BimboInventoryDemand

Juliana Yamauti

14/10/2020

# Grupo Bimbo Inventory Demand

Dataset: <https://www.kaggle.com/c/grupo-bimbo-inventory-demand>

### The goal in this project is to create a develop a model to accurately forecast inventory demand based on historical sales data

Loading necessary packages:

library(data.table)  
library(dplyr)  
library(caret)  
library(ggplot2)  
library(reshape2)  
library(MLmetrics)

Loading aditional datasets:

df\_cliente <- fread("cliente\_tabla.csv", header = TRUE, sep = ",", encoding = "UTF-8")  
head(df\_cliente)

## Cliente\_ID NombreCliente  
## 1: 0 SIN NOMBRE  
## 2: 1 OXXO XINANTECATL  
## 3: 2 SIN NOMBRE  
## 4: 3 EL MORENO  
## 5: 4 SDN SER DE ALIM CUERPO SA CIA DE INT  
## 6: 4 SDN SER DE ALIM CUERPO SA CIA DE INT

dim(df\_cliente)

## [1] 935362 2

df\_produto <- fread("producto\_tabla.csv", header = TRUE, sep = ",", encoding = "UTF-8")  
head(df\_produto)

## Producto\_ID NombreProducto  
## 1: 0 NO IDENTIFICADO 0  
## 2: 9 Capuccino Moka 750g NES 9  
## 3: 41 Bimbollos Ext sAjonjoli 6p 480g BIM 41  
## 4: 53 Burritos Sincro 170g CU LON 53  
## 5: 72 Div Tira Mini Doradita 4p 45g TR 72  
## 6: 73 Pan Multigrano Linaza 540g BIM 73

dim(df\_produto)

## [1] 2592 2

df\_town <- fread("town\_state.csv", header = TRUE, sep = ",", encoding = "UTF-8")  
head(df\_town)

## Agencia\_ID Town State  
## 1: 1110 2008 AG. LAGO FILT MÉXICO, D.F.  
## 2: 1111 2002 AG. AZCAPOTZALCO MÉXICO, D.F.  
## 3: 1112 2004 AG. CUAUTITLAN ESTADO DE MÉXICO  
## 4: 1113 2008 AG. LAGO FILT MÉXICO, D.F.  
## 5: 1114 2029 AG.IZTAPALAPA 2 MÉXICO, D.F.  
## 6: 1116 2011 AG. SAN ANTONIO MÉXICO, D.F.

dim(df\_town)

## [1] 790 3

Loading train dataset:

df\_train <- fread("train.csv", header = TRUE, sep = ",", encoding = "UTF-8")  
head(df\_train)

## Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID Venta\_uni\_hoy  
## 1: 3 1110 7 3301 15766 1212 3  
## 2: 3 1110 7 3301 15766 1216 4  
## 3: 3 1110 7 3301 15766 1238 4  
## 4: 3 1110 7 3301 15766 1240 4  
## 5: 3 1110 7 3301 15766 1242 3  
## 6: 3 1110 7 3301 15766 1250 5  
## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil  
## 1: 25.14 0 0 3  
## 2: 33.52 0 0 4  
## 3: 39.32 0 0 4  
## 4: 33.52 0 0 4  
## 5: 22.92 0 0 3  
## 6: 38.20 0 0 5

dim(df\_train)

## [1] 74180464 11

df\_train dataset has 74.180.464 observations and 11 variables. Since the dataset is too big, we’re going to get a 100.000 rows’ sample

df\_sample <- sample\_n(df\_train, size = 100000)  
dim(df\_sample)

## [1] 100000 11

# Removing df\_train object  
rm(df\_train)  
  
# Saving the sample into "AmostraBimbo.csv" so we don't have to load train dataset again  
write.csv(df\_sample, "AmostraBimbo.csv")  
  
# Reading the sample file  
df\_sample <- fread("AmostraBimbo.csv", header = TRUE, sep = ",", encoding = "UTF-8")  
head(df\_sample)

## V1 Semana Agencia\_ID Canal\_ID Ruta\_SAK Cliente\_ID Producto\_ID Venta\_uni\_hoy  
## 1: 1 6 1636 1 1112 1106211 3270 2  
## 2: 2 8 1625 1 1292 422131 1109 6  
## 3: 3 5 1330 1 1264 204979 41938 1  
## 4: 4 4 1350 1 8011 1198764 1232 2  
## 5: 5 9 3214 1 1607 597550 303 3  
## 6: 6 3 1602 1 1201 1326576 3631 2  
## Venta\_hoy Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil  
## 1: 20.94 0 0.00 2  
## 2: 90.06 1 15.01 5  
## 3: 9.91 0 0.00 1  
## 4: 36.48 0 0.00 2  
## 5: 13.62 0 0.00 3  
## 6: 32.70 0 0.00 2

# Removing column #1 with row number  
df\_sample$V1 <- NULL  
  
# Convert df\_sample to dataframe  
class(df\_sample)

## [1] "data.table" "data.frame"

df\_sample <- as.data.frame(df\_sample)

