# 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 = ",", encodi
ng = "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 = ",", encod
ing = "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
```

```r
dim(df_produto)
```

```
## [1] 2592    2
```

```r
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.
```

```r
dim(df_town)
```

```
## [1] 790   3
```

Loading train dataset:

```r
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_un
i_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 trai
n 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.   :    0.0
0
##  1st Qu.:  359942   1st Qu.: 1242   1st Qu.:   2.000   1st Qu.:   16.7
6
##  Median : 1206731   Median :30549   Median :   3.000   Median :   30.0
0
##  Mean   : 1812460   Mean   :20910   Mean   :   7.329   Mean   :   68.4
9
##  3rd Qu.: 2377992   3rd Qu.:37519   3rd Qu.:   7.000   3rd Qu.:   56.5
8
##  Max.   :10351790   Max.   :49994   Max.   :2400.000   Max.   :42667.1
2
##  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 123
5 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 110
5 1408 ...
##  $ Cliente_ID       : int  1106211 422131 204979 1198764 597550 132657
6 2337024 4489686 85669 204084 ...
##  $ Producto_ID      : int  3270 1109 41938 1232 303 3631 1240 1230 106
4 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 ...
```

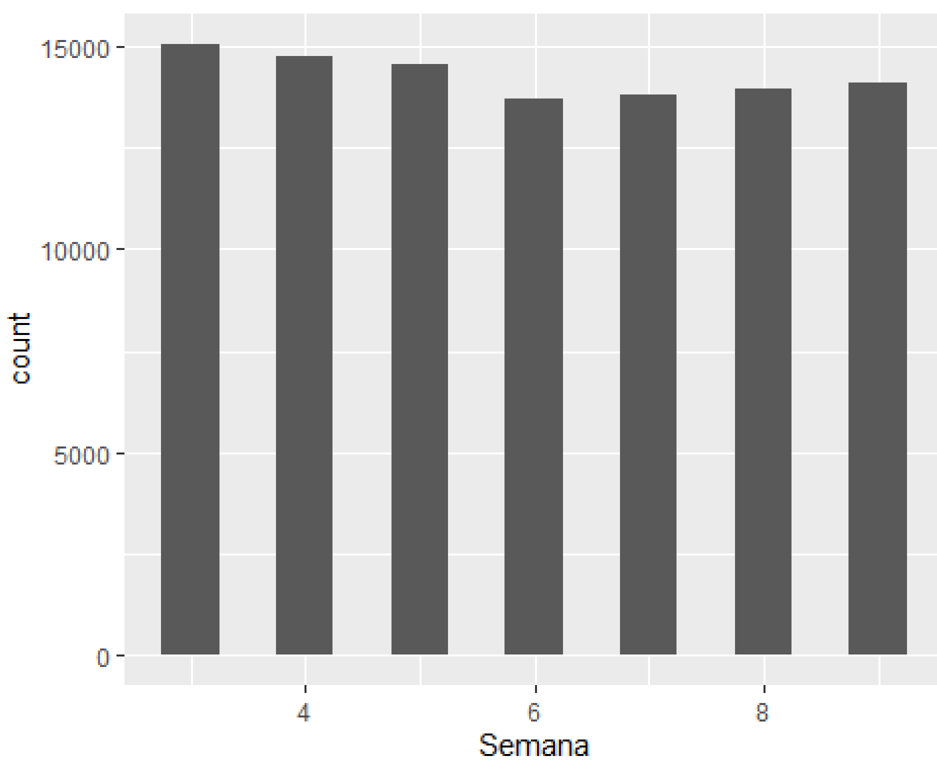
```r
# Checking missing values
colSums(is.na(df_sample))
```

```
##            Semana        Agencia_ID          Canal_ID          Ruta_SA
K
##                 0                 0                 0                 
0
##        Cliente_ID       Producto_ID     Venta_uni_hoy         Venta_ho
y
##                 0                 0                 0                 
0
##   Dev_uni_proxima       Dev_proxima Demanda_uni_equil 
##                 0                 0                 0
```

There are no missing values in this sample dataset

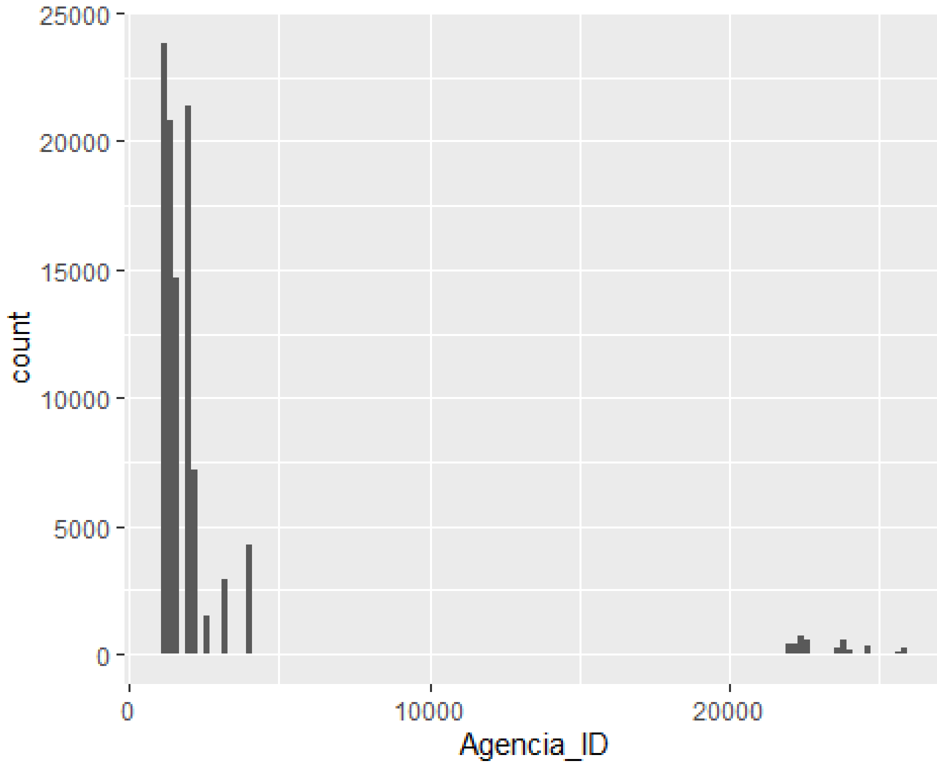
"Semana" distribution:

