Prevendo Demanda de Estoque com Base em Vendas

Sobre

Projeto com feedback 02 do curso Big Data Analytics com R e Microsoft Azure Machine da Formação Cientista de Dados da Data Science Academy.

Dataset disponibilizado no kaggle pelo grupo Bimbo no desafio Grupo Bimbo Inventory Demand - Maximize sales and minimize returns of bakery goods.

Link do dataset: https://www.kaggle.com/c/grupo-bimbo-inventory-demand

Objetivo: Construir um modelo de machine learning para prever com precisão a demanda de estoque com base nos dados históricos de vendas.

Baixando pacotes necessários

```
library(kableExtra)
library(data.table)
library(stringr)
library(dplyr)
library(naniar)
library(ggplot2)
library(gridExtra)
library(psych)
library(ggcorrplot)
library(randomForest)
library(RColorBrewer)
```

Amostra Dataset Principal

```
memory.limit(9999999999)

df <- fread('train.csv')
df <- as.data.frame(df)

cliente <- read.csv('cliente_tabla.csv', sep = ',', header = T, stringsAsFactors = F)

produto <- read.csv('producto_tabla.csv', sep = ',', header = T, stringsAsFactors = F)

local <- read.csv('town_state.csv', sep = ',', header = T, stringsAsFactors = F)

head(df[,1:6]) %>%
    kbl(caption = 'Amostra - Dataset Principal - Colunas 1 à 6') %>%
```

```
    kable_paper('striped',full_width = F) %>%
    row_spec(0, bold = T) %>%
    footnote('Grupo Bimbo - Prevendo Estoque')

head(df[,7:11]) %>%
    kbl(caption = 'Amostra - Dataset Principal - Colunas 7 à 11') %>%
    kable_paper('striped',full_width = F) %>%
    row_spec(0, bold = T) %>%
    footnote('Grupo Bimbo - Prevendo Estoque')
```

Dicionário de dados

```
dic <- data.frame(variavel = c('Semana', 'Agencia_ID', 'Canal_ID',
                               'Ruta_SAK', 'Cliente_ID', 'NombreCliente',
                               'Producto_ID', 'NombreProducto',
                               'Venta_uni_hoy', 'Venta_hoy',
                               'Dev_uni_proxima', 'Dev_proxima',
                               'Demanda_uni_equil'),
                  descricao = c('número do dia da semana', 'ID do depósito de vendas',
                                'ID do canal de vendas', 'ID da rota', 'ID do cliente',
                                'nome do cliente', 'ID do produto', 'nome do produto',
                                'quantidade de vendas da semana',
                                'valor do total de vendas da semana',
                                'unidades retornadas próxima semana',
                                'valor do total retornados próxima semana',
                                'valor da demana (variável target)'),
                  tipo = c('inteiro', 'inteiro', 'inteiro', 'inteiro', 'inteiro',
                           'string', 'inteiro', 'string', 'inteiro', 'double',
                           'inteiro', 'double', 'fator'),
                  valores_permitidos = c('números', 'números', 'números', 'números',
                                         'números', 'texto', 'números', 'texto',
                                         'números', 'valores numéricos', 'números',
                                         'valores numéricos', 'números'))

dic %>%
    rename('Variável' = variavel,
           'Descrição' = descricao,
           'Tipo de dado' = tipo,
           'Valores permitidos' = valores_permitidos) %>%
    kbl() %>% kable_paper('striped',full_width = F) %>%
    row_spec(0, bold = T)
```

Variável	Descrição	Tipo de dado	Valores permitidos
Semana	número do dia da semana	inteiro	números
Agencia_ID	ID do depósito de vendas	inteiro	números
Canal_ID	ID do canal de vendas	inteiro	números
Ruta_SAK	ID da rota	inteiro	números
Cliente_ID	ID do cliente	inteiro	números
NombreCliente	nome do cliente	string	texto
Producto_ID	ID do produto	inteiro	números
NombreProducto	nome do produto	string	texto
Venta_uni_hoy	quantidade de vendas da semana	inteiro	números
Venta_hoy	valor do total de vendas da semana	double	valores numéricos
Dev_uni_proxima	unidades retornadas próxima semana	inteiro	números
Dev_proxima	valor do total retornados próxima semana	double	valores numéricos
Demanda_uni_equil	valor da demana (variável target)	fator	números

Pré-processamento de dados

Alterando variáveis categóricas

(Agencia_ID, Canal_ID, Ruta_SAK, Cliente_ID, Producto_ID)

```
fatores <- c('Agencia_ID', 'Canal_ID', 'Ruta_SAK', 'Cliente_ID', 'Producto_ID')
for (i in colnames(df)) {
    if (i %in% fatores) {
        df[,i] <- as.factor(df[,i])
    }
}
```

Datas - criando coluna com nome e transformando em fator

(Segunda à Sexta - 1 à 7)

```
memory.limit(9999999999)

df$Semana_Ext <- factor(df$Semana, labels = c("quinta", "sexta", "sábado", "domingo",
                                              "segunda", "terça", "quarta"))
df$Semana <- as.factor(df$Semana)
```

Dataset Cliente - alterando coluna de fator e apresentando amostra

```
cliente$Cliente_ID <- as.factor(cliente$Cliente_ID)

head(cliente) %>%
  kbl(caption = 'Amostra - Dataset de Cliente') %>%
  kable_paper('striped',full_width = F) %>%
  row_spec(0, bold = T) %>%
  footnote('Grupo Bimbo - Prevendo Estoque')
```

Table 1: Amostra - Dataset de Cliente

Cliente_ID	NombreCliente
0	SIN NOMBRE
1	OXXO XINANTECATL
2	SIN NOMBRE
3	EL MORENO
4	SDN SER DE ALIM CUERPO SA CIA DE INT
4	SDN SER DE ALIM CUERPO SA CIA DE INT

Note:
Grupo Bimbo - Prevendo Estoque

Table 2: Amostra - Dataset de Produto

Producto_ID	NombreProducto
0	NO IDENTIFICADO 0
9	Capuccino Moka 750g NES 9
41	Bimbollos Ext sAjonjoli 6p 480g BIM 41
53	Burritos Sincro 170g CU LON 53
72	Div Tira Mini Doradita 4p 45g TR 72
73	Pan Multigrano Linaza 540g BIM 73

