Prevendo Demanda Estoque

Jefferson Navarausckas

19/06/2019

1 - Carregando as bibliotecas necessarias

```
library("data.table")
```

```
## Warning: package 'data.table' was built under R version 3.5.2
```

```
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
##
## 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("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
library("corrplot")
```

```
## corrplot 0.84 loaded
```

```
library("caTools")
```

```
## Warning: package 'caTools' was built under R version 3.5.2
```

```
library("neuralnet")
```

```
## Warning: package 'neuralnet' was built under R version 3.5.2
```

```
##
## Attaching package: 'neuralnet'
```

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

2 - Coleta de dados

```
system.time(dados_cidades <- fread("town_state.csv"))
```

```
##    user  system elapsed
##   0.004   0.001   0.009
```

```r
system.time(dados_vendas <- fread("train.csv"))
```

```
##    user  system elapsed
##  27.662  16.307  48.728
```

3 - Analise exploratória inicial dos dados

Como o volume de dados é muito grande para capacidade da minha maquina, utilizarei uma amostra desses dados, utilizando 500 mil observacoes.

```r
dados_vendas <- dados_vendas %>%
  sample_n(size = 500000)

head(dados_vendas)
```

```
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID Venta_uni_hoy
## 1      4       4051        1     2004     617459       35456             7
## 2      3       2229        1     1212    1562616        2233             2
## 3      3       1315        1     4503    4632720        1240             1
## 4      6       1911        1     1158     105224        1064             1
## 5      9       2211        1     2810     861546       43069             2
## 6      5       1121        1     1418    1273210        1242             8
##   Venta_hoy Dev_uni_proxima Dev_proxima Demanda_uni_equil
## 1     31.78               0           0                 7
## 2     39.88               0           0                 2
## 3      8.38               0           0                 1
## 4     16.67               0           0                 1
## 5     14.82               0           0                 2
## 6     61.12               0           0                 8
```

```r
head(dados_cidades)
```

```
##    Agencia_ID                  Town            State
## 1:       1110    2008 AG. LAGO FILT     MÉXICO, D.F.
## 2:       1111 2002 AG. AZCAPOTZALCO     MÉXICO, D.F.
## 3:       1112   2004 AG. CUAUTITLAN ESTADO DE MÉXICO
## 4:       1113    2008 AG. LAGO FILT     MÉXICO, D.F.
## 5:       1114  2029 AG.IZTAPALAPA 2     MÉXICO, D.F.
## 6:       1116  2011 AG. SAN ANTONIO     MÉXICO, D.F.
```

```r
dim(dados_vendas)
```

```
## [1] 500000     11
```

```r
dim(dados_cidades)
```

```
## [1] 790   3
```

```r
str(dados_vendas)
```

```
## 'data.frame':    500000 obs. of  11 variables:
##  $ Semana          : int  4 3 3 6 9 5 5 3 3 9 ...
##  $ Agencia_ID      : int  4051 2229 1315 1911 2211 1121 2627 1245 1126 1635 ...
##  $ Canal_ID        : int  1 1 1 1 1 1 4 1 1 1 ...
##  $ Ruta_SAK        : int  2004 1212 4503 1158 2810 1418 6611 2821 1223 1018 ...
##  $ Cliente_ID      : int  617459 1562616 4632720 105224 861546 1273210 2359335 303856 1126138 145098
##  $ Producto_ID     : int  35456 2233 1240 1064 43069 1242 35456 43069 1109 2233 ...
##  $ Venta_uni_hoy   : int  7 2 1 1 2 8 1 5 2 1 ...
##  $ Venta_hoy       : num  31.78 39.88 8.38 16.67 14.82 ...
```

```
##  $ Dev_uni_proxima  : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Dev_proxima      : num  0 0 0 0 0 0 0 7.41 0 0 ...
##  $ Demanda_uni_equil: int  7 2 1 1 2 8 1 4 2 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(dados_cidades)
```

```
## Classes 'data.table' and 'data.frame':   790 obs. of  3 variables:
##  $ Agencia_ID: int  1110 1111 1112 1113 1114 1116 1117 1118 1119 1120 ...
##  $ Town      : chr  "2008 AG. LAGO FILT" "2002 AG. AZCAPOTZALCO" "2004 AG. CUAUTITLAN" "2008 AG. LAGO
##  $ State     : chr  "MÉXICO, D.F." "MÉXICO, D.F." "ESTADO DE MÉXICO" "MÉXICO, D.F." ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
any(is.na(dados_vendas))
```

