Projeto2 - Previsão de demanda Grupo Bimbo

Thiago Barral

1/23/2020

Carregando pacotes necessários para a avaliação

```
library(data.table)
library(dplyr)
```

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

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

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

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

```
library(DataExplorer)
library(ggplot2)
library(biglmm)
```

```
## Loading required package: DBI
```

```
library(Metrics)
```

Carregando e limpando os dados

De acordo com a descrição dos dados as variáveis "Venta_uni_hoy" e "Venta_hoy" representam a mesma informação porém em unidades diferentes, o mesmo ocorre com as variáveis "Dev_uni_proxima" e "Dev_proxima". Dessa forma para essa análise deve ser retirada uma das variáveis que representam a mesma informação. Então serão retiradas as variáveis "Venta_hoy" e "Dev_proxima". A variável "Ruta_SAK" corresponde a forma como o produto chegou até o cliente dessa forma não será relevante para a previsão de demanda.

```
train <- fread(file = 'train.csv', header = TRUE, verbose = TRUE, data.table = FALSE)
```

```
##   omp_get_num_procs()            8
##   R_DATATABLE_NUM_PROCS_PERCENT  unset (default 50)
##   R_DATATABLE_NUM_THREADS        unset
##   omp_get_thread_limit()         2147483647
##   omp_get_max_threads()          8
##   OMP_THREAD_LIMIT               unset
##   OMP_NUM_THREADS                unset
##   RestoreAfterFork               true
##   data.table is using 4 threads. See ?setDTthreads.
## Input contains no \n. Taking this to be a filename to open
## [01] Check arguments
##   Using 4 threads (omp_get_max_threads()=8, nth=4)
##   NAstrings = [<<NA>>]
##   None of the NAstrings look like numbers.
##   show progress = 0
##   0/1 column will be read as integer
## [02] Opening the file
##   Opening file train.csv
##   File opened, size = 2.980GB (3199358223 bytes).
##   Memory mapped ok
## [03] Detect and skip BOM
## [04] Arrange mmap to be \0 terminated
##   \n has been found in the input and different lines can end with different line endings (e.g. mixed
## [05] Skipping initial rows if needed
##   Positioned on line 1 starting: <<Semana,Agencia_ID,Canal_ID,Rut>>
## [06] Detect separator, quoting rule, and ncolumns
##   Detecting sep automatically ...
##   sep=','  with 100 lines of 11 fields using quote rule 0
##   Detected 11 columns on line 1. This line is either column names or first data row. Line starts as:
##   Quote rule picked = 0
##   fill=false and the most number of columns found is 11
## [07] Detect column types, good nrow estimate and whether first row is column names
##   'header' changed by user from 'auto' to true
##   Number of sampling jump points = 100 because (3199358222 bytes from row 1 to eof) / (2 * 4296 jump
##   Type codes (jump 000)    : 55555557575  Quote rule 0
##   Type codes (jump 100)    : 55555557575  Quote rule 0
##   =====
##   Sampled 10047 rows (handled \n inside quoted fields) at 101 jump points
##   Bytes from first data row on line 2 to the end of last row: 3199358093
##   Line length: mean=43.10 sd=1.46 min=39 max=60
##   Estimated number of rows: 3199358093 / 43.10 = 74227396
##   Initial alloc = 81650135 rows (74227396 + 9%) using bytes/max(mean-2*sd,min) clamped between [1.1*
##   =====
## [08] Assign column names
## [09] Apply user overrides on column types
##   After 0 type and 0 drop user overrides : 55555557575
## [10] Allocate memory for the datatable
##   Allocating 11 column slots (11 - 0 dropped) with 81650135 rows
## [11] Read the data
##   jumps=[0..3052), chunk_size=1048282, total_size=3199358093
## Read 74180464 rows x 11 columns from 2.980GB (3199358223 bytes) file in 01:22.649 wall clock time
## [12] Finalizing the datatable
##   Type counts:
##          9 : int32     '5'
##          2 : float64   '7'
## =============================
##    0.001s (  0%) Memory map 2.980GB file
##    1.051s (  1%) sep=',' ncol=11 and header detection
##    0.000s (  0%) Column type detection using 10047 sample rows
##    0.450s (  1%) Allocation of 81650135 rows x 11 cols (3.954GB) of which 74180464 ( 91%) rows used
##   81.147s ( 98%) Reading 3052 chunks (0 swept) of 1.000MB (each chunk 24305 rows) using 4 threads
##    +   75.088s ( 91%) Parse to row-major thread buffers (grown 0 times)
##    +    1.214s (  1%) Transpose
##    +    4.845s (  6%) Waiting
##    0.000s (  0%) Rereading 0 columns due to out-of-sample type exceptions
##   82.649s        Total
```

```r
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
```

```r
train$Venta_hoy <- NULL
train$Dev_proxima <- NULL
train$Ruta_SAK <- NULL
```