```r
ggplot(data = df_sample) +
  geom_histogram(mapping = aes(x = Semana), binwidth = 0.5)
```



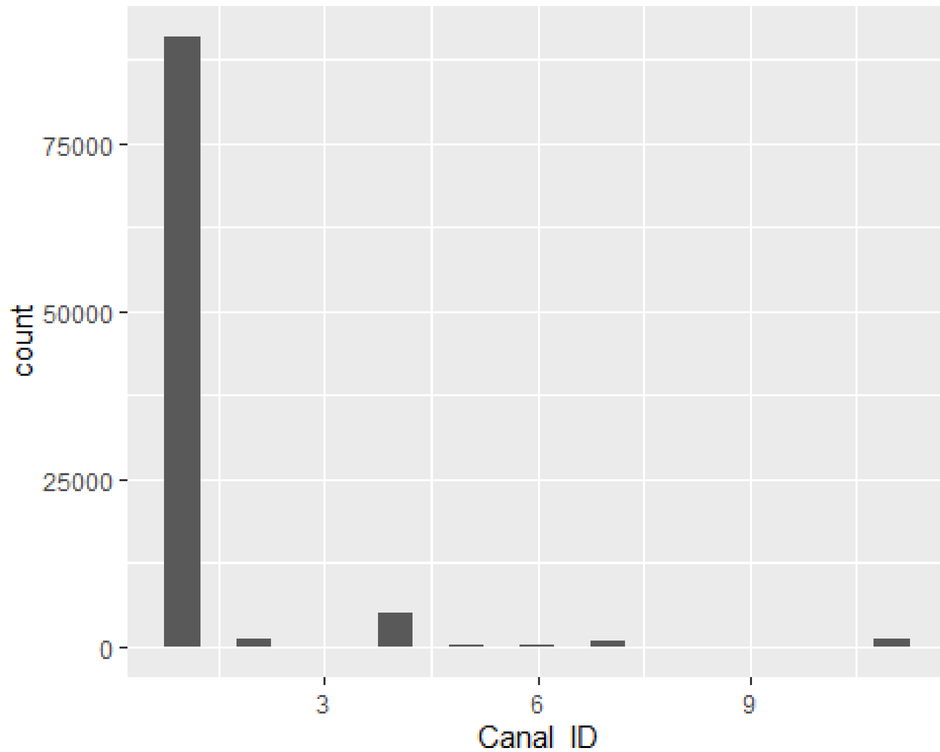

"Agencia_ID" distribution:

```r
ggplot(data = df_sample) +
  geom_histogram(mapping = aes(x = Agencia_ID), binwidth = 200)
```



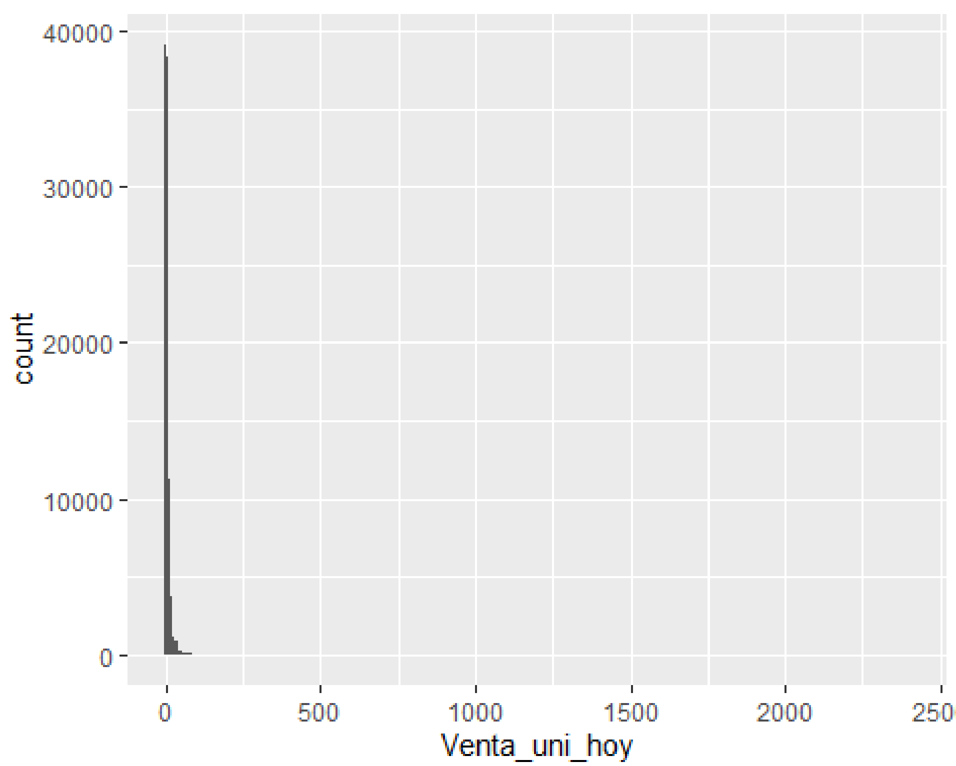

"Canal_ID" distribution

