GrupoBimboEDA

Anandan Balaji

Contents

1 Grupo Bimbo Exploratory Data Analysis

Objective: Predict the demand based on historical sales data.

1.1 About Dataset & Data fields

The dataset is available at www.kaagle.com

Some of the important data fields:

- Semana — Week number (From Thursday to Wednesday)
- Agencia_ID — Sales Depot ID
- Canal_ID — Sales Channel ID
- Ruta_SAK — Route ID (Several routes = Sales Depot)
- Cliente_ID — Client ID
- NombreCliente — Client name
- Producto_ID — Product ID
- NombreProducto — Product Name
- **Venta_uni_hoy — Sales unit this week** (integer)
- Venta_hoy — Sales this week (unit: pesos)
- Dev_uni_proxima — Returns unit next week (integer)
- Dev_proxima — Returns next week (unit: pesos)
- **Demanda_uni_equil — Adjusted Demand** (integer) (This is the target you will predict)

1.2 Evaluation

The evaulation metric for this competition is Root Mean Squared Logarithmic Error.

1.3 Train dataset

```
library("data.table")
system.time(train <- fread("./new_train.csv", header = TRUE))
```

```
##
Read 0.0% of 11127070 rows
Read 12.1% of 11127070 rows
Read 24.6% of 11127070 rows
Read 37.5% of 11127070 rows
Read 50.1% of 11127070 rows
Read 63.1% of 11127070 rows
Read 76.0% of 11127070 rows
Read 89.0% of 11127070 rows
Read 11127070 rows and 11 (of 11) columns from 0.424 GB file in 00:00:11
```

```
##    user  system elapsed
##   9.354   1.214  30.255
```

```
system.time(test <- fread("./test.csv", header = TRUE))
```

```
##
Read 21.0% of 6999251 rows
Read 53.1% of 6999251 rows
Read 84.9% of 6999251 rows
Read 6999251 rows and 7 (of 7) columns from 0.234 GB file in 00:00:05
```

```
##    user  system elapsed
##   3.906   0.583  16.138
```

```
system.time(product <- fread("./producto_tabla.csv", header=TRUE))
```

```
##    user  system elapsed
##   0.003   0.001   0.071
```

```
#structure of train
str(train)
```

```
## Classes 'data.table' and 'data.frame':   11127070 obs. of  11 variables:
##  $ Semana          : int  5 3 4 9 6 5 3 6 4 9 ...
##  $ Agencia_ID      : int  1550 1236 2234 1220 1227 1656 1612 1152 1118 1346 ...
##  $ Canal_ID        : int  1 1 1 1 1 1 1 4 1 1 ...
##  $ Ruta_SAK        : int  1007 1235 1139 1611 1016 2827 1137 6607 1415 1012 ...
##  $ Cliente_ID      : int  1193749 89902 494471 4356501 785027 2205160 1177048 2175684 438774 131975
##  $ Producto_ID     : int  46772 1238 1687 4245 42122 36610 41938 40447 1212 49972 ...
```

```
##  $ Venta_uni_hoy    : int  3 3 1 1 10 40 3 20 10 1 ...
##  $ Venta_hoy        : num  29.2 29.5 19 10.5 304 ...
##  $ Dev_uni_proxima  : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Dev_proxima      : num  0 9.83 0 0 0 0 0 0 0 0 ...
##  $ Demanda_uni_equil: int  3 2 1 1 10 40 3 20 10 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
## number of observations
nrow(train)
```

```
## [1] 11127070
```

```
## get the weekly data - ie., number of transactions in that particular week.
table(train$Semana)
```

```
##
##       3       4       5       6       7       8       9
## 1674671 1653019 1591599 1529012 1558120 1559940 1560709
```

```
## get the demand info for every week
## tapply(X-vector, Index-variable, function)
##
tapply(train$Demanda_uni_equil, train$Semana, sum)
```

```
##        3        4        5        6        7        8        9
## 11641144 11952952 11634998 11087442 11514373 11350413 11269605
```

```
## number of unque products
length(unique(train$Producto_ID))
```

```
## [1] 1626
```

```
# which is the highest demand product
prod_results <- tapply(train$Demanda_uni_equil, train$Producto_ID, sum)
prod_results <- sort(prod_results, decreasing = TRUE)

highest_demand_prod <- prod_results[1]
highest_demand_prod
```

```
##    2425
## 3558441
```

```
## the most popular product is

str(product)
```

```
## Classes 'data.table' and 'data.frame':   2592 obs. of  2 variables:
##  $ Producto_ID   : int  0 9 41 53 72 73 98 99 100 106 ...
##  $ NombreProducto: chr  "NO IDENTIFICADO 0" "Capuccino Moka 750g NES 9" "Bimbollos Ext sAjonjoli 6p 4
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
product$NombreProducto[2425]
```

```
## [1] "Tortilla Hna RB 10p 260g DH 47840"
```