Note:
Grupo Bimbo - Prevendo Estoque

Dataset Produto - alterando coluna de fator e apresentando amostra

```
produto$Producto_ID <- as.factor(produto$Producto_ID)

head(produto) %>%
  kbl(caption = 'Amostra - Dataset de Produto') %>%
  kable_paper('striped',full_width = F) %>%
  row_spec(0, bold = T) %>%
  footnote('Grupo Bimbo - Prevendo Estoque')
```

Dataset Local - alterando coluna de fator, acertando nome de país (México) e apresentando amostra

```
local$Agencia_ID <- as.factor(local$Agencia_ID)
local$State <- str_replace_all(string = local$State, pattern = 'Ã‰', replacement = 'é')
local$State <- as.factor(local$State)

head(local) %>%
  kbl(caption = 'Amostra - Dataset de Local') %>%
  kable_paper('striped',full_width = F) %>%
  row_spec(0, bold = T) %>%
  footnote('Grupo Bimbo - Prevendo Estoque')
```

Table 3: Amostra - Dataset de Local

Agencia_ID	Town	State
1110	2008 AG. LAGO FILT	MéXICO, D.F.
1111	2002 AG. AZCAPOTZALCO	MéXICO, D.F.
1112	2004 AG. CUAUTITLAN	ESTADO DE MéXICO
1113	2008 AG. LAGO FILT	MéXICO, D.F.
1114	2029 AG.IZTAPALAPA 2	MéXICO, D.F.
1116	2011 AG. SAN ANTONIO	MéXICO, D.F.

Note:
Grupo Bimbo - Prevendo Estoque

Table 4: Amostra - Dataset Final - Colunas 1 à 6

Semana	Agencia_ID	Canal_ID	Ruta_SAK	Cliente_ID	Producto_ID
7	1275	4	6614	2391881	972
7	2032	1	1252	287003	1109
9	1310	1	1261	4502462	2425
9	1118	1	1001	1892686	1109
8	2057	1	1274	1983475	1284
5	1127	1	1406	349696	1240

Redução do dataset

Tendo em vista a lentidão nas etapas de pré-processamento proporcionada pelo tamanho do dataset, optei por trabalhar uma amostra do mesmo para que não ocorram demoras e/ou problemas na análise exploratória e no treinamento do algoritmo

```
df_amostra <- df[sample(1:nrow(df), 100000),]
```

União de datasets

A fim de montar um dataset único de trabalho, realizei os joins de acordo com as chaves (Cliente_ID, Producto_ID, Agencia_ID)

```
df_geral <- df_amostra %>%
    inner_join(cliente, by = 'Cliente_ID') %>%
    inner_join(produto, by = 'Producto_ID') %>%
    inner_join(local, by = 'Agencia_ID')

head(df_geral[,1:6]) %>%
  kbl(caption = 'Amostra - Dataset Final - Colunas 1 à 6') %>%
  kable_paper('striped',full_width = F) %>%
  row_spec(0, bold = T)
```

```
head(df_geral[,7:12]) %>%
  kbl(caption = 'Amostra - Dataset Final - Colunas 7 à 12') %>%
  kable_paper('striped',full_width = F) %>%
  row_spec(0, bold = T)
```

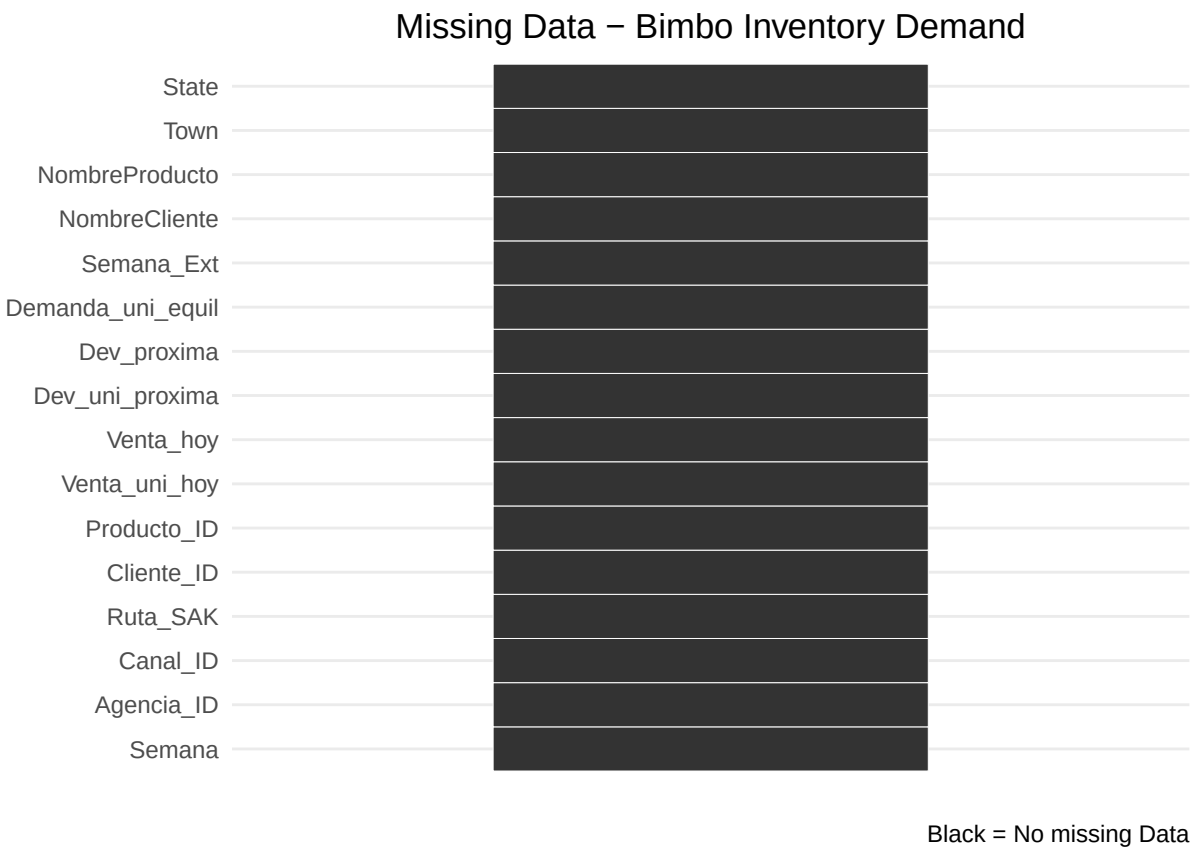
Table 5: Amostra - Dataset Final - Colunas 7 à 12

