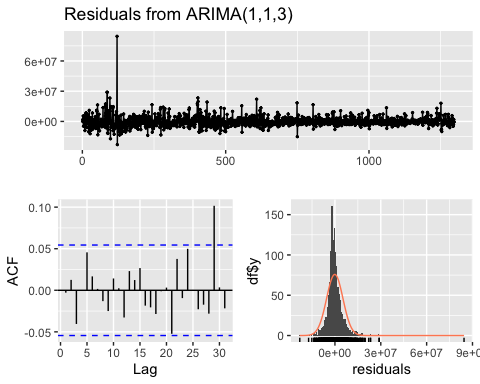
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,3) with drift  
## Q\* = 12.289, df = 5, p-value = 0.03103  
##   
## Model df: 5. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,3) with drift  
## Q\* = 12.289, df = 5, p-value = 0.03103



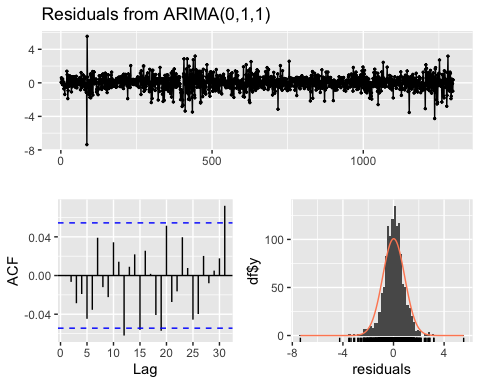
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 7.1813, df = 6, p-value = 0.3044  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 7.1813, df = 6, p-value = 0.3044



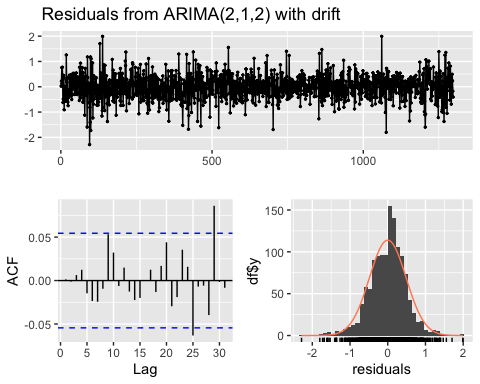
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,0,0) with non-zero mean  
## Q\* = 7.3155, df = 6, p-value = 0.2926  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(4,0,0) with non-zero mean  
## Q\* = 7.3155, df = 6, p-value = 0.2926



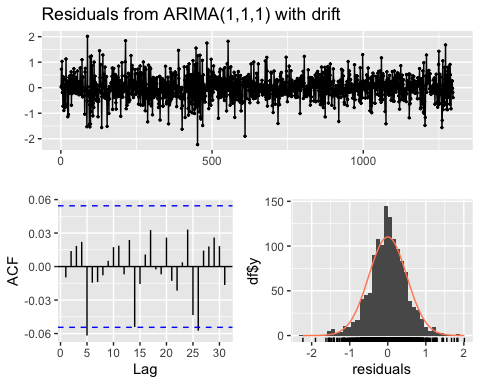
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 6.7384, df = 6, p-value = 0.3457  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,3)  
## Q\* = 6.7384, df = 6, p-value = 0.3457



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1)  
## Q\* = 10.229, df = 9, p-value = 0.3323  
##   
## Model df: 1. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1)  
## Q\* = 10.229, df = 9, p-value = 0.3323



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 7.2131, df = 6, p-value = 0.3016  
##   
## Model df: 4. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,2) with drift  
## Q\* = 7.2131, df = 6, p-value = 0.3016



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1) with drift  
## Q\* = 7.464, df = 8, p-value = 0.4875  
##   
## Model df: 2. Total lags used: 10  
##   
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1) with drift  
## Q\* = 7.464, df = 8, p-value = 0.4875

## list\_vars ses\_MAPE holt\_MAPE arima\_MAPE  
## S01.Var01 "S01.Var01" 9.165355 9.564382 16.36583   
## S01.Var02 "S01.Var02" 29.97856 29.93368 33.34665   
## S02.Var02 "S02.Var02" 29.72826 29.75418 51.60134   
## S02.Var03 "S02.Var03" 16.28359 16.30047 16.19708   
## S03.Var05 "S03.Var05" 15.91855 20.21204 27.66093   
## S03.Var07 "S03.Var07" 14.79226 17.98505 25.89001   
## S04.Var01 "S04.Var01" 23.41803 23.46231 33.20539   
## S04.Var02 "S04.Var02" 28.35688 28.36019 74.57495   
## S05.Var02 "S05.Var02" 25.68933 25.68588 31.70286   
## S05.Var03 "S05.Var03" 9.078404 9.012719 9.119705   
## S06.Var05 "S06.Var05" 5.887448 5.888391 11.08622   
## S06.Var07 "S06.Var07" 5.837965 5.84199 11.04406