1.4 Test Dataset

```
# let's look at the test dataset

str(test)
```

```
## Classes 'data.table' and 'data.frame':   6999251 obs. of  7 variables:
##  $ id         : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Semana     : int  11 11 10 11 11 11 11 10 10 11 ...
##  $ Agencia_ID : int  4037 2237 2045 1227 1219 1146 2057 1612 1349 1461 ...
##  $ Canal_ID   : int  1 1 1 1 1 4 1 1 1 1 ...
##  $ Ruta_SAK   : int  2209 1226 2831 4448 1130 6601 4507 2837 1223 1203 ...
##  $ Cliente_ID : int  4639078 4705135 4549769 4717855 966351 1741414 4659766 4414012 397854 1646915 .
##  $ Producto_ID: int  35305 1238 32940 43066 1277 972 1232 35305 1240 43203 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
# look at the week info
table(test$Semana)
```

```
##
##      10      11
## 3538385 3460866
```

```
#look at the products, are all of them available in training dataset?
train_prods <- unique(train$Producto_ID)

test_prods <- unique(test$Producto_ID)

# are the products equal
setequal(train_prods, test_prods)
```

```
## [1] FALSE
```

```
# number of products equal
length(intersect(train_prods, test_prods))
```

```
## [1] 1450
```

```
# get the new products in test dataset
new_prods_in_test <- setdiff(test_prods, train_prods)

# number of new products in test dataset
length(new_prods_in_test)
```

```
## [1] 72
```

```
## look at Agency ID (Sales Depot ID)
setequal(train$Agencia_ID, test$Agencia_ID)
```

```
## [1] TRUE
```

```
## Channel ID (Sales chaneel ID)
setequal(train$Canal_ID, test$Canal_ID)
```

```
## [1] TRUE
```

```
## Route ID ( Ruta_SAK)
setequal(train$Ruta_SAK, test$Ruta_SAK)
```

```
## [1] FALSE
```

```
intersect_routes <- intersect(train$Ruta_SAK, test$Ruta_SAK)

#new routes in test
test_new_routes <- setdiff(test$Ruta_SAK, train$Ruta_SAK)

is.element(test_new_routes[1], test$Ruta_SAK)
```

```
## [1] TRUE
```

```
## examine the client IDs
setequal(train$Cliente_ID, test$Cliente_ID)
```

```
## [1] FALSE
```

```
new_test_clients <- setdiff(test$Cliente_ID, train$Cliente_ID)

# how many new clients?
length(new_test_clients)
```

```
## [1] 38124
```

```
is.element(new_test_clients[1], test$Cliente_ID)
```

```
## [1] TRUE
```

