Kaggle Competition - Grupo Bimbo Inventory Demand

loaded packages

```
library(markdown)
library(knitr)
library(DescTools)
library(data.table)
library(dplyr)
```

Data Exploration

In this competition, you will forecast the demand of a product for a given week, at a particular store. The dataset you are given consists of 9 weeks of sales transactions in Mexico. Every week, there are delivery trucks that deliver products to the vendors. Each transaction consists of sales and returns. Returns are the products that are unsold and expired. The demand for a product in a certain week is defined as the sales this week subtracted by the return next week.

The train and test dataset are split based on time, as well as the public and private leaderboard dataset split.

File descriptions

1. train.csv — the training set
2. test.csv — the test set
3. sample_submission.csv — a sample submission file in the correct format
4. cliente_tabla.csv — client names (can be joined with train/test on Cliente_ID)
5. producto_tabla.csv — product names (can be joined with train/test on Producto_ID)
6. town_state.csv — town and state (can be joined with train/test on Agencia_ID)

loaded data

```
train <- fread("train.csv", stringsAsFactors=TRUE, data.table = FALSE)
```

```
##
Read 0.0% of 74180464 rows
Read 4.1% of 74180464 rows
Read 8.1% of 74180464 rows
Read 12.0% of 74180464 rows
Read 15.9% of 74180464 rows
Read 19.9% of 74180464 rows
Read 23.8% of 74180464 rows
Read 27.7% of 74180464 rows
Read 31.7% of 74180464 rows
Read 35.7% of 74180464 rows
Read 39.6% of 74180464 rows
Read 43.5% of 74180464 rows
Read 47.3% of 74180464 rows
Read 51.1% of 74180464 rows
Read 54.9% of 74180464 rows
Read 58.5% of 74180464 rows
Read 62.4% of 74180464 rows
Read 66.3% of 74180464 rows
Read 70.3% of 74180464 rows
Read 74.1% of 74180464 rows
Read 78.0% of 74180464 rows
Read 81.9% of 74180464 rows
Read 85.6% of 74180464 rows
Read 89.4% of 74180464 rows
Read 93.2% of 74180464 rows
Read 97.0% of 74180464 rows
Read 74180464 rows and 11 (of 11) columns from 2.980 GB file in 00:00:32
```

```
test <- fread("test.csv", stringsAsFactors=TRUE, data.table = FALSE)
client <- fread("cliente_tabla.csv", stringsAsFactors=TRUE, data.table = FALSE)
product <- fread("producto_tabla.csv", stringsAsFactors=TRUE, data.table = FALSE)
sample.sub <- fread("sample_submission.csv", stringsAsFactors=TRUE, data.table = FALSE)
town.state <- fread("town_state.csv", stringsAsFactors=TRUE, data.table = FALSE)
```

Data Fields

1. Semana — Week number (From Thursday to Wednesday)
2. Agencia_ID — Sales Depot ID
3. Canal_ID — Sales Channel ID
4. Ruta_SAK — Route ID (Several routes = Sales Depot)
5. Cliente_ID — Client ID
6. NombreCliente — Client name
7. Producto_ID — Product ID
8. NombreProducto — Product Name
9. Venta_uni_hoy — Sales unit this week (integer)
10. Venta_hoy — Sales this week (unit: pesos)
11. Dev_uni_proxima — Returns unit next week (integer)
12. Dev_proxima — Returns next week (unit: pesos)
13. Demanda_uni_equil — Adjusted Demand (integer) (This is the target you will predict)

Things to note:

There may be products in the test set that don’t exist in the train set. This is the expected behavior of inventory data, since there are new products being sold all the time. Your model should be able to accommodate this.
There are duplicate Cliente_ID’s in cliente_tabla, which means one Cliente_ID may have multiple NombreCliente that are very similar. This is due to the NombreCliente being noisy and not standardized in the raw data, so it is up to you to decide how to clean up and use this information.
The adjusted demand (Demanda_uni_equil) is always >= 0 since demand should be either 0 or a positive value. The reason that Venta_uni_hoy - Dev_uni_proxima sometimes has negative values is that the returns records sometimes carry over a few weeks.