Dividindo o objeto "train" pelas semanas

O objeto "train" é muito grande sendo oneroso trabalhar com ele, então o objeto foi dividido pelas semanas, sendo da semana 3 a semana 8, para treino do modelo de machine learning e a semana 9 sendo para testee a valiação do modelo.

```r
train_sample3 <- dplyr::filter(train, train$Semana == 3)
```

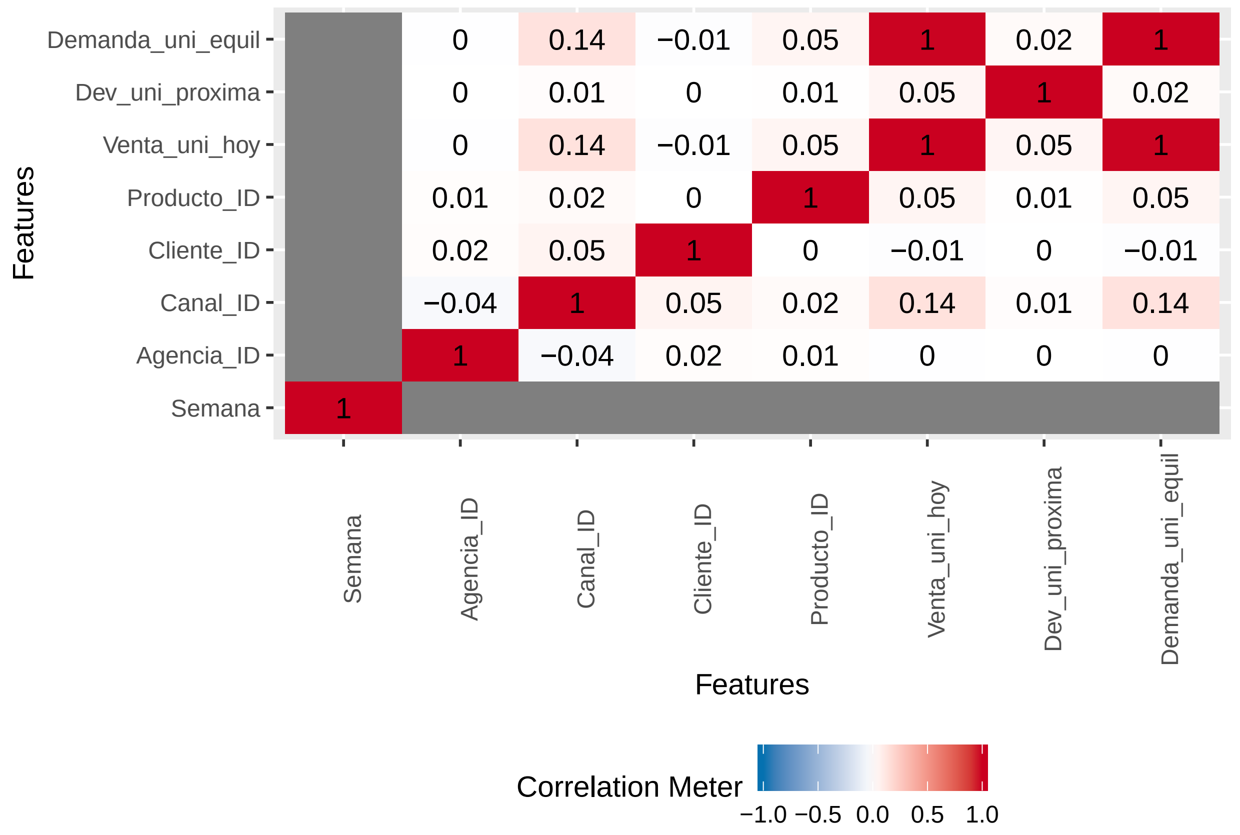
Análise de correlação

Algumas variáveis apresentam uma correlação mais forte que outras, ou seja, algumas irão apresentar uma maior relevância para a previsão de demanda. A semana 3 foi utilizada como amostra para representar todo restante.

```r
plot_correlation(data = train_sample3, type = 'all')
```

```
## Warning in cor(x = structure(list(Semana = c(3L, 3L, 3L, 3L, 3L, 3L, 3L, : the
## standard deviation is zero
```

```
## Warning: Removed 14 rows containing missing values (geom_text).
```