## EDA - Exploratory Data Analysis

# Checking dataset statistics  
summary(df\_sample)

## Semana Agencia\_ID Canal\_ID Ruta\_SAK   
## Min. :3.000 Min. : 1110 Min. : 1.000 Min. : 1   
## 1st Qu.:4.000 1st Qu.: 1311 1st Qu.: 1.000 1st Qu.:1162   
## Median :6.000 Median : 1613 Median : 1.000 Median :1286   
## Mean :5.947 Mean : 2513 Mean : 1.384 Mean :2117   
## 3rd Qu.:8.000 3rd Qu.: 2036 3rd Qu.: 1.000 3rd Qu.:2803   
## Max. :9.000 Max. :25759 Max. :11.000 Max. :9840   
## Cliente\_ID Producto\_ID Venta\_uni\_hoy Venta\_hoy   
## Min. : 60 Min. : 72 Min. : 0.000 Min. : 0.00   
## 1st Qu.: 359942 1st Qu.: 1242 1st Qu.: 2.000 1st Qu.: 16.76   
## Median : 1206731 Median :30549 Median : 3.000 Median : 30.00   
## Mean : 1812460 Mean :20910 Mean : 7.329 Mean : 68.49   
## 3rd Qu.: 2377992 3rd Qu.:37519 3rd Qu.: 7.000 3rd Qu.: 56.58   
## Max. :10351790 Max. :49994 Max. :2400.000 Max. :42667.12   
## Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil   
## Min. : 0.0000 Min. : 0.000 Min. : 0.000   
## 1st Qu.: 0.0000 1st Qu.: 0.000 1st Qu.: 2.000   
## Median : 0.0000 Median : 0.000 Median : 3.000   
## Mean : 0.1204 Mean : 1.188 Mean : 7.247   
## 3rd Qu.: 0.0000 3rd Qu.: 0.000 3rd Qu.: 6.000   
## Max. :330.0000 Max. :2897.400 Max. :2400.000

# Checking datatypes  
str(df\_sample)

## 'data.frame': 100000 obs. of 11 variables:  
## $ Semana : int 6 8 5 4 9 3 7 9 5 4 ...  
## $ Agencia\_ID : int 1636 1625 1330 1350 3214 1602 1212 2264 1235 1123 ...  
## $ Canal\_ID : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ruta\_SAK : int 1112 1292 1264 8011 1607 1201 1420 1228 1105 1408 ...  
## $ Cliente\_ID : int 1106211 422131 204979 1198764 597550 1326576 2337024 4489686 85669 204084 ...  
## $ Producto\_ID : int 3270 1109 41938 1232 303 3631 1240 1230 1064 1284 ...  
## $ Venta\_uni\_hoy : int 2 6 1 2 3 2 7 2 3 18 ...  
## $ Venta\_hoy : num 20.94 90.06 9.91 36.48 13.62 ...  
## $ Dev\_uni\_proxima : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ Dev\_proxima : num 0 15 0 0 0 ...  
## $ Demanda\_uni\_equil: int 2 5 1 2 3 2 7 2 3 18 ...

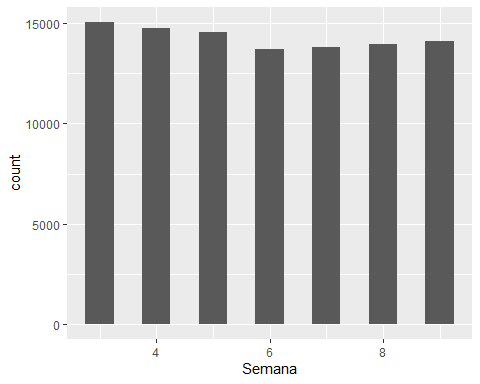
# Checking missing values  
colSums(is.na(df\_sample))

## Semana Agencia\_ID Canal\_ID Ruta\_SAK   
## 0 0 0 0   
## Cliente\_ID Producto\_ID Venta\_uni\_hoy Venta\_hoy   
## 0 0 0 0   
## Dev\_uni\_proxima Dev\_proxima Demanda\_uni\_equil   
## 0 0 0

There are no missing values in this sample dataset

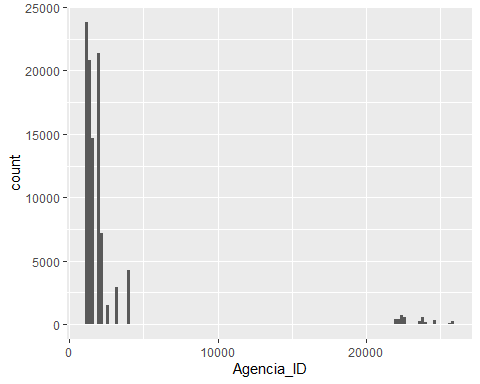
“Semana” distribution:

ggplot(data = df\_sample) +  
 geom\_histogram(mapping = aes(x = Semana), binwidth = 0.5)



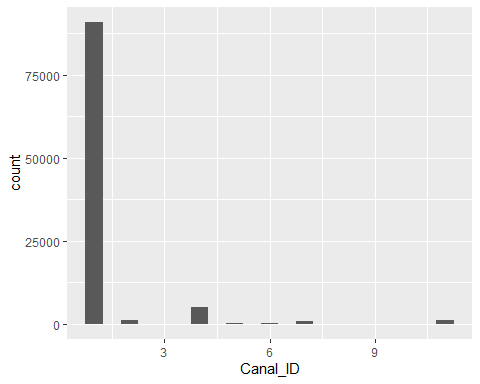
“Agencia\_ID” distribution:

ggplot(data = df\_sample) +  
 geom\_histogram(mapping = aes(x = Agencia\_ID), binwidth = 200)