train data

```
dim(train)
```

```
## [1] 74180464       11
```

```
colnames(train) #data for train data set
```

```
##  [1] "Semana"            "Agencia_ID"        "Canal_ID"
##  [4] "Ruta_SAK"          "Cliente_ID"        "Producto_ID"
##  [7] "Venta_uni_hoy"     "Venta_hoy"         "Dev_uni_proxima"
## [10] "Dev_proxima"       "Demanda_uni_equil"
```

```
head(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
```

test data

```
dim(test)
```

```
## [1] 6999251       7
```

```
colnames(test)  #data for test data set
```

```
## [1] "id"          "Semana"      "Agencia_ID"  "Canal_ID"    "Ruta_SAK"
## [6] "Cliente_ID"  "Producto_ID"
```

```
head(test)
```

```
##   id Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID
## 1  0     11       4037        1     2209    4639078       35305
## 2  1     11       2237        1     1226    4705135        1238
## 3  2     10       2045        1     2831    4549769       32940
## 4  3     11       1227        1     4448    4717855       43066
## 5  4     11       1219        1     1130     966351        1277
## 6  5     11       1146        4     6601    1741414         972
```

common parameters between train and test

```
colnames(train)[colnames(train) %in% colnames(test)]
```

```
## [1] "Semana"      "Agencia_ID"  "Canal_ID"    "Ruta_SAK"    "Cliente_ID" 
## [6] "Producto_ID"
```

town_state
Town and State data were added to train/test data.
Information about Town/State was stored as integer.

```
head(town.state)
```

```
##   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.
```

```
train$Town <- rep(NA,nrow(train))
loop <- length(town.state$Agencia_ID)

for (i in 1:loop){
  train$Town[which(train$Agencia_ID == town.state$Agencia_ID[i])] <- town.state$Town[town.state$A
gencia_ID == town.state$Agencia_ID[i]]
}
gc()
```

```
##             used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells    758474   40.6    1442291   77.1    940480   50.3
## Vcells 553950896 4226.4  958321926 7311.5 926312353 7067.3
```

```
test$Town <- rep(NA,nrow(test))
loop <- length(town.state$Agencia_ID)

for (i in 1:loop){
  test$Town[which(test$Agencia_ID == town.state$Agencia_ID[i])] <- town.state$Town[town.state$Age
ncia_ID == town.state$Agencia_ID[i]]
}
gc()
```

```
##            used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells    758525   40.6    1442291   77.1    940480   50.3
## Vcells 557450591 4253.1  958321926 7311.5 957363879 7304.2
```

```
train$State <- rep(NA,nrow(train))
loop <- length(town.state$Agencia_ID)

for (i in 1:loop){
  train$State[which(train$Agencia_ID == town.state$Agencia_ID[i])] <- town.state$State[town.state$Ag
encia_ID == town.state$Agencia_ID[i]]
}
gc()
```

```
##            used   (Mb) gc trigger   (Mb)   max used   (Mb)
## Ncells    758561   40.6    1442291   77.1     940480   50.3
## Vcells 594540867 4536.0 1150066311 8774.4 1115719761 8512.3
```

```
test$State <- rep(NA,nrow(test))
loop <- length(town.state$Agencia_ID)

for (i in 1:loop){
  test$State[which(test$Agencia_ID == town.state$Agencia_ID[i])] <- town.state$State[town.state$Age
ncia_ID == town.state$Agencia_ID[i]]
}
gc()
```

```
##            used   (Mb) gc trigger   (Mb)   max used   (Mb)
## Ncells    758603   40.6    1442291   77.1     940480   50.3
## Vcells 598040563 4562.7 1150066311 8774.4 1148719651 8764.1
```

cliente_tabla

```
head(client)
```

```
##   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
```

```
counts <- client %>% group_by(NombreCliente) %>% summarise(Count = length(Cliente_ID))
counts <- counts[order(-counts$Count),]
head(counts,10)
```

```
## Source: local data frame [10 x 2]
##
##      NombreCliente  Count
##             (fctr)  (int)
## 1  NO IDENTIFICADO 281670
## 2           LUPITA   4863
## 3             MARY   3016
## 4      LA PASADITA   2426
## 5     LA VENTANITA   2267
## 6   LA GUADALUPANA   1299
## 7             ROSY   1245
## 8             ALEX   1242
## 9             GABY   1238
## 10    LA ESCONDIDA   1216
```

```
sum(counts$Count[counts$Count == 1])/nrow(client) # the percentage of not duplicated name
```

```
## [1] 0.294888
```

About 70 % of client ID was duplicated. It will be better to remove client ID as a predictor variable.