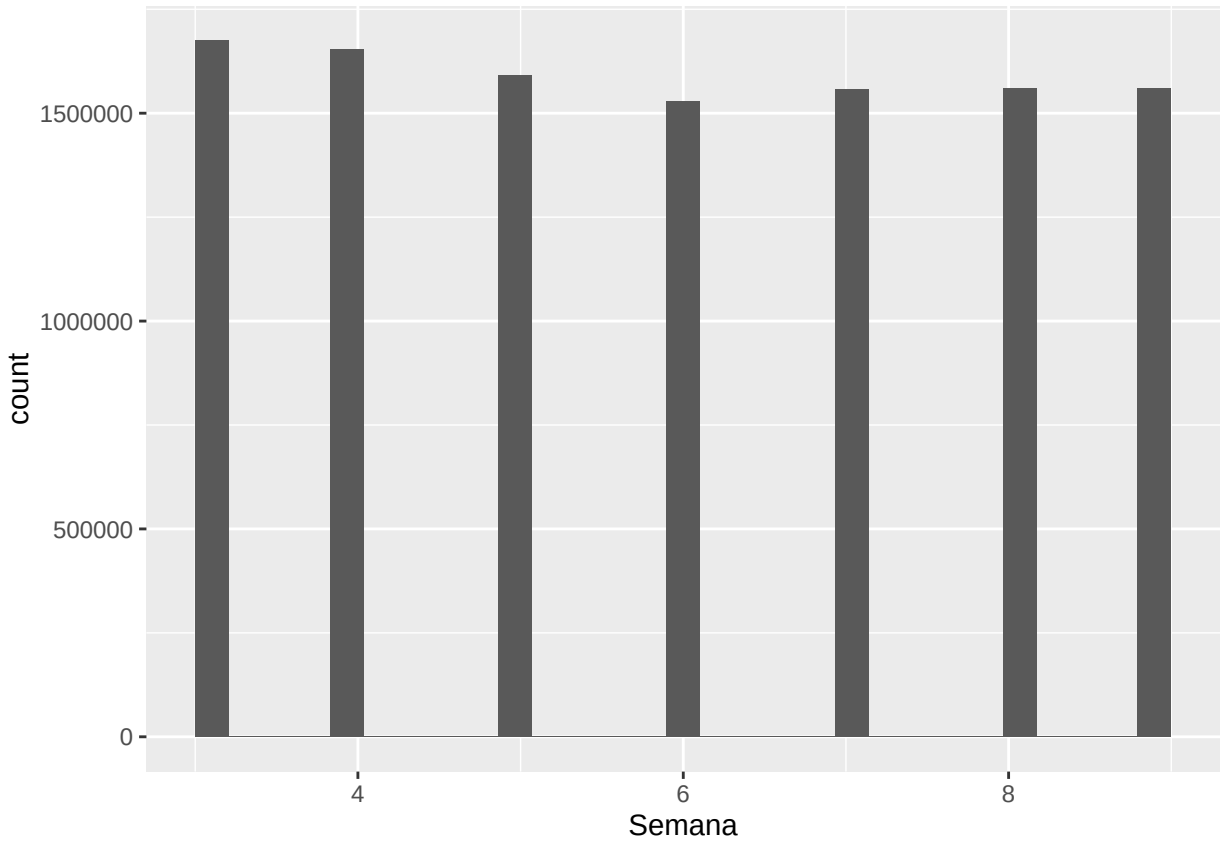
1.5 Plots

```
library("ggplot2")
```

1.6 Weekly Transactions.