Venta_uni_hoy	Venta_hoy	Dev_uni_proxima	Dev_proxima	Demanda_uni_equil	Semana_Ext
1	19.56	0	0.00	1	segunda
1	15.01	2	30.02	0	segunda
12	54.00	0	0.00	12	quarta
7	105.07	0	0.00	7	quarta
14	42.28	0	0.00	14	terça
7	58.66	0	0.00	7	sábado

Data Missing

Abaixo podemos ver que o dataset final de trabalho não apresenta nenhum dado faltante

```
gg_miss_which(df_geral) +
    labs(title = 'Missing Data - Bimbo Inventory Demand',
         caption = 'Black = No missing Data') +
    theme(plot.title = element_text(hjust = 0.5))
```



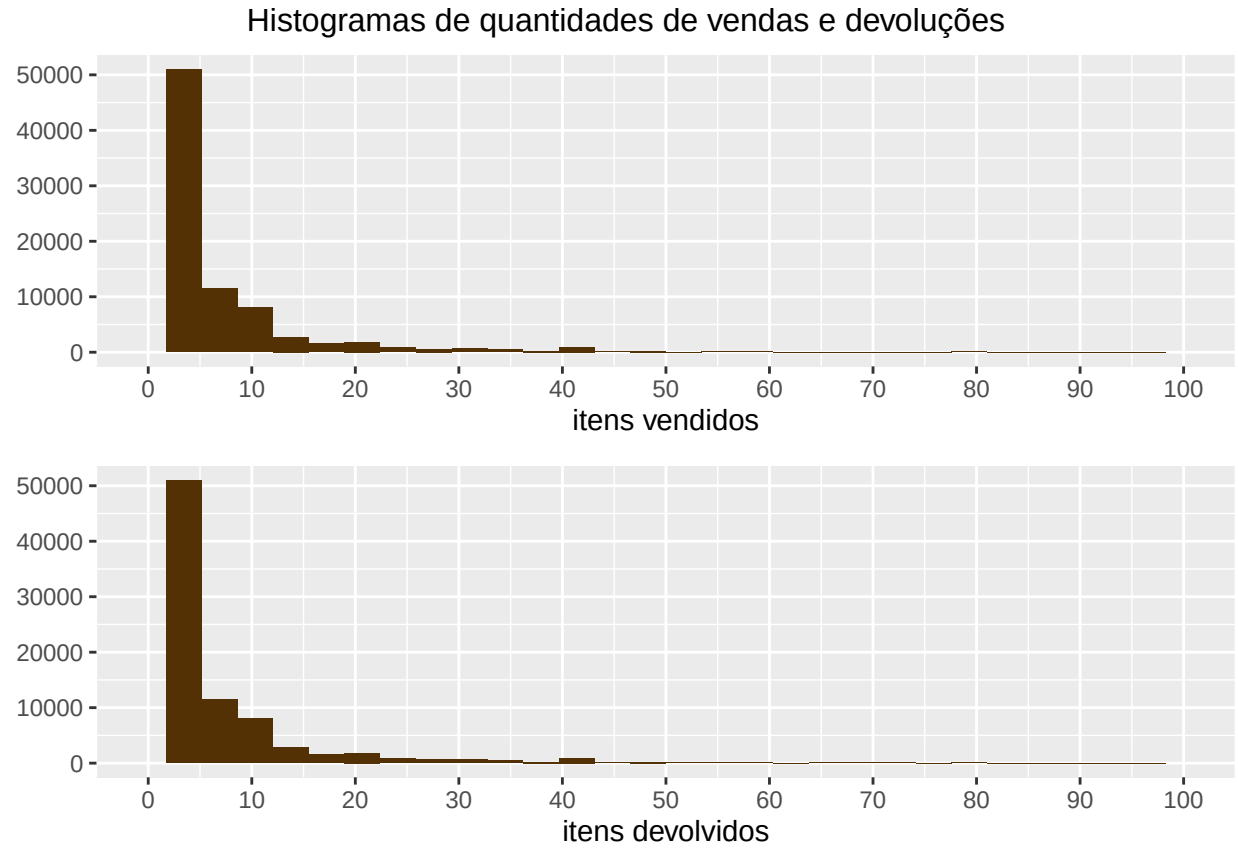
Análise Exploratória de Dados

Histogramas de vendas e devoluções - Qual é a quantidade de itens vendidos e devolvidos mais recorrente?

```r
ven_hist <- df_geral %>%
    filter(Venta_uni_hoy >= 0)  %>%
    ggplot(aes(Venta_uni_hoy)) + geom_histogram(fill = '#543005') +
    scale_x_continuous(limits = c(0, 100),
                       breaks = seq(0, 100, 10)) +
    xlab('itens vendidos') + ylab(NULL)

dev_hist <- df_geral %>%
    filter(Dev_uni_proxima >= 0)  %>%
    ggplot(aes(Venta_uni_hoy)) + geom_histogram(fill = '#543005') +
    scale_x_continuous(limits = c(0, 100),
                       breaks = seq(0, 100, 10)) +
    xlab('itens devolvidos') + ylab(NULL)

grid.arrange(ven_hist, dev_hist, ncol = 1, nrow = 2,
             top = 'Histogramas de quantidades de vendas e devoluções')
```



As quantidades vendidas e devolvidas não divergem muito, ficando em sua maioria entre 3 e 15 unidades.