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

```
any(is.na(dados_cidades))
```

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

4 - Analise, tratamento e transformacao dos dados

Alterando nome das variaveis

```
colnames(dados_vendas) <- c('dia','loja','id_canal_venda','id_rota','id_cliente','id_produto','unidade_
colnames(dados_cidades) <- c('loja','cidade','estado')
```

Incluindo uma coluna com os dias da semana e criando os fatores dos mesmos. Apos a criação dos fatores verificamos se há valores NA no dataset

```
dados_vendas$dia_semana = sapply(dados_vendas$dia, function(x){
  ifelse(x == 3 , "Quinta",
         ifelse(x == 4 , "Sexta",
                ifelse(x == 5 , "Sabado",
                       ifelse(x == 6 , "Domingo",
                              ifelse(x == 7 , "Segunda",
                                     ifelse(x == 8 , "Terca","Quarta")))))})

dados_vendas$dia_semana <- factor(dados_vendas$dia_semana, levels = c("Quinta", "Sexta", "Sabado", "Dom

any(is.na(dados_vendas))
```

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

Medidas de Tendência Central das variáveis numericas. Com base nesta analise identificamos que os valores da media e mediana estao muito distantes e tambem a diferenca entre o 3 quartil e o valor maximo.

```
summary(dados_vendas$unidade_venda)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.000    2.000    3.000    7.286    7.000 3164.000
```

```
summary(dados_vendas$unidade_dev_next_week)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.0000   0.0000   0.0000   0.1186   0.0000 240.0000
```

```
summary(dados_vendas$demanda_ajustada)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.000    2.000    3.000    7.203    6.000 3164.000
```

Analisando Quantil e Percentil das variaveis

```
quantil_vd <- quantile(dados_vendas$unidade_venda)
quantil_dv <- quantile(dados_vendas$unidade_dev_next_week)
quantil_da <- quantile(dados_vendas$demanda_ajustada)

quantile(dados_vendas$unidade_venda, probs = c(0.01, 0.99))
```

```
##  1% 99%
##   1  65
```

```
quantile(dados_vendas$unidade_venda, seq( from = 0, to = 1, by = 0.10))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##    0    1    2    2    3    3    4    6    8   14 3164
```

```
quantile(dados_vendas$unidade_dev_next_week, probs = c(0.01, 0.99))
```

```
##  1% 99%
##   0   3
```

```
quantile(dados_vendas$unidade_dev_next_week, seq( from = 0, to = 1, by = 0.10))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##    0    0    0    0    0    0    0    0    0    0  240
```

```
quantile(dados_vendas$demanda_ajustada, probs = c(0.01, 0.99))
```

```
##  1% 99%
##   0  64
```

```
quantile(dados_vendas$demanda_ajustada, seq( from = 0, to = 1, by = 0.10))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##    0    1    2    2    3    3    4    6    8   14 3164
```

Analisando a diferença entre Q3 e Q1

```
IQR(dados_vendas$unidade_venda)
```

```
## [1] 5
```

```
IQR(dados_vendas$unidade_dev_next_week)
```

```
## [1] 0
```

```
IQR(dados_vendas$demanda_ajustada)
```

```
## [1] 4
```