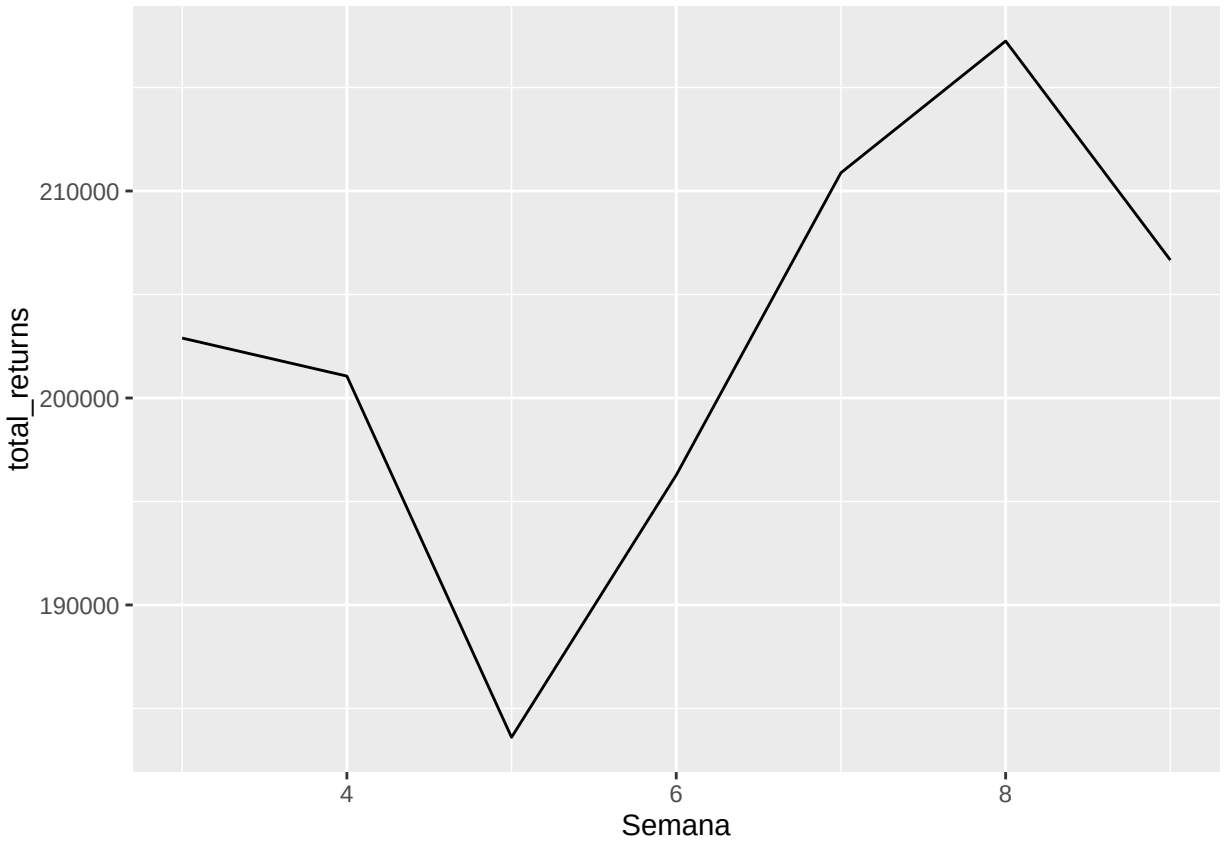
```
# per week, h
ggplot(train, aes(x = Semana)