	Semana	Agencia_ID	Canal_ID	Cliente_ID	Producto_ID	Venta_uni_hoy	Dev_uni_proxima	Demanda_uni_equil
Demanda_uni_equil		0	0.14	−0.01	0.05	1	0.02	1
Dev_uni_proxima		0	0.01	0	0.01	0.05	1	0.02
Venta_uni_hoy		0	0.14	−0.01	0.05	1	0.05	1
Producto_ID		0.01	0.02	0	1	0.05	0.01	0.05
Cliente_ID		0.02	0.05	1	0	−0.01	0	−0.01
Canal_ID		−0.04	1	0.05	0.02	0.14	0.01	0.14
Agencia_ID		1	−0.04	0.02	0.01	0	0	0
Semana	1							

Como observado através do gráfico de correlação, as variáveis "Agencia_ID" e "Cliente_ID" não serão representativas podendo ser retiradas.

```r
train_sample3$Agencia_ID <- NULL
train_sample3$Cliente_ID <- NULL
train$Agencia_ID <- NULL
train$Cliente_ID <- NULL
```

```r
train_sample3 <- dplyr::filter(train, train$Semana == 3)
train_sample4 <- dplyr::filter(train, train$Semana == 4)
train_sample5 <- dplyr::filter(train, train$Semana == 5)
train_sample6 <- dplyr::filter(train, train$Semana == 6)
train_sample7 <- dplyr::filter(train, train$Semana == 7)
train_sample8 <- dplyr::filter(train, train$Semana == 8)
test <- dplyr::filter(train, train$Semana == 9)
rm(train)
```

Análise exploratória

A análise exploratória foi realizada para cada objeto, ou seja, por semana.

```r
table(train_sample3$Dev_uni_proxima)
```

```
##
##        0        1        2        3        4        5        6        7
## 10815787   174159    71531    30105    17844    13715     7966     4145
##        8        9       10       11       12       13       14       15
##     4223     2652     4811     1560     1990     1058     1034     1436
##     16    17    18    19    20    21    22    23
##    767   622   723   427  1113   411   349   289
##     24    25    26    27    28    29    30    31
##    449   354   225   248   248   147   452   149
##     32    33    34    35    36    37    38    39
##    150   117   123   144   139   106    87    85
##     40    41    42    43    44    45    46    47
##    756    67   187    73    63    78    61    50
##     48    49    50    51    52    53    54    55
##     85    31   115    46    51    39    53    36
##     56    57    58    59    60    61    62    63
##     37    13    25    32    72    28    31    32
##     64    65    66    67    68    69    70    71
##     36    33    28    11    25    22    37    10
##     72    73    74    75    76    77    78    79
##     22    17    11    19    17    15    12    12
##     80    81    82    83    84    85    86    87
##    117    12    13    10    24    11    14     9
##     88    89    90    91    92    94    95    96
##      7    12    29     6     8    14     9    16
##     97    98    99   100   101   102   103   104
##      5    13     7    23    10     9     2     9
##    105   106   107   108   109   110   111   112
##      5     6     1     8     3    15     4     8
##    113   114   115   117   118   119   120   121
##      2     7    10     4     1     6    46     2
##    122   123   124   125   126   127   128   129
##      5     5     3     1     5     4     6     2
##    130   131   132   133   134   135   136   137
##      3     1     8     2     5     8     4     1
##    138   139   140   141   142   143   144   145
##      3     4     5     5     4     2     6     2
##    148   149   150   152   153   154   155   156
##      2     3    12     3     2     2     1     1
##    157   158   160   161   165   166   167   168
##      1     3    35     4     4     1     1     3
##    169   171   172   175   176   177   178   179
##      1     1     1     1     1     1     1     1
##    180   181   183   184   185   186   187   188
##      6     2     1     4     2     3     4     1
##    190   191   192   195   196   198   199   200
##      3     1     2     2     1     2     1    12
##    202   207   208   209   210   211   212   214
##      1     1     1     1     2     1     2     1
##    216   219   220   222   225   226   228   229
##      1     1     2     3     2     1     1     1
##    230   231   234   235   237   238   239   240
##      1     1     1     1     2     1     1     3
##    241   243   250   252   253   256   258   260
##      1     1     1     2     1     2     1     3
##    264   265   268   273   274   280   281   284
##      1     1     3     1     1     6     1     1
##    285   288   291   295   300   310   316   320
##      1     1     1     1     3     2     1     3
##      323      324      326      330      336      341      345      346
##        1        1        1        2        1        1        1        3
##      348      353      360      370      376      384      392      400
##        1        1        4        1        1        1        1        1
##      404      410      440      470      474      500      509      520
##        1        1        1        1        1        1        1        2
##      555      570      576      587      591      600      608      660
##        1        1        1        1        1        2        1        1
##      672      697      705      750      778      800      807      900
##        1        1        1        1        1        1        1        1
##      904     1000     1008     1100     1160     1230     1250     1258
##        1        1        1        1        1        1        1        1
##     1370     1630     2400     2418     4104    16345
##        1        1        1        1        1        1
```

```r
summary(train_sample3$Dev_uni_proxima)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##     0.000     0.000     0.000     0.115     0.000 16345.000
```

```r
quantile(train_sample3$Dev_uni_proxima, .999)
```

```
## 99.9%
##    16
```

```r
any(is.na(train_sample3))
```

