DATA 624 Spring 2019: Project-1

Ahmed Sajjad, Harpreet Shoker, Jagruti Solao, Chad Smith, Todd Weigel

March 19, 2019

#### Introduction

Data given is a de-identified Excel spreadsheet. Assignment is to perform the appropriate analysis to forecast several series for 140 periods.We are having 1622 periods for our analysis.We need to create model and forecast values for the following series.

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

Loading all the required libraries.

library(ggplot2)  
library(tidyr)

## Warning: package 'tidyr' was built under R version 3.4.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.2

library(gridExtra)

Read in data, which is just the Excel sheet saved as a CSV.

data <- read.csv("Project1data.csv",header=TRUE)  
data$SeriesInd <- as.Date(data$SeriesInd,origin = "1899-12-30")  
project\_in\_df <- data

As part of data cleaning lets remove forecast cells for now

project\_in\_df <- data[1:9732,]

Lets have a look at data

summary(project\_in\_df)

## SeriesInd group Var01 Var02 Var03 Var05 Var07   
## Min. :2011-05-06 S01:1622 Min. : 9.03 Min. : 1339900 Min. : 8.82 Min. : 8.99 Min. : 8.92   
## 1st Qu.:2012-12-10 S02:1622 1st Qu.: 23.10 1st Qu.: 12520675 1st Qu.: 22.59 1st Qu.: 22.91 1st Qu.: 22.88   
## Median :2014-07-25 S03:1622 Median : 38.44 Median : 21086550 Median : 37.66 Median : 38.05 Median : 38.05   
## Mean :2014-07-23 S04:1622 Mean : 46.98 Mean : 37035741 Mean : 46.12 Mean : 46.55 Mean : 46.56   
## 3rd Qu.:2016-03-01 S05:1622 3rd Qu.: 66.78 3rd Qu.: 42486700 3rd Qu.: 65.88 3rd Qu.: 66.38 3rd Qu.: 66.31   
## Max. :2017-10-13 S06:1622 Max. :195.18 Max. :480879500 Max. :189.36 Max. :195.00 Max. :189.72   
## NA's :14 NA's :2 NA's :26 NA's :26 NA's :26

From the above results can see missing values in all the variables,we need to impute NA's with some value. Var01 has 14,Var02 has 2,Var03 has 26,Var05 has 26 and Var07 has 26 missing values.

Here we will be imputing these values with the next value in sequence, e.g., if row 1600 is NA for Var01, take value from row 1601 hopefully, since these are stock values, the value from the next day for the few missing values we have, should be close enough

project\_in\_df[rowSums(is.na(project\_in\_df))>0,]

## SeriesInd group Var01 Var02 Var03 Var05 Var07  
## 118 2011-06-03 S06 NA NA NA NA NA  
## 4769 2014-07-01 S05 NA NA NA NA NA  
## 9217 2017-06-11 S03 NA 42343600 NA NA NA  
## 9218 2017-06-11 S02 NA 38160300 NA NA NA  
## 9219 2017-06-11 S01 NA 7329600 NA NA NA  
## 9220 2017-06-11 S06 NA 19885500 NA NA NA  
## 9221 2017-06-11 S05 NA 16610900 NA NA NA  
## 9222 2017-06-11 S04 NA 9098800 NA NA NA  
## 9223 2017-06-12 S03 NA 50074700 NA NA NA  
## 9224 2017-06-12 S02 NA 45801300 NA NA NA  
## 9225 2017-06-12 S01 NA 6121400 NA NA NA  
## 9226 2017-06-12 S06 NA 32570900 NA NA NA  
## 9227 2017-06-12 S05 NA 19331600 NA NA NA  
## 9228 2017-06-12 S04 NA 11188200 NA NA NA  
## 9637 2017-09-19 S03 95.43 32026000 NA NA NA  
## 9638 2017-09-19 S02 13.26 19465000 NA NA NA  
## 9639 2017-09-19 S01 58.83 6337000 NA NA NA  
## 9640 2017-09-19 S06 49.21 13222800 NA NA NA  
## 9641 2017-09-19 S05 90.40 13191900 NA NA NA  
## 9642 2017-09-19 S04 36.72 34330700 NA NA NA  
## 9643 2017-09-22 S03 97.19 38018600 NA NA NA  
## 9644 2017-09-22 S02 13.20 16234300 NA NA NA  
## 9645 2017-09-22 S01 59.28 3690900 NA NA NA  
## 9646 2017-09-22 S06 48.88 10644000 NA NA NA  
## 9647 2017-09-22 S05 89.90 11766100 NA NA NA  
## 9648 2017-09-22 S04 36.95 7785800 NA NA NA

while(nrow(project\_in\_df[rowSums(is.na(project\_in\_df))>0,]) > 0)  
{   
   
 project\_in\_df <- transmute(project\_in\_df,   
 SeriesInd = SeriesInd,group,  
 Var01 = if\_else(is.na(Var01), lead(Var01), Var01),  
 Var02 = if\_else(is.na(Var02), lead(Var02), Var02),  
 Var03 = if\_else(is.na(Var03), lead(Var03), Var03),  
 Var05 = if\_else(is.na(Var05), lead(Var05), Var05),  
 Var07 = if\_else(is.na(Var07), lead(Var07), Var07))  
 project\_in\_df[rowSums(is.na(project\_in\_df))>0,]  
}  
summary(project\_in\_df)