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



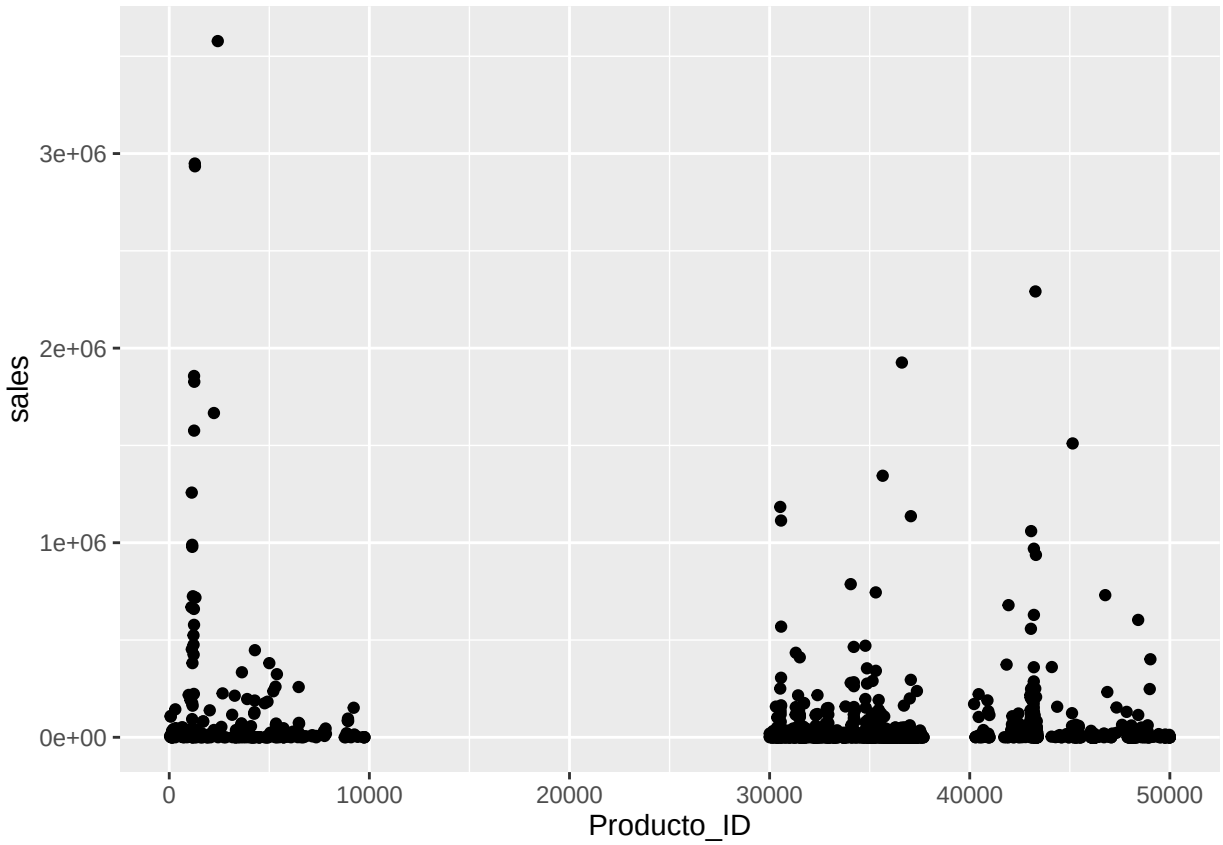
1.7 Returns per week