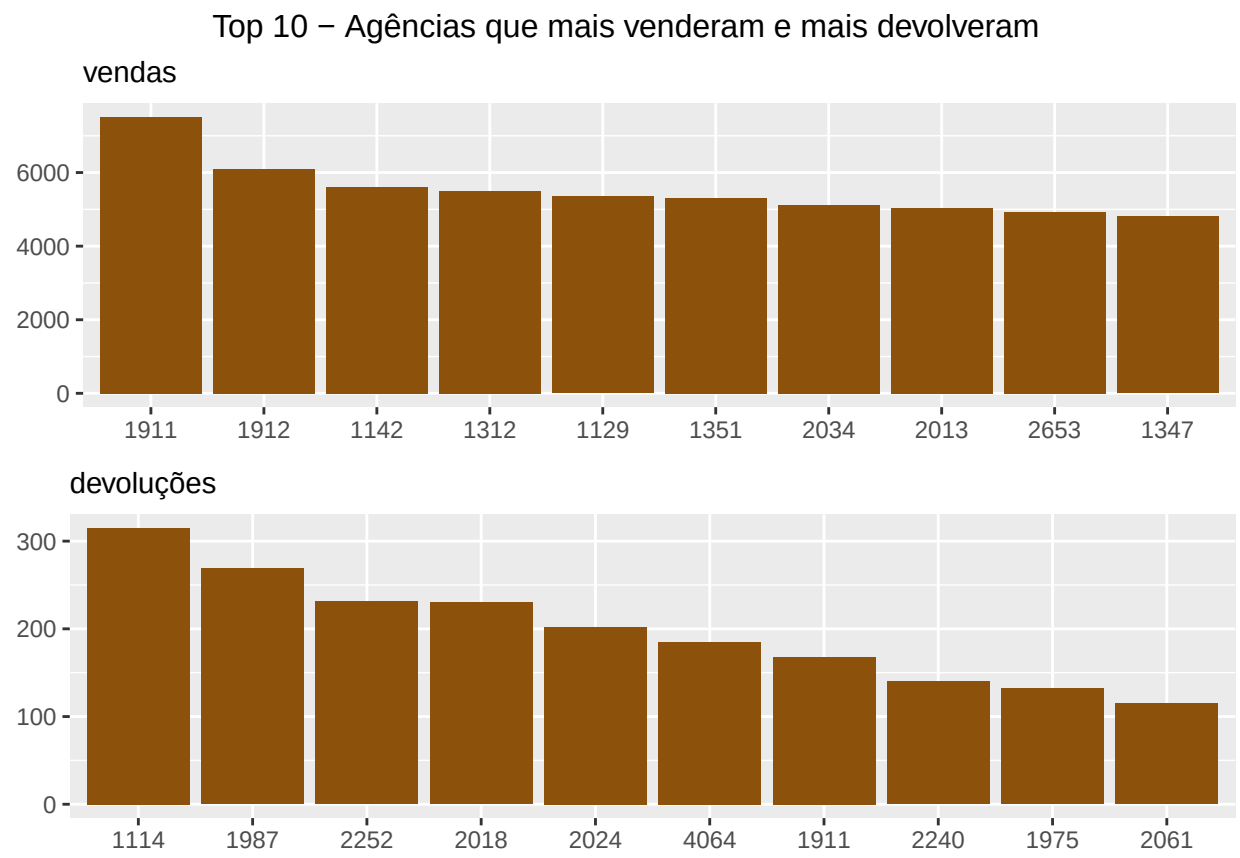
Quantidade de vendas e devoluções por agencia (top10)

```r
ven_agencia <- df_geral %>%
    group_by(Agencia_ID) %>%
    summarise(qtd = sum(Venta_uni_hoy)) %>%
    slice_max(qtd, n = 10) %>%
    ggplot(aes(x = reorder(Agencia_ID, -qtd), y = qtd)) +
```

```
    geom_col(fill = '#8C510A') +
    labs(subtitle = 'vendas') +
    xlab(NULL) + ylab(NULL)

dev_agencia <- df_geral %>%
    group_by(Agencia_ID) %>%
    summarise(qtd = sum(Dev_uni_proxima)) %>%
    slice_max(qtd, n = 10) %>%
    ggplot(aes(x = reorder(Agencia_ID, -qtd), y = qtd)) +
    geom_col(fill = '#8C510A') +
    labs(subtitle = 'devoluções') +
    xlab(NULL) + ylab(NULL)

grid.arrange(ven_agencia, dev_agencia, ncol = 1, nrow = 2,
             top = 'Top 10 - Agências que mais venderam e mais devolveram')
```



As agências que mais vendem não são as que mais devolvem. Da lista do top 10 de cada categoria, observamos que apenas a 2034 aparece nas duas, sendo a 7ª que mais vende e a 6ª que mais devolve.

Valor de vendas e devoluções por canais

```
df_geral %>%
    group_by(Canal_ID) %>%
    summarise(valor = sum(Venta_hoy)) %>%
    slice_max(valor, n = 5) %>%
    rename('R$' = valor) %>%
```