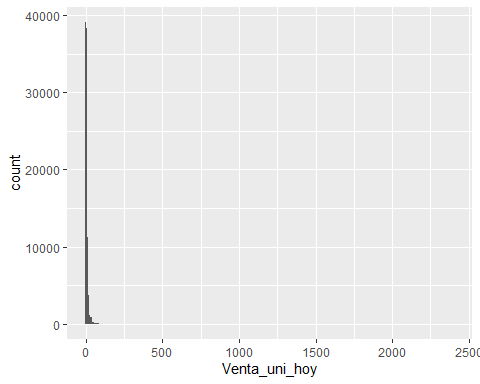
“Canal\_ID” distribution

ggplot(data = df\_sample) +  
 geom\_histogram(mapping = aes(x = Canal\_ID), binwidth = 0.5)



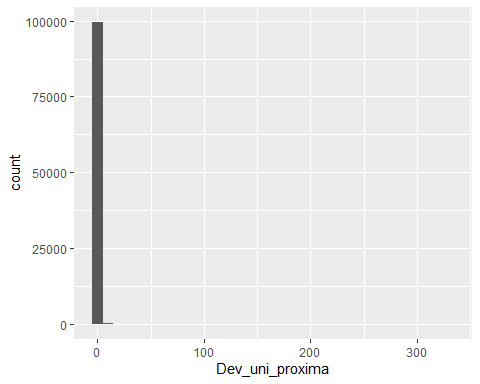
“Venta\_uni\_hoy” distribution

ggplot(data = df\_sample) +  
 geom\_histogram(mapping = aes(x = Venta\_uni\_hoy), binwidth = 5)



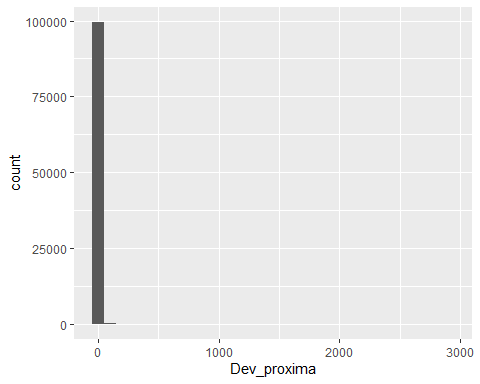
“Dev\_uni\_proxima” distribution

ggplot(data = df\_sample) +  
 geom\_histogram(mapping = aes(x = Dev\_uni\_proxima), binwidth = 10)



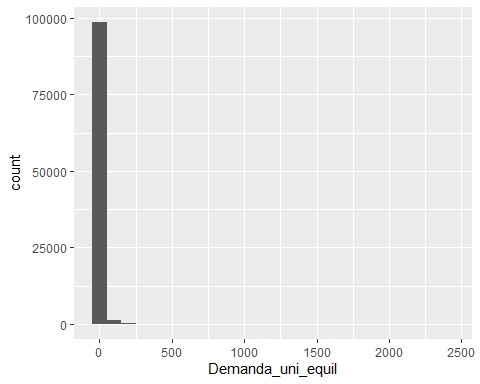
“Dev\_proxima” distribution

ggplot(data = df\_sample) +  
 geom\_histogram(mapping = aes(x = Dev\_proxima), binwidth = 100)



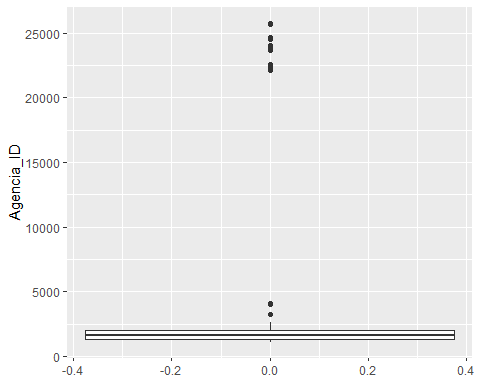
“Demanda\_uni\_equil” distribution

ggplot(data = df\_sample) +  
 geom\_histogram(mapping = aes(x = Demanda\_uni\_equil), binwidth = 100)

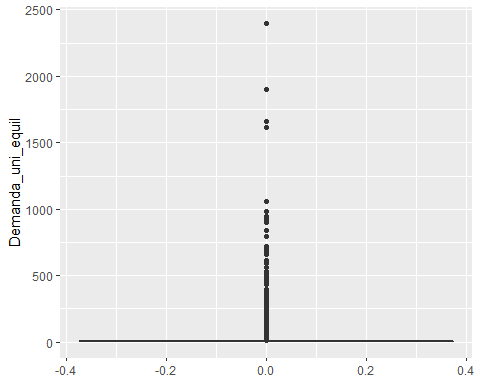


Checking outliers by “Agencia\_ID”

