Mini Projeto - Data Science Academy

André Campos da Silva

21 de Novembro, 2020

Projeto - Demanda de Estoque

Construir um modelo de analise que analise os dados históricos com as demandas de estoque e seja capaz de fazer novas previsões de demanda de estoque com dados fornecidos futuramente

https://www.kaggle.com/c/grupo-bimbo-inventory-demand/data (https://www.kaggle.com/c/grupo-bimbo-inventory-demand/data)

Coletando os dados

```
# Carrego os pacotes necessários para o projeto
library('tidyverse')
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library('caret')
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library('ROSE')
```

```
## Loaded ROSE 0.0-3
```

```
library('data.table')
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library('gridExtra')
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library('randomForest')
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library('DMwR')
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library('gridExtra')
library('caTools')
library('e1071')
library('rpart')
```

```
# Carrego os dados de treino que sera tratado e usado para a analise e treinamento.

client_tbl <- read_csv('Dados/cliente_tabla.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Cliente_ID = col_double(),
##   NombreCliente = col_character()
## )
```

```
produto_tbl <- read_csv('Dados/producto_tabla.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Producto_ID = col_double(),
##   NombreProducto = col_character()
## )
```

```
estado_tbl <- read_csv('Dados/town_state.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Agencia_ID = col_double(),
##   Town = col_character(),
##   State = col_character()
## )
```

```
train <- read_csv('Dados/train_sample.csv')
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Semana = col_double(),
##   Agencia_ID = col_double(),
##   Canal_ID = col_double(),
##   Ruta_SAK = col_double(),
##   Cliente_ID = col_double(),
##   Producto_ID = col_double(),
##   Venta_uni_hoy = col_double(),
##   Venta_hoy = col_double(),
##   Dev_uni_proxima = col_double(),
##   Dev_proxima = col_double(),
##   Demanda_uni_equil = col_double()
## )
```

```
# Faço uma verificação do formato dos dados e das primeiras linhas e verifico se
# exisite algum valor nulo que precise se tratado.
glimpse(train)
```

```
## Rows: 118,688
## Columns: 11
## $ Semana            <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3...
## $ Agencia_ID        <dbl> 1110, 1110, 1110, 1110, 1110, 1110, 1110, 1110, 1...
## $ Canal_ID          <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 11, 1, 1, 1, 1, ...
## $ Ruta_SAK          <dbl> 3301, 3302, 3306, 3308, 3309, 3311, 3312, 3316, 3...
## $ Cliente_ID        <dbl> 50395, 99974, 4316770, 1355493, 124805, 16119, 96...
## $ Producto_ID       <dbl> 47611, 31719, 5328, 36410, 37057, 2233, 31392, 34...
## $ Venta_uni_hoy     <dbl> 14, 5, 14, 2, 23, 28, 1, 2, 10, 2, 4, 18, 2, 7, 1...
## $ Venta_hoy         <dbl> 240.10, 37.95, 114.10, 26.00, 172.50, 558.32, 22....
## $ Dev_uni_proxima   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Dev_proxima       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Demanda_uni_equil <dbl> 14, 5, 14, 2, 23, 28, 1, 2, 10, 2, 4, 18, 2, 7, 1...
```

```
glimpse(client_tbl)
```

```
## Rows: 935,362
## Columns: 2
## $ Cliente_ID    <dbl> 0, 1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ...
## $ NombreCliente <chr> "SIN NOMBRE", "OXXO XINANTECATL", "SIN NOMBRE", "EL M...
```

```
glimpse(produto_tbl)
```

```
## Rows: 2,592
## Columns: 2
## $ Producto_ID    <dbl> 0, 9, 41, 53, 72, 73, 98, 99, 100, 106, 107, 108, 10...
## $ NombreProducto <chr> "NO IDENTIFICADO 0", "Capuccino Moka 750g NES 9", "B...
```

```
glimpse(estado_tbl)
```

```
## Rows: 790
## Columns: 3
## $ Agencia_ID <dbl> 1110, 1111, 1112, 1113, 1114, 1116, 1117, 1118, 1119, 11...
## $ Town       <chr> "2008 AG. LAGO FILT", "2002 AG. AZCAPOTZALCO", "2004 AG....
## $ State      <chr> "MÉXICO, D.F.", "MÉXICO, D.F.", "ESTADO DE MÉXICO", "MÉX...
```

```
head(train)
```

```
## # A tibble: 6 x 11
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Venta_uni_hoy
##    <dbl>      <dbl>    <dbl>    <dbl>      <dbl>       <dbl>         <dbl>
## 1      3       1110        7     3301      50395       47611            14
## 2      3       1110        7     3302      99974       31719             5
## 3      3       1110        7     3306    4316770        5328            14
## 4      3       1110        7     3308    1355493       36410             2
## 5      3       1110        7     3309     124805       37057            23
## 6      3       1110        7     3311      16119        2233            28
## # ... with 4 more variables: Venta_hoy <dbl>, Dev_uni_proxima <dbl>,
## #   Dev_proxima <dbl>, Demanda_uni_equil <dbl>
```

```
head(client_tbl)
```

```
## # A tibble: 6 x 2
##   Cliente_ID NombreCliente
##        <dbl> <chr>
## 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
```

```
head(produto_tbl)
```

```
## # A tibble: 6 x 2
##   Producto_ID NombreProducto
##         <dbl> <chr>
## 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
```

```
head(estado_tbl)
```

```
## # A tibble: 6 x 3
##   Agencia_ID Town                  State
##        <dbl> <chr>                 <chr>
## 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.
```

```
any(is.na(train))
```

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

```
any(is.na(client_tbl))
```

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

```
any(is.na(produto_tbl))
```

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

```
any(is.na(estado_tbl))
```

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

Tratamento dos dados

```
# Formula para tirar os espaços entre nomes.
tira_espaco <- function(x){
    str_replace_all(x,' ', '_')
  }
```

```
# Tiro os espaços em todos os campos de todas as tabelas.
client_tbl$NombreCliente <- sapply(client_tbl$NombreCliente, tira_espaco)
produto_tbl$NombreProducto <- sapply(produto_tbl$NombreProducto, tira_espaco)
estado_tbl$Town <- sapply(estado_tbl$Town , tira_espaco)
estado_tbl$State <- sapply(estado_tbl$State , tira_espaco)
```

```
# Crio um novo dataset onde faço os joins entre as tabelas para a analise exploratória
train2 <- train %>%
  left_join(client_tbl, by = 'Cliente_ID') %>%
  left_join(produto_tbl, by = 'Producto_ID') %>%
  left_join(estado_tbl, by = 'Agencia_ID')
```

```
# Retiro as variaveis ID desse dataset pois não é necessário para a análise.
train2$Agencia_ID = NULL
train2$Canal_ID = NULL
train2$Ruta_SAK = NULL
train2$Cliente_ID = NULL
train2$Producto_ID = NULL
names(train2)
```

```
##  [1] "Semana"            "Venta_uni_hoy"     "Venta_hoy"
##  [4] "Dev_uni_proxima"   "Dev_proxima"       "Demanda_uni_equil"
##  [7] "NombreCliente"     "NombreProducto"    "Town"
## [10] "State"
```

```
head(train2)
```

```
## # A tibble: 6 x 10
##   Semana Venta_uni_hoy Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equ~
##    <dbl>         <dbl>     <dbl>           <dbl>       <dbl>            <dbl>
## 1      3            14     240.                0           0               14
## 2      3             5      38.0               0           0                5
## 3      3            14     114.                0           0               14
## 4      3             2      26                 0           0                2
## 5      3            23     172.                0           0               23
## 6      3            28     558.                0           0               28
## # ... with 4 more variables: NombreCliente <chr>, NombreProducto <chr>,
## #   Town <chr>, State <chr>
```

```
any(is.na(train2))
```