Table 6: Valor das vendas dos canais mais exitosos

Canal_ID	R$
1	4305721.8
2	1232421.9
4	639784.2
5	244737.9
11	220848.0

Table 7: Valor das devoluções dos canais menos exitosos

Canal_ID	R$
1	80869.56
2	30371.98
5	16737.87
4	3488.44
11	3458.57

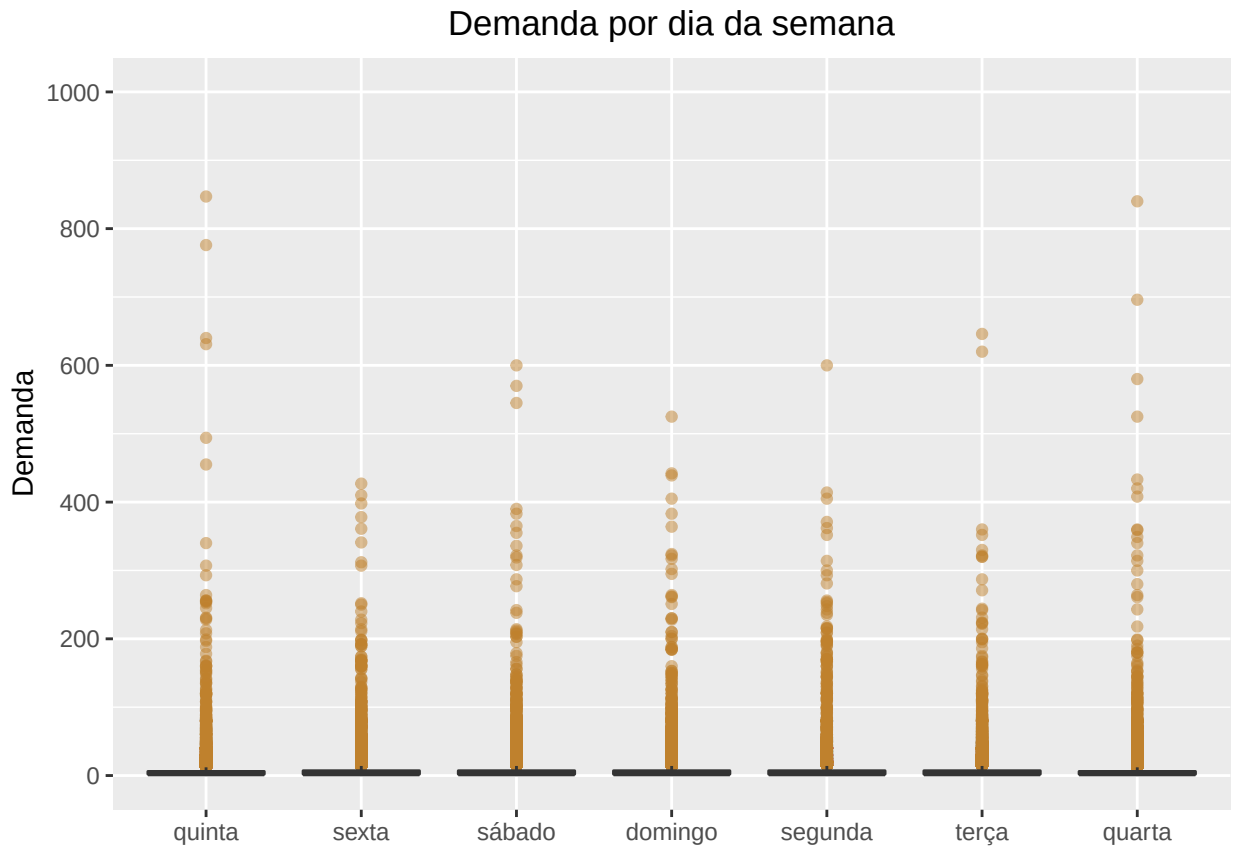
```
    kbl(caption = 'Valor das vendas dos canais mais exitosos') %>%
    kable_paper('striped',full_width = F) %>%
    row_spec(0, bold = T)
```

```
df_geral %>%
    group_by(Canal_ID) %>%
    summarise(valor = sum(Dev_proxima)) %>%
    slice_max(valor, n = 5) %>%
    rename('R$' = valor) %>%
    kbl(caption = 'Valor das devoluções dos canais menos exitosos') %>%
    kable_paper('striped',full_width = F) %>%
    row_spec(0, bold = T)
```

Nos dois quadros acima observamos que os valores das vendas são muito maiores que os das devoluções.

Demanda por dia da semana

```
df_geral %>%
    ggplot(aes(x = Semana_Ext, y = Demanda_uni_equil)) +
    geom_boxplot(outlier.colour = '#BF812D', outlier.alpha = .5) +
    xlab(NULL) + ylab('Demanda') + ggtitle('Demanda por dia da semana') +
    scale_y_continuous(limits = c(0, 1000),
                       breaks = seq(0, 1000, 200)) +
    theme(plot.title = element_text(hjust = 0.5))
```



A demanda diária parece seguir o mesmo padrão e a mesma média, com alguns outliers (atentar sábado e domingo) que podem ser relevantes.

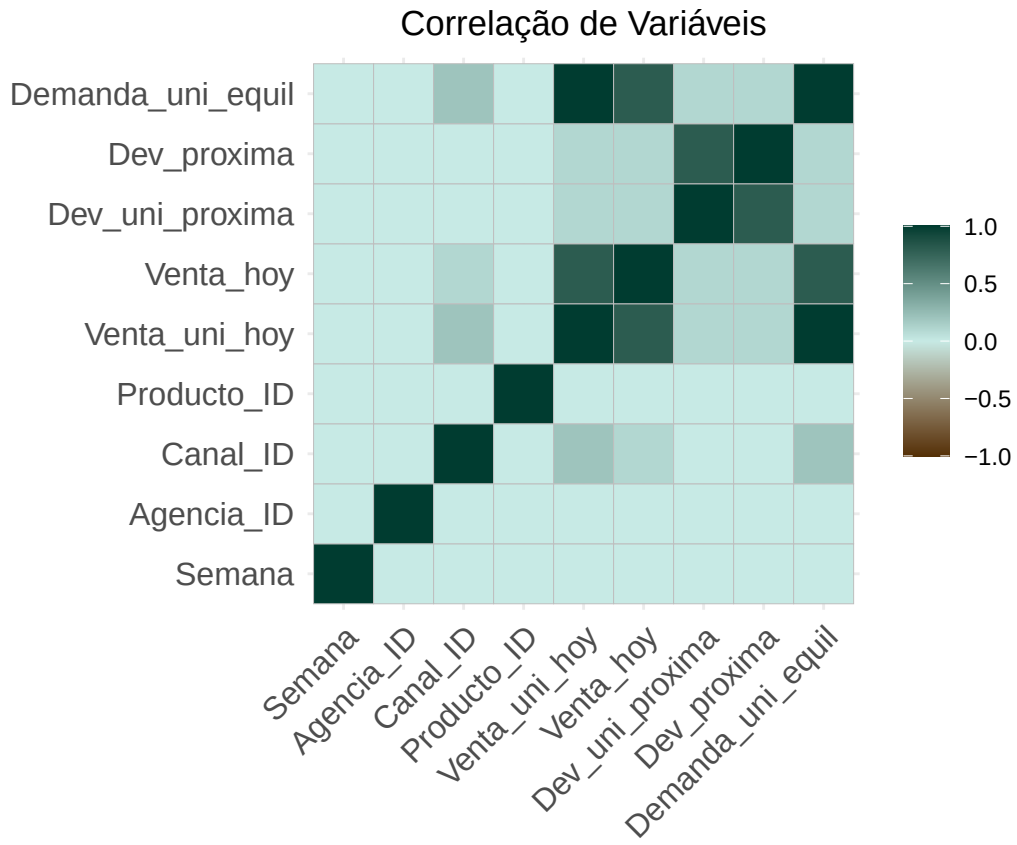
Correlação de Variáveis

```
df_cor_temp <- df_geral

df_cor_temp$Semana <- as.numeric(df_cor_temp$Semana)
df_cor_temp$Agencia_ID <- as.numeric(df_cor_temp$Agencia_ID)
df_cor_temp$Canal_ID <- as.numeric(df_cor_temp$Canal_ID)
df_cor_temp$Producto_ID <- as.numeric(df_cor_temp$Producto_ID)

df_cor <- round(cor(df_cor_temp[,c(1:3, 6:11)]), 1)

ggcorrplot(df_cor, title = 'Correlação de Variáveis', legend.title = NULL,
           colors = c("#543005", "#C7EAE5", "#003C30")) +
    theme(plot.title = element_text(hjust = 0.5))
```