Verificando outliers abaixo e acima

Abaixo utilizaremos a formula Q1 - 1,5 * IQR

Acima utilizaremos a formula Q3 + 1,5 * IQR

```
quantil_vd[[2]] - (1.5 * IQR(dados_vendas$unidade_venda))
```

```
## [1] -5.5
```

```
quantil_dv[[2]] - (1.5 * IQR(dados_vendas$unidade_dev_next_week))
```

```
## [1] 0
```

```
quantil_da[[2]] - (1.5 * IQR(dados_vendas$demanda_ajustada))
```

```
## [1] -4
```

```
quantil_vd[[4]] + (1.5 * IQR(dados_vendas$unidade_venda))
```

```
## [1] 14.5
```

```
quantil_dv[[4]] + (1.5 * IQR(dados_vendas$unidade_dev_next_week))
```

```
## [1] 0
```

```
quantil_da[[4]] + (1.5 * IQR(dados_vendas$demanda_ajustada))
```

```
## [1] 12
```

Analisando desvio padrao

```
sd(dados_vendas$unidade_venda)
```

```
## [1] 20.57554
```

```
sd(dados_vendas$unidade_dev_next_week)
```

```
## [1] 1.598223
```

```
sd(dados_vendas$demanda_ajustada)
```

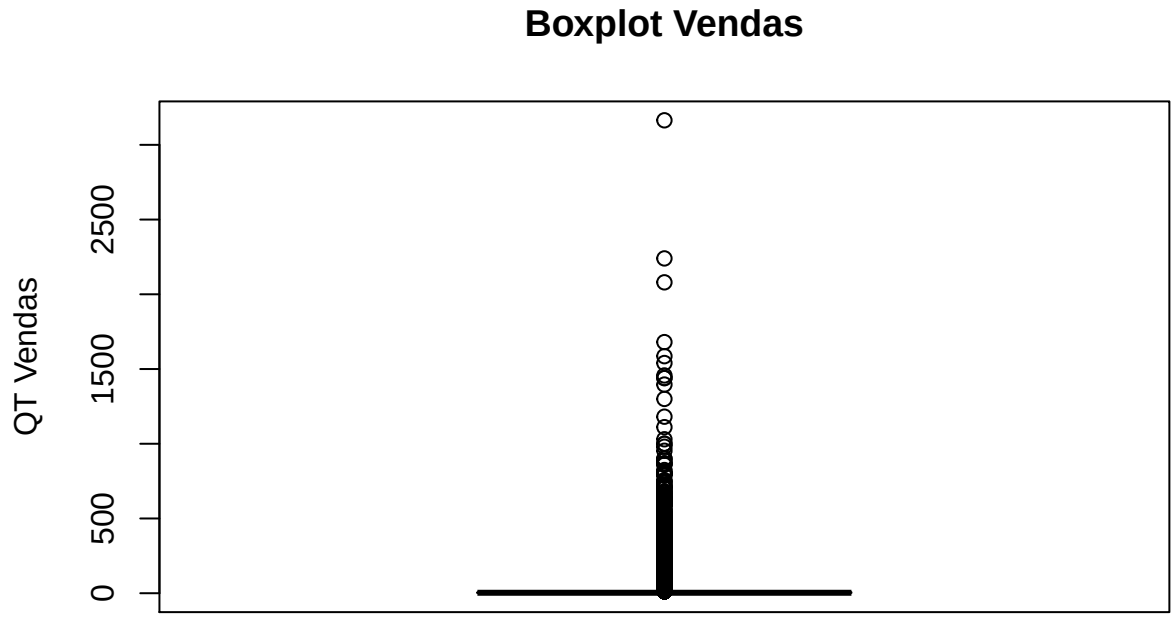
```
## [1] 20.42661
```

Criando Boxplot

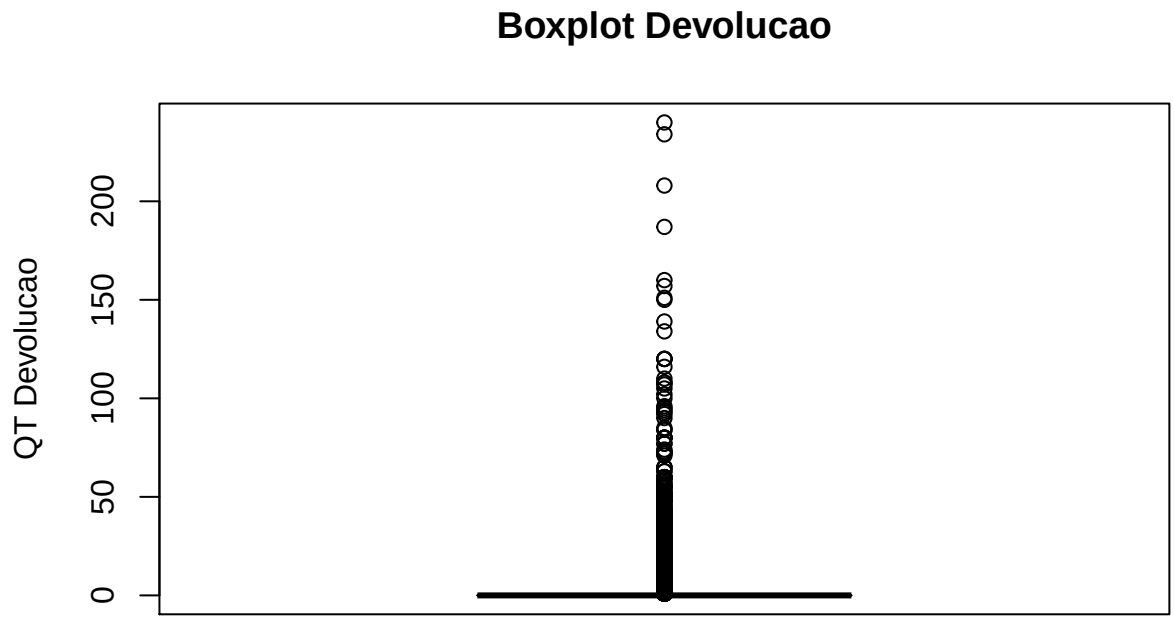
Observe que pelo boxplot podemos identificar muitos outiers

Leitura de Baixo para Cima - Q1, Q2 (Mediana) e Q3

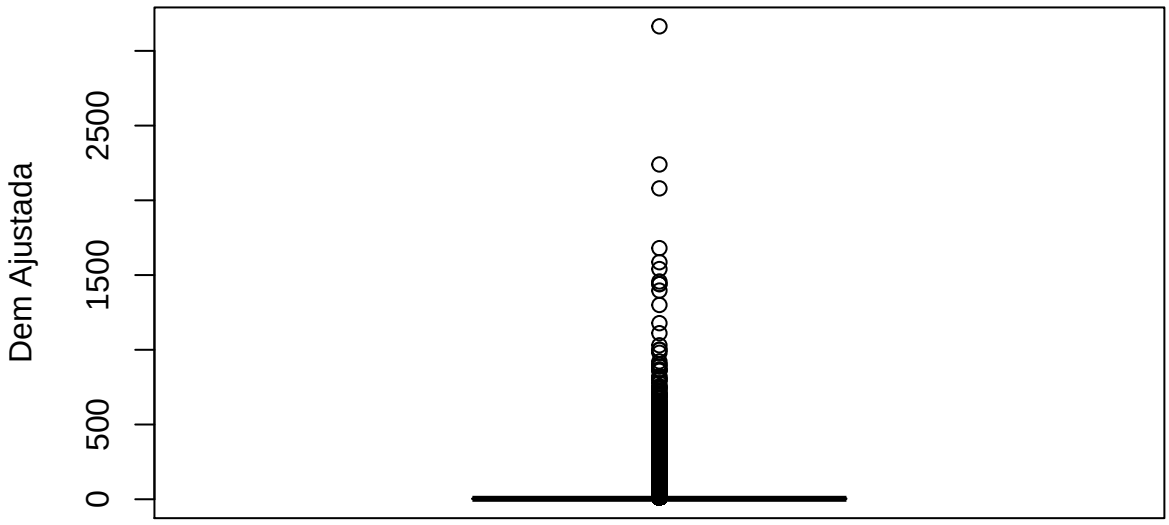
```
boxplot(dados_vendas$unidade_venda, main = "Boxplot Vendas", ylab = "QT Vendas")
```




```
boxplot(dados_vendas$unidade_dev_next_week, main = "Boxplot Devolucao", ylab = "QT Devolucao")
```




```
boxplot(dados_vendas$demanda_ajustada, main = "Boxplot Demanda Ajustada", ylab = "Dem Ajustada")
```



Criando um modelo SEM tratar nenhuma variavel para avaliar a acuracia antes e depois do tratamento das variaveis.