producto_tabla

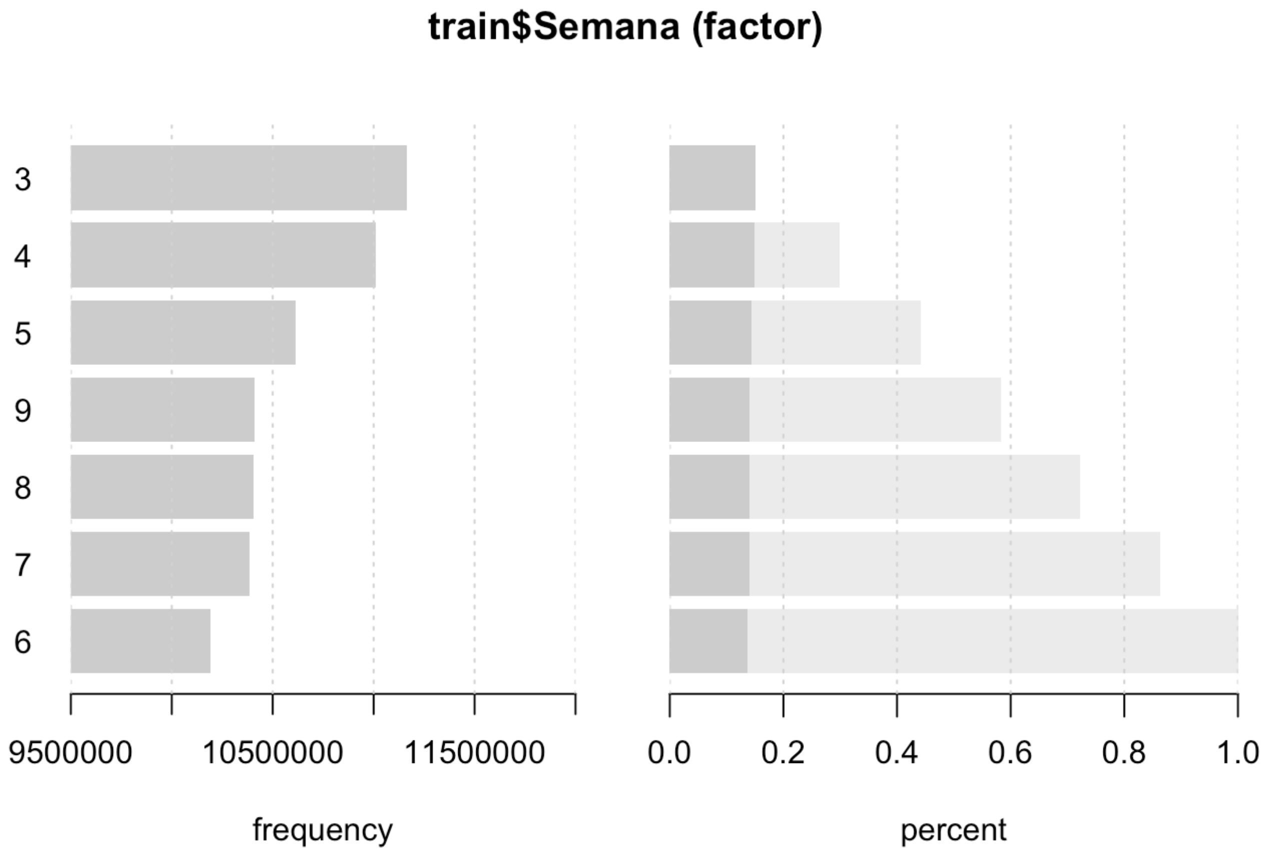
```
head(product)
```

```
##   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
```

Replacement will be not needed. No duplicate ID was found.