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

```r
any(train_sample3$Dev_uni_proxima<0)
```

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

```r
table(train_sample4$Dev_uni_proxima)
```

```
##
##        0        1        2        3        4        5        6        7
## 10668140   167414    69439    29631    17526    13225     8150     4201
##        8        9       10       11       12       13       14       15
##     4349     2827     5088     1652     2073     1027     1099     1587
##       16       17       18       19       20       21       22       23
##      864      589      684      487     1194      429      371      300
##       24       25       26       27       28       29       30       31
##      437      383      258      270      254      188      467      138
##       32       33       34       35       36       37       38       39
##      180      166      137      163      149       88      103       88
##       40       41       42       43       44       45       46       47
##      964       75      171       82       81       92       57       79
##       48       49       50       51       52       53       54       55
##      109       38      120       51       63       46       52       45
##       56       57       58       59       60       61       62       63
##       45       31       33       32       87       18       30       27
##       64       65       66       67       68       69       70       71
##       42       26       34       25       37       18       32       21
##       72       73       74       75       76       77       78       79
##       33       18       17       22       19       16       25       18
##       80       81       82       83       84       85       86       87
##      174       20        6        7       23       19       10       13
##        88        89        90        91        92        93        94        95
##         9         7        21        10         6         8        15        11
##        96        97        98        99       100       101       102       103
##        11         9         1         6        37         4        13         4
##       104       105       106       107       108       109       110       111
##         5         5         9         9        10         5        16         4
##       112       113       114       115       116       117       118       119
##         5         4         4         6         4         3         4         6
##       120       121       122       123       124       125       126       127
##        56         5         5         5         2         6         6         2
##       128       129       130       131       132       133       134       135
##         5         2         8         1         4         3         4         5
##       136       137       138       140       141       142       143       144
##         1         6         3         7         5         3         4         6
##       145       146       147       148       149       150       151       152
##         1         1         3         5         1         7         1         2
##       153       154       155       156       158       159       160       162
##         2         1         3         3         1         2        20         3
##       163       164       165       166       167       168       170       171
##         1         2         3         1         1         6         2         1
##       174       175       176       177       179       180       181       182
##         1         2         2         1         2         2         1         3
##       184       186       187       188       189       191       192       193
##         1         3         2         2         1         1         1         2
##       194       195       198       200       205       207       208       210
##         1         1         2        12         4         1         1         2
##       212       214       215       216       217       218       219       220
##         1         1         1         2         1         1         1         3
##       223       225       228       230       232       233       237       239
##         1         2         2         1         1         1         1         1
##       240       241       242       244       245       246       250       255
##         4         1         1         1         1         2         4         1
##       258       260       269       270       272       277       279       280
##         3         2         1         2         1         1         1         1
##       282       284       288       291       293       294       300       302
##         2         1         1         1         1         1         2         1
##       303       310       316       320       322       324       325       332
##         1         1         1         4         1         3         1         1
##       345       355       360       362       371       380       398       400
##         1         1         3         2         1         1         1         1
##       402       416       420       450       452       456       485       489
##         1         1         1         1         1         1         1         1
##       496       500       510       520       534       551       603       619
##         1         1         1         2         1         1         1         1
##       648       656       667       709       713       720       800       807
##         1         1         1         1         1         2         2         1
##       843       845       854       909      1146      1189      1232      1288
##         1         1         1         1         1         1         1         1
##      1371      1550      1588      1853      1973      3838      3960      5030
##         1         1         1         1         1         1         1         1
##     12760
##         1
```

```
summary(train_sample4$Dev_uni_proxima)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##     0.00     0.00     0.00     0.12     0.00 12760.00
```

```
quantile(train_sample4$Dev_uni_proxima, .999)
```

```
## 99.9% 
##    17
```

```
any(is.na(train_sample4))
```

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

```
any(train_sample4$Dev_uni_proxima<0)
```