```r
ggplot(data = df_sample) +
  geom_histogram(mapping = aes(x = Canal_ID), binwidth = 0.5)
```



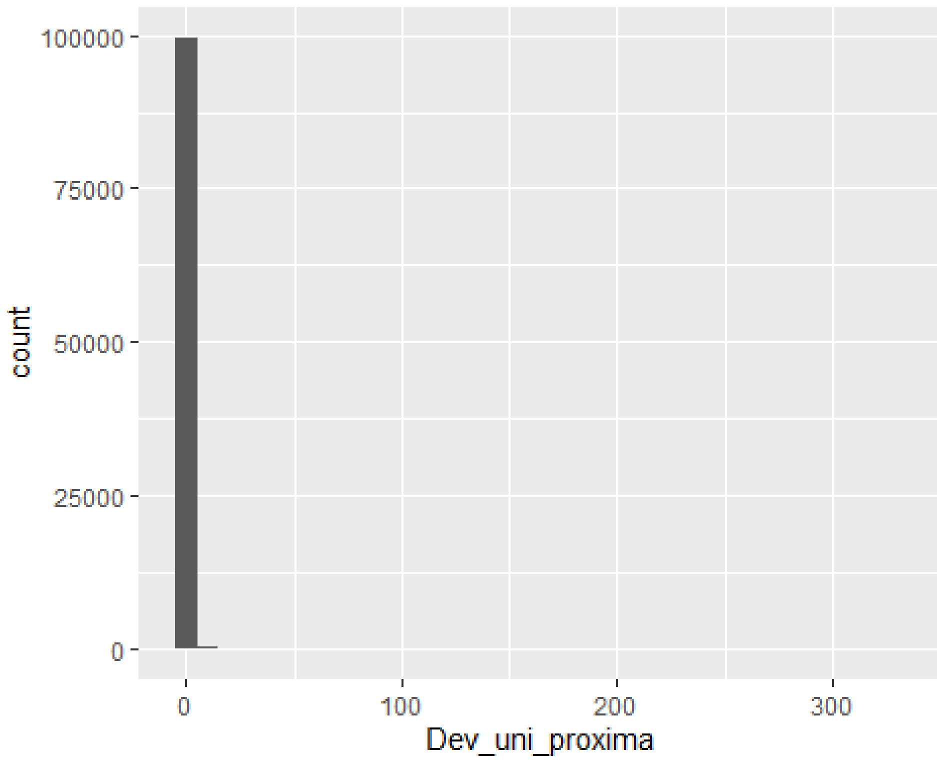

"Venta_uni_hoy" distribution

```
ggplot(data = df_sample) +
  geom_histogram(mapping = aes(x = Venta_uni_hoy), binwidth = 5)
```



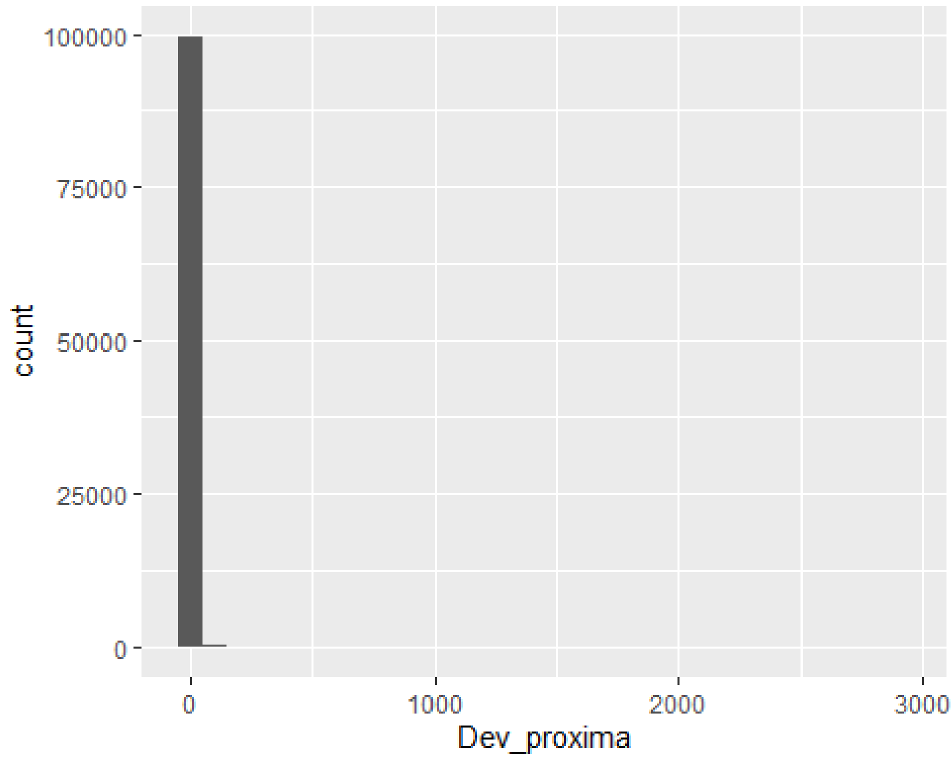
“Dev_uni_proxima” distribution

```r
ggplot(data = df_sample) +
  geom_histogram(mapping = aes(x = Dev_uni_proxima), binwidth = 10)
```



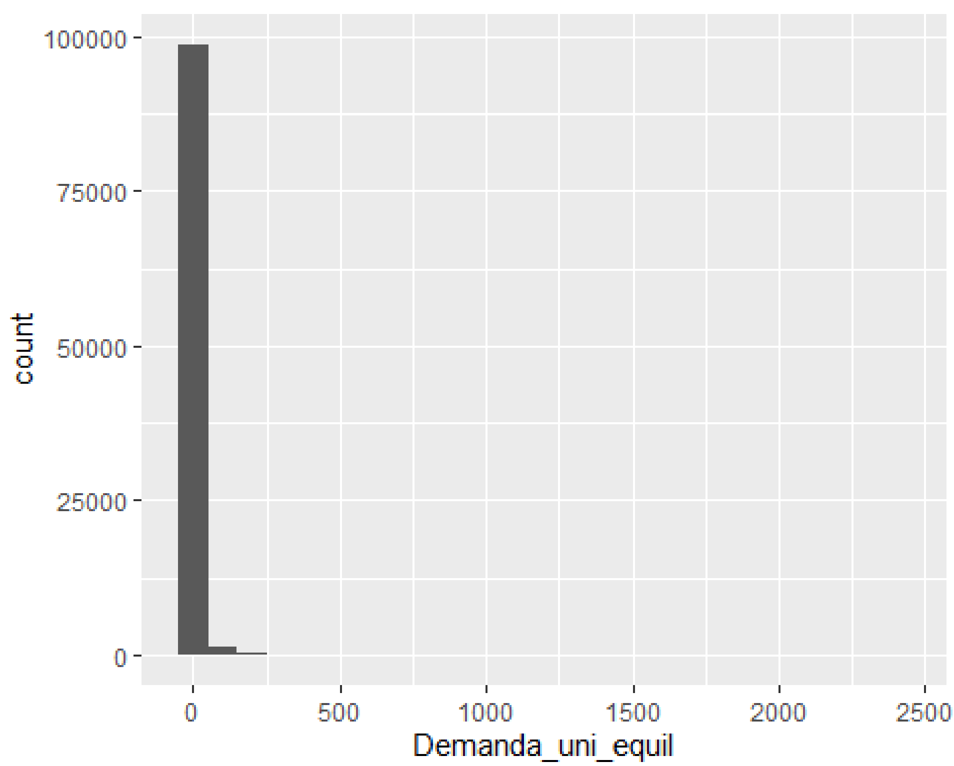

“Dev_proxima” distribution