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

```
str(train2)
```

```
## tibble [119,633 x 10] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Semana           : num [1:119633] 3 3 3 3 3 3 3 3 3 3 ...
##  $ Venta_uni_hoy    : num [1:119633] 14 5 14 2 23 28 1 2 10 2 ...
##  $ Venta_hoy        : num [1:119633] 240 38 114 26 172 ...
##  $ Dev_uni_proxima  : num [1:119633] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Dev_proxima      : num [1:119633] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Demanda_uni_equil: num [1:119633] 14 5 14 2 23 28 1 2 10 2 ...
##  $ NombreCliente    : Named chr [1:119633] "BOLICHE_POLANCO" "LA_PERLA" "NO_IDENTIFICADO" "CA
FETERIA_LA_CAFETA" ...
##   ..- attr(*, "names")= chr [1:119633] "BOLICHE POLANCO" "LA PERLA" "NO IDENTIFICADO" "CAFETE
RIA LA CAFETA" ...
##  $ NombreProducto   : Named chr [1:119633] "Bimbollos_FS_8p_450g_BIM_47611" "Mantecadas_2p_10
5g_TR_31719" "Submarinos_Vainilla_3p_105g_SP_MLA_5328" "Tortilla_Consumos_24p_577g_TR_36410" ...
##   ..- attr(*, "names")= chr [1:119633] "Bimbollos FS 8p 450g BIM 47611" "Mantecadas 2p 105g T
R 31719" "Submarinos Vainilla 3p 105g SP MLA 5328" "Tortilla Consumos 24p 577g TR 36410" ...
##  $ Town             : Named chr [1:119633] "2008_AG._LAGO_FILT" "2008_AG._LAGO_FILT" "2008_A
G._LAGO_FILT" "2008_AG._LAGO_FILT" ...
##   ..- attr(*, "names")= chr [1:119633] "2008 AG. LAGO FILT" "2008 AG. LAGO FILT" "2008 AG. LA
GO FILT" "2008 AG. LAGO FILT" ...
##  $ State            : Named chr [1:119633] "MÉXICO,_D.F." "MÉXICO,_D.F." "MÉXICO,_D.F." "MÉXI
CO,_D.F." ...
##   ..- attr(*, "names")= chr [1:119633] "MÉXICO, D.F." "MÉXICO, D.F." "MÉXICO, D.F." "MÉXICO,
D.F." ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Semana = col_double(),
##   ..   Agencia_ID = col_double(),
##   ..   Canal_ID = col_double(),
##   ..   Ruta_SAK = col_double(),
##   ..   Cliente_ID = col_double(),
##   ..   Producto_ID = col_double(),
##   ..   Venta_uni_hoy = col_double(),
##   ..   Venta_hoy = col_double(),
##   ..   Dev_uni_proxima = col_double(),
##   ..   Dev_proxima = col_double(),
##   ..   Demanda_uni_equil = col_double()
##   .. )
```

Analise Exploratória

```
# Medidas de Tendência Central
summary(train2[c('Venta_uni_hoy','Venta_hoy','Dev_uni_proxima','Dev_proxima','Demanda_uni_equil'
)])
```

```
##  Venta_uni_hoy        Venta_hoy        Dev_uni_proxima     Dev_proxima
##  Min.   :   0.000   Min.   :    0.00   Min.   :  0.0000   Min.   :    0.000
##  1st Qu.:   2.000   1st Qu.:   16.76   1st Qu.:  0.0000   1st Qu.:    0.000
##  Median :   3.000   Median :   30.00   Median :  0.0000   Median :    0.000
##  Mean   :   7.324   Mean   :   68.06   Mean   :  0.1311   Mean   :    1.351
##  3rd Qu.:   7.000   3rd Qu.:   56.58   3rd Qu.:  0.0000   3rd Qu.:    0.000
##  Max.   :4796.000   Max.   :26857.60   Max.   :854.0000   Max.   :11921.840
##  Demanda_uni_equil
##  Min.   :   0.00
##  1st Qu.:   2.00
##  Median :   3.00
##  Mean   :   7.24
##  3rd Qu.:   7.00
##  Max.   :4796.00
```

```
# Total de unidades vendidas e valor total das vendas agrupado por semana.

# Existe um padrão entre a quantidade de vendas em unidades com a receite recebida por semana, a
ssim como as devoluções em unidades com os prejuizos referentes a essas devoluções.
train2 %>%
  select(Semana, Venta_uni_hoy, Venta_hoy,Dev_uni_proxima,Dev_proxima)%>%
  group_by(Semana)%>%
  summarise(Total_Unidades = sum(Venta_uni_hoy),
            Total_Lucro = sum(Venta_hoy),
            Total_Unidades_Devolvidas = sum(Dev_uni_proxima),
            Total_Devolução = sum(Dev_proxima))
```

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

```
## # A tibble: 7 x 5
##   Semana Total_Unidades Total_Lucro Total_Unidades_Devolvidas Total_Devolução
##    <dbl>          <dbl>       <dbl>                     <dbl>           <dbl>
## 1      3         129307    1252802.                      2271          22260.
## 2      4         129009    1202533.                      2730          31165.
## 3      5         122650    1109069.                      1534          16315.
## 4      6         121626    1122520.                      2144          23343.
## 5      7         128848    1232956.                      2224          22169.
## 6      8         126893    1136397.                      2171          21044.
## 7      9         117841    1085307.                      2607          25302.
```

```
p1 <- train2 %>%
  select(Semana, Venta_uni_hoy)%>%
  group_by(Semana)%>%
  summarise(Total_Unidades = sum(Venta_uni_hoy)) %>%
  ggplot(aes(x =as.factor(Semana), y =Total_Unidades)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Unidades vendidas por semana',
       x = 'semana', y = 'Quantidade')
```

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

```
p2 <- train2 %>%
  select(Semana, Venta_hoy)%>%
  group_by(Semana)%>%
  summarise(Total_Lucro = sum(Venta_hoy)) %>%
  ggplot(aes(x= as.factor(Semana), y =Total_Lucro)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'total de lucro',
       x = 'semana', y = 'Lucro-$')
```

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

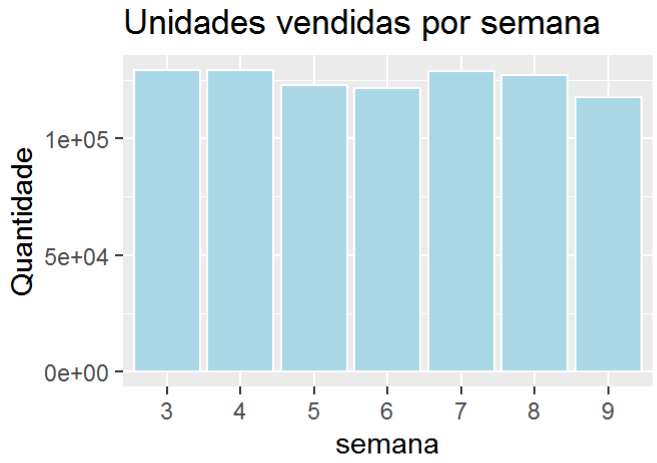
```
p3 <- train2 %>%
  select(Semana, Dev_uni_proxima)%>%
  group_by(Semana)%>%
  summarise(Total_Unidades_Devolvidas = sum(Dev_uni_proxima)) %>%
  ggplot(aes(x =as.factor(Semana), y =Total_Unidades_Devolvidas)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Total de devolução - Unit.',
       x = 'semana', y = 'Quantidade')
```

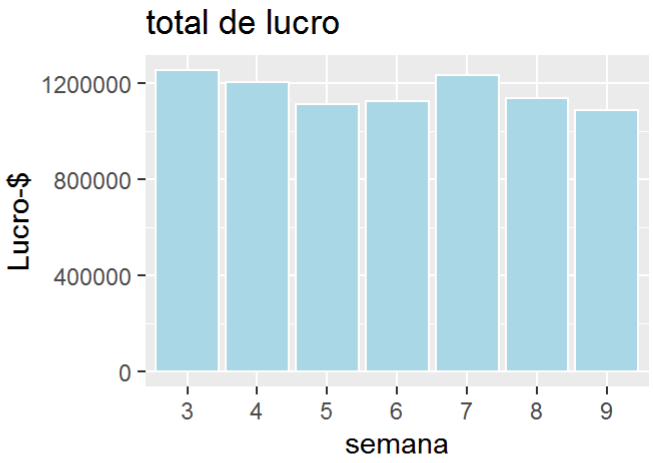
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

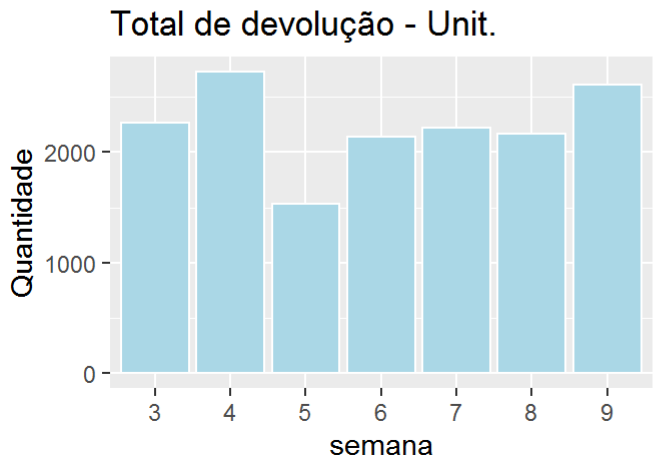
```
p4 <- train2 %>%
  select(Semana, Dev_proxima)%>%
  group_by(Semana)%>%
  summarise(Total_Devolução = sum(Dev_proxima)) %>%
  ggplot(aes(x= as.factor(Semana), y =Total_Devolução)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Total de Prejuízo ',
       x = 'semana', y = 'Prejuízo-$')
```