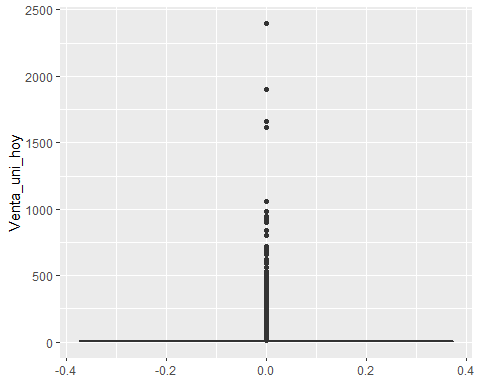
ggplot(data = df\_sample, mapping = aes(y = Agencia\_ID)) +  
 geom\_boxplot()



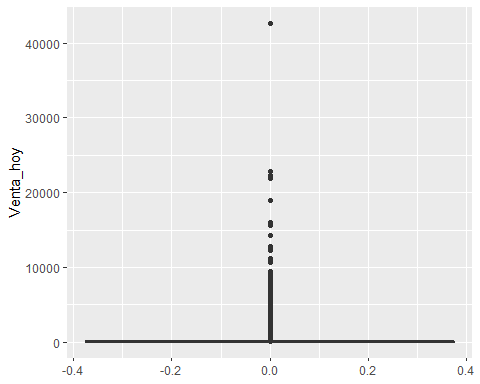
ggplot(data = df\_sample, mapping = aes(y = Demanda\_uni\_equil)) +  
 geom\_boxplot()



ggplot(data = df\_sample, mapping = aes(y = Venta\_uni\_hoy)) +  
 geom\_boxplot()



ggplot(data = df\_sample, mapping = aes(y = Venta\_hoy)) +  
 geom\_boxplot()



It seems like observation 3885 is an outlier so we are going to remove this line

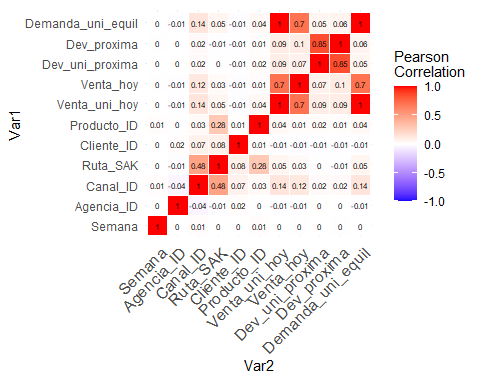
df\_sample <- df\_sample[-c(3885), ]

Checking correlation between variables

col\_num <- sapply(df\_sample, is.numeric)  
data\_cor <- cor(df\_sample[,col\_num])  
melted\_cormat <- melt(data\_cor)  
head(melted\_cormat)

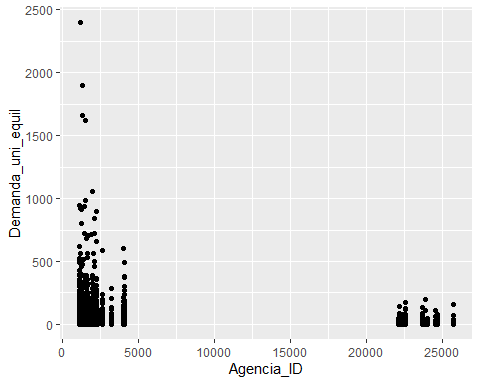
## Var1 Var2 value  
## 1 Semana Semana 1.0000000000  
## 2 Agencia\_ID Semana -0.0006255043  
## 3 Canal\_ID Semana 0.0133575223  
## 4 Ruta\_SAK Semana -0.0011637943  
## 5 Cliente\_ID Semana 0.0006834967  
## 6 Producto\_ID Semana 0.0143361179

ggplot(data = melted\_cormat, aes(Var2, Var1, fill = value))+  
 geom\_tile(color = "white")+  
 scale\_fill\_gradient2(low = "blue", high = "red", mid = "white",   
 midpoint = 0, limit = c(-1,1), space = "Lab",   
 name="Pearson\nCorrelation") +  
 theme\_minimal()+   
 theme(axis.text.x = element\_text(angle = 45, vjust = 1,   
 size = 12, hjust = 1))+  
 coord\_fixed()+  
 geom\_text(aes(Var2, Var1, label = round(value,2)), color = "black", size = 2)



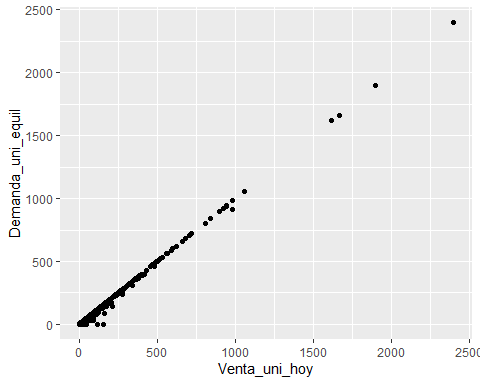
Correlation between “Agencia\_ID” and “Demanda\_uni\_equil”

ggplot(data = df\_sample) +  
 geom\_point(mapping = aes(x = Agencia\_ID, y = Demanda\_uni\_equil))