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

```
table(train_sample5$Dev_uni_proxima)
```

```
## 
##        0        1        2        3        4        5        6        7 
## 10292311   158652    64464    28226    16774    12749     8007     4030 
##        8        9       10       11       12       13       14       15 
##     4407     2657     4977     1548     1995     1037     1047     1423 
##       16       17       18       19       20       21       22       23 
##      833      604      660      486     1148      391      367      273 
##       24       25       26       27       28       29       30       31 
##      407      342      237      220      239      163      436      131 
##       32       33       34       35       36       37       38       39 
##      156      135       94      135      167       83       84       70 
##       40       41       42       43       44       45       46       47 
##      904       75      165       48       71       77       49       63 
##       48       49       50       51       52       53       54       55 
##       79       43      106       33       50       28       62       32 
##       56       57       58       59       60       61       62       63 
##       28       25       16       24       73       16       26       26 
##       64       65       66       67       68       69       70       71 
##       32       39       21       18       19       12       29       19 
##       72       73       74       75       76       77       78       79 
##       30        8        7       20        9       16       12       15 
##       80       81       82       83       84       85       86       87 
##      146        8       14        3       23        8        8        8 
##       88       89       90       91       92       93       94       95 
##        8        7       18        2        8       11        7        7 
##       96       97       98       99      100      101      102      103 
##        7        8        7        7       27        4        7        7 
##      104      105      106      107      108      109      110      111 
##        3        8        5        2        9        4       15        2 
##      112      113      114      115      116      117      118      119 
##        4        3        4        3        3        4        5        2 
##      120      121      122      123      124      125      126      127 
##       52        5        3        4        3        3        4        3 
##      128      129      130      131      132      133      134      135 
##        6        2        4        5        6        3        2        5 
##      136      137      138      139      140      141      142      143 
##        2        3        1        1        6        4        2        2
##      144      145      147      149      150      151      152      153
##        4        3        1        1        5        2        2        3
##      154      155      156      157      158      159      160      161
##        1        1        3        1        1        2       18        2
##      162      164      165      168      169      170      171      175
##        2        3        1        5        2        3        1        4
##      176      177      178      179      180      183      184      185
##        1        1        4        1        8        1        3        1
##      186      187      188      189      190      191      192      196
##        2        2        2        2        4        1        2        1
##      197      198      200      201      203      204      205      206
##        1        1        8        2        2        1        1        1
##      209      210      211      212      213      218      220      225
##        2        3        3        1        2        1        1        1
##      230      234      235      236      240      241      242      245
##        2        1        2        1        7        1        1        1
##      246      250      270      273      280      282      284      285
##        1        2        1        1        5        2        1        1
##      288      289      297      300      302      303      310      314
##        6        1        1        3        1        1        1        1
##      315      317      319      320      336      342      346      364
##        1        1        3        4        1        1        1        1
##      368      370      377      396      400      406      427      432
##        1        1        1        1        3        1        1        1
##      440      443      450      480      494      497      500      523
##        3        1        1        1        1        1        1        1
##      530      547      642      656      680      720      750      868
##        1        1        1        1        2        3        1        1
##      896      980     1080     1128     1175     1200     1241     1440
##        1        1        1        1        1        1        1        1
##     1740     2100     5400     5795
##        1        1        1        2
```

```r
summary(train_sample5$Dev_uni_proxima)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.000    0.000    0.000    0.115    0.000 5795.000
```

```r
quantile(train_sample5$Dev_uni_proxima, .999)
```

```
## 99.9%
##    16
```

```r
any(is.na(train_sample5))
```

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

```r
any(train_sample5$Dev_uni_proxima<0)
```