## Appendix 1: R Code

knitr::opts\_chunk$set(fig.path='Figs/', echo=FALSE, warning=FALSE, message=FALSE, cache=TRUE)  
library(httr)  
library(kableExtra)  
library(fpp2)  
library(imputeTS)  
library(tidyverse)  
library(urca)  
library(ggfortify)  
library(gridExtra)  
library(scales)  
library(flextable)  
set.seed(123)  
  
github\_link <- "https://github.com/klgriffen96/summer23\_data624/raw/main/project\_1/Data%20Set%20for%20Class.xls"  
temp\_file <- tempfile(fileext = ".xls")  
req <- GET(github\_link,   
 # write result to disk  
 write\_disk(path = temp\_file))  
  
df <- readxl::read\_excel(temp\_file)  
  
head(df, 10) |>  
 flextable()  
#Split the data into data frames by category and var name  
  
df\_long <- df %>% gather(key, value, -SeriesInd, -category)  
split\_data <- split(df\_long, f=list(df\_long$category, df\_long$key))  
  
  
  
#put the combo into a list so it can be run through  
list\_vars <- c(  
 "S01.Var01", "S01.Var02",  
 "S02.Var02", "S02.Var03",  
 "S03.Var05", "S03.Var07",  
 "S04.Var01", "S04.Var02",  
 "S05.Var02", "S05.Var03",  
 "S06.Var05", "S06.Var07"  
)  
  
  
#select list items based on the list vars and then turn each list item into a clean ts  
myts <- lapply(split\_data[list\_vars], function(x) {  
 x %>%   
 dplyr::select(value) %>%   
 slice(1:1622) %>% #removes the missing values we need to predict  
 ts() %>%   
 tsclean(lambda = "auto") %>%  
 na\_ma()   
})  
  
lapply(list\_vars, function(var) {  
 ts <- myts[[var]]  
 plot <- autoplot(ts) + ggtitle(paste("Plot for", var))  
 print(plot)  
})  
test\_split <- function(x){  
 # Determine the index to split the time series into train and test sets  
 split.index <- floor(0.8 \* length(ts)) # 80% for training, 20% for testing  
   
 # Split the time series into train and test sets  
 train <- window(ts, end = split.index)  
 test <- window(ts, start = split.index + 1)  
   
 # Set the horizon  
 horizon <- length(test)  
  
 return(list(train, test, horizon))  
}  
# Create the empty vectors  
smooth\_results <- vector(mode = "list", length = length(myts))  
ses\_MAPE <- vector("numeric", length = length(list\_vars))  
holt\_MAPE <- vector("numeric", length = length(list\_vars))  
ses\_p <- vector("numeric", length = length(list\_vars))  
holt\_p <- vector("numeric", length = length(list\_vars))  
  
# Create a function to test the ses and holt forecasts with test and train data  
ses\_test <- function(x, i) {  
 ts <- diff(x)  
 ts <- x  
 # Determine the index to split the time series into train and test sets  
 split.index <- floor(0.8 \* length(ts)) # 80% for training, 20% for testing  
   
 # Split the time series into train and test sets  
 train <- window(ts, end = split.index)  
 test <- window(ts, start = split.index + 1)  
   
 # Set the horizon  
 horizon <- length(test)  
   
 # Ses fit with training data  
 ses.fit <- ses(train, h = horizon)  
 ses.p <- Box.test(residuals(ses.fit))$p.value  
   
 # Test with test data  
 ses\_res <- accuracy(ses.fit, test)['Test set', 'MAPE']  
   
  
 # Holt fit with training data  
 holt.fit <- holt(train, damped = TRUE, h = horizon)  
 holt.p <- Box.test(residuals(holt.fit))$p.value  
  
 # Test with test data  
 holt\_res <- accuracy(holt.fit, test)['Test set', 'MAPE']  
   
 result <- list(ses\_MAPE = ses\_res,   
 ses.p = ses.p,   
 holt\_MAPE = holt\_res,   
 holt.p=holt.p)  
   
 # plot  
 p <- autoplot(train) +  
 autolayer(ses.fit, series = "ses") +  
 autolayer(holt.fit, alpha = 0.4, series = "holt") +  
 autolayer(test, series = "test data") +  
 ggtitle(list\_vars[i])  
   
 print(p)  
   
 return(result)  
}  
  
for (i in seq\_along(myts)) {  
 result<- ses\_test(myts[[i]], i)  
 ses\_MAPE[i] <- result[1]  
 ses\_p[i] <- result[2]  
 holt\_MAPE[i] <- result[3]  
 holt\_p[i] <- result[4]  
}  
  
# Create the empty vectors  
arima\_MAPE <- vector(mode = "list", length = length(myts))  
  
# Create a function to test the ses and holt forecasts with test and train data  
arima\_test <- function(x, i) {  
   
 # Determine the index to split the time series into train and test sets  
 split.index <- floor(0.8 \* length(x)) # 80% for training, 20% for testing  
   
 # Split the time series into train and test sets  
 train <- window(x, end = split.index)  
 test <- window(x, start = split.index + 1)  
   
 # Set the horizon  
 horizon <- length(test)  
   
 #auto arima fit  
 arima.fc <- train %>%   
 auto.arima() %>%  
 forecast(h=horizon)  
   
 # test results  
 result <- accuracy(arima.fc, test)['Test set', 'MAPE']  
  
   
 p <- checkresiduals(arima.fc)  
  
 print(p)  
   
 return(result)  
}  
  
  
## Run   
arima\_MAPE <- lapply(myts, arima\_test)  
  
  
results\_df <- cbind(  
 list\_vars,  
 ses\_MAPE,  
 holt\_MAPE,  
 arima\_MAPE  
)  
# Print the results as a table  
results\_df # %>% flextable()