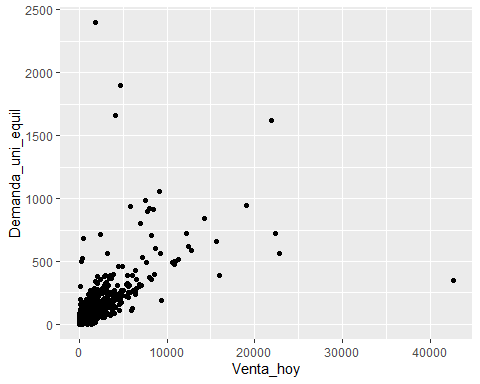
Correlation between “Venta\_uni\_hoy” and “Demanda\_uni\_equil”

ggplot(data = df\_sample) +  
 geom\_point(mapping = aes(x = Venta\_uni\_hoy, y = Demanda\_uni\_equil))



Correlation between “Venta\_hoy” and “Demanda\_uni\_equil”

ggplot(data = df\_sample) +  
 geom\_point(mapping = aes(x = Venta\_hoy, y = Demanda\_uni\_equil))



### Using dplyr to group/join data and get some insights

Top 10 sum of “Demanda\_uni\_equil” by State

df\_sample %>%  
 inner\_join(df\_town, by = 'Agencia\_ID') %>%  
 select(State, Demanda\_uni\_equil) %>%  
 group\_by(State) %>%  
 summarize(ave\_Demanda = sum(Demanda\_uni\_equil)) %>%  
 arrange(desc(ave\_Demanda))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 33 x 2  
## State ave\_Demanda  
## <chr> <int>  
## 1 ESTADO DE MÉXICO 102500  
## 2 MÉXICO, D.F. 85269  
## 3 JALISCO 67539  
## 4 NUEVO LEÓN 38991  
## 5 GUANAJUATO 36980  
## 6 VERACRUZ 36438  
## 7 PUEBLA 34670  
## 8 MICHOACÁN 28524  
## 9 SONORA 20883  
## 10 CHIHUAHUA 20829  
## # ... with 23 more rows

Top 10 sum of “Demanda\_uni\_equil” by Town

df\_sample %>%  
 inner\_join(df\_town, by = 'Agencia\_ID') %>%  
 select(Town, Demanda\_uni\_equil) %>%  
 group\_by(Town) %>%  
 summarize(ave\_Demanda = sum(Demanda\_uni\_equil)) %>%  
 arrange(desc(ave\_Demanda))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 255 x 2  
## Town ave\_Demanda  
## <chr> <int>  
## 1 2013 AG. MEGA NAUCALPAN 13460  
## 2 2011 AG. SAN ANTONIO 11725  
## 3 2029 AG.IZTAPALAPA 2 9339  
## 4 2309 NORTE 8365  
## 5 2088 AG. CEYLAN 8196  
## 6 2041 AG. TULTITLAN 7331  
## 7 2293 GRANJAS MARINELA 6997  
## 8 2252 AGUASCALIENTES SIGLO XXI 6838  
## 9 2251 AGUASCALIENTES NORTE 6819  
## 10 2017 AG. SANTA CLARA 6785  
## # ... with 245 more rows

Top 10 sum of “Demanda\_uni\_equil” by NombreCliente

df\_sample %>%  
 inner\_join(df\_cliente, by = 'Cliente\_ID') %>%  
 select(NombreCliente, Demanda\_uni\_equil) %>%  
 group\_by(NombreCliente) %>%  
 summarize(ave\_Demanda = sum(Demanda\_uni\_equil)) %>%  
 arrange(desc(ave\_Demanda))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 43,982 x 2  
## NombreCliente ave\_Demanda  
## <chr> <int>  
## 1 NO IDENTIFICADO 112859  
## 2 PUEBLA REMISION 22794  
## 3 LUPITA 3041  
## 4 YOLANDA JUAREZ RAMIREZ 2400  
## 5 QUERETARO DE ARTEAGA REMISION 2180  
## 6 MARY 1915  
## 7 AUTOBUSES DE LA PIEDAD PACIFICO 1898  
## 8 PRIMERA PLUS 1664  
## 9 OXXO SINALOA 1627  
## 10 LA PASADITA 1292  
## # ... with 43,972 more rows

Top 10 sum of “Demanda\_uni\_equil” by NombreProducto

df\_sample %>%  
 inner\_join(df\_produto, by = 'Producto\_ID') %>%  
 select(NombreProducto, Demanda\_uni\_equil) %>%  
 group\_by(NombreProducto) %>%  
 summarize(ave\_Demanda = sum(Demanda\_uni\_equil)) %>%  
 arrange(desc(ave\_Demanda))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 975 x 2  
## NombreProducto ave\_Demanda  
## <chr> <int>  
## 1 Nito 1p 62g Central BIM 2425 33034  
## 2 Rebanada 2p 55g BIM 1284 27179  
## 3 Nito 1p 62g BIM 1278 25470  
## 4 Gansito 1p 50g MTB MLA 43285 20432  
## 5 Bolsa Mini Rocko 40p 13g CU MLA 36610 17993  
## 6 Donas Azucar 4p 105g BIM 1250 17187  
## 7 Mantecadas Vainilla 4p 125g BIM 1240 15827  
## 8 Donitas Espolvoreadas 6p 105g BIM 1242 13999  
## 9 Polvoroncitos Panera 40p 16 25g TR 45143 13499  
## 10 Pan Blanco 640g BIM 2233 13034  
## # ... with 965 more rows

