Predição de demanda grupo Bimbo

luiz felipe

12/21/2020

Introdução

O trabalho consiste na análise de dados de demanda de clientes do grupo Bimbo afim de prever a demanda futura. Os dados usados foram clientes, rotas, produtos, depósito. O cálculo da demanda do produto foi feito subtraindo o quanto foi vendido do quanto foi devolvido da semana anterior. Os dados estão disponíveis no seguinte link:

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

```
# Carregando pacotes
library(data.table)
library(readr)
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(stringr)
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library(caret)
```

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

```r
# Carregando os dados do grupo Bimbo
SampleSale <- read_csv("BimboData.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   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()
## )
```

```r
SampleSale <- SampleSale[,-1]
```

Etapa exploratória dos dados

Os dados serão investigados utilizando ferramentas de estatística de descritiva e ferramentas para avaliar como os dados foram armazenados pelo R.

```r
# Visualização geral dos dados
head(SampleSale)
```

```
## # 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      9       1330        1     1252     118797        1242             2
## 2      7       1217        1     1629     112328       35452             3
## 3      8       2030        1     2012    4233888       43064             2
## 4      5       2214        1     1224    8235885        1109             2
## 5      4       1697        1     1149    1526691        1284             6
## 6      4       4046        1     2133    1310060       30552             3
## # ... with 4 more variables: Venta_hoy <dbl>, Dev_uni_proxima <dbl>,
## #   Dev_proxima <dbl>, Demanda_uni_equil <dbl>
```

```r
# Tipos de dados
str(SampleSale)
```

```
## tibble [100,000 x 11] (S3: tbl_df/tbl/data.frame)
##  $ Semana          : num [1:100000] 9 7 8 5 4 4 9 8 8 9 ...
##  $ Agencia_ID      : num [1:100000] 1330 1217 2030 2214 1697 ...
##  $ Canal_ID        : num [1:100000] 1 1 1 1 1 1 1 4 1 1 ...
##  $ Ruta_SAK        : num [1:100000] 1252 1629 2012 1224 1149 ...
##  $ Cliente_ID      : num [1:100000] 118797 112328 4233888 8235885 1526691 ...
##  $ Producto_ID     : num [1:100000] 1242 35452 43064 1109 1284 ...
##  $ Venta_uni_hoy   : num [1:100000] 2 3 2 2 6 3 2 4 4 1 ...
##  $ Venta_hoy       : num [1:100000] 15.3 13.3 16.3 30 18.1 ...
##  $ Dev_uni_proxima : num [1:100000] 3 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ Dev_proxima      : num [1:100000] 22.9 0 0 0 0 ...
##  $ Demanda_uni_equil: num [1:100000] 0 3 2 2 6 3 2 4 4 1 ...
```

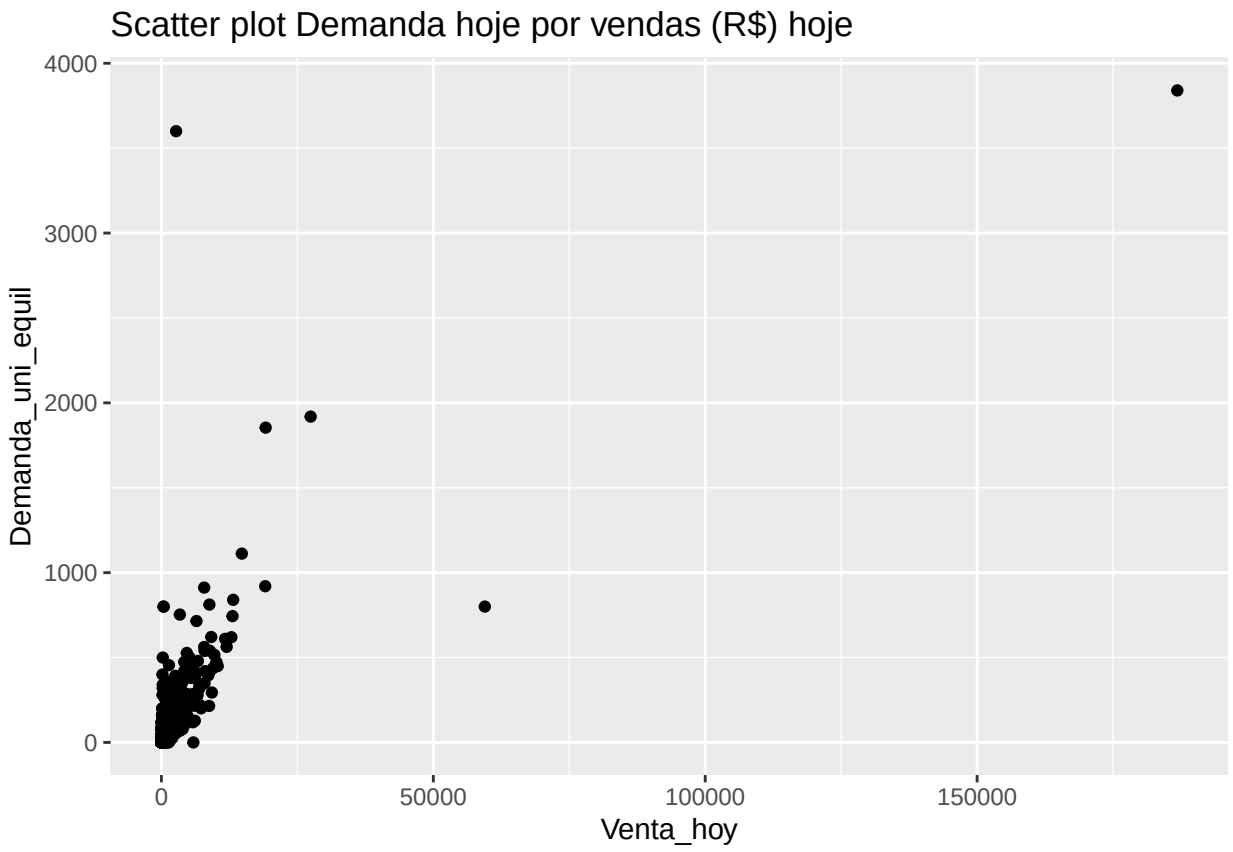
```
# Verificando se há valores NA
sapply(SampleSale, function(x) sum(is.na(x)))
```

```
##            Semana        Agencia_ID          Canal_ID          Ruta_SAK
##                 0                 0                 0                 0
##        Cliente_ID       Producto_ID     Venta_uni_hoy         Venta_hoy
##                 0                 0                 0                 0
##   Dev_uni_proxima       Dev_proxima Demanda_uni_equil
##                 0                 0                 0
```

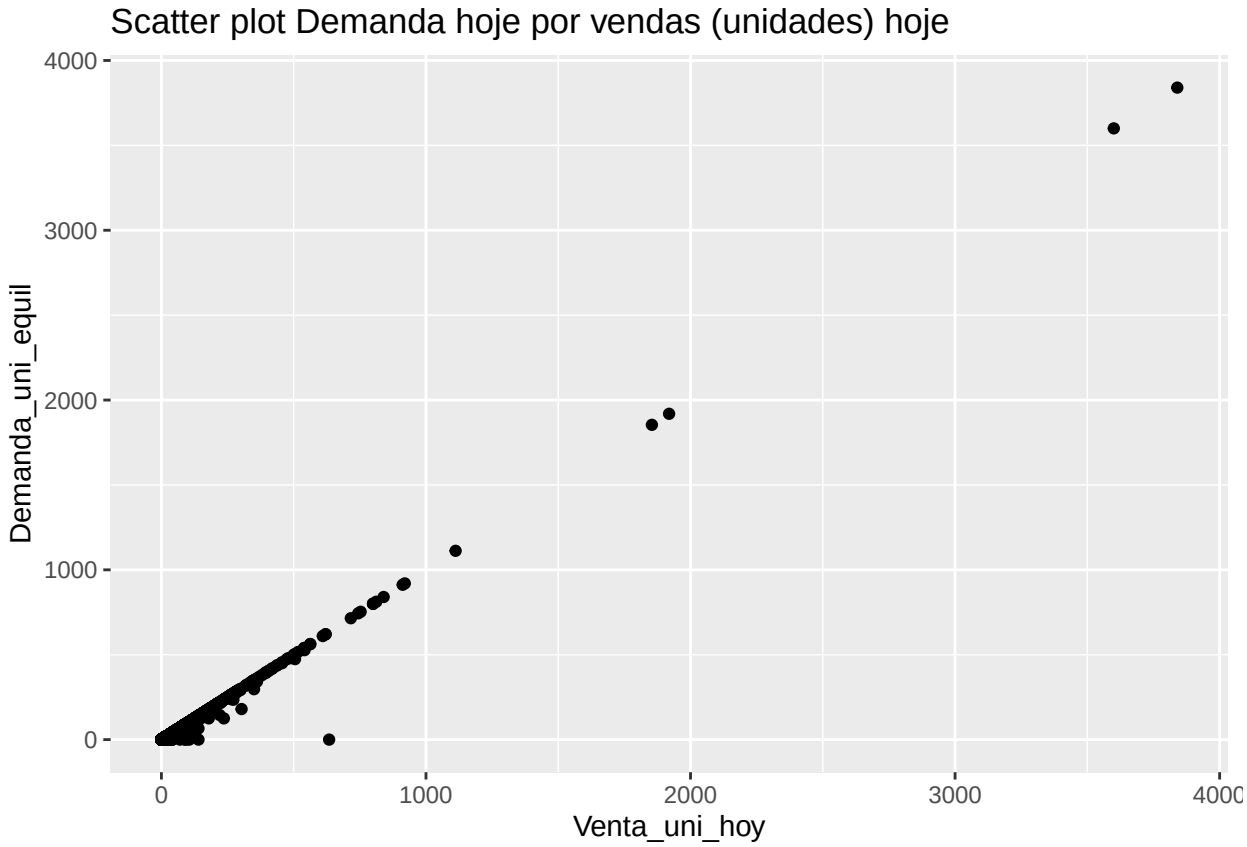
Variáveis quantitativas

As varíaveis quantitativas serão exploradas por meio de ferramentas gráficas e estatística descritiva.