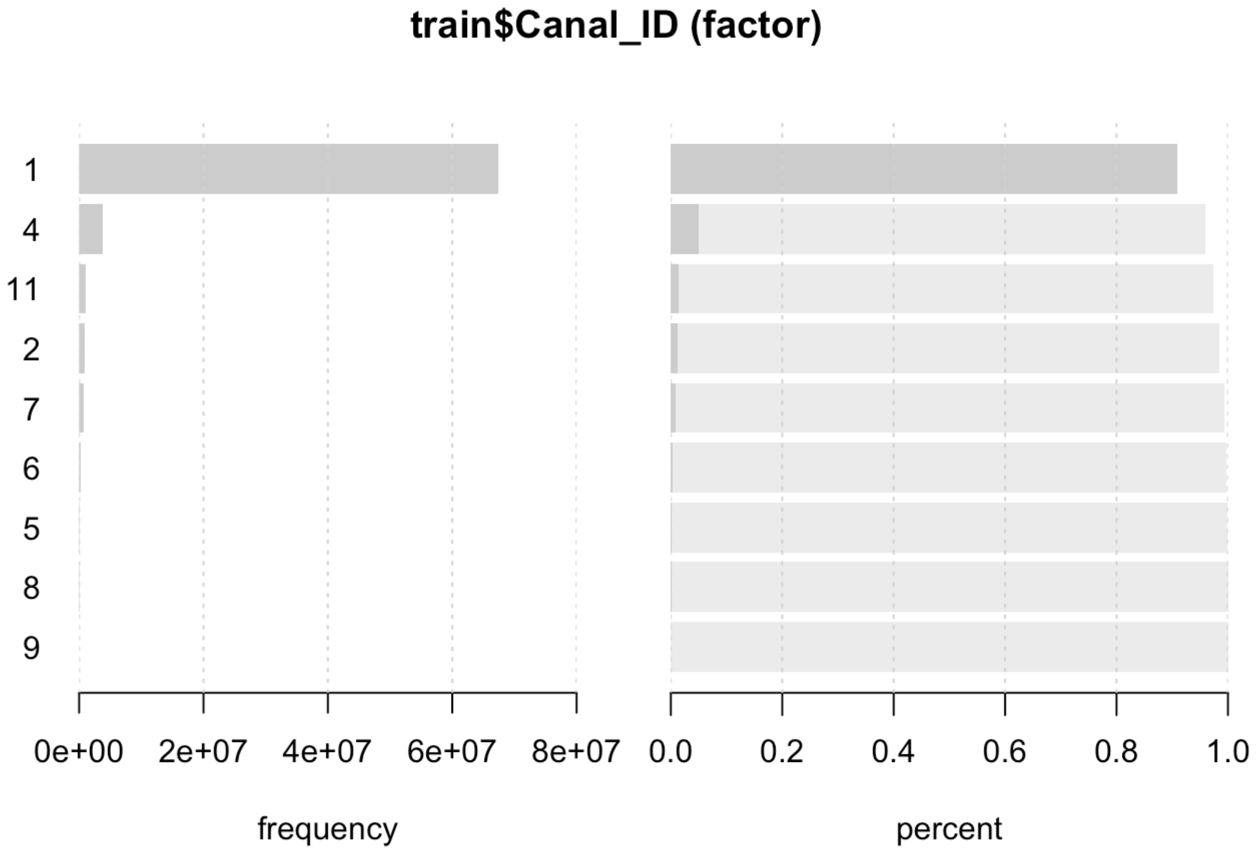
```
Desc(train$Semana, plotit = TRUE, digits=7)
```

```
## ------------------------------------------------------------------------
## train$Semana (factor)
##
##   length       n   NAs unique levels  dupes
##   7e+07  7e+07     0  7e+00  7e+00      y
##
##   level   freq         perc  cumfreq        cumperc
## 1     3  1e+07  15.0514116%   1e+07   15.0514116%
## 2     4  1e+07  14.8416340%   2e+07   29.8930457%
## 3     5  1e+07  14.3102327%   3e+07   44.2032784%
## 4     9  1e+07  14.0316095%   4e+07   58.2348878%
## 5     8  1e+07  14.0291223%   5e+07   72.2640101%
## 6     7  1e+07  13.9967431%   6e+07   86.2607532%
## 7     6  1e+07  13.7392468%   7e+07  100.0000000%
```



