Data624 - Project 1

Esteban Aramayo, Coffy Andrews-Guo, LeTicia Cancel, Joseph Connolly, Ian Costello

21 June, 2022

Table of Contents

Required Libraries

#libraries  
library(readxl)  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.8  
## v tidyr 1.2.0 v stringr 1.4.0  
## v readr 2.1.2 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggplot2)  
library(patchwork)  
library(fpp2)

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

## -- Attaching packages ---------------------------------------------- fpp2 2.4 --

## v forecast 8.16 v expsmooth 2.3   
## v fma 2.4

##

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

#library(RANN)  
#library(VIM)

# Project Summary

De-identified data was provided to conduct a series of six forecasts of different variables of a provided data set. There are two major requirements of this project:

1. This written report
2. The forecasts and error rates

df <- read\_excel("data.xls")  
head(df)

## # A tibble: 6 x 7  
## SeriesInd category Var01 Var02 Var03 Var05 Var07  
## <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 40669 S03 30.6 123432400 30.3 30.5 30.6  
## 2 40669 S02 10.3 60855800 10.0 10.2 10.3  
## 3 40669 S01 26.6 10369300 25.9 26.2 26.0  
## 4 40669 S06 27.5 39335700 26.8 27.0 27.3  
## 5 40669 S05 69.3 27809100 68.2 68.7 69.2  
## 6 40669 S04 17.2 16587400 16.9 16.9 17.1

## Data Exploration

The data set does not appear to have any distinguishing labels that would indicate anything about the source or purpose of the data set. Under normal circumstances, context about data and use case is important for forecasting. Context may help identify the kinds of methods and models that would be best suited to produce an accurate result - following the “no free lunch” principle.

Starting with some data exploration using summary()

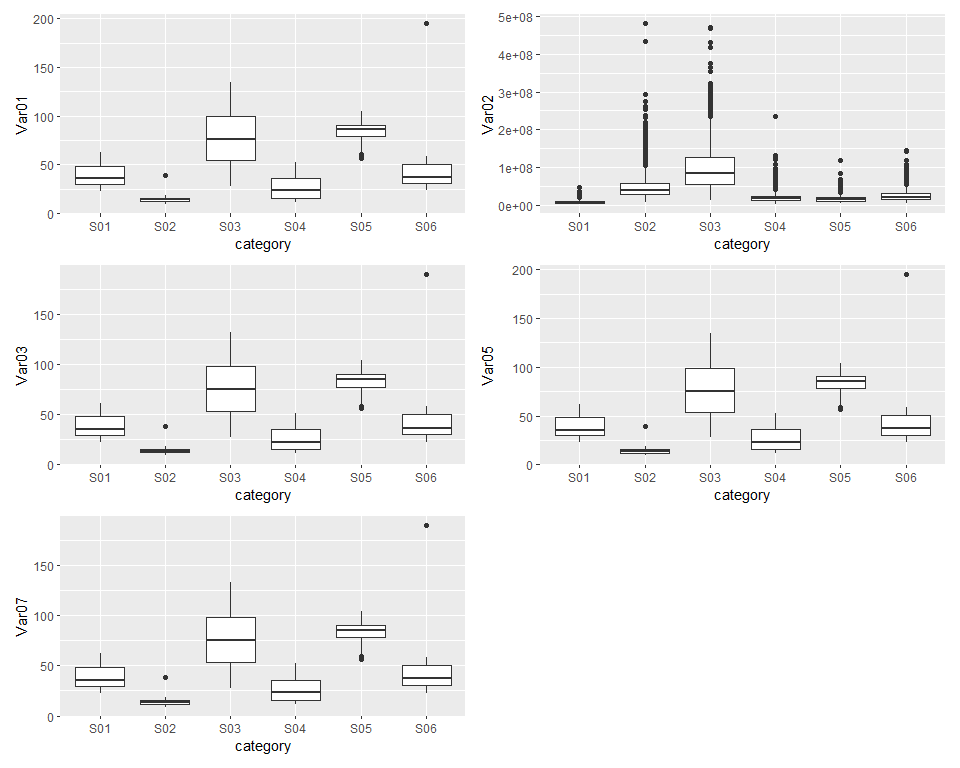
each predictor has NA values but I’m not sure what proportion of the data is NA yet. Column Var02’s values differ the most when compared to the other Var## columns. So we will see how that impacts the data.