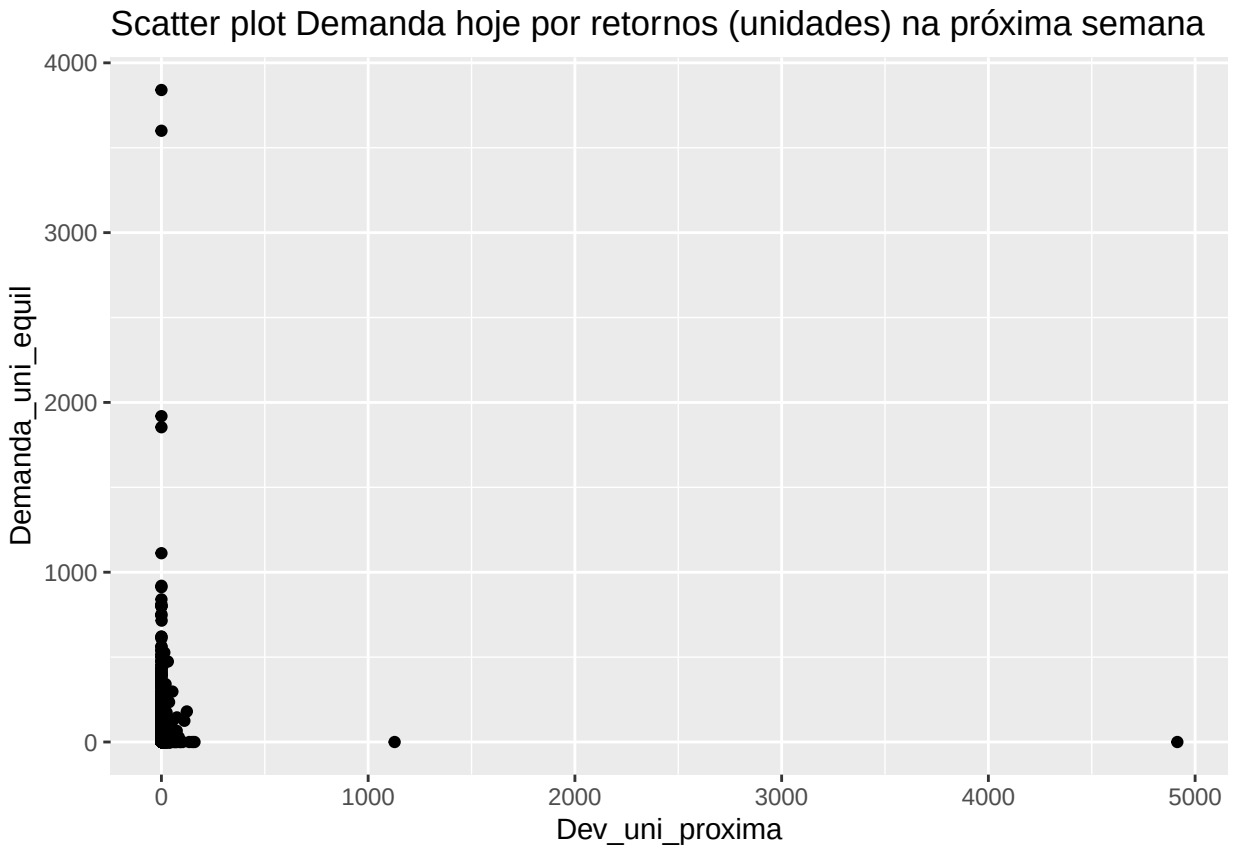
```
# As variáveis Venta_uni_hoy, Venta_hoy, Dev_uni_proxima, Dev_proxima, Demanda_uni_equil
# são numéricas e as outras categóricas
# Vamos avaliar as variáveis neméricas
NumVar <- SampleSale[,c("Venta_uni_hoy", "Venta_hoy", "Dev_uni_proxima", "Dev_proxima",
                        "Demanda_uni_equil")]

# Verificando a relação entre as variáveis
# É possível ver as que existem outliers e as variáveis são correlacionados
ggplot(data = NumVar) +
  geom_point(mapping = aes(x = Venta_hoy, y = Demanda_uni_equil)) +
  ggtitle("Scatter plot Demanda hoje por vendas (R$) hoje")
```



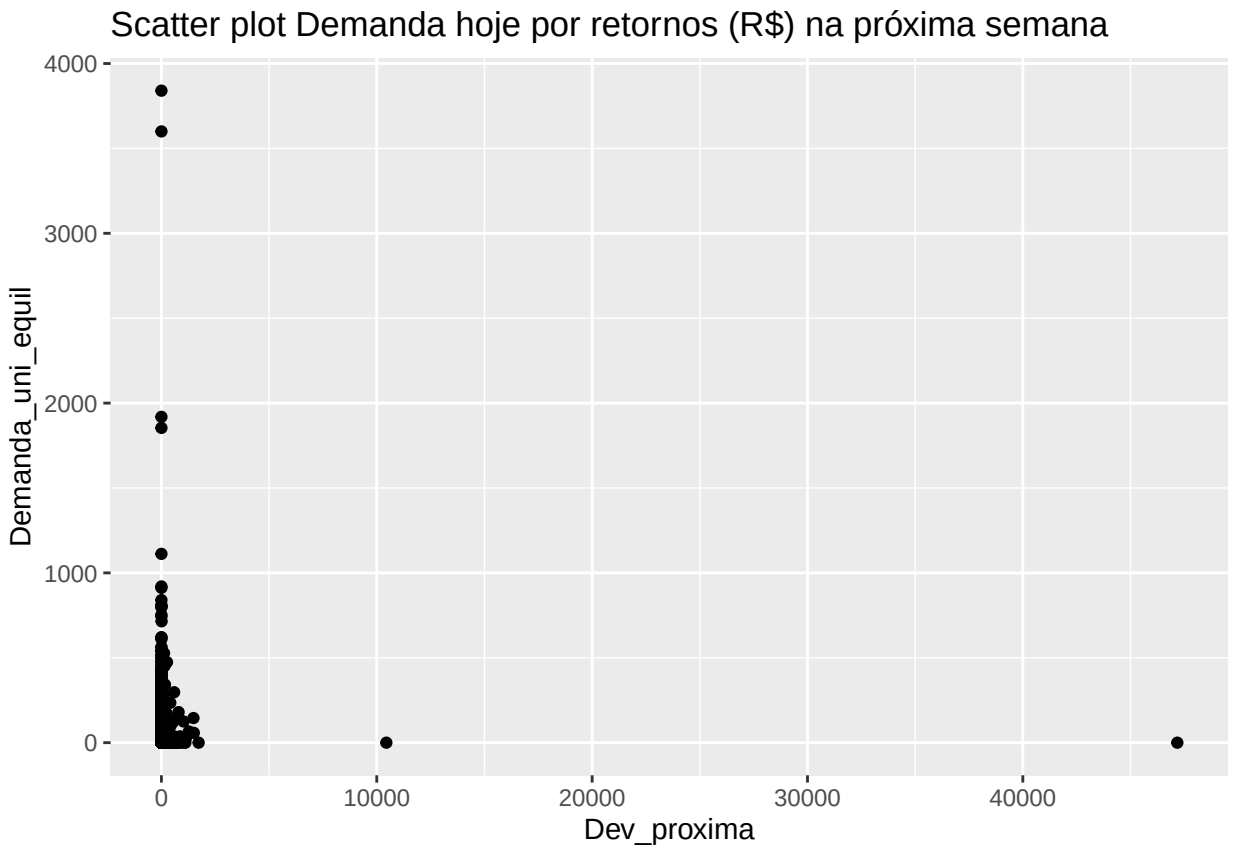
```r
# Relacionamento altamente linear
ggplot(data = NumVar) +
  geom_point(mapping = aes(x = Venta_uni_hoy, y = Demanda_uni_equil)) +
  ggtitle("Scatter plot Demanda hoje por vendas (unidades) hoje")
```