Searching for distinct values in town

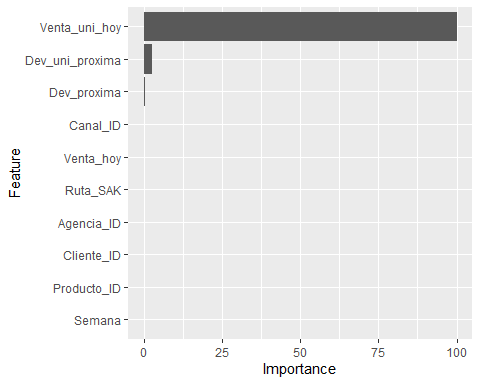
unique(df\_town$State)

## [1] "MÉXICO, D.F." "ESTADO DE MÉXICO" "HIDALGO"   
## [4] "Queretaro de Arteaga" "PUEBLA" "OAXACA"   
## [7] "MORELOS" "GUERRERO" "TLAXCALA"   
## [10] "JALISCO" "COLIMA" "ZACATECAS"   
## [13] "NAYARIT" "SAN LUIS POTOSÍ" "AGUASCALIENTES"   
## [16] "MICHOACÁN" "TAMAULIPAS" "NUEVO LEÓN"   
## [19] "COAHUILA" "CHIHUAHUA" "DURANGO"   
## [22] "SONORA" "BAJA CALIFORNIA NORTE" "SINALOA"   
## [25] "BAJA CALIFORNIA SUR" "VERACRUZ" "GUANAJUATO"   
## [28] "QUERETARO" "TABASCO" "YUCATÁN"   
## [31] "CAMPECHE" "QUINTANA ROO" "CHIAPAS"

Most important variables for the model using varImp

In this plot we can see variable importance for predicting Demanda\_uni\_equil

modelo <- train(Demanda\_uni\_equil ~ ., data = df\_sample, method = "lm")  
vImp <- varImp(modelo)   
  
# In this plot we can see variable importance for predicting Demanda\_uni\_equil  
ggplot(vImp) +  
 geom\_bar(stat='identity')



vImp

## lm variable importance  
##   
## Overall  
## Venta\_uni\_hoy 1.000e+02  
## Dev\_uni\_proxima 2.612e+00  
## Dev\_proxima 4.457e-01  
## Canal\_ID 2.171e-02  
## Venta\_hoy 1.893e-02  
## Ruta\_SAK 9.627e-03  
## Agencia\_ID 2.946e-03  
## Cliente\_ID 6.445e-04  
## Producto\_ID 8.449e-05  
## Semana 0.000e+00

Separating data into train/test

linha <- sample(1:nrow(df\_sample), 0.7 \* nrow(df\_sample))  
df\_train <- df\_sample[linha,]  
df\_test <- df\_sample[-linha,]  
  
dim(df\_train)

## [1] 69999 11

dim(df\_test)

## [1] 30000 11

Normalization

# Normalizing train dataset  
df\_n <- scale(df\_train[,-11])  
df\_train\_normalized <- as.data.frame(cbind(df\_n, df\_train$Demanda\_uni\_equil))  
rm(df\_n)  
colnames(df\_train\_normalized)[11] <- "Demanda\_uni\_equil"  
  
# Normalizing test dataset  
df\_n2 <- scale(df\_test[,-11])  
df\_test\_normalized <- as.data.frame(cbind(df\_n2, df\_test$Demanda\_uni\_equil))  
rm(df\_n2)  
colnames(df\_test\_normalized)[11] <- "Demanda\_uni\_equil"

Creating the model with all variables and without pre processing

modelo\_v1 <- lm(Demanda\_uni\_equil ~ ., data = df\_train)  
  
summary(modelo\_v1)

##   
## Call:  
## lm(formula = Demanda\_uni\_equil ~ ., data = df\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.070 0.003 0.008 0.018 65.700   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.140e-03 9.215e-03 0.124 0.90158   
## Semana -7.727e-04 1.244e-03 -0.621 0.53451   
## Agencia\_ID -1.845e-07 6.221e-07 -0.297 0.76681   
## Canal\_ID 4.578e-03 2.005e-03 2.284 0.02238 \*   
## Ruta\_SAK 4.579e-07 2.012e-06 0.228 0.81997   
## Cliente\_ID 9.522e-11 1.364e-09 0.070 0.94434   
## Producto\_ID 2.439e-08 1.411e-07 0.173 0.86276   
## Venta\_uni\_hoy 9.970e-01 1.394e-04 7151.078 < 2e-16 \*\*\*  
## Venta\_hoy -3.215e-05 9.895e-06 -3.249 0.00116 \*\*   
## Dev\_uni\_proxima -5.136e-01 2.744e-03 -187.168 < 2e-16 \*\*\*  
## Dev\_proxima -8.379e-04 2.548e-04 -3.289 0.00101 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6622 on 69988 degrees of freedom  
## Multiple R-squared: 0.9992, Adjusted R-squared: 0.9992   
## F-statistic: 8.781e+06 on 10 and 69988 DF, p-value: < 2.2e-16

previsao1 <- predict(modelo\_v1, df\_test)  
  