summary(df)

## SeriesInd category Var01 Var02   
## Min. :40669 Length:10572 Min. : 9.03 Min. : 1339900   
## 1st Qu.:41303 Class :character 1st Qu.: 23.10 1st Qu.: 12520675   
## Median :41946 Mode :character Median : 38.44 Median : 21086550   
## Mean :41945 Mean : 46.98 Mean : 37035741   
## 3rd Qu.:42587 3rd Qu.: 66.78 3rd Qu.: 42486700   
## Max. :43221 Max. :195.18 Max. :480879500   
## NA's :854 NA's :842   
## Var03 Var05 Var07   
## Min. : 8.82 Min. : 8.99 Min. : 8.92   
## 1st Qu.: 22.59 1st Qu.: 22.91 1st Qu.: 22.88   
## Median : 37.66 Median : 38.05 Median : 38.05   
## Mean : 46.12 Mean : 46.55 Mean : 46.56   
## 3rd Qu.: 65.88 3rd Qu.: 66.38 3rd Qu.: 66.31   
## Max. :189.36 Max. :195.00 Max. :189.72   
## NA's :866 NA's :866 NA's :866

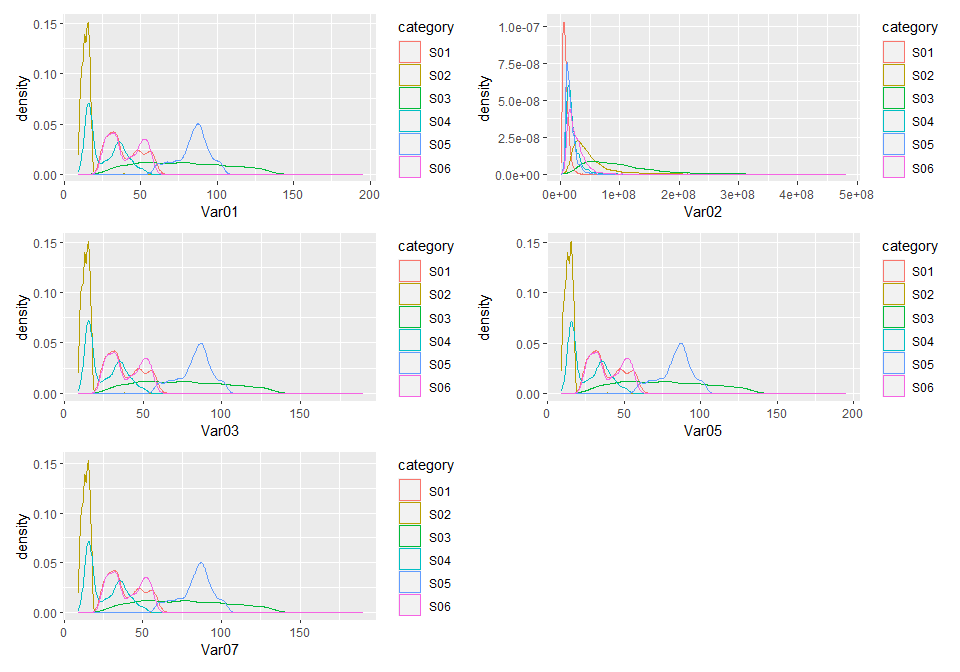
Boxplots: Var02 has the most outliers but this is also the column with the largest range in values

p1 <- ggplot(df, aes(category, Var01)) +  
 geom\_boxplot()  
p2 <- ggplot(df, aes(category, Var02)) +  
 geom\_boxplot()  
p3 <- ggplot(df, aes(category, Var03)) +  
 geom\_boxplot()  
p4 <- ggplot(df, aes(category, Var05)) +  
 geom\_boxplot()  
p5 <- ggplot(df, aes(category, Var07)) +  
 geom\_boxplot()  
  
p1+p2+p3+p4+p5+  
 plot\_layout(ncol = 2)