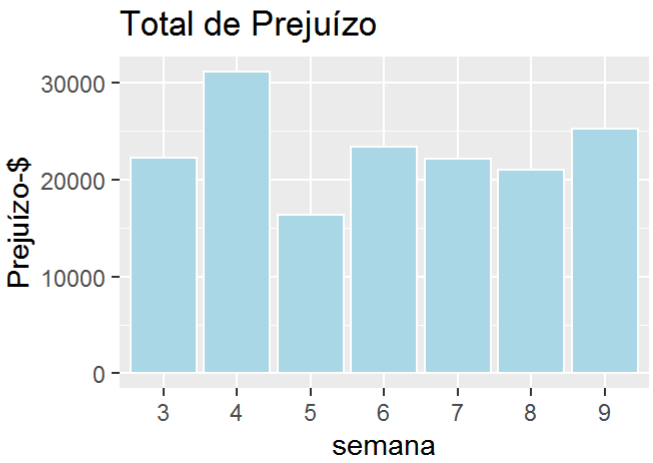
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
grid.arrange(p1,p2,p3,p4 ,nrow=2,ncol=2)
```










```
# Estado de  Mexico, Jalisco  e Mexico.DF são os Estados que mais geram vendas e lucros.
train2 %>%
  select(State, Venta_uni_hoy, Venta_hoy,Dev_uni_proxima,Dev_proxima)%>%
  group_by(State)%>%
  summarise(Total_Unidades = sum(Venta_uni_hoy),
            Total_Lucro = sum(Venta_hoy),
            Total_Unidades_Devolvidas = sum(Dev_uni_proxima),
            Total_Devolução = sum(Dev_proxima))
```

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

```
## # A tibble: 33 x 5
##    State         Total_Unidades Total_Lucro Total_Unidades_Devo~ Total_Devolução
##    <chr>                  <dbl>       <dbl>                <dbl>           <dbl>
##  1 AGUASCALIENT~          15544     128697.                  271           2603.
##  2 BAJA_CALIFOR~          24788     260012.                  598           5072.
##  3 BAJA_CALIFOR~           5989      75042.                   68            772.
##  4 CAMPECHE                6889      71338.                   58            708.
##  5 CHIAPAS                13369     142756.                  370           3188.
##  6 CHIHUAHUA              25225     261868.                  500           5319.
##  7 COAHUILA               22257     217587.                  438           4264.
##  8 COLIMA                  8314      81994.                  207           2500.
##  9 DURANGO                10681      87004.                   90            918.
## 10 ESTADO_DE_MÉ~         112428    1039019.                 1484          16614.
## # ... with 23 more rows
```

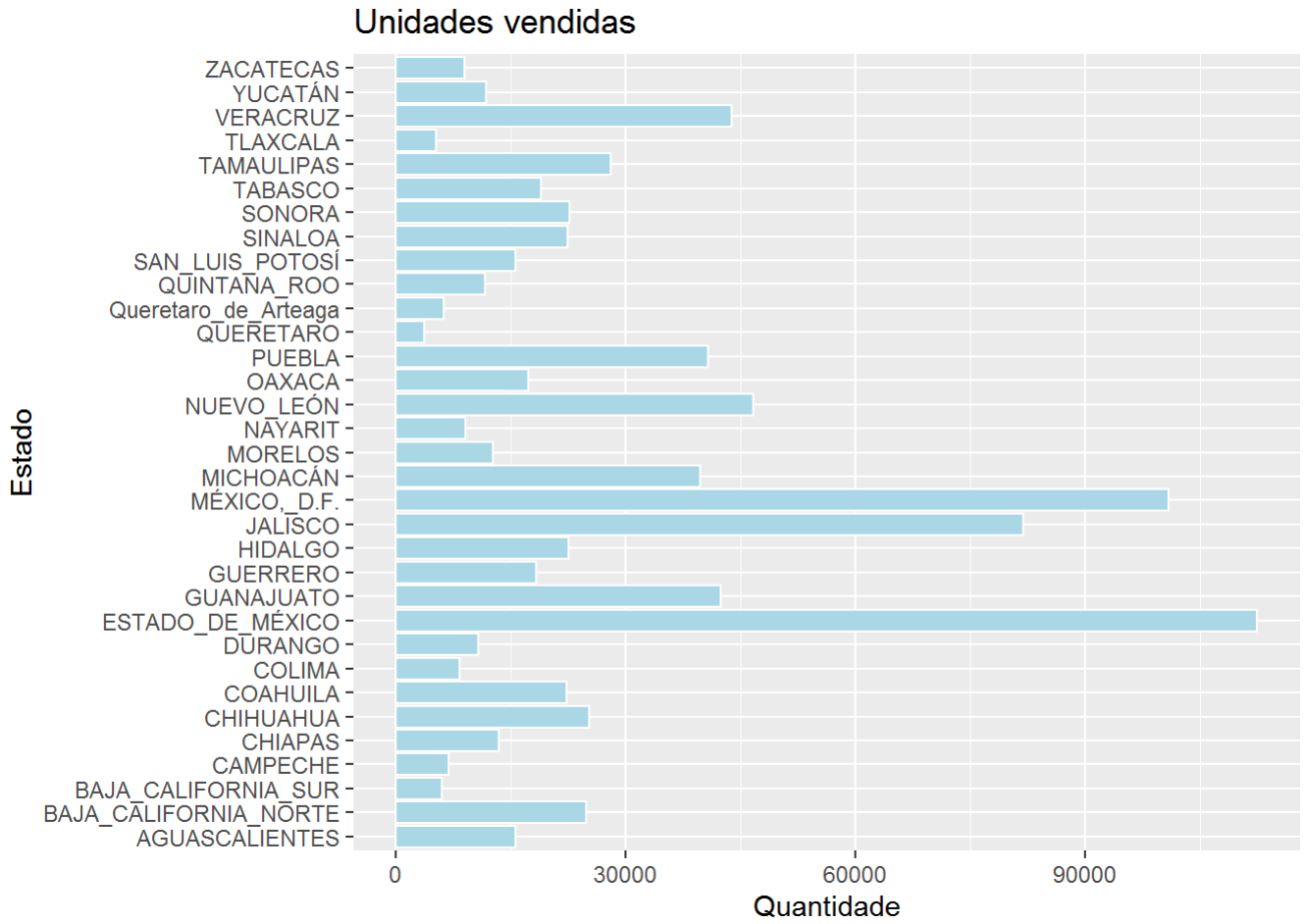
```
p5 <- train2 %>%
  select(State, Venta_uni_hoy)%>%
  group_by(State)%>%
  summarise(Total_Unidades = sum(Venta_uni_hoy)) %>%
  ggplot(aes(y =as.factor(State), x =Total_Unidades)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Unidades vendidas',
       y = 'Estado', x = 'Quantidade')
```

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

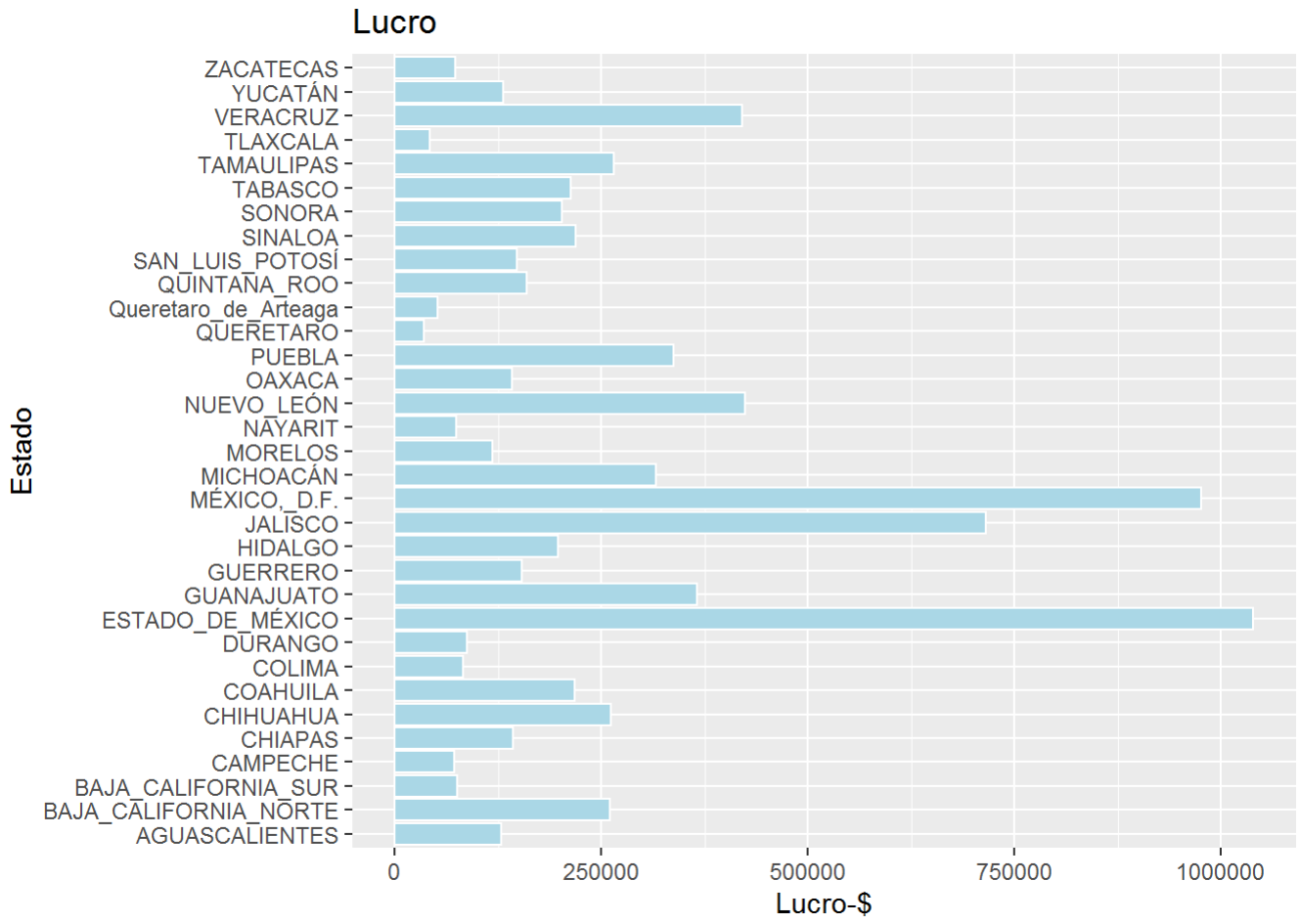
```
p6 <- train2 %>%
  select(State, Venta_hoy)%>%
  group_by(State)%>%
  summarise(Total_Lucro = sum(Venta_hoy)) %>%
  ggplot(aes(y = as.factor(State), x =Total_Lucro)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Lucro',
       y = 'Estado', x = 'Lucro-$')
```

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