```r
ggplot(data = df_sample) +
  geom_histogram(mapping = aes(x = Dev_proxima), binwidth = 100)
```



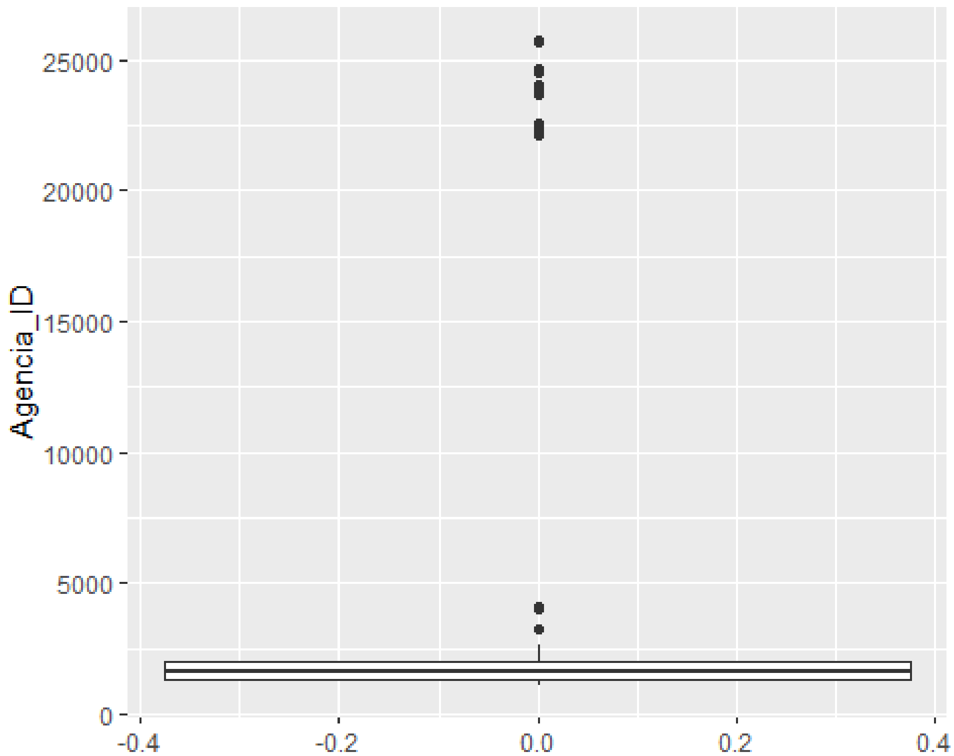

"Demanda_uni_equil" distribution