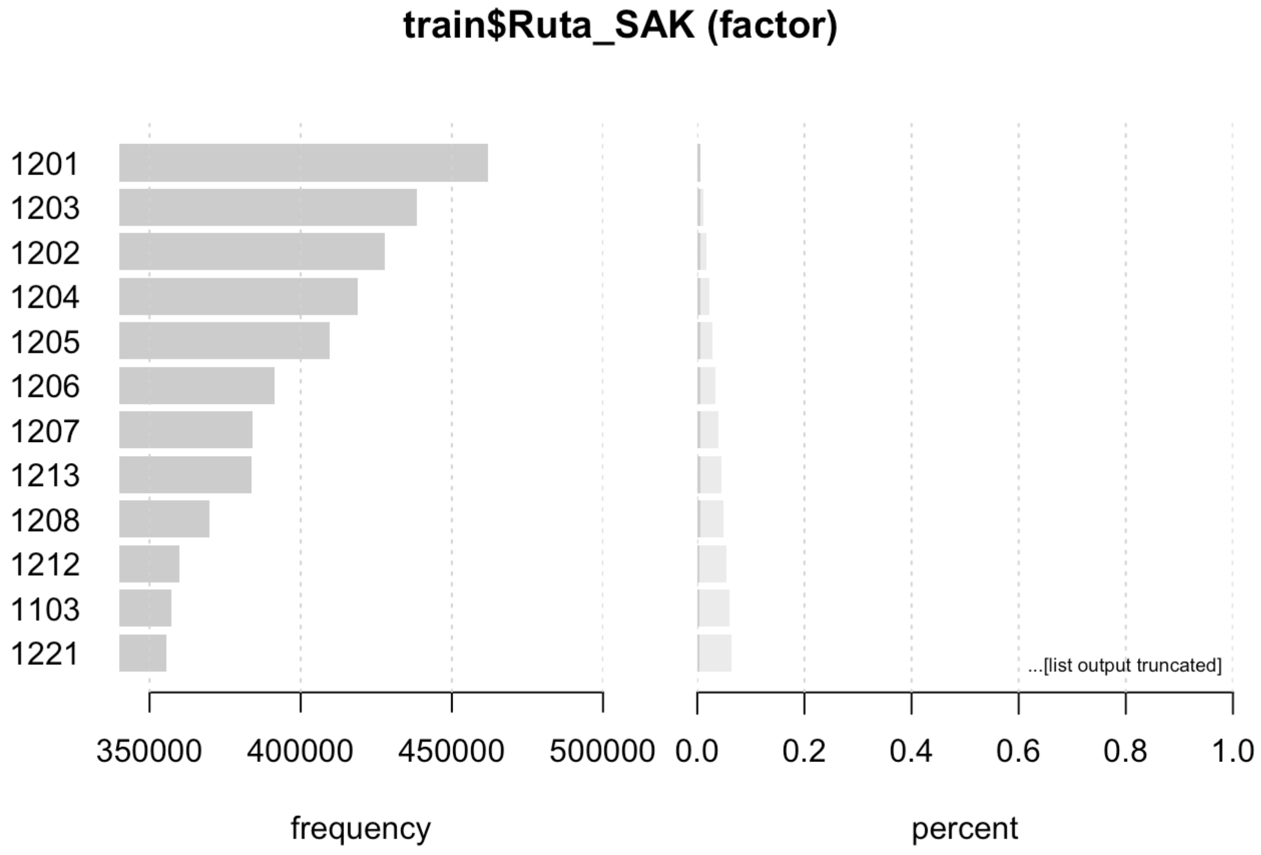
```
Desc(train$Canal_ID, plotit = TRUE, digits=9)
```

```
## ------------------------------------------------------------------------
## train$Canal_ID (factor)
##
##   length      n    NAs unique levels  dupes
##    7e+07  7e+07      0  9e+00  9e+00      y
##
##    level   freq          perc  cumfreq        cumperc
## 1      1  7e+07  90.906976532%   7e+07   90.906976532%
## 2      4  4e+06   5.065717842%   7e+07   95.972694374%
## 3     11  1e+06   1.324196354%   7e+07   97.296890729%
## 4      2  8e+05   1.131694188%   7e+07   98.428584917%
## 5      7  7e+05   0.904723378%   7e+07   99.333308295%
## 6      6  3e+05   0.379330332%   7e+07   99.712638627%
## 7      5  1e+05   0.196571971%   7e+07   99.909210598%
## 8      8  7e+04   0.090279834%   7e+07   99.999490432%
## 9      9  4e+02   0.000509568%   7e+07  100.000000000%
```