```
p5
```


Unidades vendidas
Estado
ZACATECAS
YUCATÁN
VERACRUZ
TLAXCALA
TAMAULIPAS
TABASCO
SONORA
SINALOA
SAN_LUIS_POTOSÍ
QUINTANA_ROO
Queretaro_de_Arteaga
QUERETARO
PUEBLA
OAXACA
NUEVO_LEÓN
NAYARIT
MORELOS
MICHOACÁN
MÉXICO,_D.F.
JALISCO
HIDALGO
GUERRERO
GUANAJUATO
ESTADO_DE_MÉXICO
DURANGO
COLIMA
COAHUILA
CHIHUAHUA
CHIAPAS
CAMPECHE
BAJA_CALIFORNIA_SUR
BAJA_CALIFORNIA_NORTE
AGUASCALIENTES
0
30000
60000
90000
Quantidade

p6



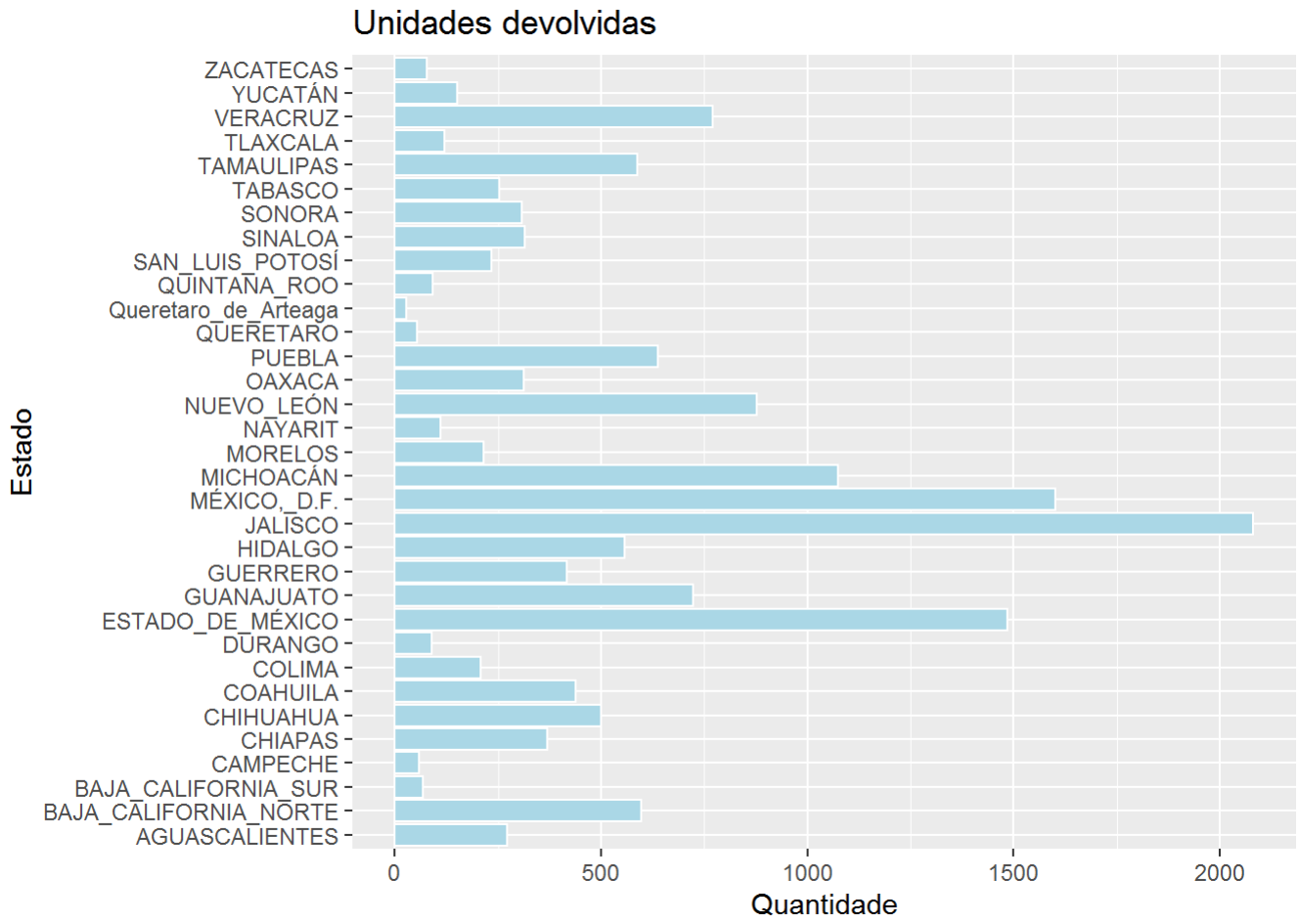