All predictors are left skewed so transformations are needed for each column

p1 <- ggplot(df, aes(Var01, color=category)) +  
 geom\_density()  
p2 <- ggplot(df, aes(Var02, color=category)) +  
 geom\_density()  
p3 <- ggplot(df, aes(Var03, color=category)) +  
 geom\_density()  
p4 <- ggplot(df, aes(Var05, color=category)) +  
 geom\_density()  
p5 <- ggplot(df, aes(Var07, color=category)) +  
 geom\_density()  
  
p1+p2+p3+p4+p5+  
 plot\_layout(ncol = 2)



The NAs make up less than 10% of all values in the data.

preProcess\_missingdata\_model <- preProcess(as.data.frame(df), method='knnImpute')  
df <- predict(preProcess\_missingdata\_model, newdata = df)  
anyNA(df)

## Data transformations

Var02 is the most skewed

library(moments)  
skewness(df$Var01, na.rm = TRUE)

## [1] 0.7387325

skewness(df$Var02, na.rm = TRUE)

## [1] 3.037223

skewness(df$Var03, na.rm = TRUE)

## [1] 0.7379661

skewness(df$Var05, na.rm = TRUE)

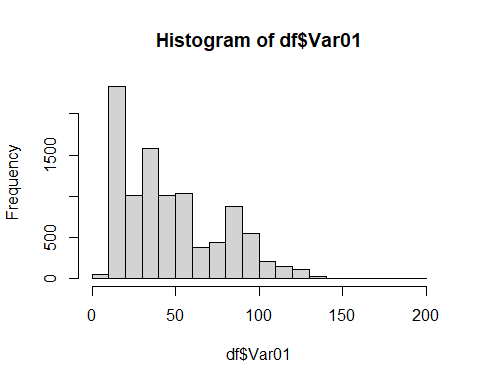
## [1] 0.7405568

skewness(df$Var07, na.rm = TRUE)

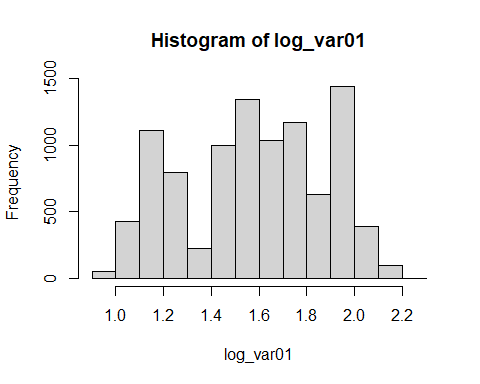
## [1] 0.7376865

Experimenting with 3 different transformations; log, square root, and cube root. The log transformation looks the most normalized

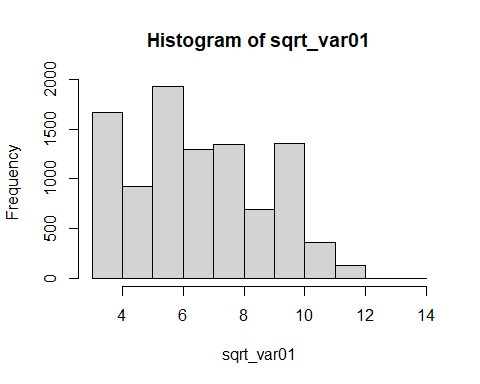
log\_var01 <- log10(df$Var01)  
sqrt\_var01 <- sqrt(df$Var01)  
cube\_var01 <- df$Var01^(1/3)  
  
hist(df$Var01)