## SeriesInd group Var01 Var02 Var03 Var05 Var07   
## Min. :2011-05-06 S01:1622 Min. : 9.03 Min. : 1339900 Min. : 8.82 Min. : 8.99 Min. : 8.92   
## 1st Qu.:2012-12-10 S02:1622 1st Qu.: 23.30 1st Qu.: 12521025 1st Qu.: 22.65 1st Qu.: 23.01 1st Qu.: 23.01   
## Median :2014-07-25 S03:1622 Median : 38.52 Median : 21091200 Median : 37.83 Median : 38.24 Median : 38.21   
## Mean :2014-07-23 S04:1622 Mean : 47.04 Mean : 37035474 Mean : 46.24 Mean : 46.67 Mean : 46.68   
## 3rd Qu.:2016-03-01 S05:1622 3rd Qu.: 67.01 3rd Qu.: 42464900 3rd Qu.: 66.25 3rd Qu.: 66.72 3rd Qu.: 66.73   
## Max. :2017-10-13 S06:1622 Max. :195.18 Max. :480879500 Max. :189.36 Max. :195.00 Max. :189.72

From the above summary results we can see all Null values are replaced

# Create separate dataframes for each group

group\_S01\_df = filter(project\_in\_df, group == 'S01')  
group\_S02\_df = filter(project\_in\_df, group == 'S02')  
group\_S03\_df = filter(project\_in\_df, group == 'S03')  
group\_S04\_df = filter(project\_in\_df, group == 'S04')  
group\_S05\_df = filter(project\_in\_df, group == 'S05')  
group\_S06\_df = filter(project\_in\_df, group == 'S06')

Checking for the missing values in our dataset

group\_S01\_df[rowSums(is.na(group\_S01\_df))>0,]

## [1] SeriesInd group Var01 Var02 Var03 Var05 Var07   
## <0 rows> (or 0-length row.names)

group\_S02\_df[rowSums(is.na(group\_S02\_df))>0,]

## [1] SeriesInd group Var01 Var02 Var03 Var05 Var07   
## <0 rows> (or 0-length row.names)

group\_S03\_df[rowSums(is.na(group\_S03\_df))>0,]

## [1] SeriesInd group Var01 Var02 Var03 Var05 Var07   
## <0 rows> (or 0-length row.names)

group\_S04\_df[rowSums(is.na(group\_S04\_df))>0,]

## [1] SeriesInd group Var01 Var02 Var03 Var05 Var07   
## <0 rows> (or 0-length row.names)

group\_S05\_df[rowSums(is.na(group\_S05\_df))>0,]

## [1] SeriesInd group Var01 Var02 Var03 Var05 Var07   
## <0 rows> (or 0-length row.names)

group\_S06\_df[rowSums(is.na(group\_S06\_df))>0,]

## [1] SeriesInd group Var01 Var02 Var03 Var05 Var07   
## <0 rows> (or 0-length row.names)

# Select relevant columns for each group  
group\_S01\_df = select (group\_S01\_df, matches("SeriesInd|datetime|group|Var01|Var02"))  
group\_S02\_df = select (group\_S02\_df, matches("SeriesInd|datetime|group|Var02|Var03"))  
group\_S03\_df = select (group\_S03\_df, matches("SeriesInd|datetime|group|Var05|Var07"))  
group\_S04\_df = select (group\_S04\_df, matches("SeriesInd|datetime|group|Var01|Var02"))  
group\_S05\_df = select (group\_S05\_df, matches("SeriesInd|datetime|group|Var02|Var03"))  
group\_S06\_df = select (group\_S06\_df, matches("SeriesInd|datetime|group|Var05|Var07"))  
# Check number of rows  
print (c(nrow(group\_S01\_df), nrow(group\_S02\_df), nrow(group\_S03\_df), nrow(group\_S04\_df), nrow(group\_S05\_df), nrow(group\_S06\_df)))

## [1] 1622 1622 1622 1622 1622 1622

# Verify dataframes  
head(project\_in\_df, 20)

## SeriesInd group Var01 Var02 Var03 Var05 Var07  
## 1 2011-05-06 S03 30.64286 123432400 30.34000 30.49000 30.57286  
## 2 2011-05-06 S02 10.28000 60855800 10.05000 10.17000 10.28000  
## 3 2011-05-06 S01 26.61000 10369300 25.89000 26.20000 26.01000  
## 4 2011-05-06 S06 27.48000 39335700 26.82000 27.02000 27.32000  
## 5 2011-05-06 S05 69.26000 27809100 68.19000 68.72000 69.15000  
## 6 2011-05-06 S04 17.20000 16587400 16.88000 16.94000 17.10000  
## 7 2011-05-07 S03 30.79857 150476200 30.46428 30.65714 30.62571  
## 8 2011-05-07 S02 11.24000 215620200 10.40000 10.45000 10.96000  
## 9 2011-05-07 S01 26.30000 10943800 25.70000 25.95000 25.86000  
## 10 2011-05-07 S06 28.24000 55416000 27.24000 27.27000 28.07000  
## 11 2011-05-07 S05 69.45000 30174700 68.80000 69.19000 69.42000  
## 12 2011-05-07 S04 17.23000 11718100 17.00000 17.22000 17.23000  
## 13 2011-05-08 S03 30.74714 138040000 30.10714 30.62571 30.13857  
## 14 2011-05-08 S02 11.46000 200070600 11.13000 11.21000 11.37000  
## 15 2011-05-08 S01 26.03000 8933800 25.56000 25.90000 25.67000  
## 16 2011-05-08 S06 28.33000 33237000 27.79000 28.03000 28.11000  
## 17 2011-05-08 S05 70.60000 35044700 69.34000 69.45000 70.02000  
## 18 2011-05-08 S04 17.30000 16422000 17.07000 17.17000 17.17000  
## 19 2011-05-09 S03 30.28572 119282800 29.86429 30.25000 30.08286  
## 20 2011-05-09 S02 11.69000 130201700 11.32000 11.46000 11.66000

head(group\_S01\_df, 20)