É possível observar que a única coluna categórica que demonstra fraca correlação com as demais numéricas é a variável Canal_ID. As variáveis numéricas apresentam médias e fortes correlações entre si.

Processo de Machine Learning

Separação de datasets de treino e teste

O dataset de trabalho é o de treino, vamos baixar o de teste para realização das previsões:

```
teste <- fread('test.csv')
```

Treinamento do algoritmo

Modelo 1 - Regressão linear Conforme visto, além das variáveis numéricas, a única categórica que parece ter alguma relação e/ou influência no conjunto de dados é a Canal_ID.

Por isso, além das numéricas, vamos utilizá-la na primeira versão do nosso modelo:

```
modelo1 <- lm(Demanda_uni_equil ~ Venta_uni_hoy +
                Venta_hoy +
                Dev_uni_proxima +
                Dev_proxima +
                Canal_ID, data = df_geral)
```

```
previsao1 <- predict(modelo1, data = teste)
```

Previsão

```
summary(modelo1)
```

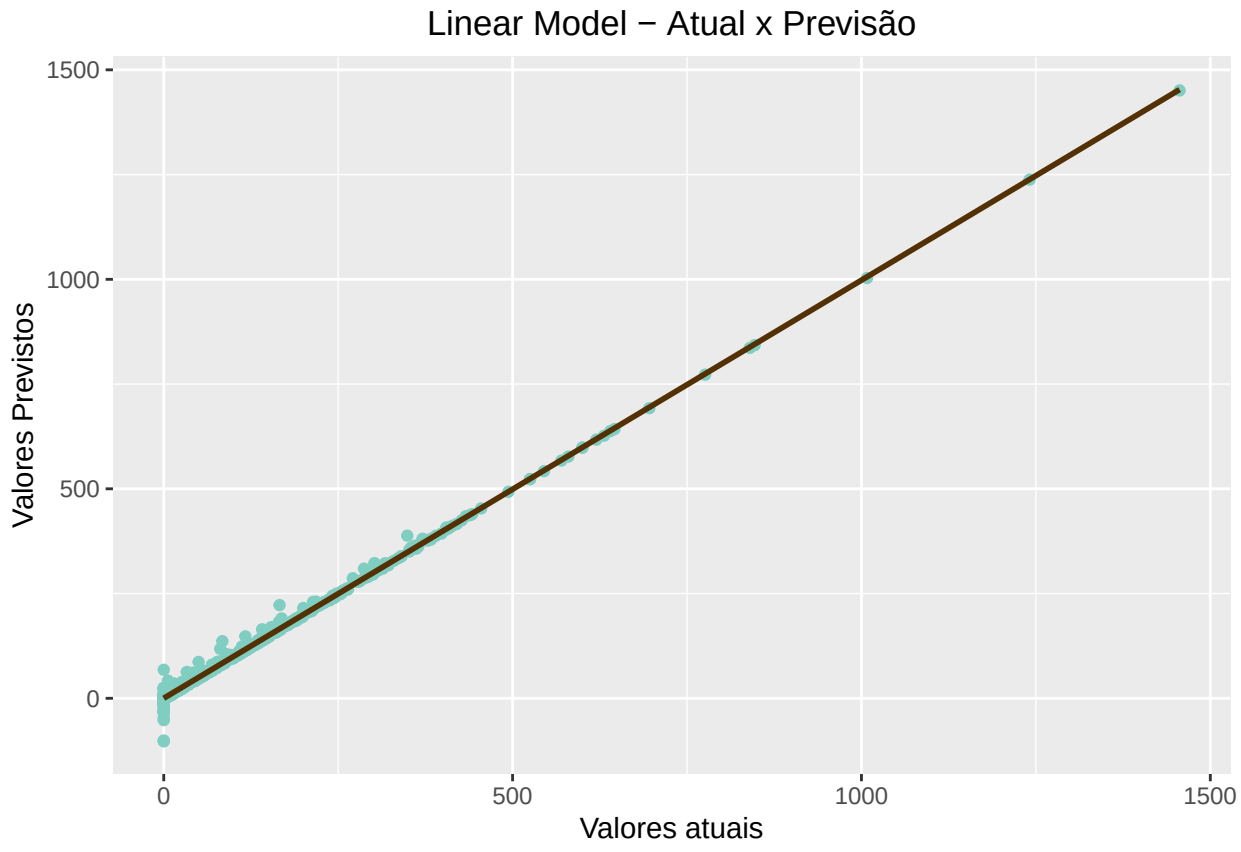
Análises

```
##
## Call:
## lm(formula = Demanda_uni_equil ~ Venta_uni_hoy + Venta_hoy +
##     Dev_uni_proxima + Dev_proxima + Canal_ID, data = df_geral)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -68.132   0.002   0.006   0.019 102.783
##
## Coefficients:
##                   Estimate Std. Error  t value Pr(>|t|)
## (Intercept)      6.248e-03  3.078e-03    2.030 0.042367 *
## Venta_uni_hoy    9.964e-01  2.694e-04 3698.959  < 2e-16 ***
## Venta_hoy       -7.284e-05  1.966e-05   -3.705 0.000211 ***
## Dev_uni_proxima -3.886e-01  2.615e-03 -148.576  < 2e-16 ***
## Dev_proxima     -9.221e-03  2.242e-04  -41.131  < 2e-16 ***
## Canal_ID2       -1.618e-01  2.837e-02   -5.704 1.18e-08 ***
## Canal_ID4        3.180e-02  1.265e-02    2.515 0.011921 *
## Canal_ID5       -3.186e+00  6.713e-02  -47.455  < 2e-16 ***
## Canal_ID6        3.055e-02  4.499e-02    0.679 0.497093
## Canal_ID7        1.638e-01  2.866e-02    5.716 1.09e-08 ***
## Canal_ID8        1.037e-01  9.452e-02    1.097 0.272792
## Canal_ID11       6.696e-02  2.409e-02    2.779 0.005450 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8807 on 100800 degrees of freedom
## Multiple R-squared:  0.9979, Adjusted R-squared:  0.9979
## F-statistic: 4.269e+06 on 11 and 100800 DF,  p-value: < 2.2e-16
```