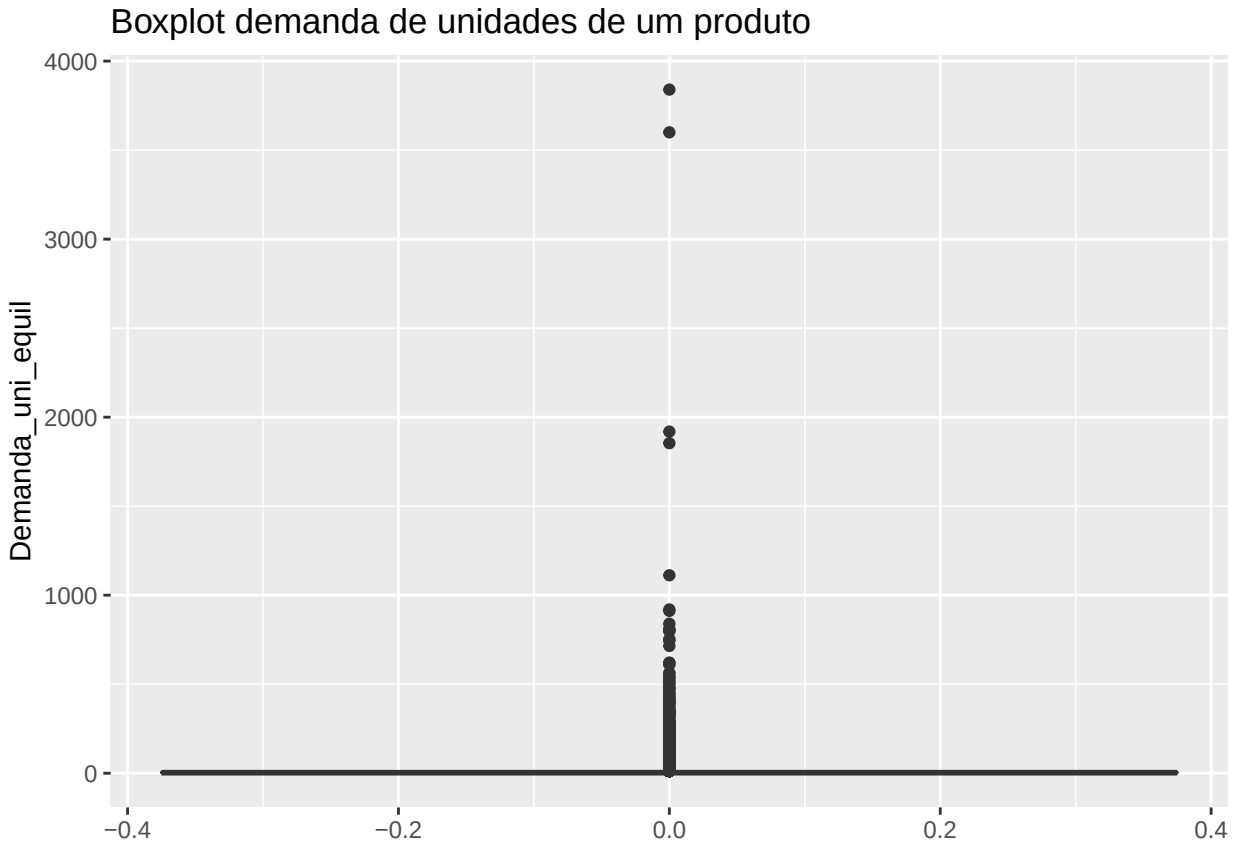

```r
# As variáveis não tem uma relação linear, pois não podem existir valores negativos de
# Demanda_uni_equil
ggplot(data = NumVar) +
  geom_point(mapping = aes(x = Dev_uni_proxima, y = Demanda_uni_equil)) +
  ggtitle("Scatter plot Demanda hoje por retornos (unidades) na próxima semana")
```




```
# Nenhuma relação linear
ggplot(data = NumVar) +
  geom_point(mapping = aes(x = Dev_proxima, y = Demanda_uni_equil)) +
  ggtitle("Scatter plot Demanda hoje por retornos (R$) na próxima semana")
```

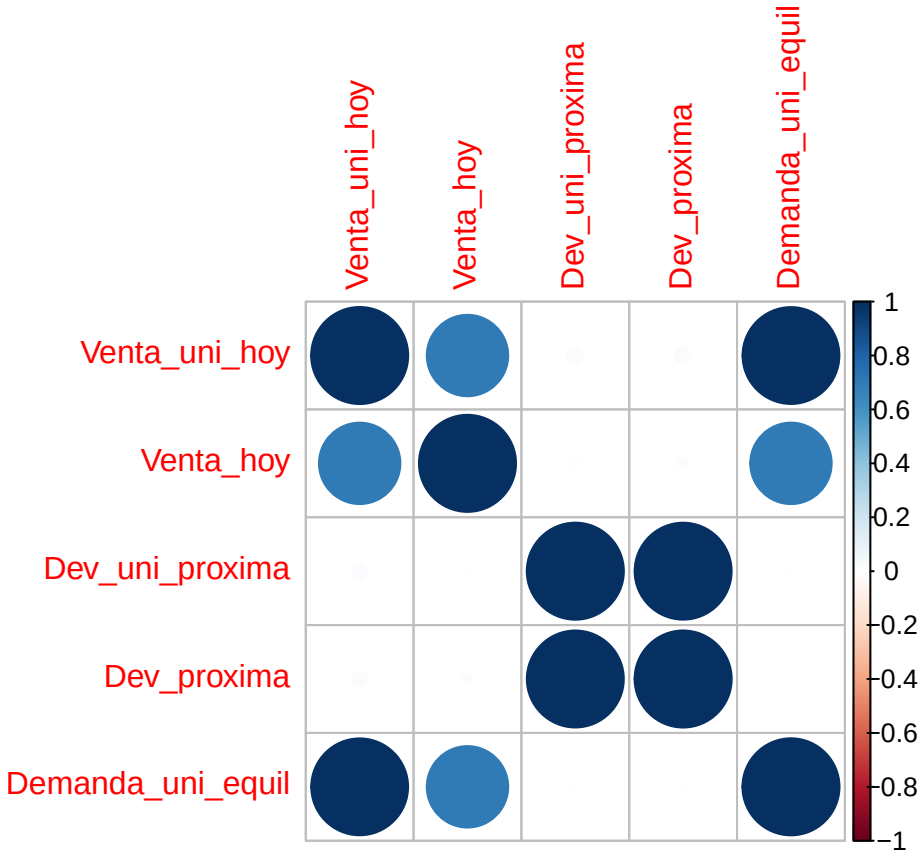



```
# Gráfico de boxplot para verificar a dispersão dos dados
ggplot(data = NumVar) +
  geom_boxplot(mapping = aes(y = Demanda_uni_equil)) +
  ggtitle("Boxplot demanda de unidades de um produto")
```



A fim de confirmar o que é possível ver por meio dos gráficos, vamos gerar uma matriz de correlação.

```
# Avaliando a relação entre variáveis numéricas
M <- cor(NumVar)
corrplot(M, method = "circle")
```



Variáveis qualitativas

As varíaveis qualitativas serão exploradas por meio de ferramentas gráficas e essas variáveis serão transformadas no tipo factor.

```
# criando um dataframe sem as variáveis quantitativas

CatVar <- SampleSale[,-c(7,8,9,10)]

# Visualizando os dados
head(CatVar)
```

```
## # A tibble: 6 x 7
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Demanda_uni_equil
##    <dbl>      <dbl>    <dbl>    <dbl>      <dbl>       <dbl>             <dbl>
## 1      9       1330        1     1252     118797        1242                 0
## 2      7       1217        1     1629     112328       35452                 3
## 3      8       2030        1     2012    4233888       43064                 2
## 4      5       2214        1     1224    8235885        1109                 2
## 5      4       1697        1     1149    1526691        1284                 6
## 6      4       4046        1     2133    1310060       30552                 3
```

```
# Transformando as variáveis categóricas em factor
Cat <- data.frame(lapply(CatVar[,-c(7)], factor))
str(Cat)
```

```
## 'data.frame':    100000 obs. of  6 variables:
##  $ Semana     : Factor w/ 7 levels "3","4","5","6",..: 7 5 6 3 2 2 7 6 6 7 ...
```

```
##  $ Agencia_ID : Factor w/ 536 levels "1110","1111",..: 88 50 366 416 306 506 525 155 141 3 ...
##  $ Canal_ID   : Factor w/ 9 levels "1","2","4","5",..: 1 1 1 1 1 1 1 3 1 1 ...
##  $ Ruta_SAK   : Factor w/ 1901 levels "1","2","3","4",..: 360 661 749 332 262 855 834 1528 396 638 .
##  $ Cliente_ID : Factor w/ 89405 levels "26","65","107",..: 8134 7559 72809 88800 49881 46210 7299 493
##  $ Producto_ID: Factor w/ 954 levels "53","72","73",..: 64 490 718 35 68 203 254 762 919 476 ...
```

```
# Novo dataframe com as variáveis qualitativas e o a variável target
df.Demand.Forecasting <- cbind(Cat, CatVar[,"Demanda_uni_equil"])

head(df.Demand.Forecasting)
```