## SeriesInd group Var01 Var02  
## 1 2011-05-06 S01 26.61 10369300  
## 2 2011-05-07 S01 26.30 10943800  
## 3 2011-05-08 S01 26.03 8933800  
## 4 2011-05-09 S01 25.84 10775400  
## 5 2011-05-10 S01 26.34 12875600  
## 6 2011-05-13 S01 26.49 11677000  
## 7 2011-05-14 S01 26.03 21165300  
## 8 2011-05-15 S01 25.16 18809200  
## 9 2011-05-16 S01 25.00 22908400  
## 10 2011-05-17 S01 24.77 20359100  
## 11 2011-05-21 S01 24.78 11783800  
## 12 2011-05-22 S01 24.61 13253500  
## 13 2011-05-23 S01 24.88 18407800  
## 14 2011-05-24 S01 24.17 21806400  
## 15 2011-05-27 S01 23.82 20192100  
## 16 2011-05-28 S01 23.90 19646700  
## 17 2011-05-29 S01 23.66 19535100  
## 18 2011-05-30 S01 23.51 20198600  
## 19 2011-05-31 S01 23.37 23137300  
## 20 2011-06-03 S01 23.08 13199000

head(group\_S02\_df, 20)

## SeriesInd group Var02 Var03  
## 1 2011-05-06 S02 60855800 10.05  
## 2 2011-05-07 S02 215620200 10.40  
## 3 2011-05-08 S02 200070600 11.13  
## 4 2011-05-09 S02 130201700 11.32  
## 5 2011-05-10 S02 130463000 11.46  
## 6 2011-05-13 S02 170626200 11.78  
## 7 2011-05-14 S02 162995900 11.72  
## 8 2011-05-15 S02 154527100 11.47  
## 9 2011-05-16 S02 116531200 11.51  
## 10 2011-05-17 S02 96149800 11.55  
## 11 2011-05-21 S02 65934000 11.46  
## 12 2011-05-22 S02 71649500 11.50  
## 13 2011-05-23 S02 121451400 11.01  
## 14 2011-05-24 S02 161530100 10.41  
## 15 2011-05-27 S02 121621500 10.61  
## 16 2011-05-28 S02 108250500 11.07  
## 17 2011-05-29 S02 105091600 11.22  
## 18 2011-05-30 S02 203320000 11.27  
## 19 2011-05-31 S02 159741200 10.70  
## 20 2011-06-03 S02 82748200 10.93

head(group\_S03\_df, 20)

## SeriesInd group Var05 Var07  
## 1 2011-05-06 S03 30.49000 30.57286  
## 2 2011-05-07 S03 30.65714 30.62571  
## 3 2011-05-08 S03 30.62571 30.13857  
## 4 2011-05-09 S03 30.25000 30.08286  
## 5 2011-05-10 S03 30.04286 30.28286  
## 6 2011-05-13 S03 30.40000 30.01571  
## 7 2011-05-14 S03 29.88428 29.67429  
## 8 2011-05-15 S03 29.69571 30.09286  
## 9 2011-05-16 S03 30.01571 29.91857  
## 10 2011-05-17 S03 30.13286 29.41857  
## 11 2011-05-21 S03 29.76143 30.72000  
## 12 2011-05-22 S03 30.70143 30.24714  
## 13 2011-05-23 S03 30.29714 29.72429  
## 14 2011-05-24 S03 29.54000 28.25000  
## 15 2011-05-27 S03 28.93000 29.01000  
## 16 2011-05-28 S03 29.42143 29.42000  
## 17 2011-05-29 S03 29.55000 29.69714  
## 18 2011-05-30 S03 29.27571 28.47000  
## 19 2011-05-31 S03 28.72571 27.43714  
## 20 2011-06-03 S03 27.48143 27.81857

head(group\_S04\_df, 20)

## SeriesInd group Var01 Var02  
## 1 2011-05-06 S04 17.20 16587400  
## 2 2011-05-07 S04 17.23 11718100  
## 3 2011-05-08 S04 17.30 16422000  
## 4 2011-05-09 S04 16.90 31816300  
## 5 2011-05-10 S04 16.76 15470000  
## 6 2011-05-13 S04 16.83 16181900  
## 7 2011-05-14 S04 16.86 15672400  
## 8 2011-05-15 S04 16.98 16955600  
## 9 2011-05-16 S04 17.23 16715600  
## 10 2011-05-17 S04 17.25 18415000  
## 11 2011-05-21 S04 16.96 15182600  
## 12 2011-05-22 S04 16.68 14419500  
## 13 2011-05-23 S04 16.58 21858400  
## 14 2011-05-24 S04 16.21 25132800  
## 15 2011-05-27 S04 16.11 19683700  
## 16 2011-05-28 S04 16.17 43979400  
## 17 2011-05-29 S04 16.49 41701000  
## 18 2011-05-30 S04 15.96 30159500  
## 19 2011-05-31 S04 15.67 39664600  
## 20 2011-06-03 S04 15.30 29865700

head(group\_S05\_df, 20)

## SeriesInd group Var02 Var03  
## 1 2011-05-06 S05 27809100 68.19  
## 2 2011-05-07 S05 30174700 68.80  
## 3 2011-05-08 S05 35044700 69.34  
## 4 2011-05-09 S05 27192100 69.42  
## 5 2011-05-10 S05 24891800 69.22  
## 6 2011-05-13 S05 30685000 69.65  
## 7 2011-05-14 S05 31496700 69.52  
## 8 2011-05-15 S05 24884400 69.26  
## 9 2011-05-16 S05 18630800 69.35  
## 10 2011-05-17 S05 29411900 68.65  
## 11 2011-05-21 S05 26081900 68.42  
## 12 2011-05-22 S05 34629500 67.93  
## 13 2011-05-23 S05 39114000 66.50  
## 14 2011-05-24 S05 39085500 66.00  
## 15 2011-05-27 S05 29305100 65.69  
## 16 2011-05-28 S05 34083300 65.50  
## 17 2011-05-29 S05 35723500 65.00  
## 18 2011-05-30 S05 37349800 64.57  
## 19 2011-05-31 S05 40880500 64.02  
## 20 2011-06-03 S05 37703000 65.35