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

```r
table(train_sample6$Dev_uni_proxima)
```

```
##
##       0       1       2       3       4       5       6       7       8       9
## 9834238  178723   71036   31104   17995   13378    8644    4241    4674    2820
##      10      11      12      13      14      15      16      17      18      19
##    5086    1693    2174    1094    1117    1612     850     640     790     478
##      20     21     22     23     24     25     26     27     28     29
##    1181    427    371    329    483    381    301    252    248    185
##      30     31     32     33     34     35     36     37     38     39
##     476    165    184    138    128    135    162    115    115     81
##      40     41     42     43     44     45     46     47     48     49
##     895     66    159     68     87     88     61     75     93     55
##      50     51     52     53     54     55     56     57     58     59
##     118     53     81     44     57     36     44     43     31     28
##      60     61     62     63     64     65     66     67     68     69
##      83     28     25     31     28     28     28     21     24     26
##      70     71     72     73     74     75     76     77     78     79
##      37     20     28     15     16     20     12     18     18     13
##      80     81     82     83     84     85     86     87     88     89
##     172      6     10     10     25      7     11      8     12      3
##      90     91     92     93     94     95     96     97     98     99
##      16      6      4     11     14     12     13      8      4      5
##     100    101    102    103    104    105    106    107    108    109
##      30      6      7      4     17      8      6      5      8      4
##     110    111    112    113    114    115    116    117    118    119
##      12      1      9      5      4      8      7      7      3      6
##     120    121    122    123    124    125    126    127    128    129
##      52      3      2      1      4      3      9      3      4      3
##     130    131    132    133    134    135    137    138    139    140
##       5      2      8      1      2      3      6      2      5      4
##     141    142    144    145    146    147    148    150    151    152
##       3      2      5      1      3      2      3      6      2      4
##     154    155    156    157    158    160    161    162    163    165
##       1      2      2      1      2     16      3      2      1      3
##     166    167    168    169    170    171    172    173    174    175
##       1      4      7      2      5      2      2      1      2      4
##     176    178    180    181    182    184    186    187    188    189
##       2      1      3      2      1      1      2      2      3      1
##     190    192    193    194    195    198    200    202    203    208
##       1      4      2      1      1      1     10      2      1      1
##     209    210    212    213    216    218    219    220    226    229
##       1      3      1      2      1      1      1      1      2      1
##     230    231    234    235    240    242    244    249    250    251
##       1      2      1      1     11      1      1      1      1      1
##     260    263    266    267    268    270    272    273    280    282
##       1      1      1      1      1      1      1      1      5      2
##     288    289    291    297    298    300    305    309    310    312
##       2      1      1      1      1      3      1      1      1      1
##     321    323    330    347    350    353    360    364    373    376
##       1      1      1      1      1      1      2      1      1      1
##     381    388    390    394    396    400    410    415    416    428
##       1      1      1      1      1      3      1      1      1      1
##     435    440    444    446    453    463    470    480    482    489
##       1      1      1      1      1      1      1      2      1      1
##     520    521    532    549    598    627    705    720    819    882
##       2      1      1      1      1      1      1      1      1      1
##     960    980   1017   1200   1258   1484   1550   1680   1848   2044
##       2      1      1      1      1      1      1      1      1      1
##    3600
##       1
```

```r
summary(train_sample6$Dev_uni_proxima)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00    0.00    0.00    0.13    0.00 3600.00
```

```r
quantile(train_sample6$Dev_uni_proxima, .999)
```

```
## 99.9% 
##    18
```

```r
any(is.na(train_sample6))
```

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

```r
any(train_sample6$Dev_uni_proxima<0)
```

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

```r
table(train_sample7$Dev_uni_proxima)
```

```
## 
##       0       1       2       3       4       5       6       7       8       9 
## 9995019  193473   78631   33797   19528   14660    9191    4518    4837    3031 
##      10      11      12      13      14      15      16      17      18      19 
##    5527    1780    2283    1167    1168    1608     899     664     769     488 
##      20      21      22      23      24      25      26      27      28      29 
##    1236     417     436     361     471     407     284     250     233     185 
##      30      31      32      33      34      35      36      37      38      39 
##     520     157     184     140     146     132     153      79     126      97 
##      40      41      42      43      44      45      46      47      48      49 
##    1021      76     167      69      74      82      60      85      95      40 
##      50      51      52      53      54      55      56      57      58      59 
##     115      37      92      45      60      34      50      20      42      28 
##      60      61      62      63      64      65      66      67      68      69 
##      92      18      31      23      25      42      31      23      21      12 
##      70      71      72      73      74      75      76      77      78      79 
##      40       9      22      18      14      23      18      11      16      13 
##      80      81      82      83      84      85      86      87      88      89 
##     161       6      17       7      28      17      19      12       8       8 
##      90      91      92      93      94      95      96      97      98      99 
##      27       6      10       8      20       7      13       9       8       6 
##     100     101     102     103     104     105     106     107     108     109 
##      39       5       1       4      24       5       5       4       9       5 
##     110     111     112     113     114     115     116     117     118     119 
##      11       6       6       2       9       4       5       7       2       6 
##     120     121     122     123     124     125     126     127     128     129 
##      65       4       4       4       3       1       5       5       5       2 
##     130     131     132     134     135     137     138     139     140     141 
##       6       4      11       3       2       3       4       3       5       5 
##     142     143     144     145     147     148     149     150     151     152 
##       4       1       1       9       3       3       4       7       2       4 
##     153     155     156     157     158     160     161     163     164     165 
##       2       5       9       2       5      27       3       2       1       2 
##     166     168     171     172     174     175     176     179     180     181 
##       1       7       1       1       1       1       2       1       9       1 
##     184     185     186     187     188     189     191     192     193     195 
##       2       1       1       1       2       1       2       3       2       1
##     198     199     200     201     203     206     208     209     210     213
##       1       1      15       1       1       2       5       1       3       1
##     214     216     218     219     220     221     223     226     227     228
##       1       1       1       2       2       1       2       2       1       1
##     230     231     233     234     235     240     241     244     245     249
##       1       1       1       1       2       8       1       2       2       1
##     251     253     255     256     257     258     259     260     261     270
##       2       1       1       1       2       2       1       2       1       2
##     272     274     276     279     281     282     284     288     289     290
##       1       1       1       1       1       1       1       3       1       1
##     301     302     304     306     308     311     312     320     324     329
##       1       1       1       1       1       2       1       3       2       1
##     332     337     344     349     350     356     358     360     370     374
##       1       1       1       1       1       1       1       2       1       1
##     377     390     400     401     404     416     417     420     423     428
##       1       1       2       1       1       1       1       1       1       1
##     443     460     463     469     470     480     487     494     500     510
##       1       1       1       1       1       1       1       1       1       1
##     528     588     596     600     632     640     682     705     720     740
##       1       1       1       1       1       1       1       1       1       1
##     750     766     779     949    1000    1020    1105    1116    1148    1200
##       1       1       1       1       2       1       1       1       1       1
##    1300    1478    1551    2240    2520    2772    2800
##       1       1       1       1       1       1       1
```

```r
summary(train_sample7$Dev_uni_proxima)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
##    0.0000    0.0000    0.0000    0.1365    0.0000 2800.0000
```

```r
quantile(train_sample7$Dev_uni_proxima, .999)
```

```
## 99.9% 
##    18
```

```r
any(is.na(train_sample7))
```