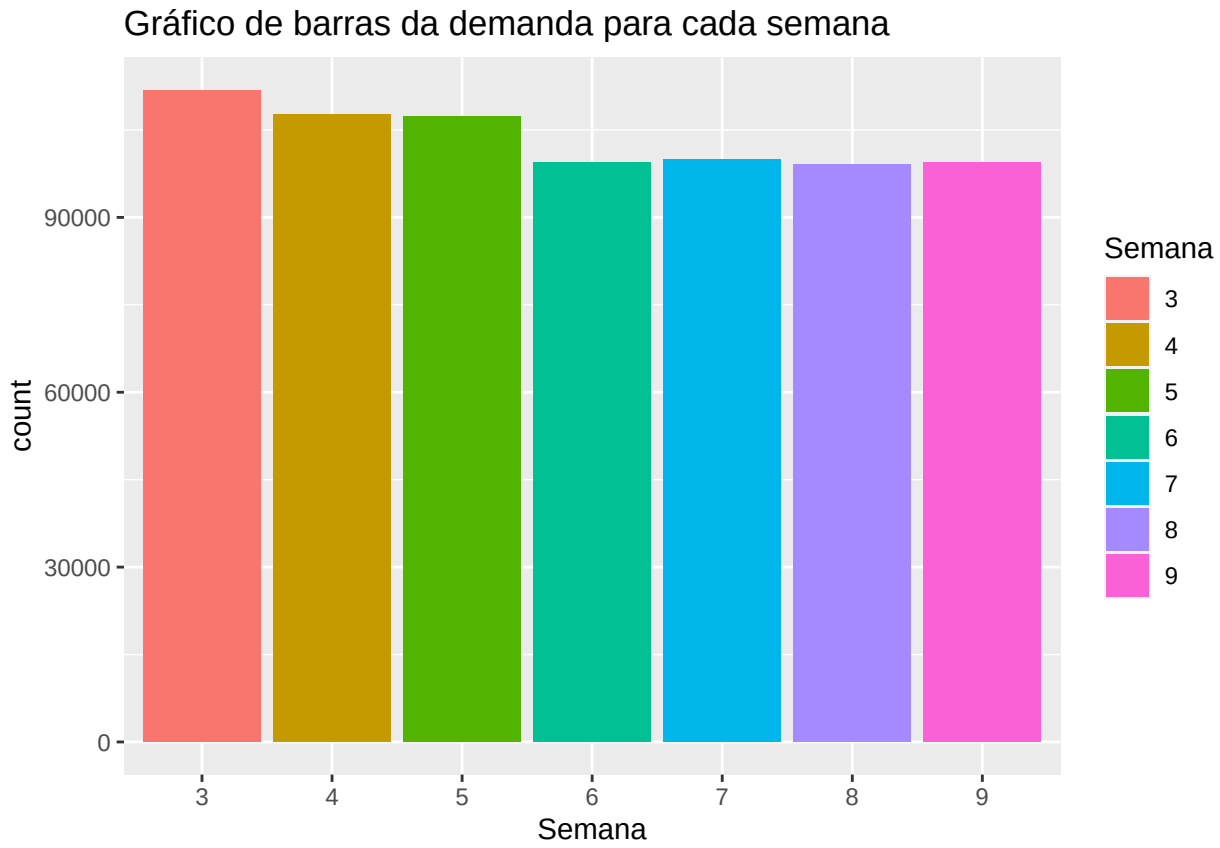
```
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Demanda_uni_equil
## 1      9       1330        1     1252     118797        1242                0
## 2      7       1217        1     1629     112328       35452                3
## 3      8       2030        1     2012    4233888       43064                2
## 4      5       2214        1     1224    8235885        1109                2
## 5      4       1697        1     1149    1526691        1284                6
## 6      4       4046        1     2133    1310060       30552                3
```

```
str(df.Demand.Forecasting)
```

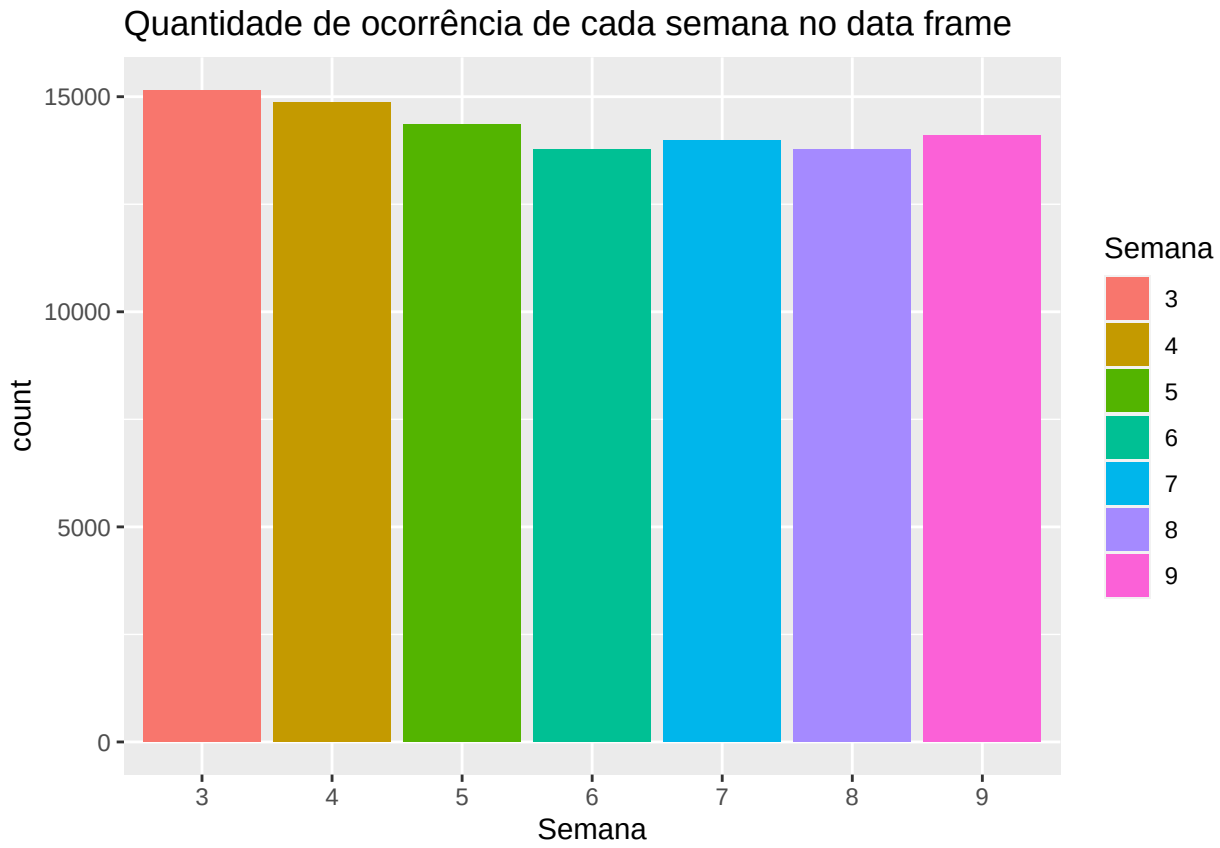
```
## 'data.frame':    100000 obs. of  7 variables:
##  $ Semana           : Factor w/ 7 levels "3","4","5","6",..: 7 5 6 3 2 2 7 6 6 7 ...
##  $ Agencia_ID       : Factor w/ 536 levels "1110","1111",..: 88 50 366 416 306 506 525 155 141 3 ...
##  $ Canal_ID         : Factor w/ 9 levels "1","2","4","5",..: 1 1 1 1 1 1 1 3 1 1 ...
##  $ Ruta_SAK         : Factor w/ 1901 levels "1","2","3","4",..: 360 661 749 332 262 855 834 1528 396
##  $ Cliente_ID       : Factor w/ 89405 levels "26","65","107",..: 8134 7559 72809 88800 49881 46210 7
##  $ Producto_ID      : Factor w/ 954 levels "53","72","73",..: 64 490 718 35 68 203 254 762 919 476 .
##  $ Demanda_uni_equil: num  0 3 2 2 6 3 2 4 4 1 ...
```

Gráfico de barras das variáveis semana e canal_id.