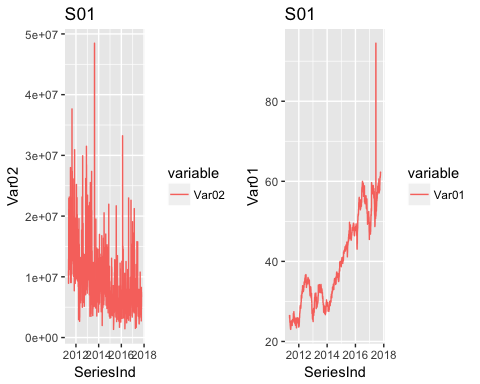
head(group\_S06\_df, 20)

## SeriesInd group Var05 Var07  
## 1 2011-05-06 S06 27.02 27.32  
## 2 2011-05-07 S06 27.27 28.07  
## 3 2011-05-08 S06 28.03 28.11  
## 4 2011-05-09 S06 28.12 29.13  
## 5 2011-05-10 S06 28.90 28.86  
## 6 2011-05-13 S06 29.09 28.80  
## 7 2011-05-14 S06 28.47 28.08  
## 8 2011-05-15 S06 27.99 28.58  
## 9 2011-05-16 S06 28.50 28.99  
## 10 2011-05-17 S06 28.82 28.08  
## 11 2011-05-21 S06 27.93 28.28  
## 12 2011-05-22 S06 28.02 27.82  
## 13 2011-05-23 S06 27.98 28.00  
## 14 2011-05-24 S06 27.97 27.26  
## 15 2011-05-27 S06 27.45 27.66  
## 16 2011-05-28 S06 27.45 26.99  
## 17 2011-05-29 S06 26.90 28.20  
## 18 2011-05-30 S06 28.28 28.45  
## 19 2011-05-31 S06 28.44 28.43  
## 20 2011-06-03 S06 65.77 66.18

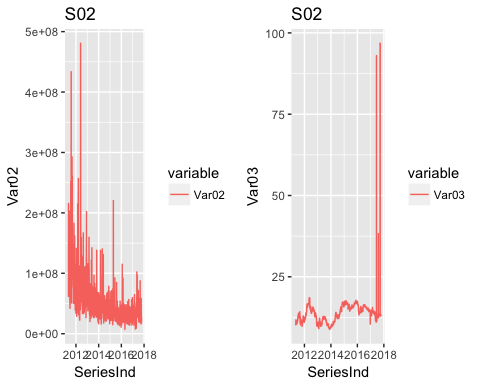
#### Exploratory Data Analysis

Lets start looking at the data. First we are making line plots for all the groups and variables we will be forecasting.

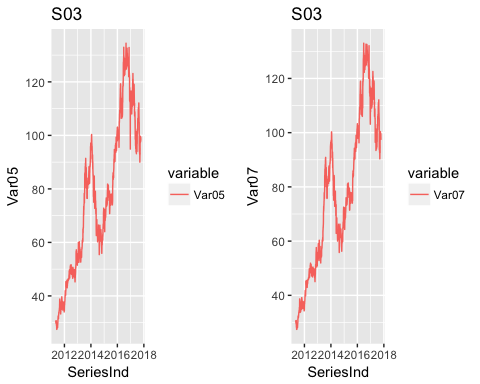
p1 <- ggplot(group\_S01\_df, aes(SeriesInd, y = Var02, color = variable)) +   
 ggtitle("S01") +  
 geom\_line(aes(y = Var02, col = "Var02"))  
  
p2 <- ggplot(group\_S01\_df, aes(SeriesInd, y = Var01, color = variable)) +   
 ggtitle("S01") +   
 geom\_line(aes(y = Var01, col = "Var01"))  
  
p3 <- ggplot(group\_S02\_df, aes(SeriesInd, y = Var02,color = variable))+  
 ggtitle("S02") +   
 geom\_line(aes(y = Var02, col = "Var02"))  
  
  
p4 <- ggplot(group\_S02\_df, aes(SeriesInd, y = Var03,color = variable))+  
 ggtitle("S02") +   
 geom\_line(aes(y = Var03, col = "Var03"))  
  
p5 <- ggplot(group\_S03\_df, aes(SeriesInd, y = Var05, color = variable)) +   
 geom\_line(aes(y = Var05, col = "Var05")) +  
 ggtitle("S03")   
  
p6 <- ggplot(group\_S03\_df, aes(SeriesInd, y = Var07, color = variable)) +   
 geom\_line(aes(y = Var07, col = "Var07")) +  
 ggtitle("S03")   
   
p7 <- ggplot(group\_S04\_df, aes(SeriesInd, y = Var01, color = variable)) +   
 geom\_line(aes(y = Var01, col = "Var01")) +   
 ggtitle("S04")   
p8 <- ggplot(group\_S04\_df, aes(SeriesInd, y = Var02,color = variable))+  
 ggtitle("S04") +   
 geom\_line(aes(y = Var02, col = "Var02"))  
  
  
p9 <- ggplot(group\_S05\_df, aes(SeriesInd, y = Var03, color = variable)) +   
 ggtitle("S05") +  
 geom\_line(aes(y = Var03, col = "Var03"))  
  
p10 <- ggplot(group\_S05\_df, aes(SeriesInd, y = Var02,color = variable))+  
 ggtitle("S05") +   
 geom\_line(aes(y = Var02, col = "Var02"))  
  
p11 <- ggplot(group\_S06\_df, aes(SeriesInd, y = Var05, color = variable)) +   
 geom\_line(aes(y = Var05, col = "Var05")) +  
 ggtitle("S06")  
  
p12 <- ggplot(group\_S06\_df, aes(SeriesInd, y = Var07, color = variable)) +   
 ggtitle("S06") +  
 geom\_line(aes(y = Var07, col = "Var07"))  
grid.arrange(p1, p2, nrow = 1)