```
ggplot(data = df_sample) +
  geom_histogram(mapping = aes(x = Demanda_uni_equil), binwidth = 100)
```

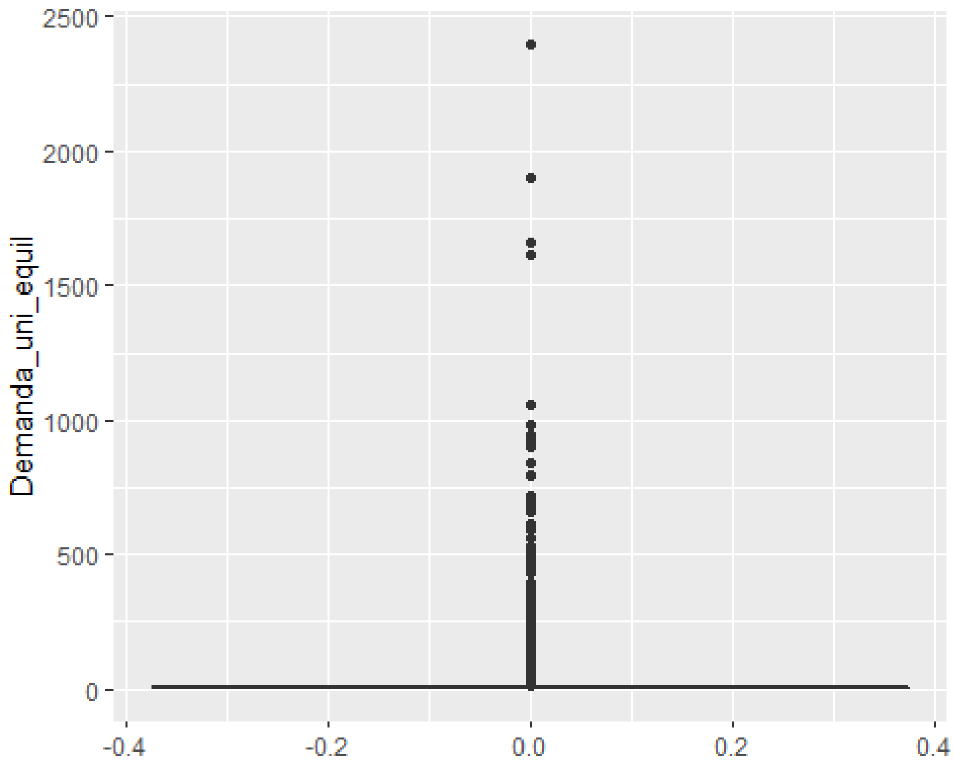


Checking outliers by “Agencia_ID”

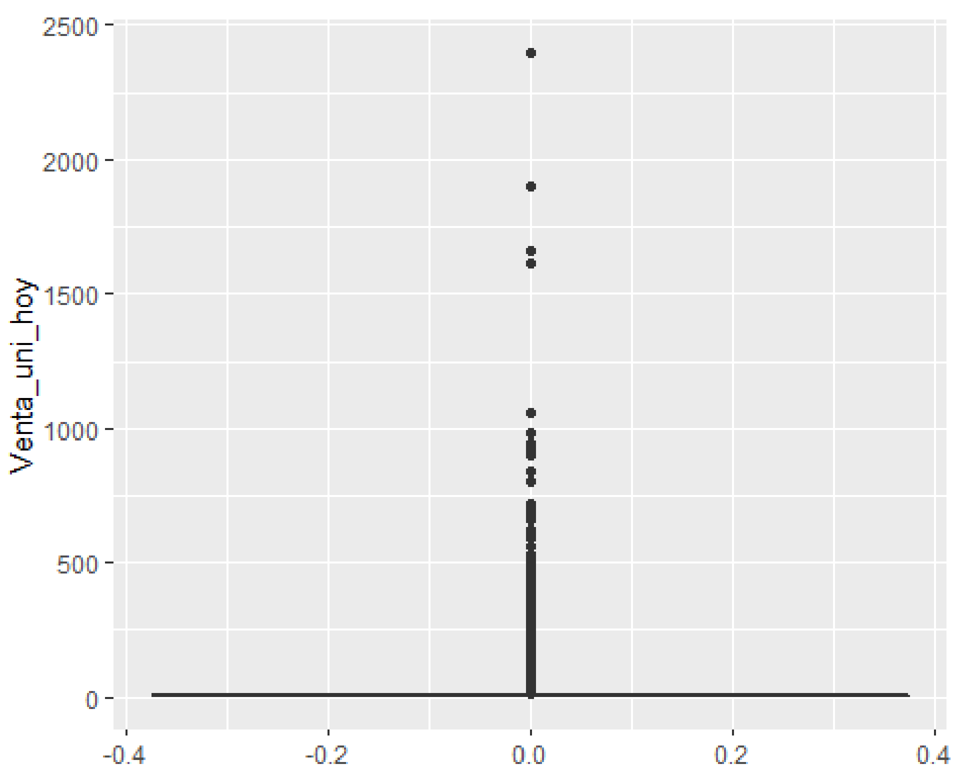
```
ggplot(data = df_sample, mapping = aes(y = Agencia_ID)) +
  geom_boxplot()
```



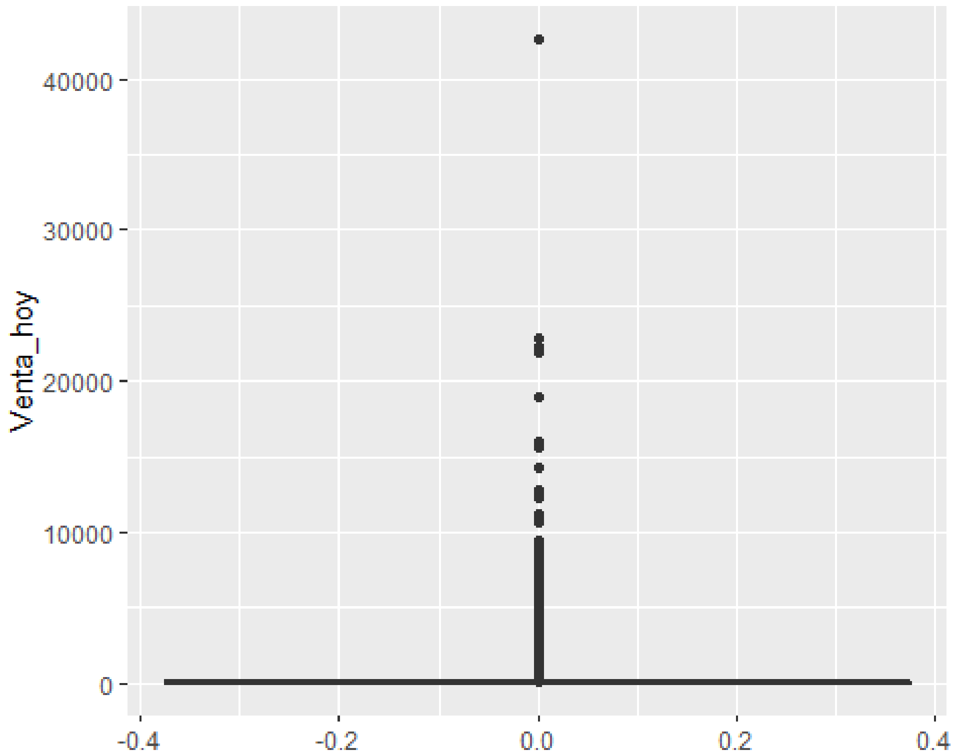

```
ggplot(data = df_sample, mapping = aes(y = Demanda_uni_equil)) +
  geom_boxplot()
```




```r
ggplot(data = df_sample, mapping = aes(y = Venta_uni_hoy)) +
  geom_boxplot()
```



```r
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

```r
df_sample <- df_sample[-c(3885), ]
```

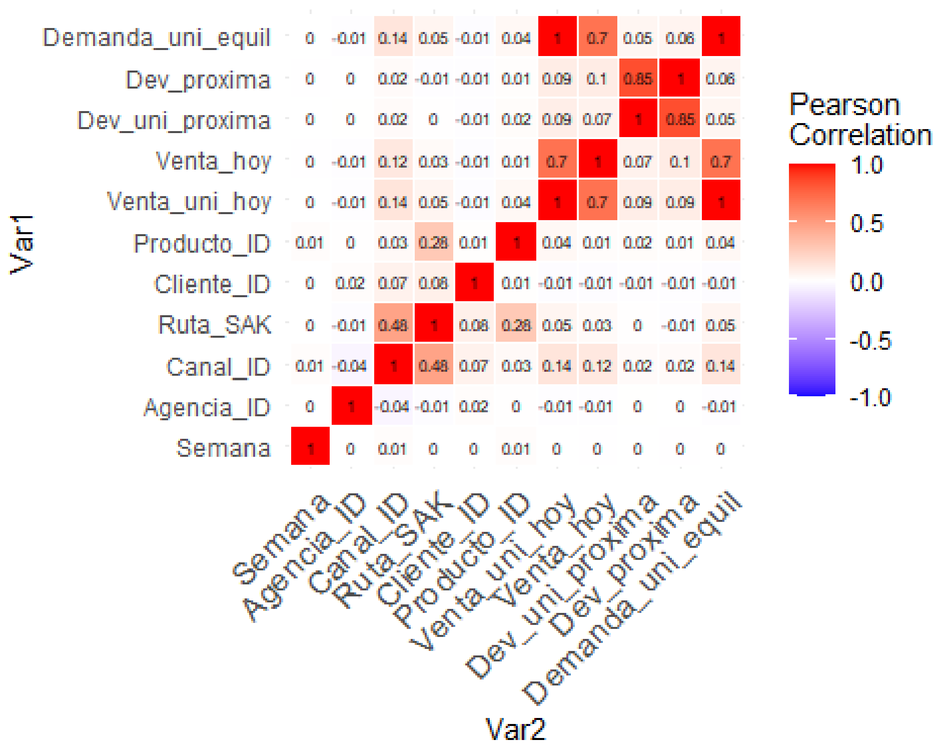
Checking correlation between variables

```r
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
```