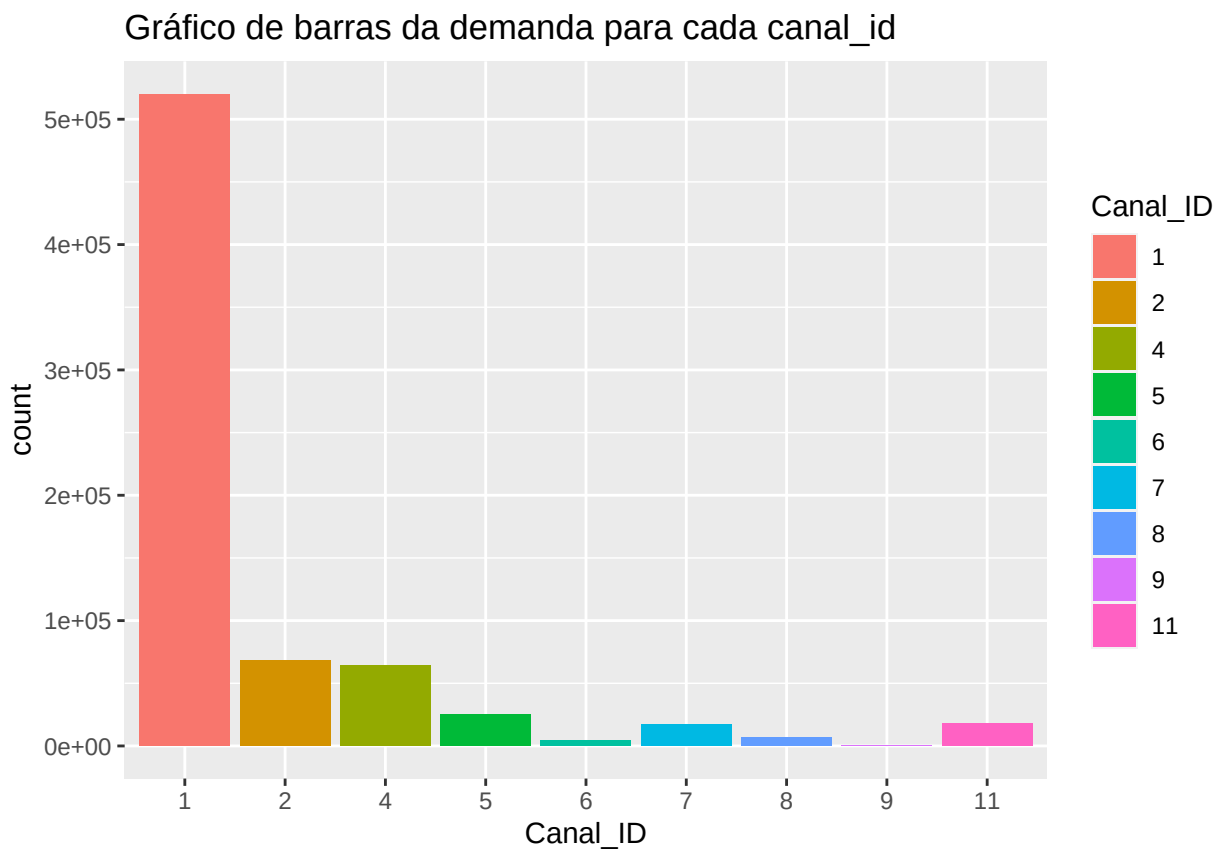
```
# Demanda_uni_equil pela semana
ggplot(data = df.Demand.Forecasting) +
  geom_bar(mapping = aes(x = Semana, weight = Demanda_uni_equil, fill = Semana)) +
  ggtitle("Gráfico de barras da demanda para cada semana")
```




```
# Quantidade de cada semana
ggplot(data = df.Demand.Forecasting) +
  geom_bar(mapping = aes(x = Semana, fill = Semana)) +
  ggtitle("Quantidade de ocorrência de cada semana no data frame")
```




```
# Demanda_uni_equil pelo Canal_ID
ggplot(data = df.Demand.Forecasting) +
  geom_bar(mapping = aes(x = Canal_ID, weight = Demanda_uni_equil, fill = Canal_ID)) +
  ggtitle("Gráfico de barras da demanda para cada canal_id")
```




```
# Quantidade de Canal_ID
ggplot(data = df.Demand.Forecasting) +
  geom_bar(mapping = aes(x = Canal_ID, fill = Canal_ID)) +
  ggtitle("Quantidade de ocorrência de cada semana no data frame")
```