```
# Por serem os estados que mais compra os produtos, consequentemente são os que mais devolvem pa
ra troca.
p7 <- train2 %>%
  select(State, Dev_uni_proxima)%>%
  group_by(State)%>%
  summarise(Total_Unidades_Devolvidas = sum(Dev_uni_proxima)) %>%
  ggplot(aes(y=as.factor(State), x =Total_Unidades_Devolvidas)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Unidades devolvidas',
       y = 'Estado', x = 'Quantidade')
```

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

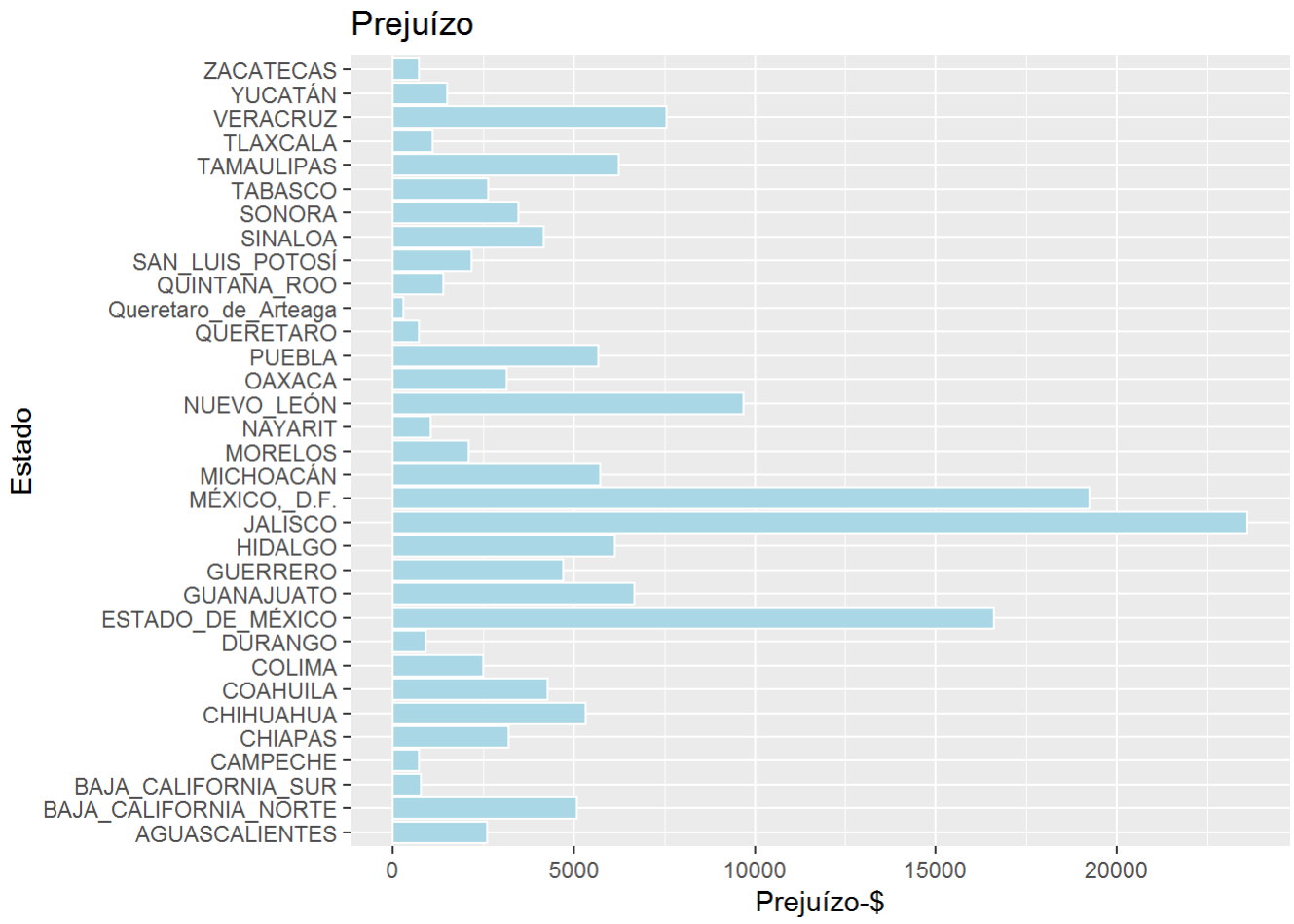
```
p8 <- train2 %>%
  select(State, Dev_proxima)%>%
  group_by(State)%>%
  summarise(Total_Devolução = sum(Dev_proxima)) %>%
  ggplot(aes(y = as.factor(State), x =Total_Devolução)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Prejuízo ',
       y = 'Estado', x = 'Prejuízo-$')
```

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

p7


Unidades devolvidas
Estado
ZACATECAS
YUCATÁN
VERACRUZ
TLAXCALA
TAMAULIPAS
TABASCO
SONORA
SINALOA
SAN_LUIS_POTOSÍ
QUINTANA_ROO
Queretaro_de_Arteaga
QUERETARO
PUEBLA
OAXACA
NUEVO_LEÓN
NAYARIT
MORELOS
MICHOACÁN
MÉXICO,_D.F.
JALISCO
HIDALGO
GUERRERO
GUANAJUATO
ESTADO_DE_MÉXICO
DURANGO
COLIMA
COAHUILA
CHIHUAHUA
CHIAPAS
CAMPECHE
BAJA_CALIFORNIA_SUR
BAJA_CALIFORNIA_NORTE
AGUASCALIENTES
0
500
1000
1500
2000
Quantidade

p8



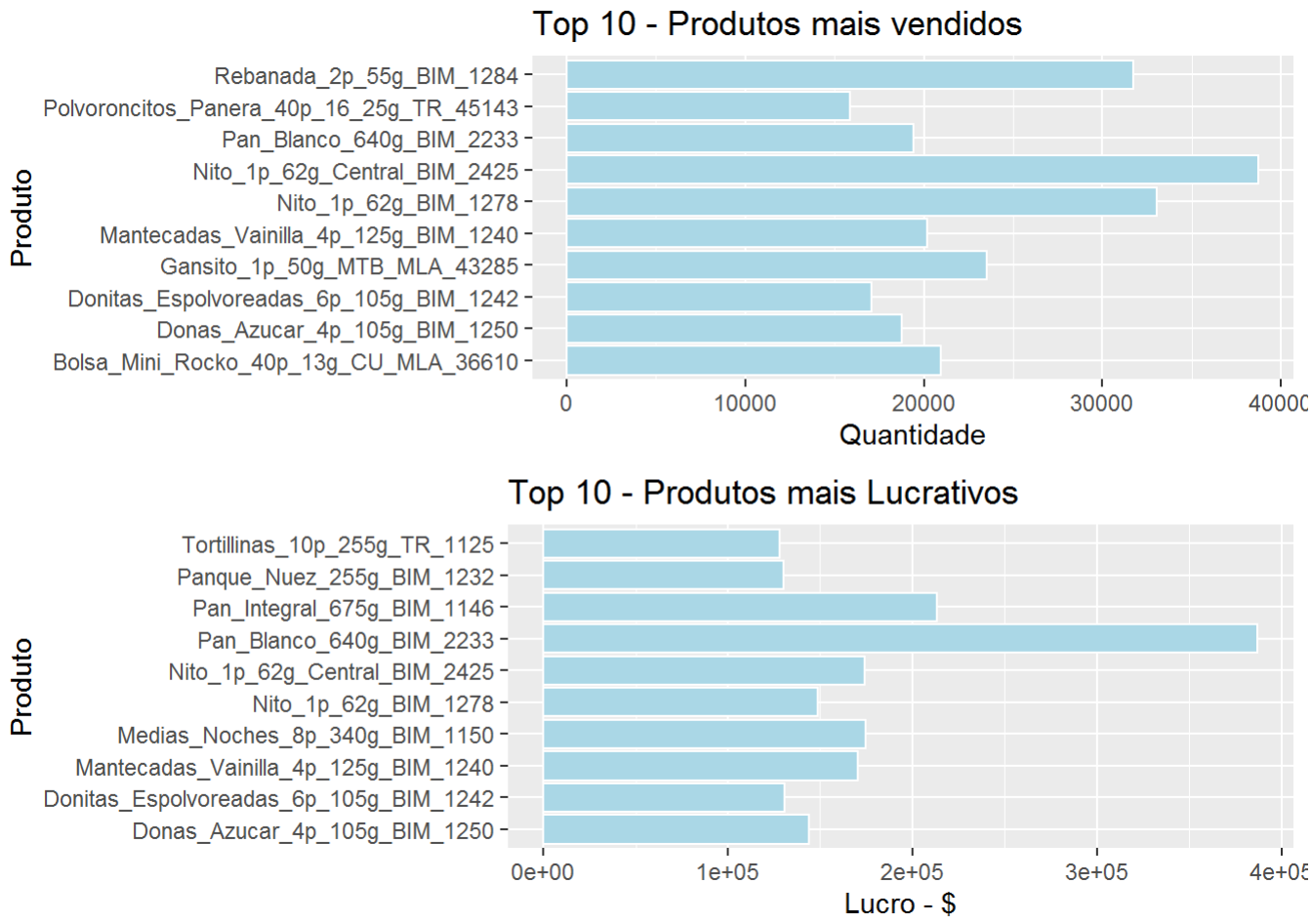