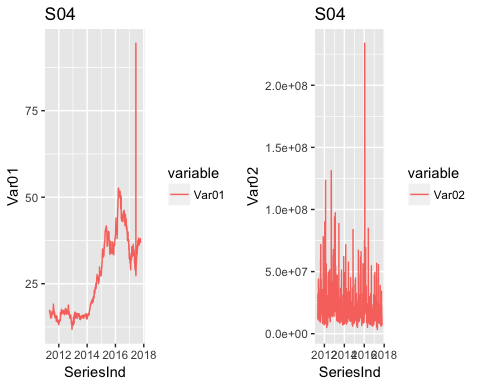
grid.arrange(p3, p4, nrow = 1)



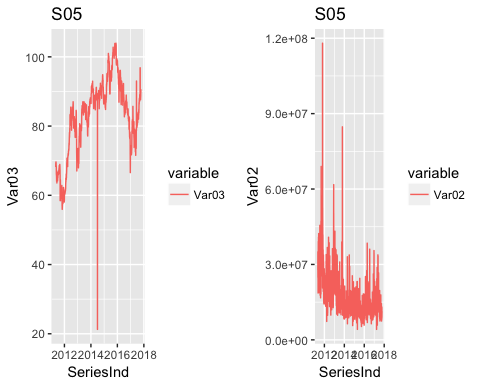
grid.arrange(p5, p6, nrow = 1)



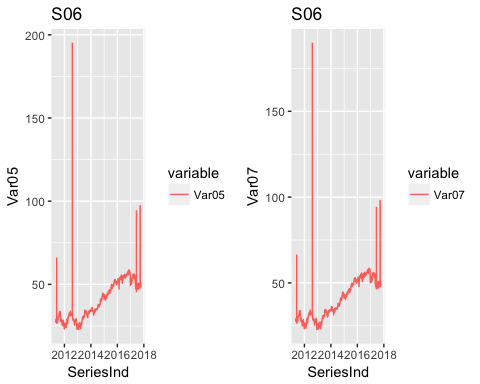
grid.arrange(p7,p8, nrow = 1)



grid.arrange(p9,p10, nrow = 1)



grid.arrange(p11,p12, nrow = 1)



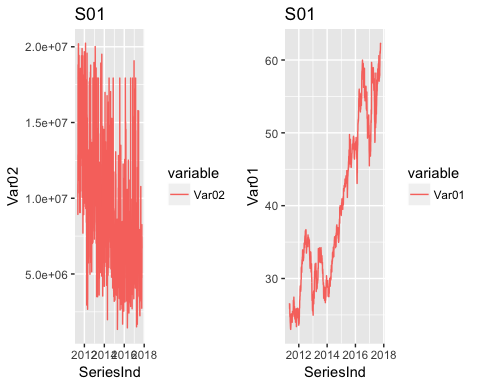
Comparing to other variable Var02 seems to be noisy than any other variable and having outliers. These outliers needs to be fixed before producing forecasts S03 and S06 variables Var05 and Var07 seems to be quite similar Also we see in S02 - Var03 and S06-Var05 and Var07 plot some outlier values that also needs to be fixed before forecasting. We can also observe some seasonality and trend pattern in Var01,Var03,Var05 and var07

Removing outliers We are using here IQR to fix the outliers in our data For missing values that lie outside the 1.5\*IQR limits, we are capping it by replacing those observations outside the lower limit with the value of 5th %ile and those that lie above the upper limit, with the value of 95th %ile.

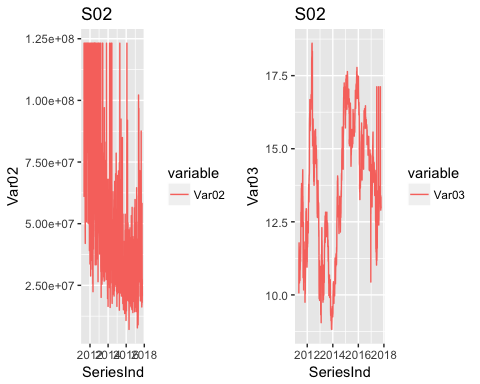
Remove\_Outlier <- function(x){  
 qnt <- quantile(x, probs=c(.25, .75), na.rm = T)  
 caps <- quantile(x, probs=c(.05, .95), na.rm = T)  
 H <- 1.5 \* IQR(x, na.rm = T)  
 x[x < (qnt[1] - H)] <- caps[1]  
 x[x > (qnt[2] + H)] <- caps[2]  
 return(x)  
}  
group\_S06\_df$Var07[which.max(group\_S06\_df$Var07)] = mean(group\_S06\_df$Var07, na.rm=TRUE)  
group\_S06\_df$Var05[which.max(group\_S06\_df$Var05)] = mean(group\_S06\_df$Var05, na.rm=TRUE)  
group\_S06\_df$Var07[which.max(group\_S06\_df$Var07)] = mean(group\_S06\_df$Var07, na.rm=TRUE)  
group\_S05\_df$Var03[which.max(group\_S05\_df$Var03)] = mean(group\_S05\_df$Var03, na.rm=TRUE)  
group\_S05\_df$Var03[which.max(group\_S05\_df$Var03)] = mean(group\_S05\_df$Var03, na.rm=TRUE)  
group\_S01\_df$Var01=Remove\_Outlier(group\_S01\_df$Var01)  
group\_S01\_df$Var02=Remove\_Outlier(group\_S01\_df$Var02)  
group\_S02\_df$Var03=Remove\_Outlier(group\_S02\_df$Var03)  
group\_S02\_df$Var02=Remove\_Outlier(group\_S02\_df$Var02)  
group\_S04\_df$Var01=Remove\_Outlier(group\_S04\_df$Var01)  
group\_S04\_df$Var02=Remove\_Outlier(group\_S04\_df$Var02)  
group\_S05\_df$Var03=Remove\_Outlier(group\_S05\_df$Var03)  
group\_S05\_df$Var02=Remove\_Outlier(group\_S05\_df$Var02)  
group\_S06\_df$Var05=Remove\_Outlier(group\_S06\_df$Var05)  
group\_S06\_df$Var07=Remove\_Outlier(group\_S06\_df$Var07)