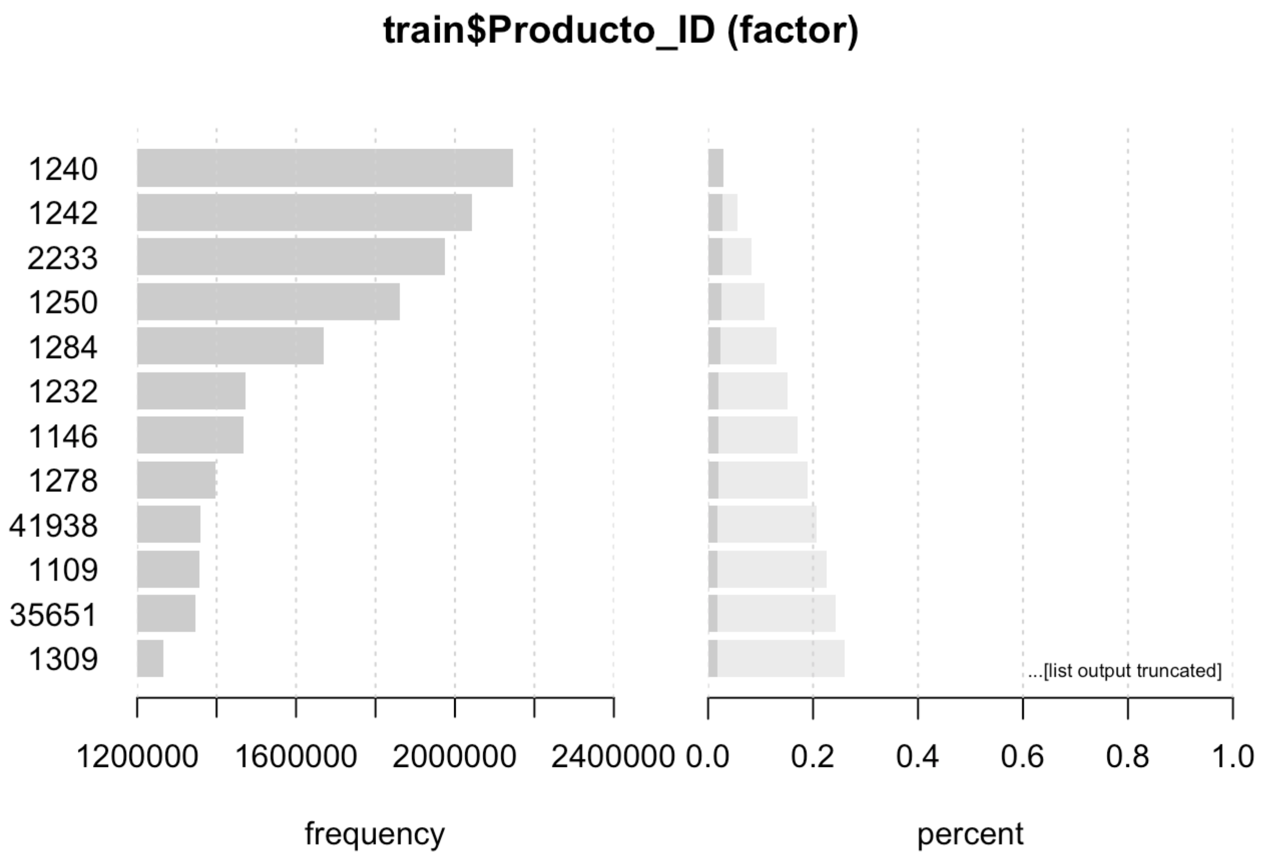
```
Desc(train$Ruta_SAK, plotit = TRUE)
```

```
## ------------------------------------------------------------------------
## train$Ruta_SAK (factor)
##
##   length      n    NAs unique levels  dupes
##    7e+07  7e+07      0  4e+03  4e+03      y
##
##     level   freq  perc  cumfreq  cumperc
## 1    1201  5e+05  0.6%    5e+05     0.6%
## 2    1203  4e+05  0.6%    9e+05     1.2%
## 3    1202  4e+05  0.6%    1e+06     1.8%
## 4    1204  4e+05  0.6%    2e+06     2.4%
## 5    1205  4e+05  0.6%    2e+06     2.9%
## 6    1206  4e+05  0.5%    3e+06     3.4%
## 7    1207  4e+05  0.5%    3e+06     4.0%
## 8    1213  4e+05  0.5%    3e+06     4.5%
## 9    1208  4e+05  0.5%    4e+06     5.0%
## 10   1212  4e+05  0.5%    4e+06     5.5%
## 11   1103  4e+05  0.5%    4e+06     5.9%
## 12   1221  4e+05  0.5%    5e+06     6.4%
## ... etc.
##  [list output truncated]
```