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

```r
any(train_sample7$Dev_uni_proxima<0)
```

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

```r
table(train_sample8$Dev_uni_proxima)
```

```
## 
##        0        1        2        3        4        5        6        7 
## 10010732   198902    80638    34273    19838    14570     9088     4807 
##        8        9       10       11       12       13       14       15 
##     4859     2941     5413     1797     2259     1191     1164     1656 
##       16       17       18       19       20       21       22       23 
##      935      670      751      472     1294      475      378      335 
##       24       25       26       27       28       29       30       31 
##      453      384      274      287      247      188      447      165 
##       32       33       34       35       36       37       38       39 
##      172      144      119      154      167      102      114       93 
##       40       41       42       43       44       45       46       47 
##     1098       63      177       72       78       76       57       75
##    48    49    50    51    52    53    54    55
##    96    49   130    44    72    44    79    39
##    56    57    58    59    60    61    62    63
##    42    29    42    18    74    24    27    19
##    64    65    66    67    68    69    70    71
##    29    36    35    16    15    15    29    12
##    72    73    74    75    76    77    78    79
##    33    24    22    23    10    16    24    16
##    80    81    82    83    84    85    86    87
##   189    14     9    10    36    19    11    15
##    88    89    90    91    92    93    94    95
##     8     8    24    12     9    12    23    10
##    96    97    98    99   100   101   102   103
##     6     7     6     5    26     8    10     6
##   104   105   106   107   108   109   110   111
##    21     8     2     4    13    10    10     4
##   112   113   114   115   116   117   119   120
##     7     8     3     5     3     4     6    59
##   122   124   125   126   127   128   129   130
##     6     2     3     8     4     3     5     8
##   131   132   134   135   136   137   138   139
##     2     6     1     4     4     1     5     1
##   140   141   142   143   144   145   146   147
##     6     4     2     4     4     5     1     1
##   148   149   150   151   152   153   156   157
##     3     2     6     2     4     3     6     4
##   158   159   160   162   163   164   165   166
##     1     2    31     1     1     3     3     3
##   167   168   169   170   171   172   173   174
##     2     6     1     3     2     1     1     1
##   175   176   177   178   180   181   182   183
##     7     1     1     2     7     1     1     1
##   184   185   187   188   189   190   191   192
##     1     2     1     3     1     1     1     4
##   194   195   196   197   198   199   200   201
##     1     2     2     3     2     2    15     1
##   202   203   204   206   207   208   209   210
##     2     1     2     1     1     6     1     2
##   212   215   216   218   220   221   223   224
##     1     1     1     1     3     1     2     1
##   225   229   230   233   235   236   238   239
##     2     1     1     1     1     2     1     1
##   240   249   250   252   253   255   256   257
##     8     1     1     2     2     1     1     1
##   258   260   263   264   270   271   273   276
##     1     3     1     1     3     1     1     1
##   280   289   290   295   300   312   317   320
##     5     1     2     1     2     4     1     1
##   325   330   334   340   342   359   360   361
##     1     1     1     3     1     1     1     1
##   362   373   397   400   403   405   406   415
##     2     1     1     1     1     2     2     1
##   416   427   431   432   440   460   480   492
##     2     1     1     1     1     1     1     2
##      500      505      530      533      564      596      599      600
##        1        1        1        1        1        1        1        1
##      629      630      647      652      739      800      833      840
##        1        1        1        1        1        1        1        2
##      985     1120     1159     1197     1240     1389     1410     1686
##        1        1        1        1        1        1        1        2
##     1708     2604     2701     3360     6768     9765
##        1        1        1        1        1        1
```

```
summary(train_sample8$Dev_uni_proxima)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00    0.14    0.00 9765.00
```

```
quantile(train_sample8$Dev_uni_proxima, .999)
```

```
## 99.9%
##    18
```

```
any(is.na(train_sample8))
```