Lets plot again plots for the variables to check if the outliers are fixed

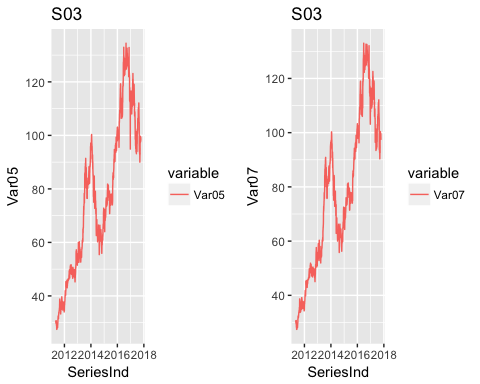
p1 <- ggplot(group\_S01\_df, aes(SeriesInd, y = Var02, color = variable)) +   
 ggtitle("S01") +  
 geom\_line(aes(y = Var02, col = "Var02"))  
  
p2 <- ggplot(group\_S01\_df, aes(SeriesInd, y = Var01, color = variable)) +   
 ggtitle("S01") +   
 geom\_line(aes(y = Var01, col = "Var01"))  
  
p3 <- ggplot(group\_S02\_df, aes(SeriesInd, y = Var02,color = variable))+  
 ggtitle("S02") +   
 geom\_line(aes(y = Var02, col = "Var02"))  
  
  
p4 <- ggplot(group\_S02\_df, aes(SeriesInd, y = Var03,color = variable))+  
 ggtitle("S02") +   
 geom\_line(aes(y = Var03, col = "Var03"))  
  
p5 <- ggplot(group\_S03\_df, aes(SeriesInd, y = Var05, color = variable)) +   
 geom\_line(aes(y = Var05, col = "Var05")) +  
 ggtitle("S03")   
  
p6 <- ggplot(group\_S03\_df, aes(SeriesInd, y = Var07, color = variable)) +   
 geom\_line(aes(y = Var07, col = "Var07")) +  
 ggtitle("S03")   
   
p7 <- ggplot(group\_S04\_df, aes(SeriesInd, y = Var01, color = variable)) +   
 geom\_line(aes(y = Var01, col = "Var01")) +   
 ggtitle("S04")   
p8 <- ggplot(group\_S04\_df, aes(SeriesInd, y = Var02,color = variable))+  
 ggtitle("S04") +   
 geom\_line(aes(y = Var02, col = "Var02"))  
  
  
p9 <- ggplot(group\_S05\_df, aes(SeriesInd, y = Var03, color = variable)) +   
 ggtitle("S05") +  
 geom\_line(aes(y = Var03, col = "Var03"))  
  
p10 <- ggplot(group\_S05\_df, aes(SeriesInd, y = Var02,color = variable))+  
 ggtitle("S05") +   
 geom\_line(aes(y = Var02, col = "Var02"))  
  
p11 <- ggplot(group\_S06\_df, aes(SeriesInd, y = Var05, color = variable)) +   
 geom\_line(aes(y = Var05, col = "Var05")) +  
 ggtitle("S06")  
  
p12 <- ggplot(group\_S06\_df, aes(SeriesInd, y = Var07, color = variable)) +   
 ggtitle("S06") +  
 geom\_line(aes(y = Var07, col = "Var07"))  
grid.arrange(p1, p2, nrow = 1)



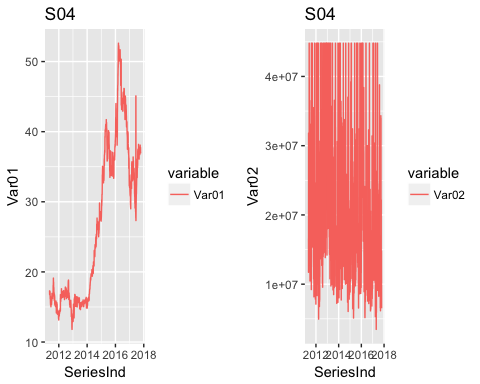
grid.arrange(p3, p4, nrow = 1)



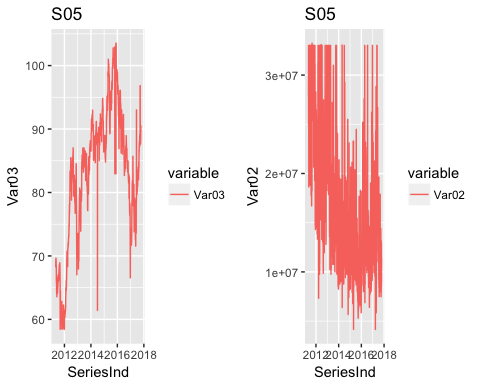
grid.arrange(p5, p6, nrow = 1)



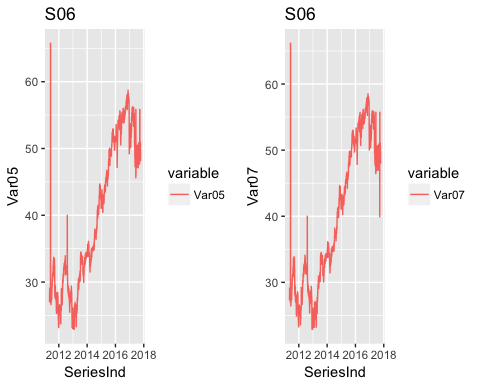
grid.arrange(p7,p8, nrow = 1)



grid.arrange(p9,p10, nrow = 1)



grid.arrange(p11,p12, nrow = 1)