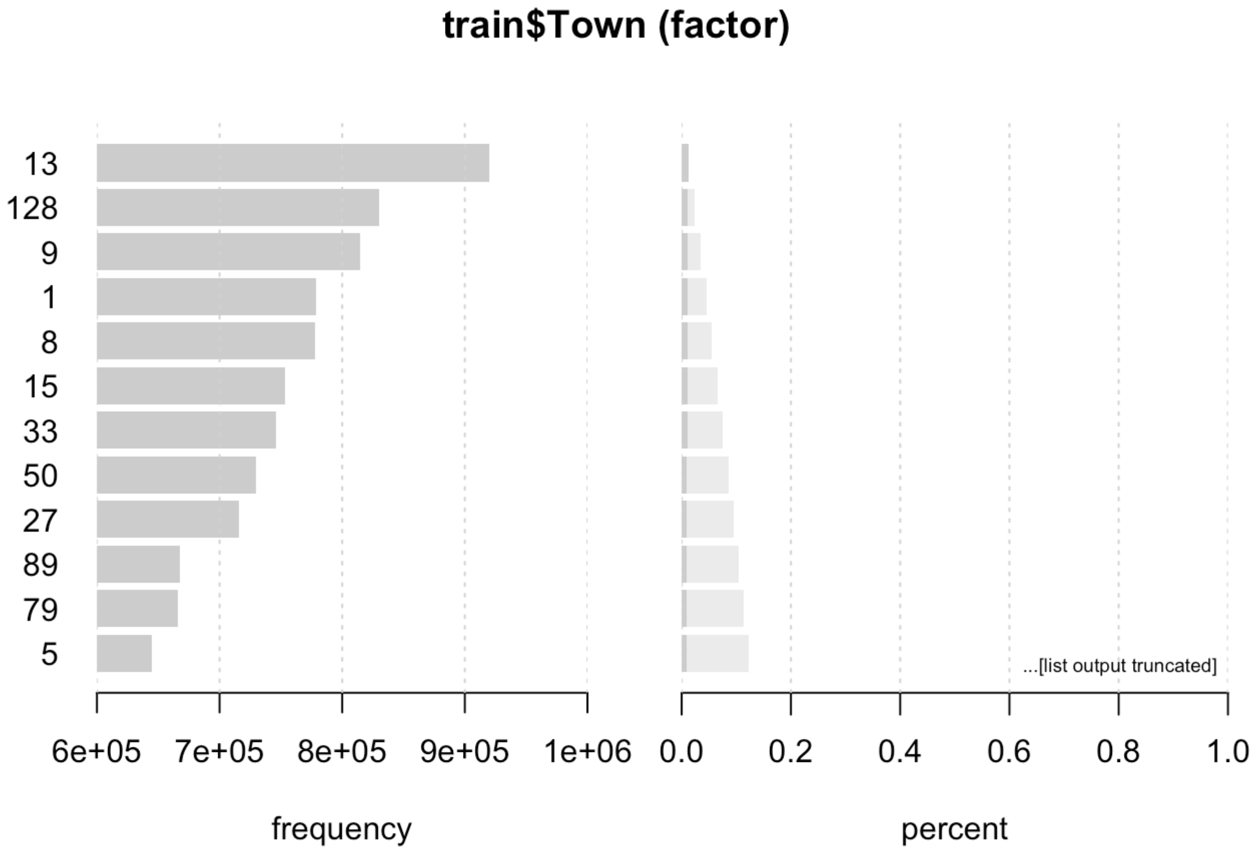
```
Desc(train$Producto_ID, plotit = TRUE)
```

```
## ------------------------------------------------------------------------
## train$Producto_ID (factor)
##
##   length      n    NAs unique levels  dupes
##    7e+07  7e+07      0  2e+03  2e+03      y
##
##     level   freq  perc  cumfreq  cumperc
## 1    1240  2e+06  2.9%    2e+06     2.9%
## 2    1242  2e+06  2.8%    4e+06     5.6%
## 3    2233  2e+06  2.7%    6e+06     8.3%
## 4    1250  2e+06  2.5%    8e+06    10.8%
## 5    1284  2e+06  2.3%    1e+07    13.1%
## 6    1232  1e+06  2.0%    1e+07    15.1%
## 7    1146  1e+06  2.0%    1e+07    17.0%
## 8    1278  1e+06  1.9%    1e+07    18.9%
## 9   41938  1e+06  1.8%    2e+07    20.8%
## 10   1109  1e+06  1.8%    2e+07    22.6%
## 11  35651  1e+06  1.8%    2e+07    24.4%
## 12   1309  1e+06  1.7%    2e+07    26.1%
## ... etc.
##  [list output truncated]
```