```
# Top 10 dos produtos mais vendidos por unidade e Top 10 dos mais Lucrativos.
p9 <- train2 %>%
  select(NombreProducto, Venta_uni_hoy)%>%
  group_by(NombreProducto)%>%
  summarise(Total_Unidades = sum(Venta_uni_hoy)) %>%
  filter(Total_Unidades >= 15800)%>%
  ggplot(aes(y = as.factor(NombreProducto), x =Total_Unidades)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Top 10 - Produtos mais vendidos ',
       y = 'Produto', x = 'Quantidade')
```

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

```
p10 <- train2 %>%
  select(NombreProducto, Venta_hoy)%>%
  group_by(NombreProducto)%>%
  summarise(Lucro= sum(Venta_hoy)) %>%
  filter(Lucro >= 125000) %>%
  ggplot(aes(y = as.factor(NombreProducto), x = Lucro)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Top 10 - Produtos mais Lucrativos ',
       y = 'Produto', x = 'Lucro - $')
```

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

```
grid.arrange(p9,p10,nrow=2,ncol=1)
```



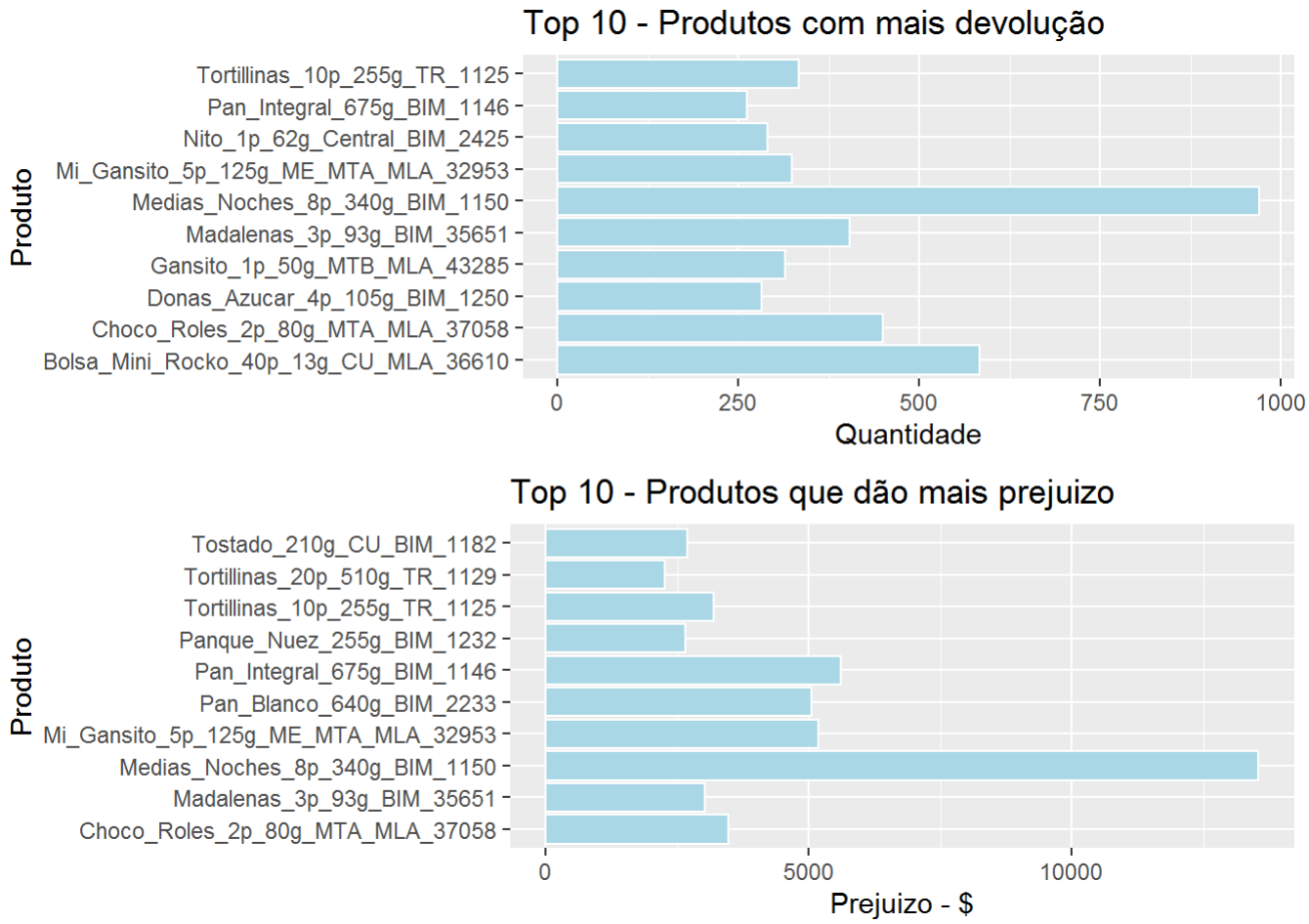

```
# Top 10 dos produtos com mais devolução e o top 10 dos que dão mais prejuízo.
p11 <- train2 %>%
  select(NombreProducto, Dev_uni_proxima)%>%
  group_by(NombreProducto)%>%
  summarise(Quantidade_devolucao = sum(Dev_uni_proxima)) %>%
  filter(Quantidade_devolucao >= 260) %>%
  ggplot(aes(y = as.factor(NombreProducto), x =  Quantidade_devolucao)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Top 10 - Produtos com mais devolução',
       y = 'Produto', x = 'Quantidade')
```

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

```
p12 <- train2 %>%
  select(NombreProducto, Dev_proxima)%>%
  group_by(NombreProducto)%>%
  summarise(Prejuizo = sum(Dev_proxima)) %>%
  filter(Prejuizo >= 2270) %>%
  ggplot(aes(y = as.factor(NombreProducto), x =  Prejuizo)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Top 10 - Produtos que dão mais prejuizo',
       y = 'Produto', x = 'Prejuizo - $')
```

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