```
# returns per week

returns <- train[, .( total_returns = sum(Dev_uni_proxima)), by = Semana]

ggplot(returns, aes(x=Semana, y = total_returns)) + geom_line()
```



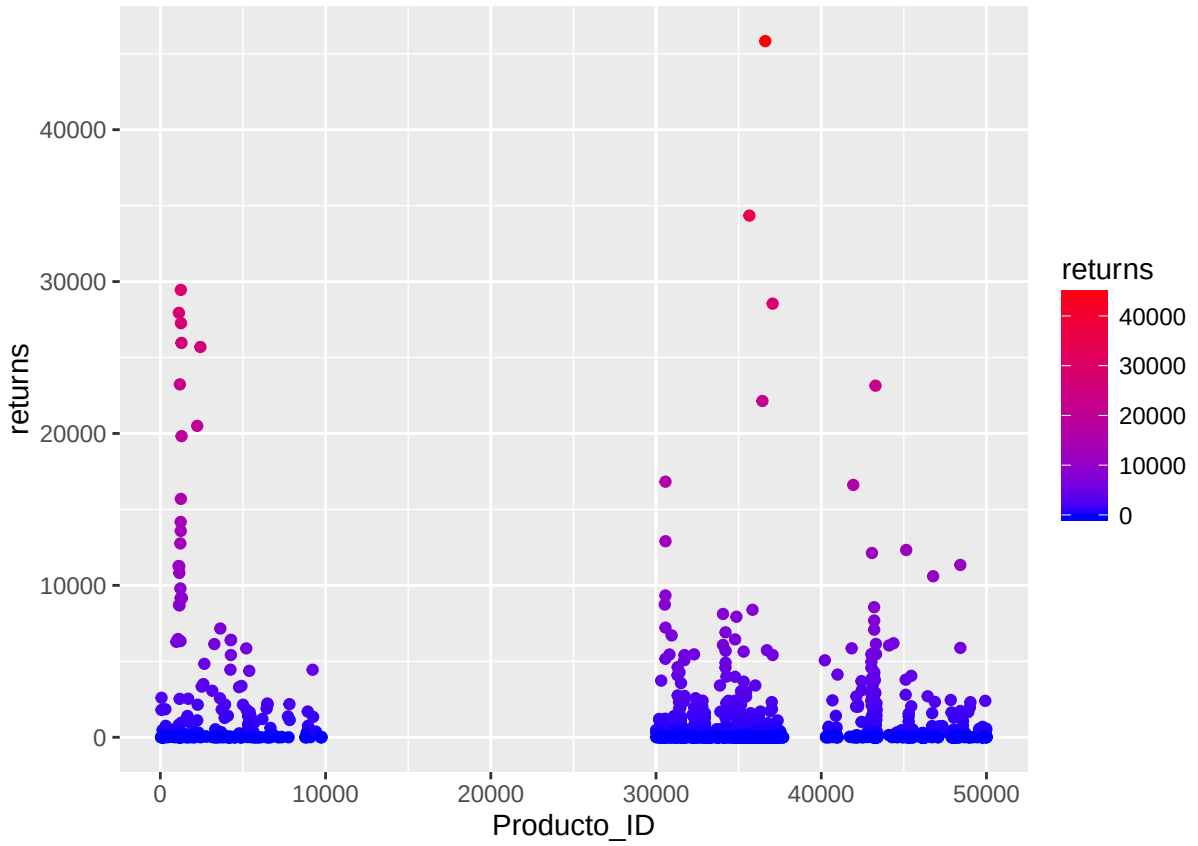
1.8 Product Sales

```
# product wise sale (units) & return info.

prodReturn <- train[, .(sales = sum(Venta_uni_hoy), returns = sum(Dev_uni_proxima)), by = Producto_ID]

# product-wise sales (units)
ggplot(prodReturn, aes(x = Producto_ID, y = sales)) + geom_point()
```



1.9 Product wise Sale vs Return