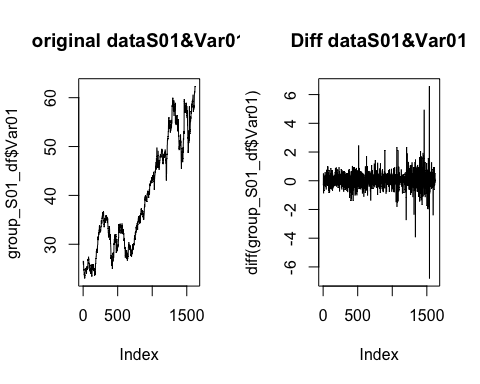
The plots looks it has removed the extreme outliers.

The plots for Var01,Var03,Var05,Var07 shows increase in general during the period of time. But there is no obvious pattern in the fluctuation. In other words, there is no seasonality, but an obvious upward trend. Also, the variance is not stable seeing from the plots and it seems to increase. Thus, we may use difference and logarithm or square root transformation on original data to stabilize the variance.

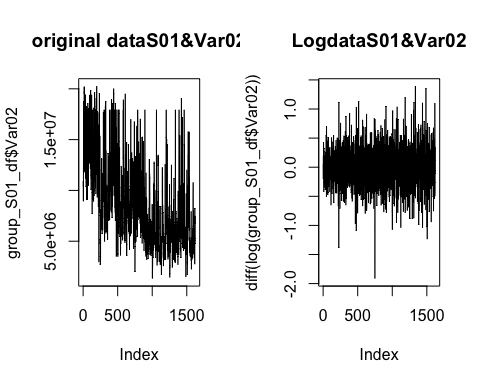
We are using here difference of each value over previous value and difference logarithm transformation for Var02

Lets create plots to have compare how the orignal data and transformed data

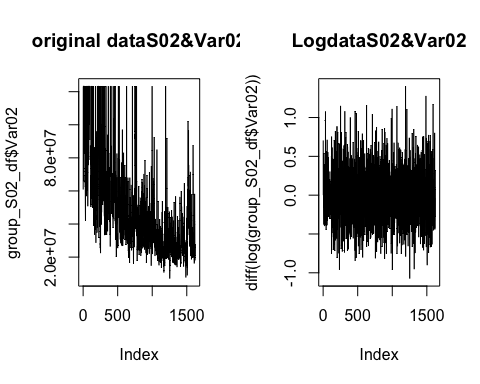
par(mfrow=c(1,2))  
p1 <- plot(group\_S01\_df$Var01, type = "l", main = "original dataS01&Var01")  
p2 <- plot(diff(group\_S01\_df$Var01), type = "l", main = "Diff dataS01&Var01")



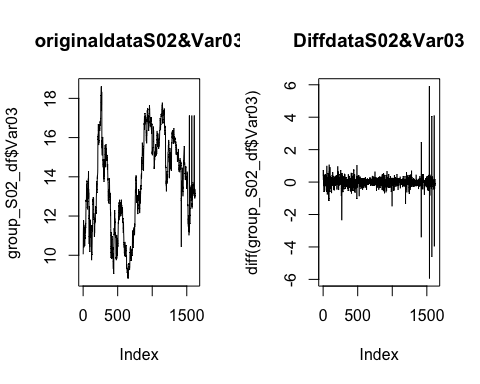
p3 <- plot(group\_S01\_df$Var02, type = "l", main = "original dataS01&Var02")  
p4 <- plot(diff(log(group\_S01\_df$Var02)), type = "l", main = "LogdataS01&Var02")



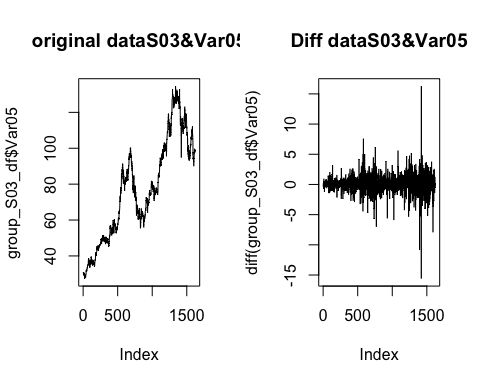
p5 <- plot(group\_S02\_df$Var02, type = "l", main = "original dataS02&Var02")  
p6 <- plot(diff(log(group\_S02\_df$Var02)), type = "l", main = "LogdataS02&Var02")



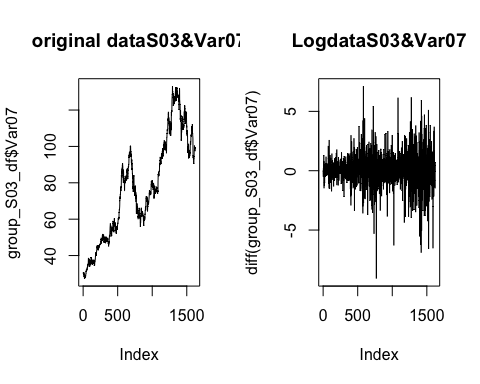
p7 <- plot(group\_S02\_df$Var03, type = "l", main = "originaldataS02&Var03")  
p8 <- plot(diff(group\_S02\_df$Var03), type = "l", main = "DiffdataS02&Var03")