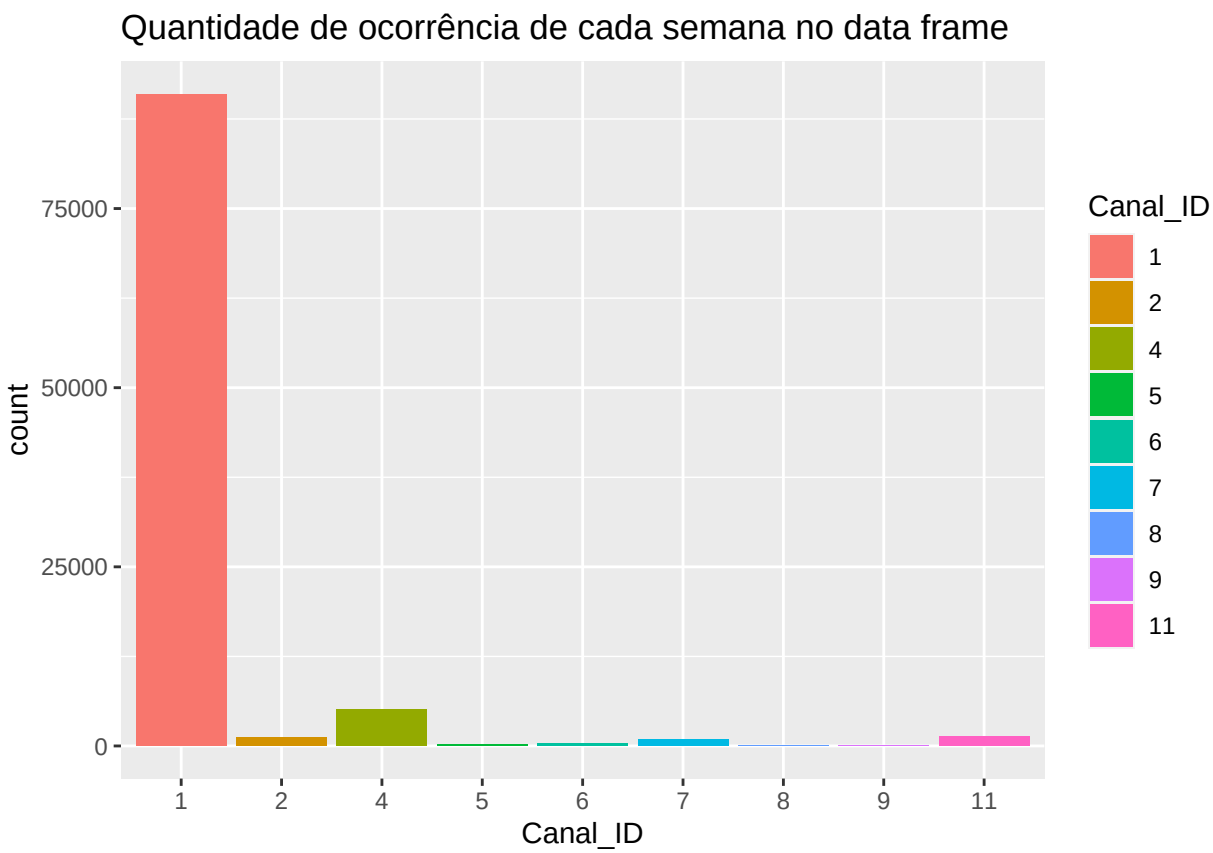


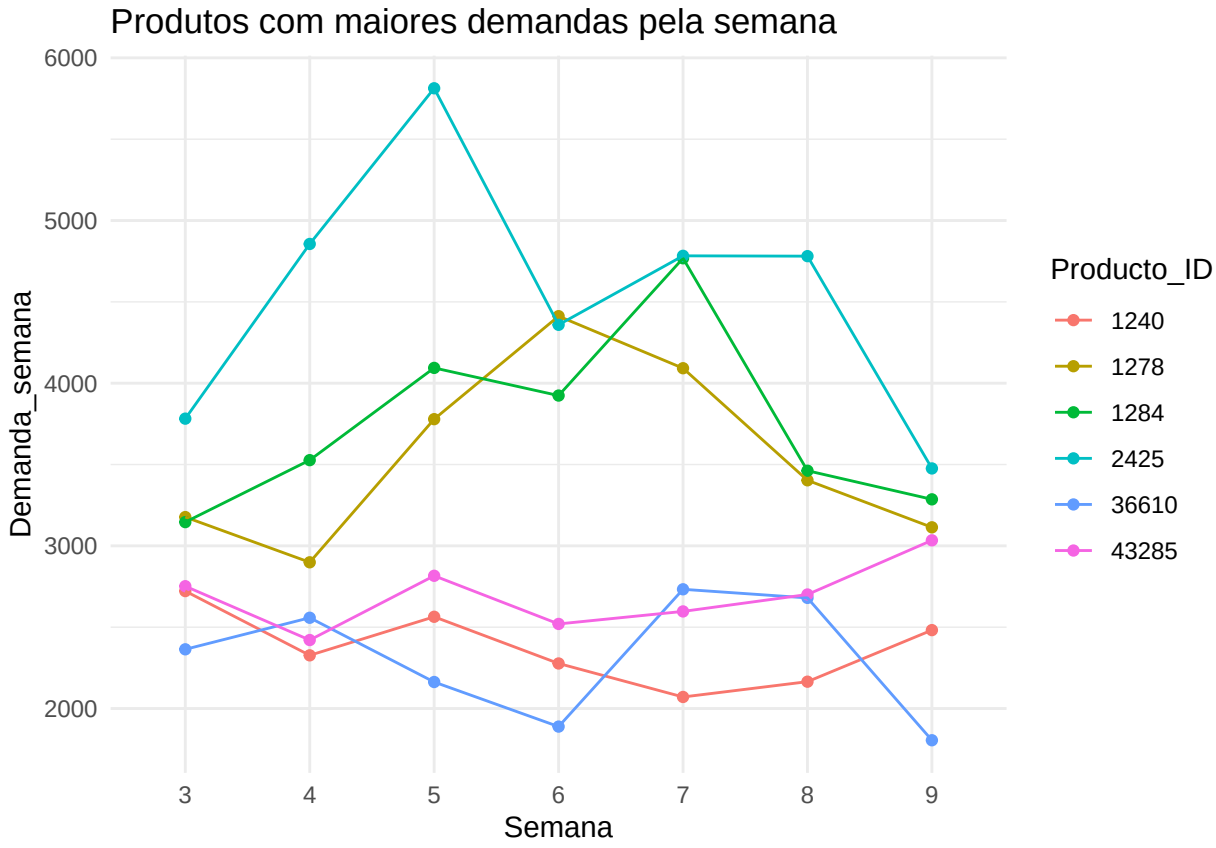
Gráfico das maiores quantidades de cada variável.

```
# Os cinco produtos com maiores demandas
Prod_5_mais <- SampleSale %>%
  group_by(Producto_ID) %>%
  summarise(Quant_vend_prod = sum(Demanda_uni_equil)) %>%
  arrange(desc(Quant_vend_prod))

head(Prod_5_mais)
```

```
## # A tibble: 6 x 2
##   Producto_ID Quant_vend_prod
##         <dbl>           <dbl>
## 1        2425           31850
## 2        1284           26207
## 3        1278           24876
## 4       43285           18841
## 5        1240           16608
## 6       36610           16192
```

```
# Gráfico dos 5 maiores produtos demandados por semana
df.Demand.Forecasting %>%
  filter(Producto_ID %in% c(2425,1284,1278,43285,1240,36610)) %>%
  group_by(Semana, Producto_ID) %>%
  arrange(Semana, Producto_ID) %>%
  summarise(Demanda_semana = sum(Demanda_uni_equil)) %>%
  ggplot(mapping = aes(x = Semana, y = Demanda_semana, group = Producto_ID,
                       colour = Producto_ID)) +
```

```
  geom_point() +
  geom_line() + theme_minimal() +
  ggtitle("Produtos com maiores demandas pela semana")
```




```
# Os cinco maiores demandas por Clientes
Cliente_5_mais <- SampleSale %>%
  group_by(Cliente_ID) %>%
  summarise(Quant_vend_cliente = sum(Demanda_uni_equil)) %>%
  arrange(desc(Quant_vend_cliente))

head(Cliente_5_mais)
```

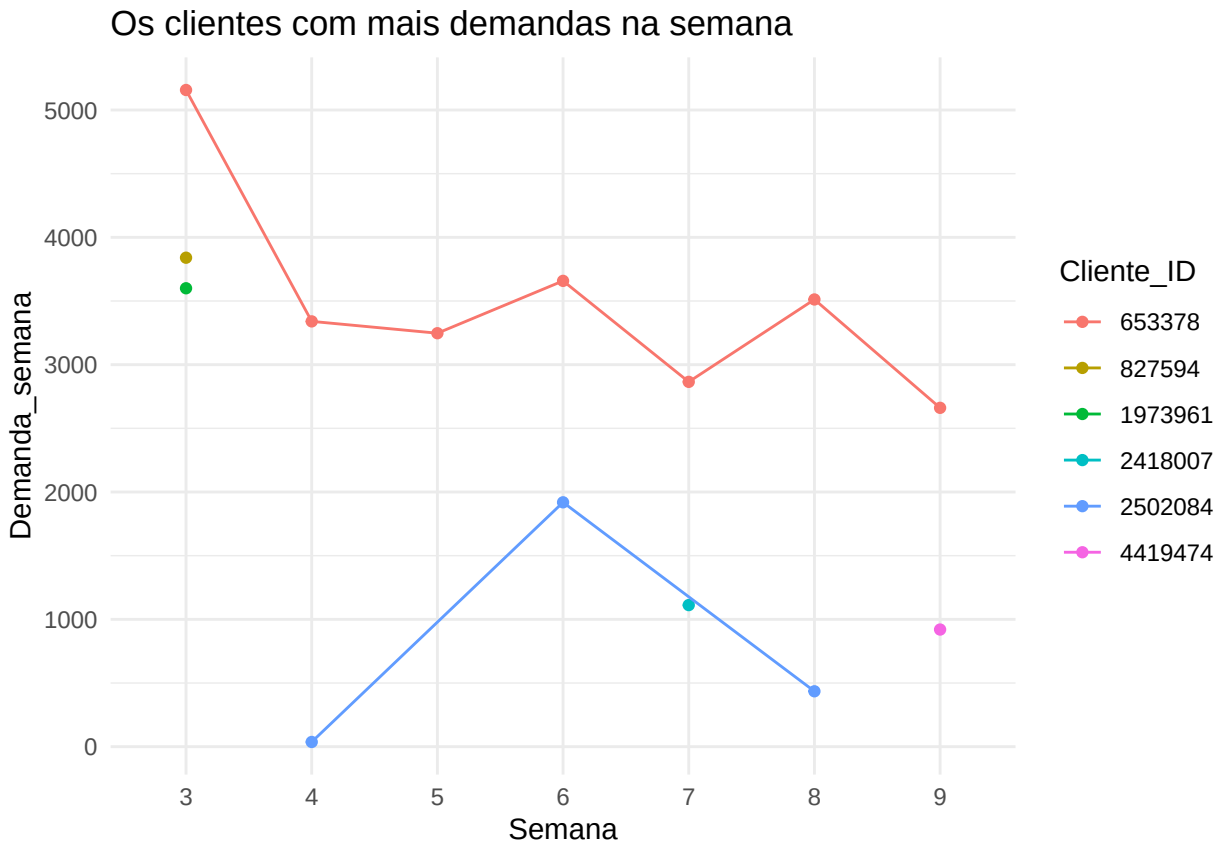
```
## # A tibble: 6 x 2
##   Cliente_ID Quant_vend_cliente
##        <dbl>              <dbl>
## 1     653378              24440
## 2     827594               3840
## 3    1973961               3600
## 4    2502084               2391
## 5    2418007               1112
## 6    4419474                920
```

```
# Os 5 maiores clientes em demandas
df_cliente_5_mais <- df.Demand.Forecasting %>%
  arrange(Semana, Cliente_ID) %>%
  group_by(Semana, Cliente_ID) %>%
  summarise(Demanda_semana = sum(Demanda_uni_equil))
```

```r
# Quantidade de semanas de cliente (só um cliente teve demanda todos os dias da semana)
df_cliente_5_mais %>%
  group_by(Cliente_ID) %>%
  summarise(quant_cliente_semana = n()) %>%
  arrange(desc(quant_cliente_semana)) %>%
  head()
```

```
## # A tibble: 6 x 2
##   Cliente_ID quant_cliente_semana
##   <fct>                     <int>
## 1 653378                        7
## 2 20993                         6
## 3 42686                         4
## 4 48160                         4
## 5 54950                         4
## 6 62633                         4
```

```r
# Gráfico dos 5 maiores demandas de clientes por semana
# (semanas sem demanda representam demanda zero, ou seja, não compra nem devolução de produto)
df.Demand.Forecasting %>%
  filter(Cliente_ID %in% c(653378, 827594, 1973961, 2502084, 2418007, 4419474)) %>%
  group_by(Semana, Cliente_ID) %>%
  arrange(Semana, Cliente_ID) %>%
  summarise(Demanda_semana = sum(Demanda_uni_equil)) %>%
  ggplot(mapping = aes(x = Semana, y = Demanda_semana, group = Cliente_ID,
                       colour = Cliente_ID)) +
  geom_point() +
  geom_line() + theme_minimal() +
  ggtitle("Os clientes com mais demandas na semana")
```