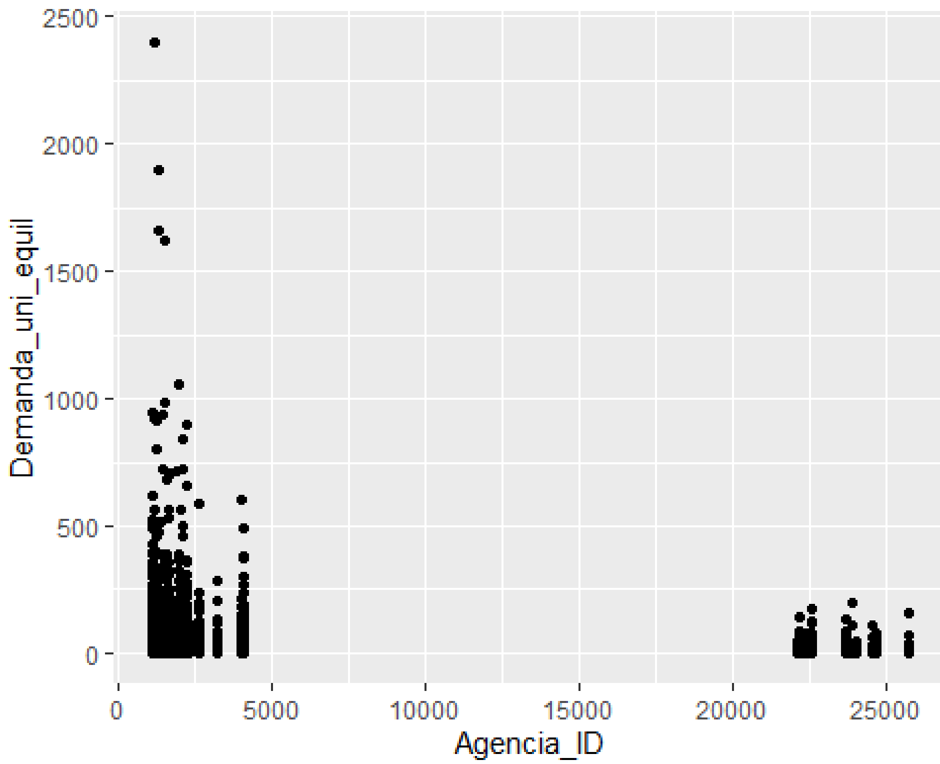
```r
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+
```

```r
  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", siz
e = 2)
```



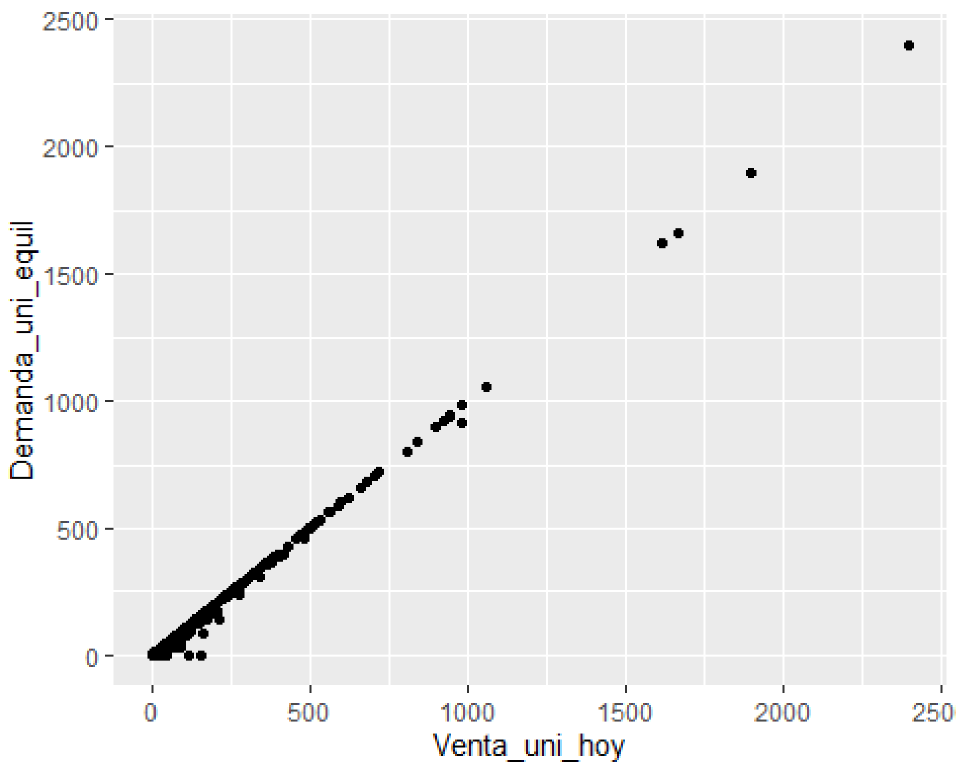

Correlation between “Agencia_ID” and “Demanda_uni_equil”

```r
ggplot(data = df_sample) +
  geom_point(mapping = aes(x = Agencia_ID, y = Demanda_uni_equil))
```