```
grid.arrange(p11,p12,nrow=2,ncol=1)
```



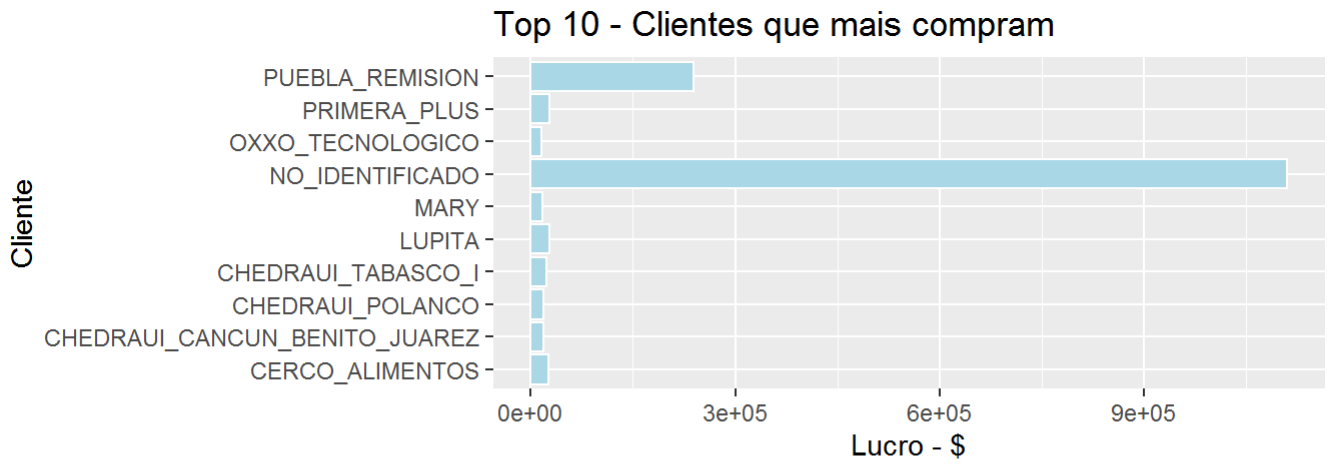
```
# Top 10 dos clientes que mais compram e os que mais geram prejuizo.
p13 <- train2 %>%
  select(NombreCliente, Venta_hoy)%>%
  group_by(NombreCliente)%>%
  summarise(Total = sum(Venta_hoy)) %>%
  filter(Total >= 15000)%>%
  ggplot(aes(y = as.factor(NombreCliente), x =  Total)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Top 10 - Clientes que mais compram',
       y = 'Cliente', x = 'Lucro - $')
```

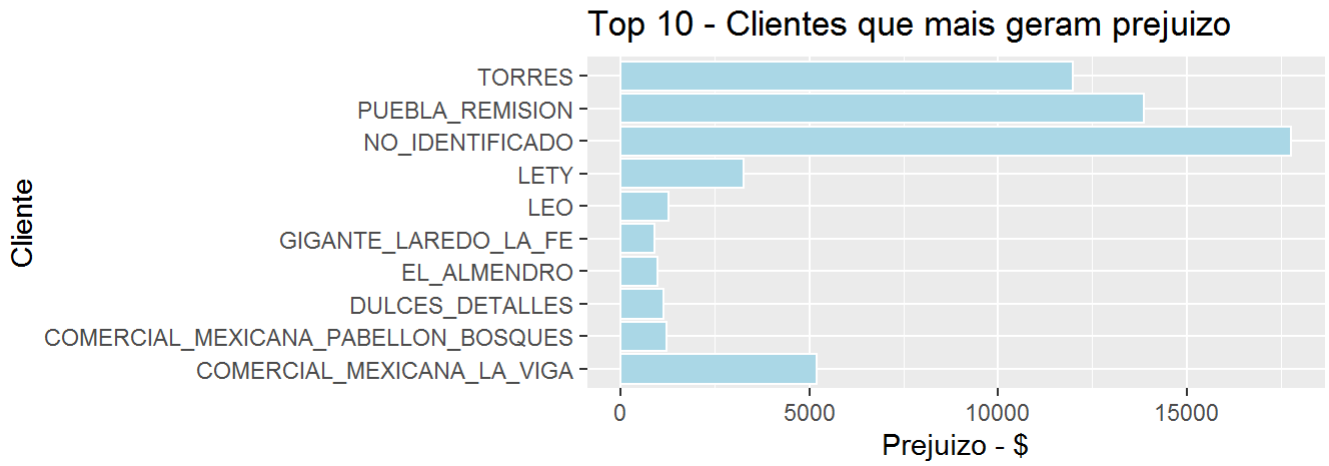
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p14 <- train2 %>%
  select(NombreCliente, Dev_proxima)%>%
  group_by(NombreCliente)%>%
  summarise(Total = sum(Dev_proxima)) %>%
  filter(Total >= 820)%>%
  ggplot(aes(y = as.factor(NombreCliente), x =  Total)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Top 10 - Clientes que mais geram prejuizo',
       y = 'Cliente', x = 'Prejuizo - $')
```

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

```
grid.arrange(p13,p14,nrow=2,ncol=1)
```

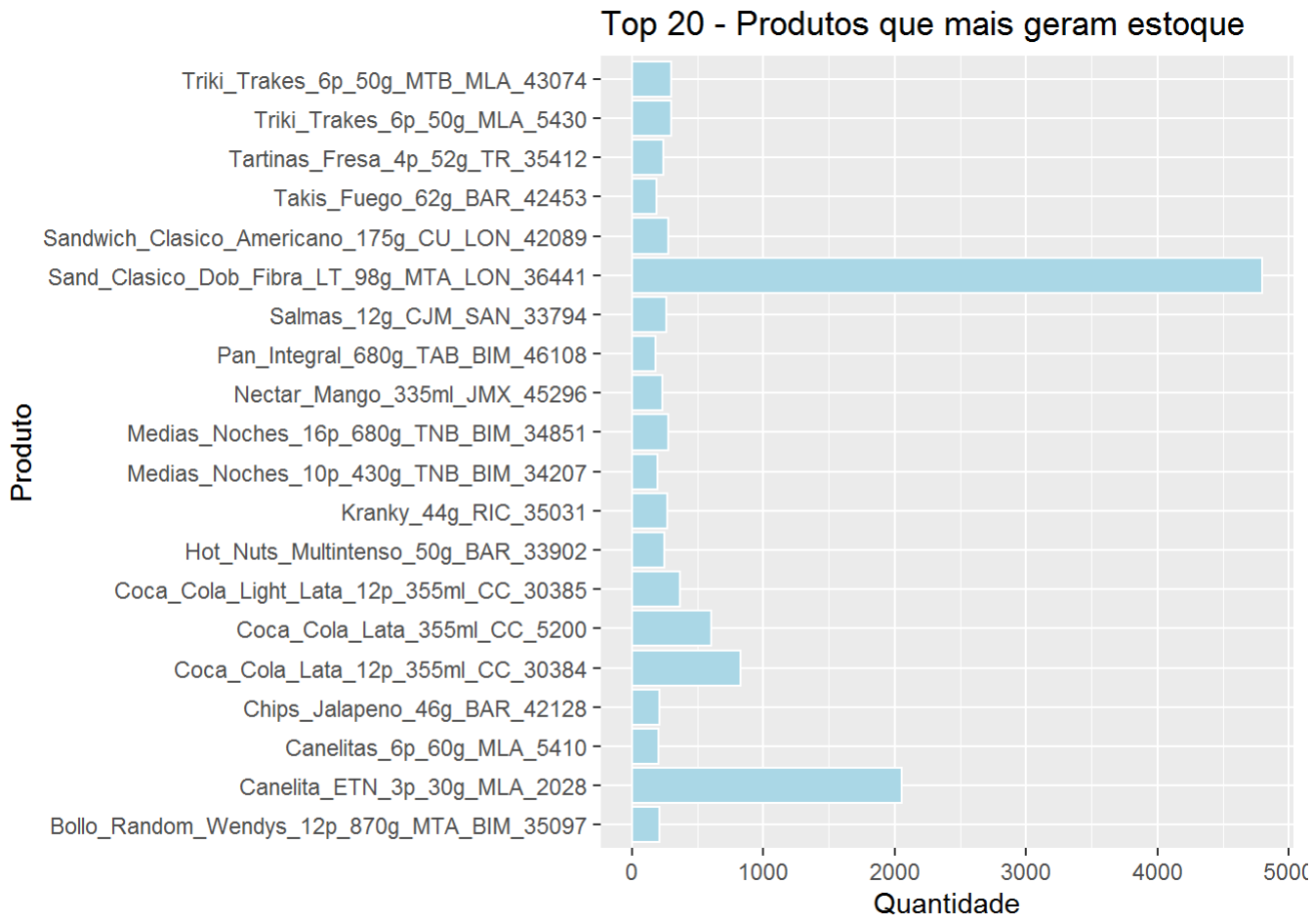





```
# Top 20 dos produtos que mais geram estoque em media
p15 <- train2 %>%
  select(NombreProducto, Demanda_uni_equil)%>%
  group_by(NombreProducto,)%>%
  summarise(Total = mean(Demanda_uni_equil))%>%
  filter(Total > 177)%>%
  ggplot(aes(y = as.factor(NombreProducto), x =  Total)) +
  geom_bar(stat = "identity",color = "white", fill = "lightblue") +
  labs(title = 'Top 20 - Produtos que mais geram estoque',
       y = 'Produto', x = 'Quantidade')
```

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

```
p15
```



Split dos dados

```
# Faço o split dos dados para treinar e testar os algoritmos de regreção.

split <- sample.split(train$Canal_ID, SplitRatio = 0.80)

trainModel = subset(train, split == TRUE)
testModel = subset(train, split == FALSE)
nrow(trainModel)
```

```
## [1] 94951
```

```
nrow(testModel)
```

```
## [1] 23737
```

```
names(train)
```

```
##  [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"
```