- Variáveis com muitos asterísticos - a análise exploratória foi bem sucedida e ajudou a pré selecionar variáveis relevantes para o modelo como, por exemplo: Venta_uni_hoy, Venta_hoy e Dev_uni_proxima.
- Multiple R-Squared altíssimo - muito próximo de 1, indicando alto nível de precisão
- p-value baixo, indicando a alta probabilidade das variáveis serem relvantes para o modelo

```
score1 <- data.frame(atual = df_geral$Demanda_uni_equil,
                     previsao = previsao1)
```

```r
ggplot(score1, aes(x = atual, y = previsao1)) +
    geom_point(color = '#80CDC1') +
    geom_smooth(method = 'lm', color = '#543005') +
    labs(title = 'Linear Model - Atual x Previsão',
         x = 'Valores atuais',
         y = 'Valores Previstos') +
    theme(plot.title = element_text(hjust = 0.5))
```



No gráfico também podemos observar a ótima performance do modelo, onde os valores previstos se alteram de forma bastante discreta dos valores atuais.

```r
modelo2 <- randomForest(Demanda_uni_equil ~ Venta_uni_hoy +
                          Venta_hoy +
                          Dev_uni_proxima +
                          Dev_proxima +
                          Canal_ID,
                          data = df_geral)
```