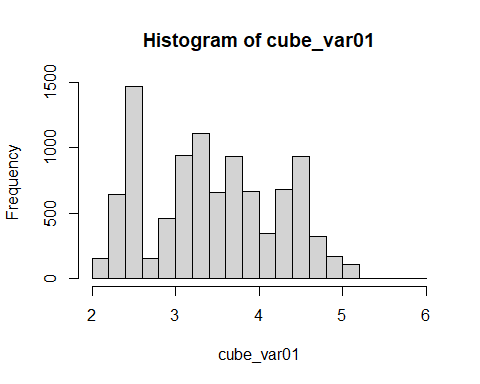
hist(log\_var01)



hist(sqrt\_var01)



hist(cube\_var01)

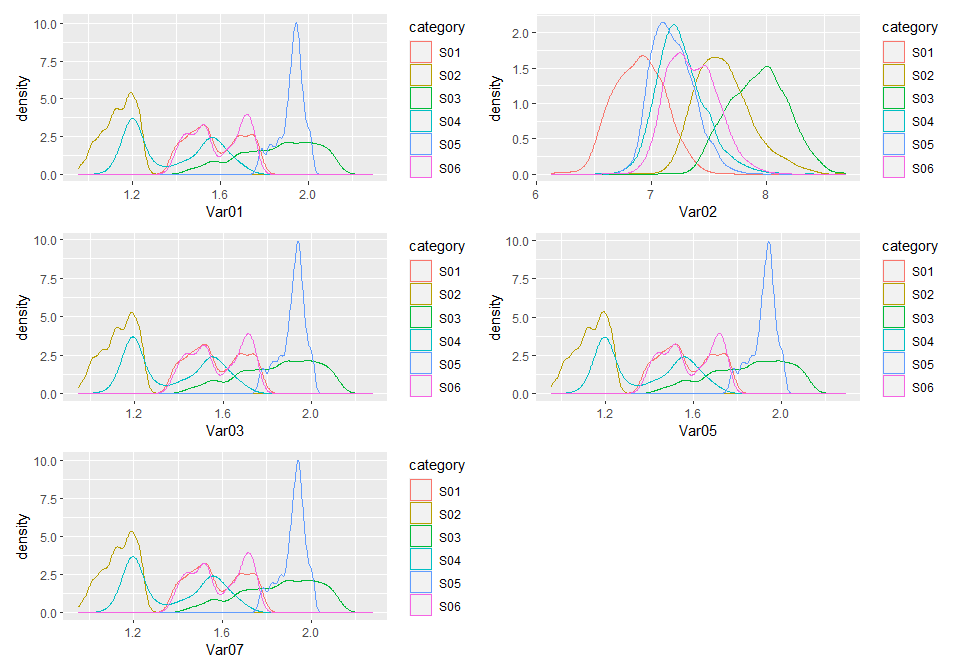


Transform all predictors using log transformation

df\_transformed <- df  
df\_transformed$Var01 <- log10(df$Var01)  
df\_transformed$Var02 <- log10(df$Var02)  
df\_transformed$Var03 <- log10(df$Var03)  
df\_transformed$Var05 <- log10(df$Var05)  
df\_transformed$Var07 <- log10(df$Var07)

New plots using transformed dataframe. Var02 is the most normalize so I will try different transformations with the other columns

p1 <- ggplot(df\_transformed, aes(Var01, color=category)) +  
 geom\_density()  
p2 <- ggplot(df\_transformed, aes(Var02, color=category)) +  
 geom\_density()  
p3 <- ggplot(df\_transformed, aes(Var03, color=category)) +  
 geom\_density()  
p4 <- ggplot(df\_transformed, aes(Var05, color=category)) +  
 geom\_density()  
p5 <- ggplot(df\_transformed, aes(Var07, color=category)) +  
 geom\_density()  
  
p1+p2+p3+p4+p5+  
 plot\_layout(ncol = 2)



s01 <- data %>% filter(group == "S01")  
s02 <- data %>% filter(group == "S02")  
s03 <- data %>% filter(group == "S03")  
s04 <- data %>% filter(group == "S04")  
s05 <- data %>% filter(group == "S05")  
s06 <- data %>% filter(group == "S06")