Algoritmos de aprendizagem

```
# Modelo com Regressão linear

modelo_v1 <- lm(Demanda_uni_equil ~ Canal_ID
                +Ruta_SAK
                +Producto_ID
                +Cliente_ID
                +Venta_uni_hoy
                +Dev_uni_proxima,
                data = trainModel)

summary(modelo_v1)
```

```
##
## Call:
## lm(formula = Demanda_uni_equil ~ Canal_ID + Ruta_SAK + Producto_ID +
##     Cliente_ID + Venta_uni_hoy + Dev_uni_proxima, data = trainModel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.904   0.019   0.027   0.040  76.136
##
## Coefficients:
##                   Estimate Std. Error   t value Pr(>|t|)
## (Intercept)     -1.819e-02  4.857e-03    -3.746  0.00018 ***
## Canal_ID        -8.008e-03  1.799e-03    -4.451 8.54e-06 ***
## Ruta_SAK         7.598e-06  1.844e-06     4.120 3.79e-05 ***
## Producto_ID     -3.477e-07  1.296e-07    -2.683  0.00729 **
## Cliente_ID       2.280e-09  1.255e-09     1.817  0.06929 .
## Venta_uni_hoy    9.970e-01  8.809e-05 11317.015  < 2e-16 ***
## Dev_uni_proxima -3.641e-01  1.276e-03  -285.351  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7068 on 94944 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.9993
## F-statistic: 2.173e+07 on 6 and 94944 DF,  p-value: < 2.2e-16
```

```
previsao_v1 <- predict(modelo_v1, testModel)
head(cbind(round(previsao_v1), testModel$Demanda_uni_equil))
```

```
##   [,1] [,2]
## 1    2    2
## 2    1    1
## 3    2    2
## 4    7    7
## 5    2    2
## 6    3    3
```

```
tail(cbind(round(previsao_v1), testModel$Demanda_uni_equil))
```

```
##       [,1] [,2]
## 23732    2    2
## 23733    3    3
## 23734    1    1
## 23735    1    1
## 23736    1    1
## 23737    1    0
```

```
# Accuracy
mae_1 = MAE(testModel$Demanda_uni_equil,round(previsao_v1))
rmse_1 = RMSE(testModel$Demanda_uni_equil,round(previsao_v1))
r2_1 = R2(testModel$Demanda_uni_equil,round(previsao_v1))

cat(" MAE:", mae_1, "\n",
    "RMSE:", rmse_1, "\n", "R-squared:", r2_1)
```

```
##  MAE: 0.07861145
##  RMSE: 2.227678
##  R-squared: 0.9847746
```

```
# Modelo com o randomForest

modelo_v2 <- randomForest(Demanda_uni_equil ~ Canal_ID
                          +Ruta_SAK
                          +Producto_ID
                          +Cliente_ID
                          +Venta_uni_hoy
                          +Dev_uni_proxima,
                          data = trainModel,
                          ntree = 40,
                          nodesize = 5)

print(modelo_v2)
```

```
##
## Call:
##  randomForest(formula = Demanda_uni_equil ~ Canal_ID + Ruta_SAK +      Producto_ID + Cliente_
ID + Venta_uni_hoy + Dev_uni_proxima,      data = trainModel, ntree = 40, nodesize = 5)
##                Type of random forest: regression
##                      Number of trees: 40
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 211.7793
##                     % Var explained: 69.14
```

```
previsao_v2 <- predict(modelo_v2, testModel)
head(cbind(round(previsao_v2), testModel$Demanda_uni_equil))
```

```
##   [,1] [,2]
## 1    3    2
## 2    2    1
## 3    2    2
## 4    7    7
## 5    2    2
## 6    3    3
```

```
tail(cbind(round(previsao_v2), testModel$Demanda_uni_equil))
```

```
##       [,1] [,2]
## 23732    2    2
## 23733    4    3
## 23734    2    1
## 23735    2    1
## 23736    2    1
## 23737    0    0
```

```
# Accuracy
mae_2 = MAE(testModel$Demanda_uni_equil,round(previsao_v2))
rmse_2 = RMSE(testModel$Demanda_uni_equil,round(previsao_v2))
r2_2 = R2(testModel$Demanda_uni_equil,round(previsao_v2))

cat(" MAE:", mae_2, "\n",
    "RMSE:", rmse_2, "\n", "R-squared:", r2_2)
```

```
##  MAE: 0.5421073
##  RMSE: 3.944554
##  R-squared: 0.9599814
```

```
# Modelo com o SVM

modelo_v3 <- svm(Demanda_uni_equil ~ Canal_ID
                 +Ruta_SAK
                 +Producto_ID
                 +Cliente_ID
                 +Venta_uni_hoy
                 +Dev_uni_proxima,
                 data = trainModel)

previsao_v3 <- predict(modelo_v3, testModel)
head(cbind(round(previsao_v3), testModel$Demanda_uni_equil))
```

```
##   [,1] [,2]
## 1    4    2
## 2    4    1
## 3    2    2
## 4    7    7
## 5    2    2
## 6    3    3
```

```
tail(cbind(round(previsao_v3), testModel$Demanda_uni_equil))
```

```
##       [,1] [,2]
## 23732    3    2
## 23733    3    3
## 23734    3    1
## 23735    2    1
## 23736    2    1
## 23737   -1    0
```

```
# Accuracy
mae_3 = MAE(testModel$Demanda_uni_equil,round(previsao_v3))
rmse_3 = RMSE(testModel$Demanda_uni_equil,round(previsao_v3))
r2_3 = R2(testModel$Demanda_uni_equil,round(previsao_v3))

cat(" MAE:", mae_3, "\n",
    "RMSE:", rmse_3, "\n", "R-squared:", r2_3)
```

```
##  MAE: 1.004508
##  RMSE: 10.00485
##  R-squared: 0.691055
```

```
# Modelo com o rpart

modelo_v4 <- rpart(Demanda_uni_equil ~ Canal_ID
                   +Ruta_SAK
                   +Producto_ID
                   +Cliente_ID
                   +Venta_uni_hoy
                   +Dev_uni_proxima,
                   data = trainModel,
                   method = 'anova')

previsao_v4 <- predict(modelo_v4, testModel,method = 'anova')
head(cbind(round(previsao_v4), testModel$Demanda_uni_equil))
```

```
##   [,1] [,2]
## 1    3    2
## 2    3    1
## 3    3    2
## 4    3    7
## 5    3    2
## 6    3    3
```

```
tail(cbind(round(previsao_v4), testModel$Demanda_uni_equil))
```

```
##       [,1] [,2]
## 23732    3    2
## 23733    3    3
## 23734    3    1
## 23735    3    1
## 23736    3    1
## 23737    3    0
```

```
printcp(modelo_v4)
```

```
##
## Regression tree:
## rpart(formula = Demanda_uni_equil ~ Canal_ID + Ruta_SAK + Producto_ID +
##     Cliente_ID + Venta_uni_hoy + Dev_uni_proxima, data = trainModel,
##     method = "anova")
##
## Variables actually used in tree construction:
## [1] Venta_uni_hoy
##
## Root node error: 65160405/94951 = 686.25
##
## n= 94951
##
##         CP nsplit rel error  xerror    xstd
## 1 0.363633      0   1.00000 1.00003 0.36321
## 2 0.245978      1   0.63637 0.87496 0.35586
## 3 0.087810      2   0.39039 0.56644 0.31377
## 4 0.059270      3   0.30258 0.36471 0.20169
## 5 0.017306      4   0.24331 0.31381 0.20130
## 6 0.016872      5   0.22600 0.28202 0.20126
## 7 0.013476      6   0.20913 0.27689 0.20126
## 8 0.010000      7   0.19565 0.27029 0.20126
```

```
# Accuracy
mae_4 = MAE(testModel$Demanda_uni_equil,round(previsao_v4))
rmse_4 = RMSE(testModel$Demanda_uni_equil,round(previsao_v4))
r2_4 = R2(testModel$Demanda_uni_equil,round(previsao_v4))

cat(" MAE:", mae_4, "\n",
    "RMSE:", rmse_4, "\n", "R-squared:", r2_4)
```

```
##  MAE: 2.246746
##  RMSE: 6.394414
##  R-squared: 0.8951853
```

```
printcp(modelo_v4)
```

```
##
## Regression tree:
## rpart(formula = Demanda_uni_equil ~ Canal_ID + Ruta_SAK + Producto_ID +
##     Cliente_ID + Venta_uni_hoy + Dev_uni_proxima, data = trainModel,
##     method = "anova")
##
## Variables actually used in tree construction:
## [1] Venta_uni_hoy
##
## Root node error: 65160405/94951 = 686.25
##
## n= 94951
##
##         CP nsplit rel error  xerror    xstd
## 1 0.363633      0   1.00000 1.00003 0.36321
## 2 0.245978      1   0.63637 0.87496 0.35586
## 3 0.087810      2   0.39039 0.56644 0.31377
## 4 0.059270      3   0.30258 0.36471 0.20169
## 5 0.017306      4   0.24331 0.31381 0.20130
## 6 0.016872      5   0.22600 0.28202 0.20126
## 7 0.013476      6   0.20913 0.27689 0.20126
## 8 0.010000      7   0.19565 0.27029 0.20126
```

Os algoritmos LM e RandomForest tiveram uma eficácia maior nas previsões com relação aos SVM e Rpart, para entregar para o cliente ficaria com mu dos dois primeiros.