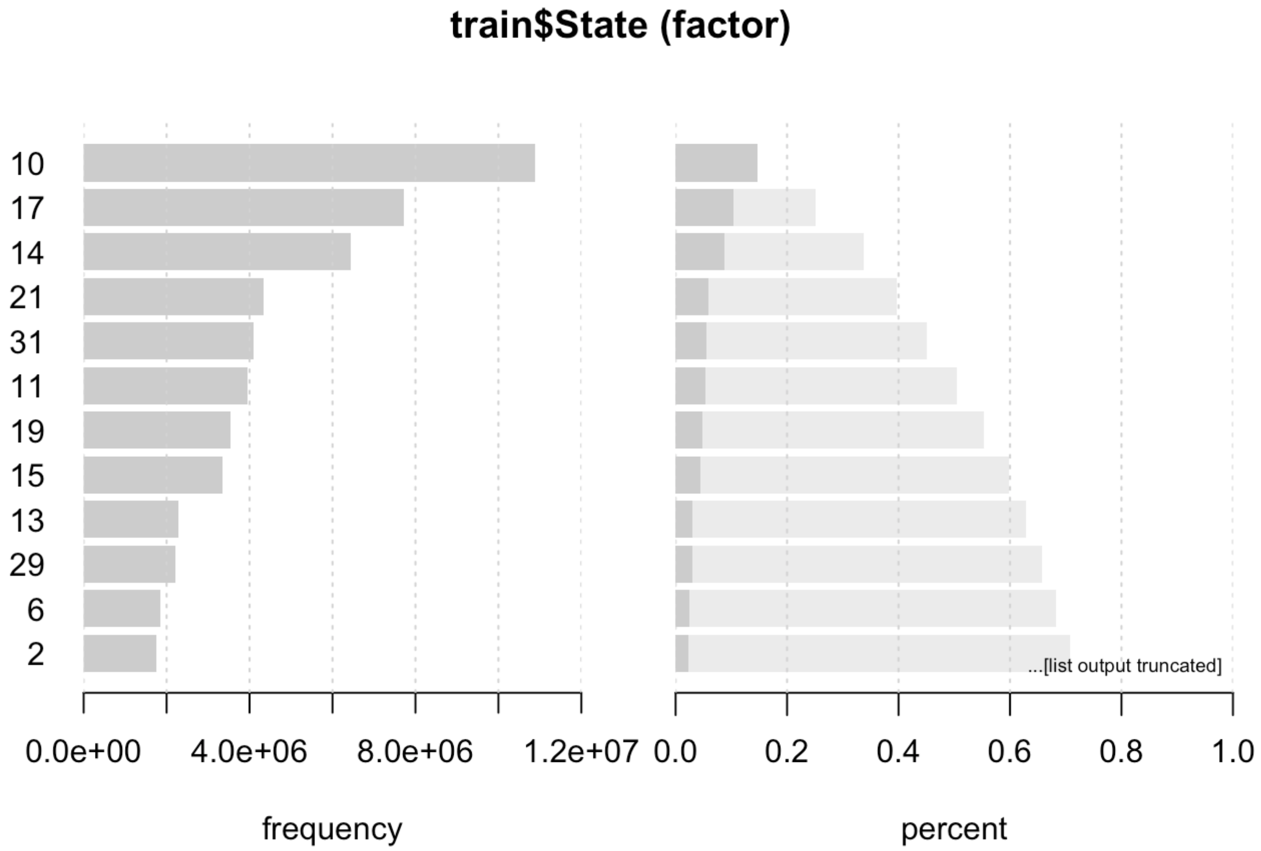
```
Desc(train$Town, plotit = TRUE)
```

```
## ------------------------------------------------------------------------
## train$Town (factor)
##
##   length      n    NAs unique levels  dupes
##   7e+07  7e+07      0  3e+02  3e+02      y
##
##     level   freq  perc  cumfreq  cumperc
## 1      13  9e+05  1.2%    9e+05     1.2%
## 2     128  8e+05  1.1%    2e+06     2.4%
## 3       9  8e+05  1.1%    3e+06     3.5%
## 4       1  8e+05  1.0%    3e+06     4.5%
## 5       8  8e+05  1.0%    4e+06     5.6%
## 6      15  8e+05  1.0%    5e+06     6.6%
## 7      33  7e+05  1.0%    6e+06     7.6%
## 8      50  7e+05  1.0%    6e+06     8.6%
## 9      27  7e+05  1.0%    7e+06     9.5%
## 10     89  7e+05  0.9%    8e+06    10.4%
## 11     79  7e+05  0.9%    8e+06    11.3%
## 12      5  6e+05  0.9%    9e+06    12.2%
## ... etc.
##  [list output truncated]
```



```
Desc(train$State, plotit = TRUE)
```

```
## ------------------------------------------------------------------------
## train$State (factor)
##
##   length      n    NAs unique levels  dupes
##    7e+07  7e+07      0  3e+01  3e+01      y
##
##     level   freq   perc  cumfreq  cumperc
## 1      10  1e+07  14.7%    1e+07    14.7%
## 2      17  8e+06  10.4%    2e+07    25.1%
## 3      14  6e+06   8.7%    3e+07    33.8%
## 4      21  4e+06   5.9%    3e+07    39.6%
## 5      31  4e+06   5.5%    3e+07    45.2%
## 6      11  4e+06   5.3%    4e+07    50.5%
## 7      19  4e+06   4.8%    4e+07    55.3%
## 8      15  3e+06   4.5%    4e+07    59.8%
## 9      13  2e+06   3.1%    5e+07    62.9%
## 10     29  2e+06   3.0%    5e+07    65.8%
## 11      6  2e+06   2.5%    5e+07    68.3%
## 12      2  2e+06   2.4%    5e+07    70.7%
## ... etc.
##  [list output truncated]
```


train$State (factor)
0.0e+00
4.0e+06
8.0e+06
1.2e+07
frequency
percent
...[list output truncated]