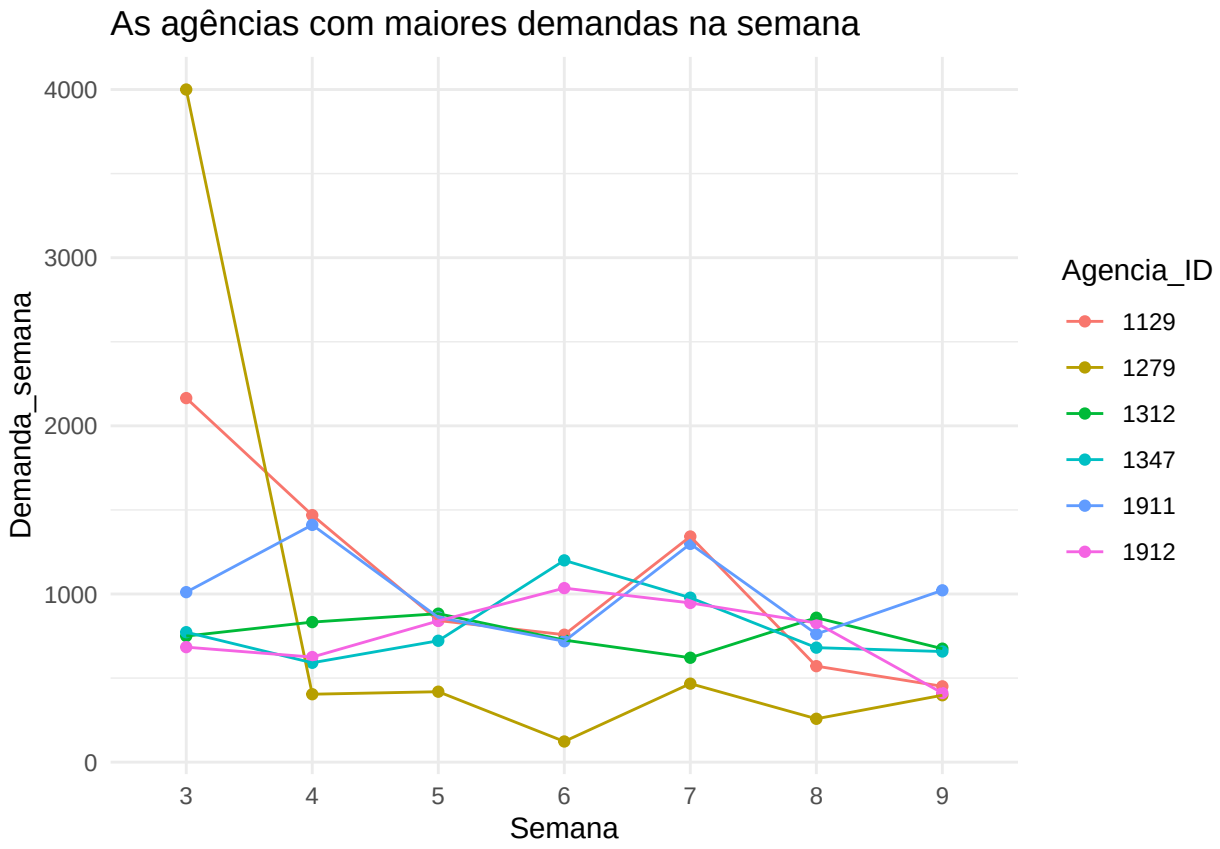



```
# Os cinco produtos com maiores demandas
Agencia_5_mais <- SampleSale %>%
  group_by(Agencia_ID) %>%
  summarise(Quant_vend_prod = sum(Demanda_uni_equil)) %>%
  arrange(desc(Quant_vend_prod))

head(Agencia_5_mais)
```

```
## # A tibble: 6 x 2
##   Agencia_ID Quant_vend_prod
##        <dbl>           <dbl>
## 1       1129            7598
## 2       1911            7084
## 3       1279            6069
## 4       1347            5603
## 5       1912            5370
## 6       1312            5347
```

```
# Gráfico dos 5 maiores demandas para a agência por semana
df.Demand.Forecasting %>%
  filter(Agencia_ID %in% c(1129,1911,1279,1347,1912,1312)) %>%
  group_by(Semana, Agencia_ID) %>%
  arrange(Semana, Agencia_ID) %>%
  summarise(Demanda_semana = sum(Demanda_uni_equil)) %>%
  ggplot(mapping = aes(x = Semana, y = Demanda_semana, group = Agencia_ID,
                       colour = Agencia_ID)) +
  geom_point() +
  geom_line() + theme_minimal() +
```

```
  ggtitle("As agências com maiores demandas na semana")
```



Engenharia de atributos

Nesta etapa, variáveis serão modificadas devido sua alta cardinalidade, por exemplo. Além disso, novas variáveis serão criadas.

```
# Engenharia de atributos - substituindo categorias com alta cardinalidade pela frequência de
# cada categoria no data frame. Variáveis a serem encodadas: Agencia_ID, Ruta_SAK, Cliente_ID,
# Producto_ID.
df.Demand.Forecasting2 <- df.Demand.Forecasting %>%
  add_count(Producto_ID, name = "Freq_produto") %>%
  add_count(Agencia_ID, name = "Freq_Agencia") %>%
  add_count(Ruta_SAK, name = "Freq_ruta") %>%
  add_count(Cliente_ID, name = "Freq_cliente")

df.Demand.Forecasting2 <- df.Demand.Forecasting2[,-c(2, 4, 5, 6)]

head(df.Demand.Forecasting2)
```

```
## # A tibble: 6 x 7
##   Semana Canal_ID Demanda_uni_equ~ Freq_produto Freq_Agencia Freq_ruta
##   <fct>  <fct>               <dbl>        <int>        <int>     <int>
## 1 9      1                       0         2726          314       357
## 2 7      1                       3          294          333        45
## 3 8      1                       2          853          561       169
## 4 5      1                       2         1890          501       430
```

```
## 5 4      1                       6         2307          311        52
## 6 4      1                       3          306          591        78
## # ... with 1 more variable: Freq_cliente <int>
```

```
# Descrição estatística das novas variáveis
summary(df.Demand.Forecasting2)
```

```
##  Semana       Canal_ID     Demanda_uni_equil   Freq_produto   Freq_Agencia
##  3:15156   1      :90984   Min.    :   0.000   Min.    :   1   Min.    :   1.0
##  4:14855   4      : 5054   1st Qu.:   2.000   1st Qu.: 265   1st Qu.: 306.0
##  5:14362   11     : 1297   Median :   3.000   Median : 813   Median : 452.0
##  6:13764   2      : 1131   Mean   :   7.248   Mean   :1035   Mean   : 458.6
##  7:13977   7      :  886   3rd Qu.:   6.000   3rd Qu.:1615   3rd Qu.: 610.0
##  8:13781   6      :  354   Max.   :3840.000   Max.   :2903   Max.   :1097.0
##  9:14105   (Other):  294
##    Freq_ruta      Freq_cliente
##  Min.   :  1.0   Min.   :  1.000
##  1st Qu.: 81.0   1st Qu.:  1.000
##  Median :163.0   Median :  1.000
##  Mean   :194.7   Mean   :  1.516
##  3rd Qu.:269.0   3rd Qu.:  1.000
##  Max.   :644.0   Max.   :169.000
##
```

```
# Removendo outliers
df.Demand.Forecasting3 <- df.Demand.Forecasting2 %>%
  filter(Demanda_uni_equil < 200)

nrow(df.Demand.Forecasting3)
```

```
## [1] 99834
```

```
summary(df.Demand.Forecasting3)
```

```
##  Semana       Canal_ID     Demanda_uni_equil  Freq_produto   Freq_Agencia
##  3:15134   1      :90956   Min.    :  0.000   Min.    :   1   Min.    :   1.0
##  4:14827   4      : 5049   1st Qu.:  2.000   1st Qu.: 268   1st Qu.: 306.0
##  5:14338   11     : 1288   Median :  3.000   Median : 813   Median : 452.0
##  6:13740   2      : 1052   Mean   :  6.602   Mean   :1037   Mean   : 459.2
##  7:13955   7      :  878   3rd Qu.:  6.000   3rd Qu.:1615   3rd Qu.: 610.0
##  8:13756   6      :  354   Max.   :198.000   Max.   :2903   Max.   :1097.0
##  9:14084   (Other):  257
##    Freq_ruta    Freq_cliente
##  Min.   :  1   Min.   :  1.000
##  1st Qu.: 81   1st Qu.:  1.000
##  Median :163   Median :  1.000
##  Mean   :195   Mean   :  1.467
##  3rd Qu.:269   3rd Qu.:  1.000
##  Max.   :644   Max.   :169.000
##
```

```
head(df.Demand.Forecasting3)
```

```
## # A tibble: 6 x 7
##   Semana Canal_ID Demanda_uni_equ~ Freq_produto Freq_Agencia Freq_ruta
##   <fct>  <fct>               <dbl>        <int>        <int>     <int>
## 1 9      1                       0         2726          314       357
```