MSE1 = MSE(y\_pred=previsao1, y\_true=df\_test$Demanda\_uni\_equil)  
MAE1 = MAE(y\_pred=previsao1, y\_true=df\_test$Demanda\_uni\_equil)  
RMSE1 = RMSE(y\_pred=previsao1, y\_true=df\_test$Demanda\_uni\_equil)  
  
#RMSLE  
predicted\_value = abs(previsao1)   
actual\_value = abs(df\_test$Demanda\_uni\_equil)  
  
SLE = (log(predicted\_value + 1) - log(actual\_value+ 1))^2  
  
RMSLE = sqrt(mean(SLE))  
  
Score1 = 1/(1+exp(RMSLE))

Creating a new dataframe with the results

result <- data.frame("modelo\_v1", "all variables + no preprocessing", summary(modelo\_v1)$r.squared, MAE1, MSE1, RMSE1, Score1)  
names(result) <-c("Model", "Variables", "R-squared", "MAE", "MSE", "RMSE", "RMSLE")

Creating the model2 with all variables + normalized data

modelo\_v2 <- lm(Demanda\_uni\_equil ~ ., data = df\_train\_normalized)  
  
summary(modelo\_v2)

##   
## Call:  
## lm(formula = Demanda\_uni\_equil ~ ., data = df\_train\_normalized)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.070 0.003 0.008 0.018 65.700   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.2429892 0.0025028 2893.923 < 2e-16 \*\*\*  
## Semana -0.0015552 0.0025038 -0.621 0.53451   
## Agencia\_ID -0.0007430 0.0025056 -0.297 0.76681   
## Canal\_ID 0.0066318 0.0029037 2.284 0.02238 \*   
## Ruta\_SAK 0.0006821 0.0029970 0.228 0.81997   
## Cliente\_ID 0.0001755 0.0025140 0.070 0.94434   
## Producto\_ID 0.0004553 0.0026340 0.173 0.86276   
## Venta\_uni\_hoy 23.5218558 0.0032893 7151.078 < 2e-16 \*\*\*  
## Venta\_hoy -0.0106534 0.0032788 -3.249 0.00116 \*\*   
## Dev\_uni\_proxima -0.7830984 0.0041839 -187.168 < 2e-16 \*\*\*  
## Dev\_proxima -0.0137619 0.0041847 -3.289 0.00101 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.6622 on 69988 degrees of freedom  
## Multiple R-squared: 0.9992, Adjusted R-squared: 0.9992   
## F-statistic: 8.781e+06 on 10 and 69988 DF, p-value: < 2.2e-16

previsao2 <- predict(modelo\_v2, df\_test\_normalized)  
  
MSE2 = MSE(y\_pred=previsao2, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
MAE2 = MAE(y\_pred=previsao2, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
RMSE2 = RMSE(y\_pred=previsao2, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
  
#RMSLE  
predicted\_value = abs(previsao2)   
actual\_value = abs(df\_test\_normalized$Demanda\_uni\_equil)  
  
SLE = (log(predicted\_value + 1) - log(actual\_value+ 1))^2  
  
RMSLE = sqrt(mean(SLE))  
  
Score2 = 1/(1+exp(RMSLE))  
  
# Creating a new dataframe with the results  
result2 <- data.frame("modelo\_v2", "all variables + normalized data", summary(modelo\_v2)$r.squared, MAE2, MSE2, RMSE2, Score2)  
names(result2) <-c("Model", "Variables", "R-squared", "MAE", "MSE", "RMSE", "RMSLE")  
newresult <- rbind(result, result2)

Creating the model3 with top 3 variables and pre processing

modelo\_v3 <- lm(Demanda\_uni\_equil ~ Venta\_uni\_hoy +  
 Dev\_uni\_proxima +  
 Dev\_proxima, data = df\_train\_normalized)  
  
previsao3 <- predict(modelo\_v3, df\_test\_normalized)  
  
MSE3 = MSE(y\_pred=previsao3, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
MAE3 = MAE(y\_pred=previsao3, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
RMSE3 = RMSE(y\_pred=previsao3, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
  
#RMSLE  
predicted\_value3 = abs(previsao3)   
actual\_value3 = abs(df\_test$Demanda\_uni\_equil)  
  
SLE3 = (log(predicted\_value3 + 1) - log(actual\_value3+ 1))^2  
RMSLE3 = sqrt(mean(SLE3))  
Score3 = 1/(1+exp(RMSLE3))  
  
result3 <- data.frame("modelo\_v3", "top 3 variables + normalized data", summary(modelo\_v3)$r.squared, MAE3, MSE3, RMSE3, Score3)  
names(result3) <-c("Model", "Variables", "R-squared", "MAE", "MSE", "RMSE", "RMSLE")  
newresult <- rbind(result, result2, result3)