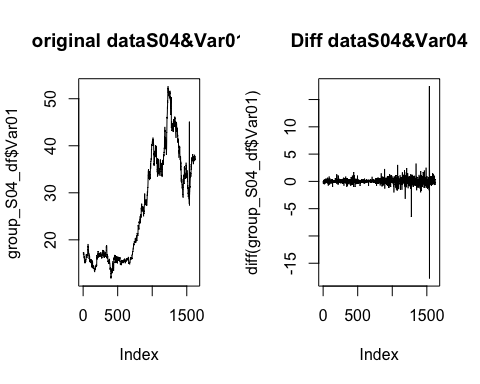
p9 <- plot(group\_S03\_df$Var05, type = "l", main = "original dataS03&Var05")  
p10 <- plot(diff(group\_S03\_df$Var05), type = "l", main = "Diff dataS03&Var05")



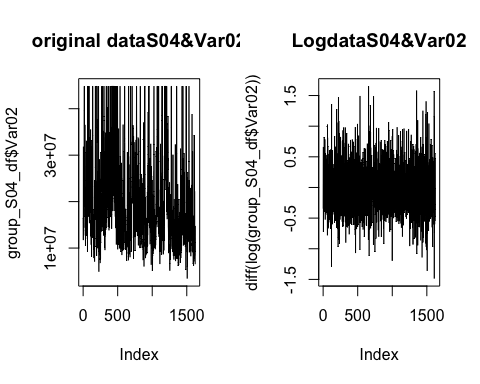
p11 <- plot(group\_S03\_df$Var07, type = "l", main = "original dataS03&Var07")  
p12 <- plot(diff(group\_S03\_df$Var07), type = "l", main = "LogdataS03&Var07")



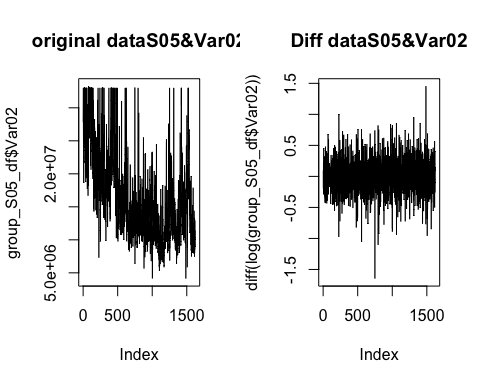
p13 <- plot(group\_S04\_df$Var01, type = "l", main = "original dataS04&Var01")  
p14 <- plot(diff(group\_S04\_df$Var01), type = "l", main = "Diff dataS04&Var04")



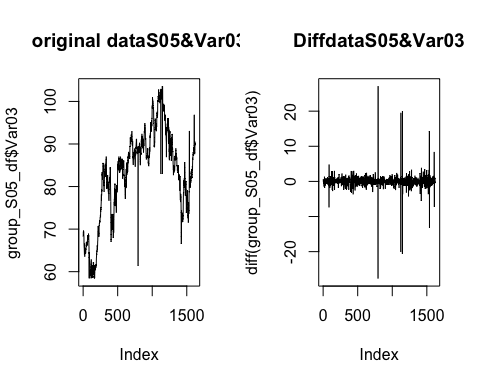
p15 <- plot(group\_S04\_df$Var02, type = "l", main = "original dataS04&Var02")  
p16 <- plot(diff(log(group\_S04\_df$Var02)), type = "l", main = "LogdataS04&Var02")



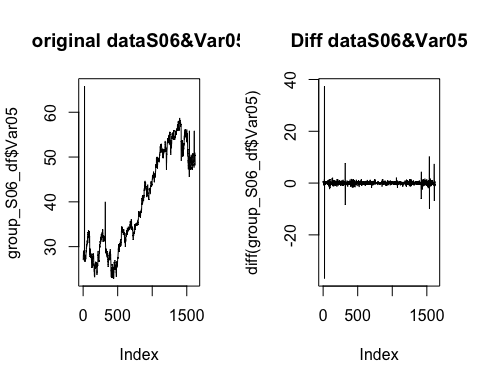
p17 <- plot(group\_S05\_df$Var02, type = "l", main = "original dataS05&Var02")  
p18 <- plot(diff(log(group\_S05\_df$Var02)), type = "l", main = "Diff dataS05&Var02")



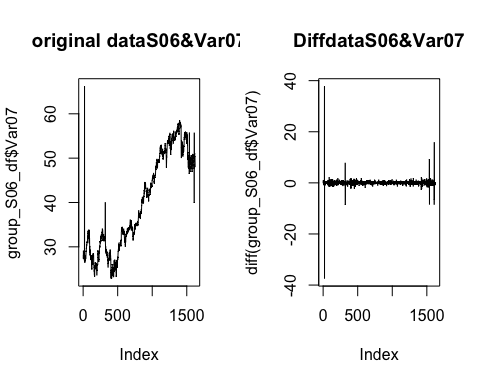
p19 <- plot(group\_S05\_df$Var03, type = "l", main = "original dataS05&Var03")  
p20 <- plot(diff(group\_S05\_df$Var03), type = "l", main = "DiffdataS05&Var03")



p21 <- plot(group\_S06\_df$Var05, type = "l", main = "original dataS06&Var05")  
p22 <- plot(diff(group\_S06\_df$Var05), type = "l", main = "Diff dataS06&Var05")



p23 <- plot(group\_S06\_df$Var07, type = "l", main = "original dataS06&Var07")  
p24 <- plot(diff(group\_S06\_df$Var07), type = "l", main = "DiffdataS06&Var07")



Data looks like it eliminated noise compared to the orginal. Data looks more stationary after applying differencing to Var01,Var03,Var05,Var07 and applying logarithm Var02.

#### Cleanup Dataset

#### Build Model

#### Evaluate Model

#### Group S01 Forecast

#### Group S02 Forecast

#### Group S03 Forecast

#### Group S04 Forecast

#### Group S05 Forecast

#### Group S06 Forecast