```
# product wise returns
ggplot(prodReturn, aes(x = Producto_ID, y = returns, color = returns)) + geom_point() +
  scale_color_gradient(low="blue", high="red")
```

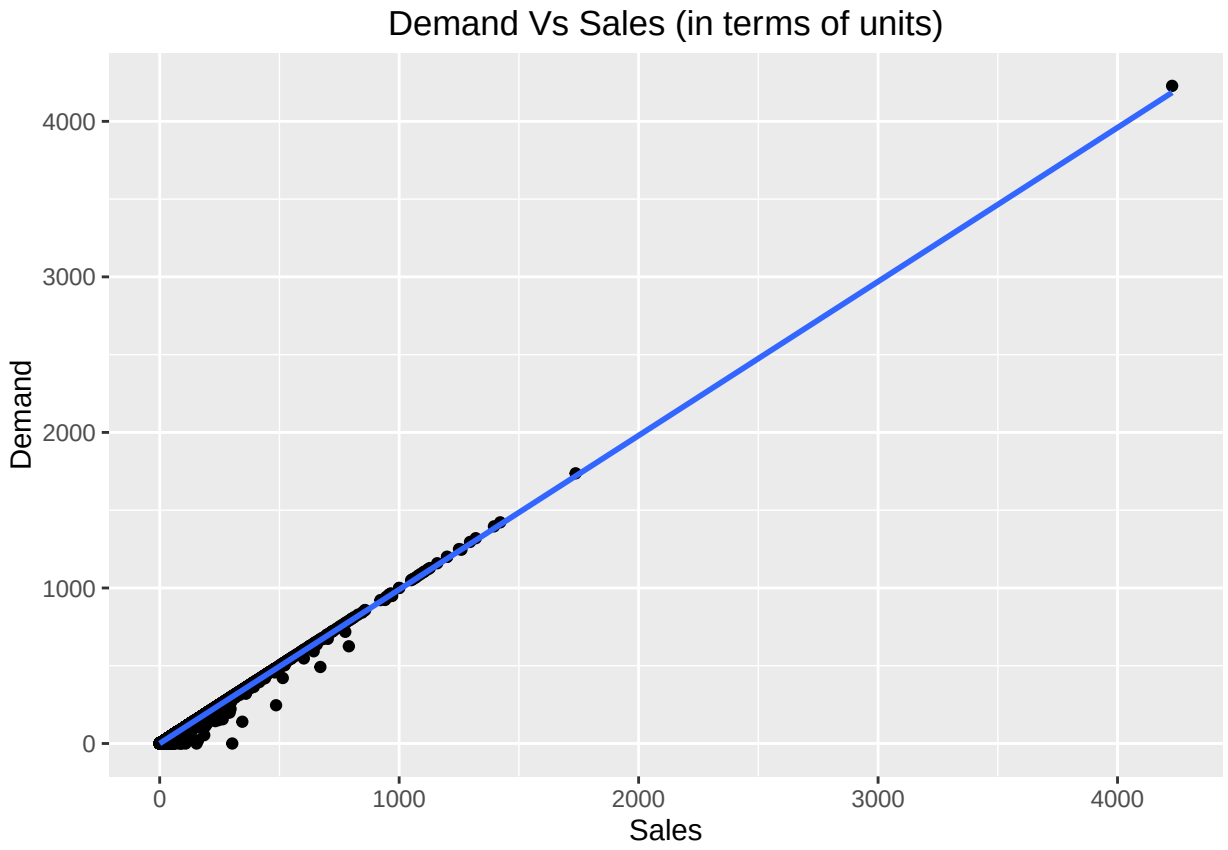


1.10 Demand Vs Sales

```
# use just a fraction to plot as the table is huge!
# demand vs sales
# Demanda_uni_quil vs Venta_hoy

library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
ggplot(train %>% sample_frac(0.05),aes(x = Venta_uni_hoy , y = Demanda_uni_equil)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_x_continuous(name = "Sales") +
  scale_y_continuous(name = "Demand") +
  ggtitle("Demand Vs Sales (in terms of units)")
```



1.11 Model

```r
#create the log. demand

train$log_demand <- log1p(train$Demanda_uni_equil)

# fit the training data
fit <- lm(Demanda_uni_equil ~ Venta_uni_hoy, data = train)
summary(fit)
```

```
##
## Call:
## lm(formula = Demanda_uni_equil ~ Venta_uni_hoy, data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -2372.50     0.03     0.04     0.07    56.16
```

```
##
## Coefficients:
##                  Estimate Std. Error   t value Pr(>|t|)
## (Intercept)   -3.746e-03  4.742e-04    -7.899 2.81e-15 ***
## Venta_uni_hoy  9.888e-01  2.019e-05 48969.146  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.503 on 11127068 degrees of freedom
## Multiple R-squared:  0.9954, Adjusted R-squared:  0.9954
## F-statistic: 2.398e+09 on 1 and 11127068 DF,  p-value: < 2.2e-16
```

```
fit2 <- lm(log_demand ~ Venta_uni_hoy, data = train)
summary(fit2)
```

```
##
## Call:
## lm(formula = log_demand ~ Venta_uni_hoy, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -95.226  -0.395  -0.128   0.370   1.457
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.453e+00  2.201e-04    6602   <2e-16 ***
## Venta_uni_hoy 2.046e-02  9.372e-06    2183   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6977 on 11127068 degrees of freedom
## Multiple R-squared:  0.2998, Adjusted R-squared:  0.2998
## F-statistic: 4.765e+06 on 1 and 11127068 DF,  p-value: < 2.2e-16
```