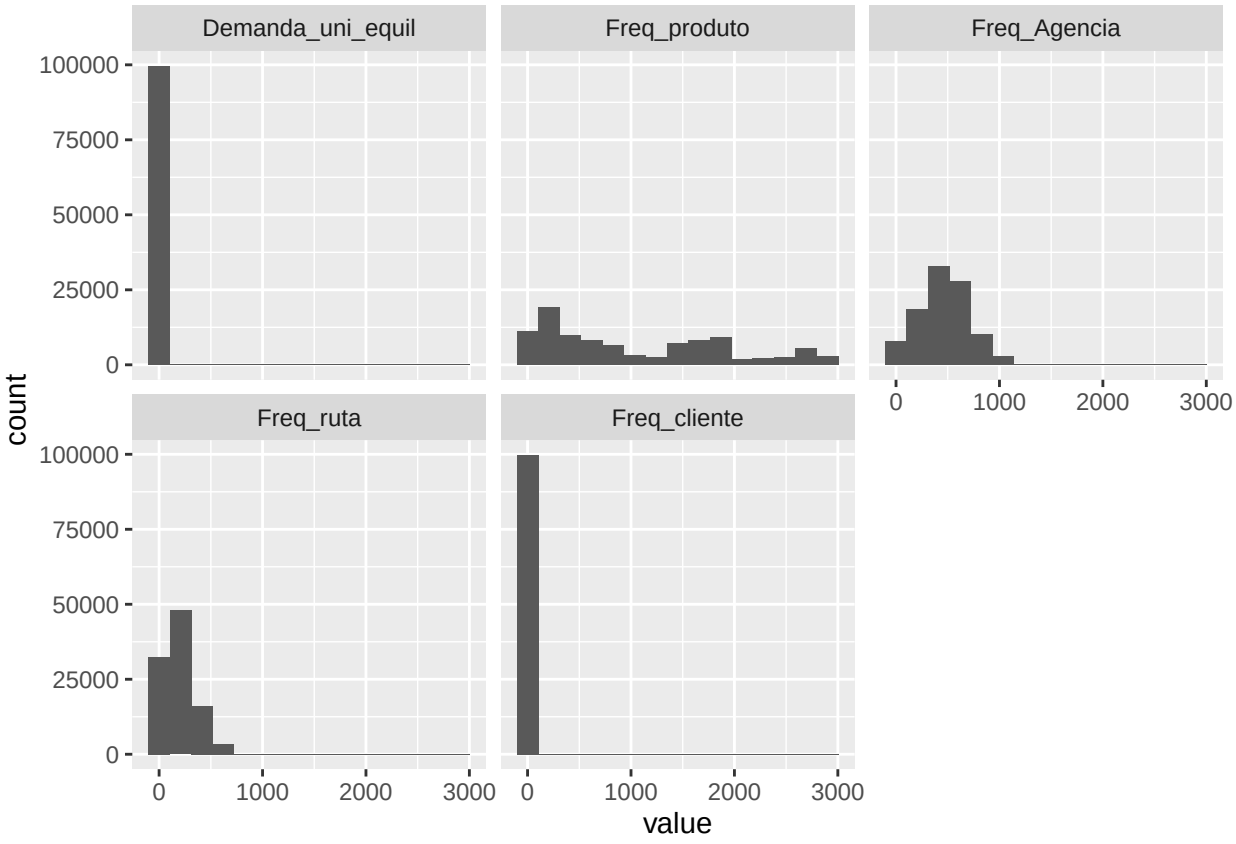
```
## 2 7      1                       3          294          333         45
## 3 8      1                       2          853          561        169
## 4 5      1                       2         1890          501        430
## 5 4      1                       6         2307          311         52
## 6 4      1                       3          306          591         78
## # ... with 1 more variable: Freq_cliente <int>
```

Analisando vários histogramas ao mesmo tempo.

```
# Histograma das novas variáveis
dfM <- melt(df.Demand.Forecasting3)
```

```
## Using Semana, Canal_ID as id variables
```

```
ggplot(dfM, mapping = aes(x=value)) +
  geom_histogram(bins = 15) +
  facet_wrap(~variable)
```



Normalização

Os dados serão normalizados, ou seja, serão transformados para valores no intervalo entre 0 e 1. Pois, neste formato, será requerido menor tempo de processamento e os algoritmos alcaçam melhores performances.

```
# Normalizando o data set (mix-max scalling)
preproc <- preProcess(df.Demand.Forecasting3[,c(1:2,4:7)], method = c("range"))
DataNorm <- predict(preproc, df.Demand.Forecasting3[,c(1:2,4:7)])

summary(DataNorm)
```

```
##  Semana        Canal_ID       Freq_produto       Freq_Agencia       Freq_ruta
```

```
##  3:15134   1      :90956   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000
##  4:14827   4      : 5049   1st Qu.:0.09201   1st Qu.:0.2783   1st Qu.:0.1244
##  5:14338   11     : 1288   Median :0.27981   Median :0.4115   Median :0.2519
##  6:13740   2      : 1052   Mean   :0.35690   Mean   :0.4181   Mean   :0.3017
##  7:13955   7      :  878   3rd Qu.:0.55617   3rd Qu.:0.5557   3rd Qu.:0.4168
##  8:13756   6      :  354   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000
##  9:14084   (Other):  257
##   Freq_cliente
##  Min.   :0.000000
##  1st Qu.:0.000000
##  Median :0.000000
##  Mean   :0.002779
##  3rd Qu.:0.000000
##  Max.   :1.000000
##
```

```
df.Demand.Norm <- cbind(DataNorm, df.Demand.Forecasting3[, 3])
head(df.Demand.Norm)
```

```
##   Semana Canal_ID Freq_produto Freq_Agencia  Freq_ruta Freq_cliente
## 1      9        1    0.9390076    0.2855839 0.55365474  0.000000000
## 2      7        1    0.1009649    0.3029197 0.06842924  0.000000000
## 3      8        1    0.2935906    0.5109489 0.26127527  0.005952381
## 4      5        1    0.6509304    0.4562044 0.66718507  0.000000000
## 5      4        1    0.7946244    0.2828467 0.07931571  0.000000000
## 6      4        1    0.1050999    0.5383212 0.11975117  0.000000000
##   Demanda_uni_equil
## 1                 0
## 2                 3
## 3                 2
## 4                 2
## 5                 6
## 6                 3
```

```
# Como existem muitos valore muito próximos a zero, então diminui eles de 1
df.Demand.Norm$Freq_cliente <- 1 - df.Demand.Norm$Freq_cliente
```

Criação do modelo

Alguns modelos como regressão linear, svm, redes neurais e árvores de decisão foram testado, entretanto, o de melhor performance foi o svm segundo a métrica RMSLE.

```
# Criando Modelo
set.seed(1)
amostra2 <- sample(1:nrow(df.Demand.Norm), 50000, replace = FALSE)

df_demanda <- df.Demand.Norm[amostra2,]

#############################################################################
######################### Cross Validation ##################################
#############################################################################
set.seed(1)
linhas <- sample(1:nrow(df_demanda), 0.7*nrow(df_demanda),
                 replace = FALSE)
```

```r
train_data <- df_demanda[linhas,]
test_data <- df_demanda[-linhas,]
head(train_data)
```

```
##       Semana Canal_ID Freq_produto Freq_Agencia  Freq_ruta Freq_cliente
## 8796       8        1   0.55547898   0.52828467 0.14152411    1.0000000
## 74619      3       11   0.07064094   0.01368613 0.09797823    1.0000000
## 94113      9        1   0.89869056   0.51277372 0.39346812    0.9940476
## 49188      9        1   0.62405238   0.69525547 0.97356143    1.0000000
## 31314      9        1   0.37215713   0.18430657 0.36391913    1.0000000
## 42918      5        1   0.29359063   0.24270073 0.15396579    1.0000000
##       Demanda_uni_equil
## 8796                  2
## 74619                90
## 94113                 5
## 49188                 6
## 31314                 0
## 42918                 1
```

```r
head(test_data)
```

```
##       Semana Canal_ID Freq_produto Freq_Agencia  Freq_ruta Freq_cliente
## 43307      4        1    0.6554101    0.5200730 0.65007776    1.0000000
## 25173      5        1    0.5000000    0.5027372 0.41057543    1.0000000
## 43809      4       11    0.2991041    0.1961679 0.00155521    1.0000000
## 92490      3        1    0.8390765    0.7463504 0.46656299    0.9940476
## 45399      9        1    0.8390765    0.6952555 0.16640747    1.0000000
## 92199      3        1    0.8390765    1.0000000 0.12130638    1.0000000
##       Demanda_uni_equil
## 43307                 1
## 25173                 9
## 43809                 7
## 92490                 2
## 45399                 5
## 92199                 1
```

```r
# Modelo support vector machine
library(e1071)

model <- svm(data = train_data, Demanda_uni_equil ~ .)
summary(model)
```

```
##
## Call:
## svm(formula = Demanda_uni_equil ~ ., data = train_data)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.05263158
##     epsilon:  0.1
##
##
```

```
## Number of Support Vectors:  24464
```

```
pred <- fitted(model)
RMSLE <- sqrt(1/length(train_data$Demanda_uni_equil)*
                sum((log(pred + 1) -
                       log(train_data$Demanda_uni_equil + 1))^2))
RMSLE
```

```
## [1] 0.7662853
```