Modelo 2 - Random Forest

```
previsao2 <- predict(modelo2, data = teste)
```

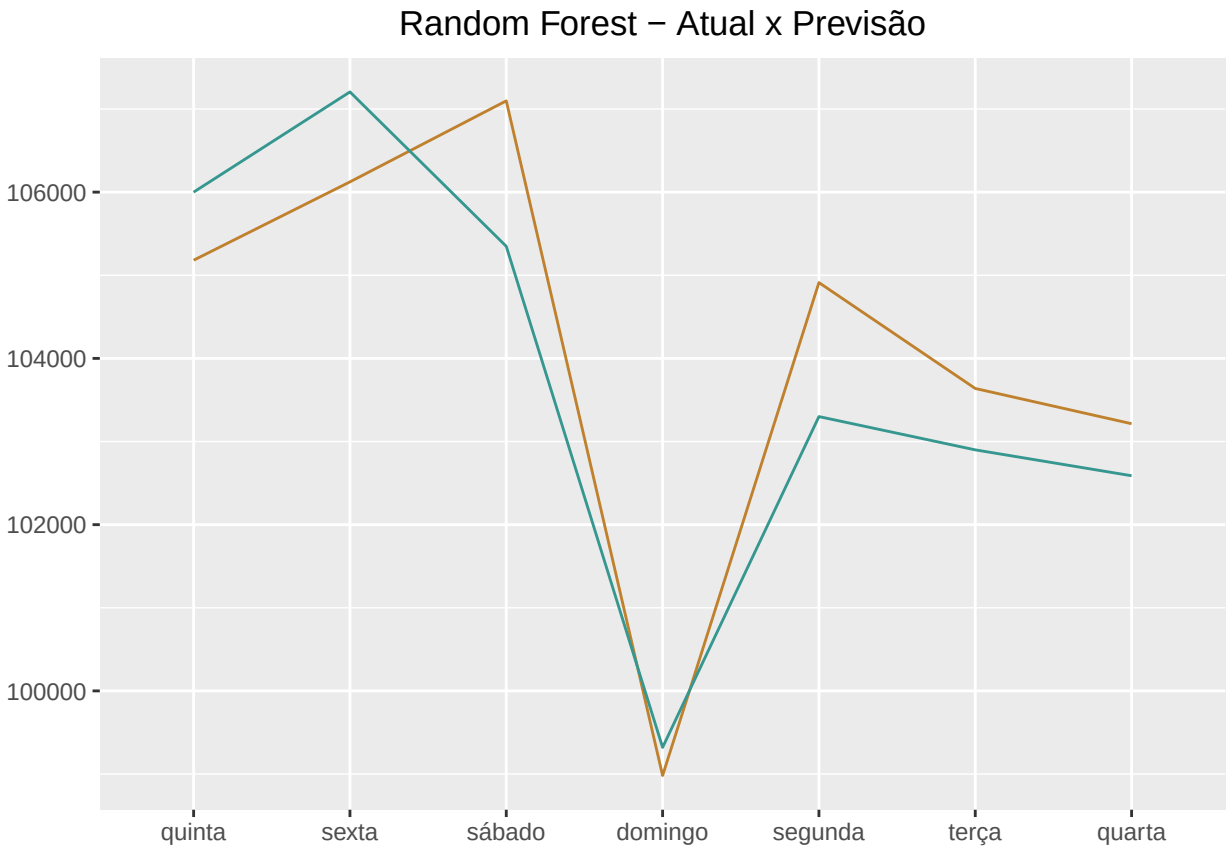
Previsão

```
print(modelo2)
```

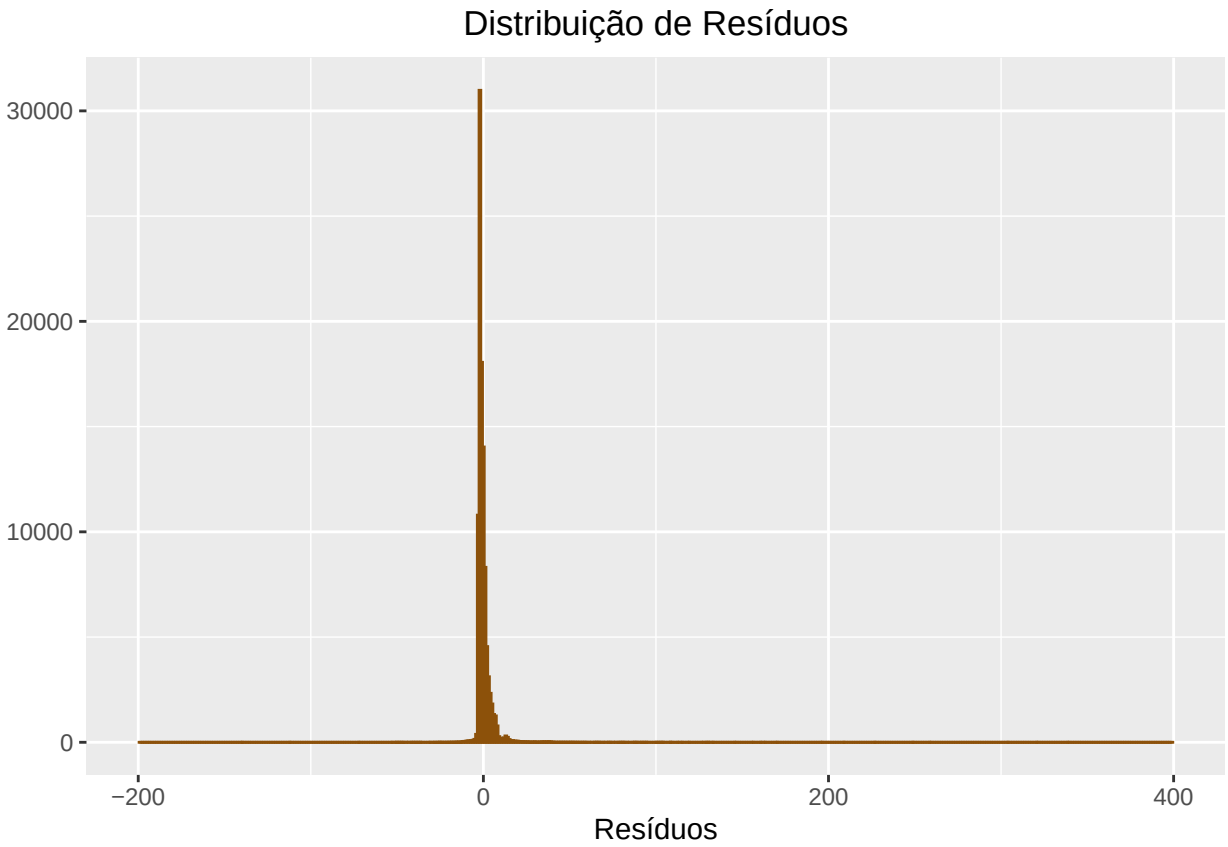
Análises

```
##
## Call:
##  randomForest(formula = Demanda_uni_equil ~ Venta_uni_hoy + Venta_hoy +      Dev_uni_proxima + Dev_p
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 49.29635
##                     % Var explained: 86.38
```

```
score2 <- data.frame(dia_da_semana = df_geral$Semana_Ext,
                     atual = df_geral$Demanda_uni_equil,
                     previsao = previsao2)
score2 %>%
    group_by(dia_da_semana) %>%
    summarise(qtd1 = sum(atual),
              qtd2 = sum(previsao)) %>%
    ungroup() %>%
    ggplot() +
    geom_line(aes(x = dia_da_semana, y = qtd1, group = 1), color = '#BF812D') +
    geom_line(aes(x = dia_da_semana, y = qtd2, group = 1), color = '#35978F') +
    labs(title = 'Random Forest - Atual x Previsão',
         x = NULL,
         y = NULL) +
    theme(plot.title = element_text(hjust = 0.5))
```




```
score2 %>%
    mutate(residuos = atual - previsao) %>%
    ggplot(aes(x = residuos)) +
    geom_histogram(binwidth = 1, fill = "#543005", color = "#8C510A") +
    scale_x_continuous(limits = c(-200, 400),
                       breaks = seq(-200, 400, 200)) +
    labs(title = 'Distribuição de Resíduos',
         x = 'Resíduos',
         y = NULL) +
    theme(plot.title = element_text(hjust = 0.5))
```



As linhas do primeiro gráfico mostram certa disparidade entre os valores atuais (linha marrom) e previstos (linha azul), porém, na distribuição dos resíduos, podemos ver que a maioria ficou em zero ou próximo de zero, levando ao entendimento que as diferenças entre valores atuais e previstos são nulos ou quase nulos.

O histograma nos fornece, também, a informação que a distribuição dos resíduos segue o formato de uma distribuição normal, dando indícios de um modelo com boa performance.

Conclusão

Apesar do bom desempenho do random forest, o modelo de regressão linear simples resultou em altíssimo R-Squared e, ao mesmo tempo, baixo p-value, indicando que seria o melhor caminho para previsão das demandas.