Creating the model4 with top 1 variables and pre processing

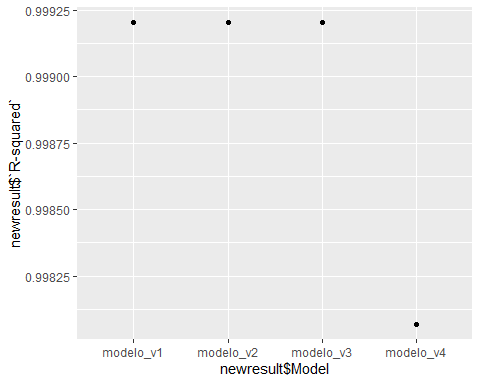
modelo\_v4 <- lm(Demanda\_uni\_equil ~ Venta\_uni\_hoy, data = df\_train\_normalized)  
  
previsao4 <- predict(modelo\_v4, df\_test\_normalized)  
  
MSE4 = MSE(y\_pred=previsao4, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
MAE4 = MAE(y\_pred=previsao4, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
RMSE4 = RMSE(y\_pred=previsao4, y\_true=df\_test\_normalized$Demanda\_uni\_equil)  
  
#RMSLE  
predicted\_value4 = abs(previsao4)   
actual\_value4 = abs(df\_test$Demanda\_uni\_equil)  
  
SLE4 = (log(predicted\_value4 + 1) - log(actual\_value4+ 1))^2  
RMSLE4 = sqrt(mean(SLE4))  
Score4 = 1/(1+exp(RMSLE4))  
  
result4 <- data.frame("modelo\_v4", "top 1 variable + normalized data", summary(modelo\_v4)$r.squared, MAE4, MSE4, RMSE4, Score4)  
names(result4) <- c("Model", "Variables", "R-squared", "MAE", "MSE", "RMSE", "RMSLE")  
newresult <- rbind(result, result2, result3, result4)

Visualizing the results from the 3 models created:

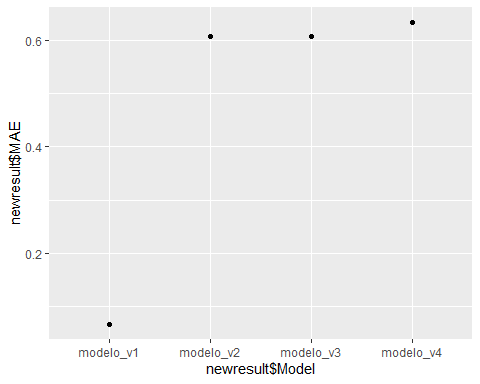
head(newresult)

## Model Variables R-squared MAE MSE  
## 1 modelo\_v1 all variables + no preprocessing 0.9992036 0.0666552 0.3293135  
## 2 modelo\_v2 all variables + normalized data 0.9992036 0.6085626 3.7878370  
## 3 modelo\_v3 top 3 variables + normalized data 0.9992034 0.6079777 3.7968868  
## 4 modelo\_v4 top 1 variable + normalized data 0.9980683 0.6341164 4.9691170  
## RMSE RMSLE  
## 1 0.5738585 0.4716284  
## 2 1.9462366 0.4505657  
## 3 1.9485602 0.4506237  
## 4 2.2291516 0.4471934

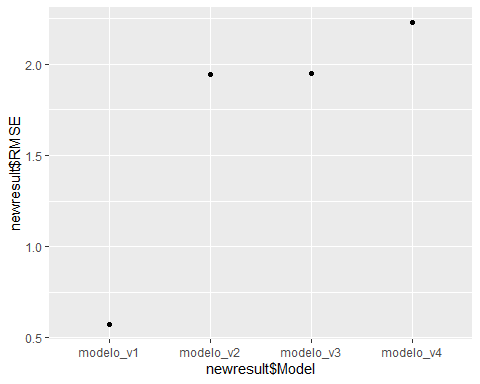
#R-Squared  
ggplot(data = newresult) +  
 geom\_point(mapping = aes(x = newresult$`Model`, y = newresult$`R-squared`))



#MAE  
ggplot(data = newresult) +  
 geom\_point(mapping = aes(x = newresult$`Model`, y = newresult$`MAE`))



#RMSE  
ggplot(data = newresult) +  
 geom\_point(mapping = aes(x = newresult$`Model`, y = newresult$`RMSE`))



## Conclusions

Since our original dataset was too big (74.180.464 observations), we decided to get a 100.000 observations sample to do our analysis;

Our target variable (the one we are trying to predict) is the ‘Demanda\_uni\_equil’;

In the correlation plot, we can see that variable ‘Venta\_uni\_hoy’ have a strong positive correlation to our target. Variable ‘Venta\_hoy’ also have a strong positive correlation with out target;

Variables ‘Dev\_proxima’ and ‘Dev\_uni\_proxima’ are strongly correlated, as is variables ‘Venta\_hoy’ and ‘Venta\_uni\_hoy’

Most of the customers are ‘Not identified’, but we can see that we have a list of our top customers and we could promote a marketing campaign for them.

We created 4 final versions of our model, and the metrics are very similar. Since all R-Squared metrics are around 99%, it means that 99% of the data fit the regression model. If we analyze the MAE (mean absolute error), ‘modelo\_v1’ is the one with the least value, which means that we can expect the least error from the forecast on average. We could continue this analysis doing other pre processing to see if the results will change.