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

```
any(train_sample8$Dev_uni_proxima<0)
```

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

Eliminando outliers

Como pode ser observado pela análise exploratória, há vários valores extremos que podem ser julgados como erros ou valores que apresentam um fato raro e pontual. Então esses valores serão retirados do objeto de treino para não interferirem negativamente no modelo de machine learning.

```
train_sample3 <- dplyr::filter(train_sample3, train_sample3$Dev_uni_proxima < 25)
train_sample4 <- dplyr::filter(train_sample4, train_sample4$Dev_uni_proxima < 30)
train_sample5 <- dplyr::filter(train_sample5, train_sample5$Dev_uni_proxima < 20)
train_sample6 <- dplyr::filter(train_sample6, train_sample6$Dev_uni_proxima < 20)
train_sample7 <- dplyr::filter(train_sample7, train_sample7$Dev_uni_proxima < 20)
train_sample8 <- dplyr::filter(train_sample8, train_sample8$Dev_uni_proxima < 20)
```

Treinando o modelo

Agora com os dados limpos e estruturados pode se realizar o treinamento do modelo. Foi utilizado o modelo de regressão linear para essa previsão. Ométodo de avliação do modelo foi o Root Mean Squared Logarithmic Error (RMSLE).

```
treino <- full_join(train_sample3, train_sample4)
```

```
## Joining, by = c("Semana", "Canal_ID", "Producto_ID", "Venta_uni_hoy",
## "Dev_uni_proxima", "Demanda_uni_equil")
```

```
rm(train_sample3)
rm(train_sample4)
treino <- full_join(treino, train_sample5)
```

```
## Joining, by = c("Semana", "Canal_ID", "Producto_ID", "Venta_uni_hoy",
## "Dev_uni_proxima", "Demanda_uni_equil")
```

```r
rm(train_sample5)
treino <- full_join(treino, train_sample6)
```

```
## Joining, by = c("Semana", "Canal_ID", "Producto_ID", "Venta_uni_hoy",
## "Dev_uni_proxima", "Demanda_uni_equil")
```

```r
rm(train_sample6)
treino <- full_join(treino, train_sample7)
```

```
## Joining, by = c("Semana", "Canal_ID", "Producto_ID", "Venta_uni_hoy",
## "Dev_uni_proxima", "Demanda_uni_equil")
```

```r
rm(train_sample7)
treino <- full_join(treino, train_sample8)
```

```
## Joining, by = c("Semana", "Canal_ID", "Producto_ID", "Venta_uni_hoy",
## "Dev_uni_proxima", "Demanda_uni_equil")
```

```r
rm(train_sample8)
treino$Semana <- NULL
fit_lm <- bigglm(formula = Demanda_uni_equil ~ Canal_ID + Producto_ID + Venta_uni_hoy + Dev_uni_proxima
actual <- test$Demanda_uni_equil
test$Demanda_uni_equil <- NULL
test$Demanda_uni_equil <- 0
test$Semana <- NULL
prev <- predict(fit_lm, test)
```

Avaliação do modelo

Algumas previsões tiveram valores negativos mas que em sua maioria foram bem próximas de zero, como nesse caso não há valores de demanda inferiores a zero então os valores negativos foram igualados a zero.

```r
any(prev < 0)
```

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

```r
df_prev <- as.data.frame(prev)
df_prev$V1[df_prev$V1 < 0] <- 0
df_actual <- as.data.frame(actual)
rmsle(actual = as.numeric(df_actual$actual), predicted = as.numeric(df_prev$V1))
```

```
## [1] 0.0586998
```