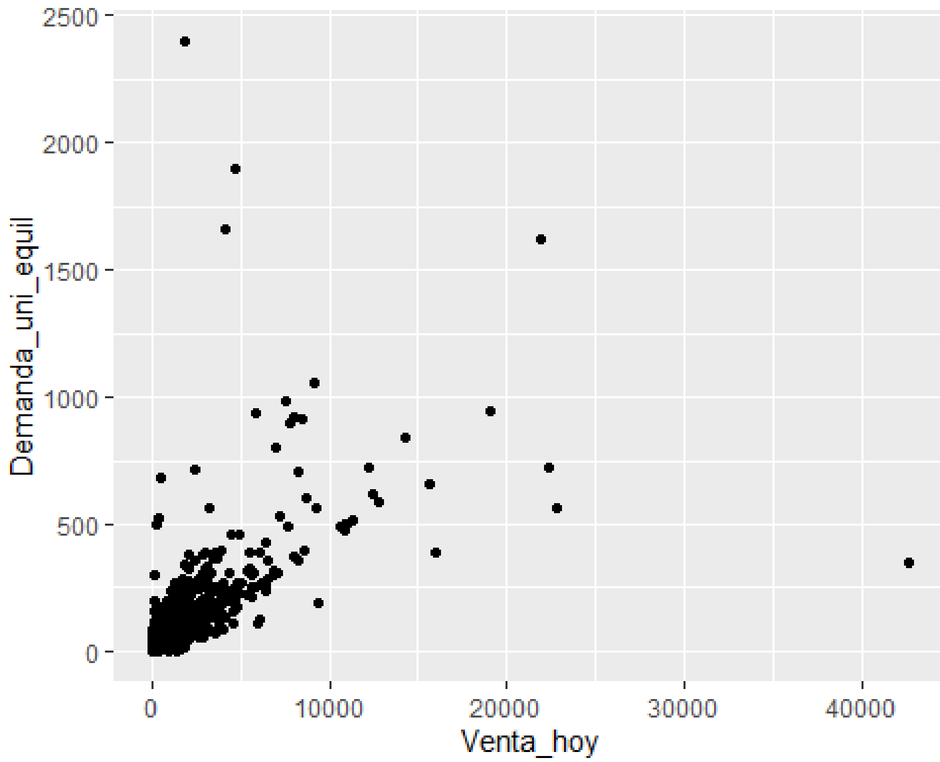

Correlation between “Venta_uni_hoy” and “Demanda_uni_equil”

```
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

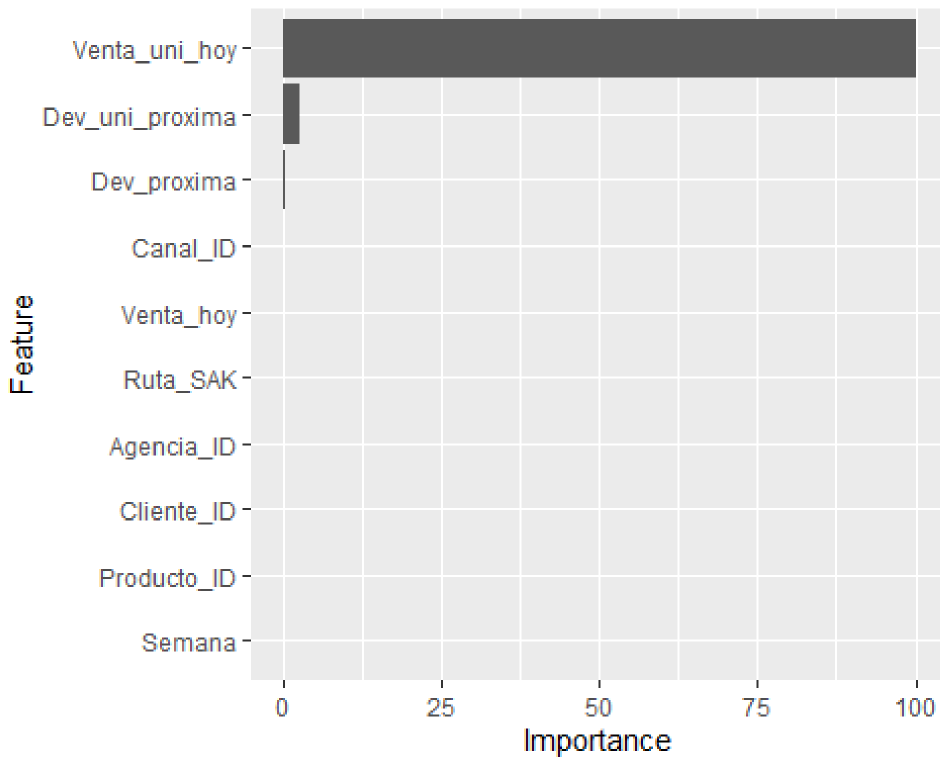
```
unique(df_town$State)
```

```
##  [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

```r
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')
```




```r
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

```r
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

```r
# Normalizing train dataset
df_n <- scale(df_train[,-11])
df_train_normalized <- as.data.frame(cbind(df_n, df_train$Demanda_uni_equ
il))
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_equi
l))
rm(df_n2)
colnames(df_test_normalized)[11] <- "Demanda_uni_equil"
```

Creating the model with all variables and without pre processing

```r
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", sum
mary(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_equi
l)

#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", sum
mary(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_equi
l)

#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", s
ummary(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

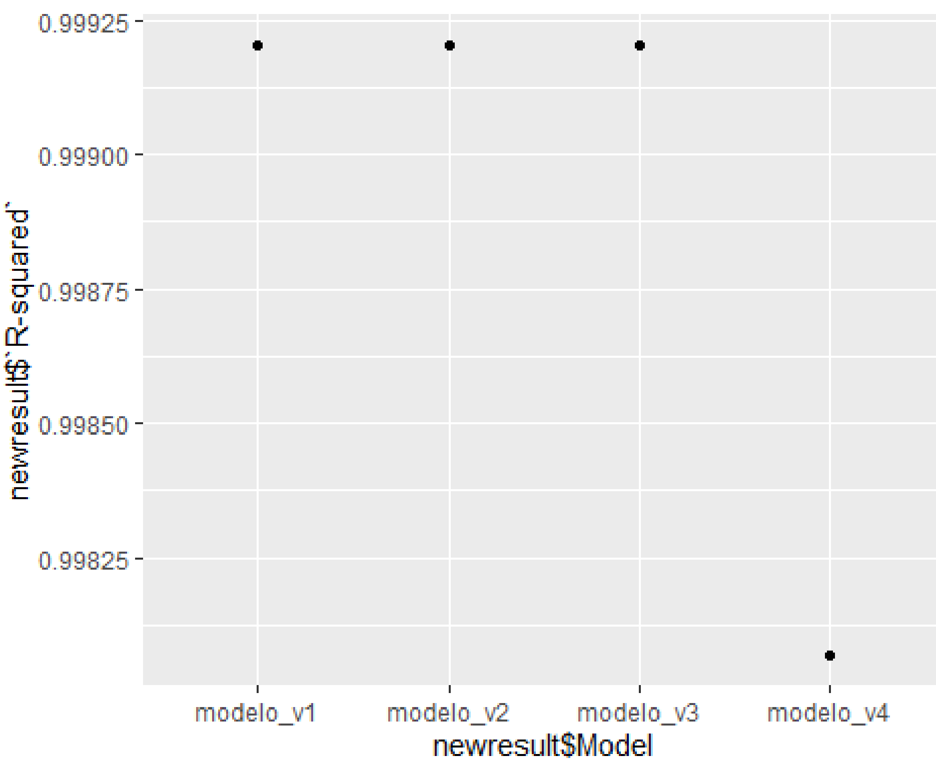
```
modelo_v4 <- lm(Demanda_uni_equil ~ Venta_uni_hoy, data = df_train_normal
ized)

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_equi
l)

#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", su
mmary(modelo_v4)$r.squared, MAE4, MSE4, RMSE4, Score4)
names(result4) <- c("Model", "Variables", "R-squared", "MAE", "MSE", "RMS
E", "RMSLE")
newresult <- rbind(result, result2, result3, result4)
```

Visualizing the results from the 3 models created:

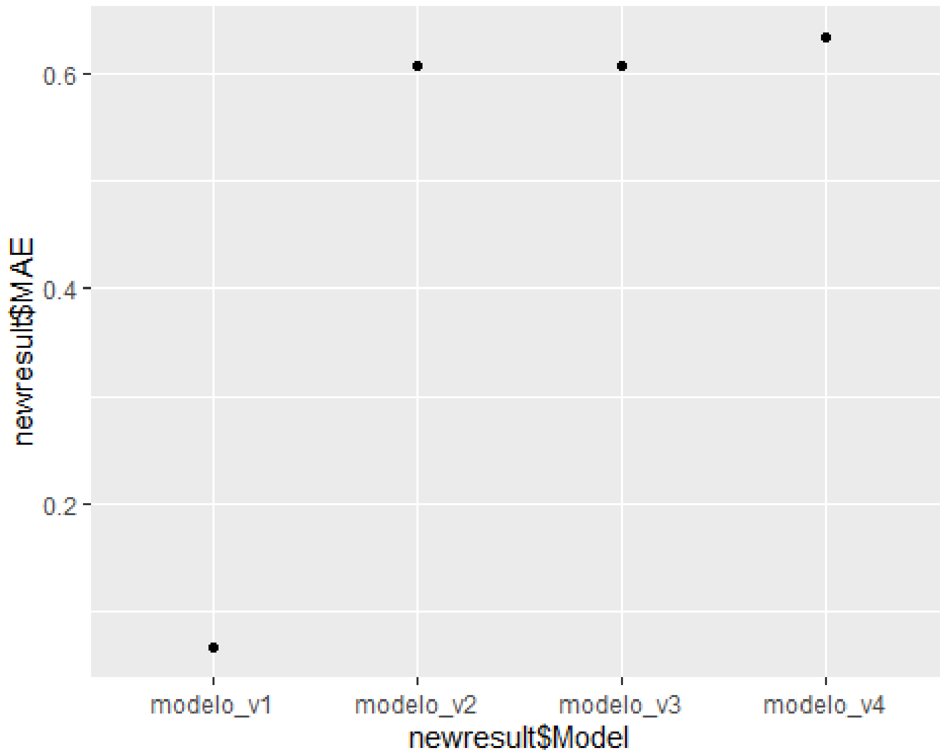
```
head(newresult)

##       Model                              Variables R-squared       MAE
MSE
```

```
## 1 modelo_v1  all variables + no preprocessing 0.9992036 0.0666552 0.32
93135
## 2 modelo_v2   all variables + normalized data 0.9992036 0.6085626 3.78
78370
## 3 modelo_v3 top 3 variables + normalized data 0.9992034 0.6079777 3.79
68868
## 4 modelo_v4  top 1 variable + normalized data 0.9980683 0.6341164 4.96
91170
##        RMSE     RMSLE
## 1 0.5738585 0.4716284
## 2 1.9462366 0.4505657
## 3 1.9485602 0.4506237
## 4 2.2291516 0.4471934
```

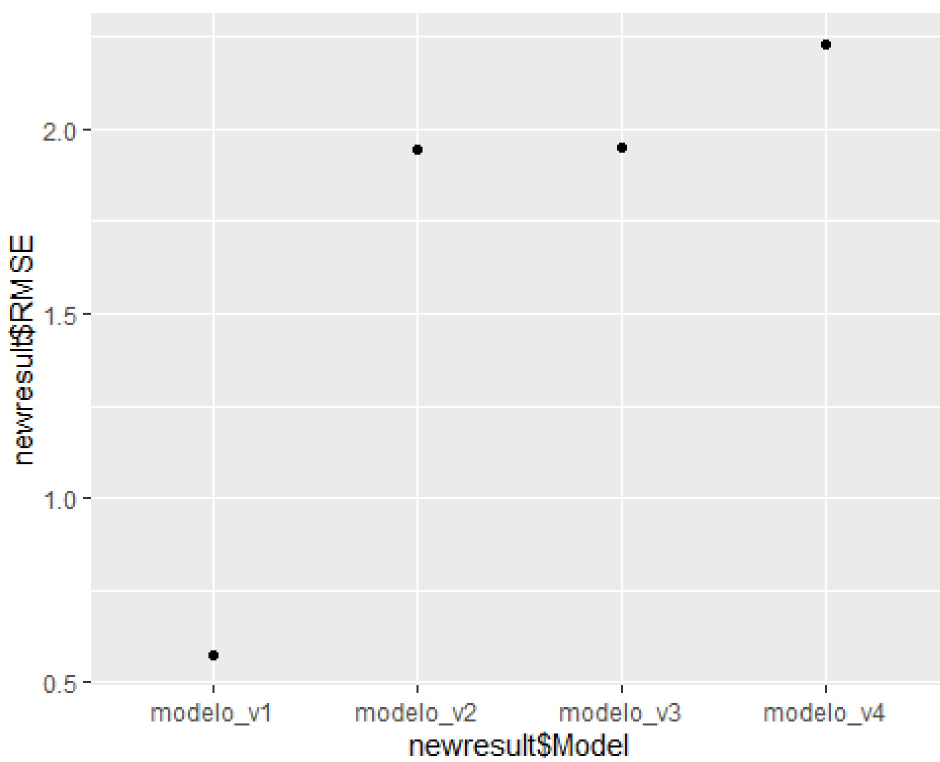
```
#R-Squared
ggplot(data = newresult) +
  geom_point(mapping = aes(x = newresult$`Model`, y = newresult$`R-square
d`))
```




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




```r
#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.