Podemos observar que a acuracia do modelo foi boa, porem há muitos graus de liberdade e muitos outliers nesses dados que ainda nao foram tratados o que pode prejudicar a generalizacao do modelo.

```
modelo_lm_v1 <- lm(demanda_ajustada ~. , data = dados_vendas)
modelo_lm_v2 <- lm(demanda_ajustada ~ dia + unidade_venda + id_produto + valor_venda + unidade_dev_next
modelo_lm_v3 <- lm(demanda_ajustada ~ dia + unidade_venda + valor_venda + unidade_dev_next_week + valor

summary(modelo_lm_v1)
```

```
##
## Call:
## lm(formula = demanda_ajustada ~ ., data = dados_vendas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -108.673   -0.001    0.007    0.021   86.571
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error   t value Pr(>|t|)
## (Intercept)           -2.489e-03  4.334e-03    -0.574  0.56575
## dia                   -6.578e-04  5.923e-04    -1.110  0.26679
## loja                  -6.001e-08  2.345e-07    -0.256  0.79801
## id_canal_venda         2.218e-03  7.580e-04     2.926  0.00343 **
## id_rota                5.822e-06  7.718e-07     7.543 4.58e-14 ***
## id_cliente             9.432e-10  5.233e-10     1.802  0.07150 .
## id_produto            -1.222e-08  5.411e-08    -0.226  0.82137
```

```
## unidade_venda          9.950e-01  7.962e-05 12497.042  < 2e-16 ***
## valor_venda            5.864e-05  5.542e-06    10.581  < 2e-16 ***
## unidade_dev_next_week -5.280e-01  9.470e-04  -557.593  < 2e-16 ***
## valor_dev_next_week    1.798e-03  8.819e-05    20.391  < 2e-16 ***
## dia_semanaSexta       -1.750e-03  3.253e-03    -0.538  0.59051
## dia_semanaSabado       2.460e-03  3.136e-03     0.784  0.43291
## dia_semanaDomingo      5.488e-04  3.137e-03     0.175  0.86110
## dia_semanaSegunda      1.126e-03  3.185e-03     0.353  0.72375
## dia_semanaTerca       -1.315e-03  3.354e-03    -0.392  0.69505
## dia_semanaQuarta              NA         NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6771 on 499984 degrees of freedom
## Multiple R-squared:  0.9989, Adjusted R-squared:  0.9989
## F-statistic: 3.03e+07 on 15 and 499984 DF,  p-value: < 2.2e-16
```

```
summary(modelo_lm_v2)
```

```
##
## Call:
## lm(formula = demanda_ajustada ~ dia + unidade_venda + id_produto +
##     valor_venda + unidade_dev_next_week + valor_dev_next_week,
##     data = dados_vendas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -108.699   -0.002    0.005    0.020   86.570
##
## Coefficients:
##                          Estimate Std. Error   t value Pr(>|t|)
## (Intercept)            1.151e-02  3.173e-03     3.628 0.000285 ***
## dia                   -6.683e-04  4.757e-04    -1.405 0.160069
## unidade_venda          9.950e-01  7.944e-05 12524.793  < 2e-16 ***
## id_produto             1.235e-07  5.141e-08     2.403 0.016272 *
## valor_venda            5.789e-05  5.539e-06    10.452  < 2e-16 ***
## unidade_dev_next_week -5.280e-01  9.471e-04  -557.494  < 2e-16 ***
## valor_dev_next_week    1.791e-03  8.818e-05    20.315  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6772 on 499993 degrees of freedom
## Multiple R-squared:  0.9989, Adjusted R-squared:  0.9989
## F-statistic: 7.574e+07 on 6 and 499993 DF,  p-value: < 2.2e-16
```

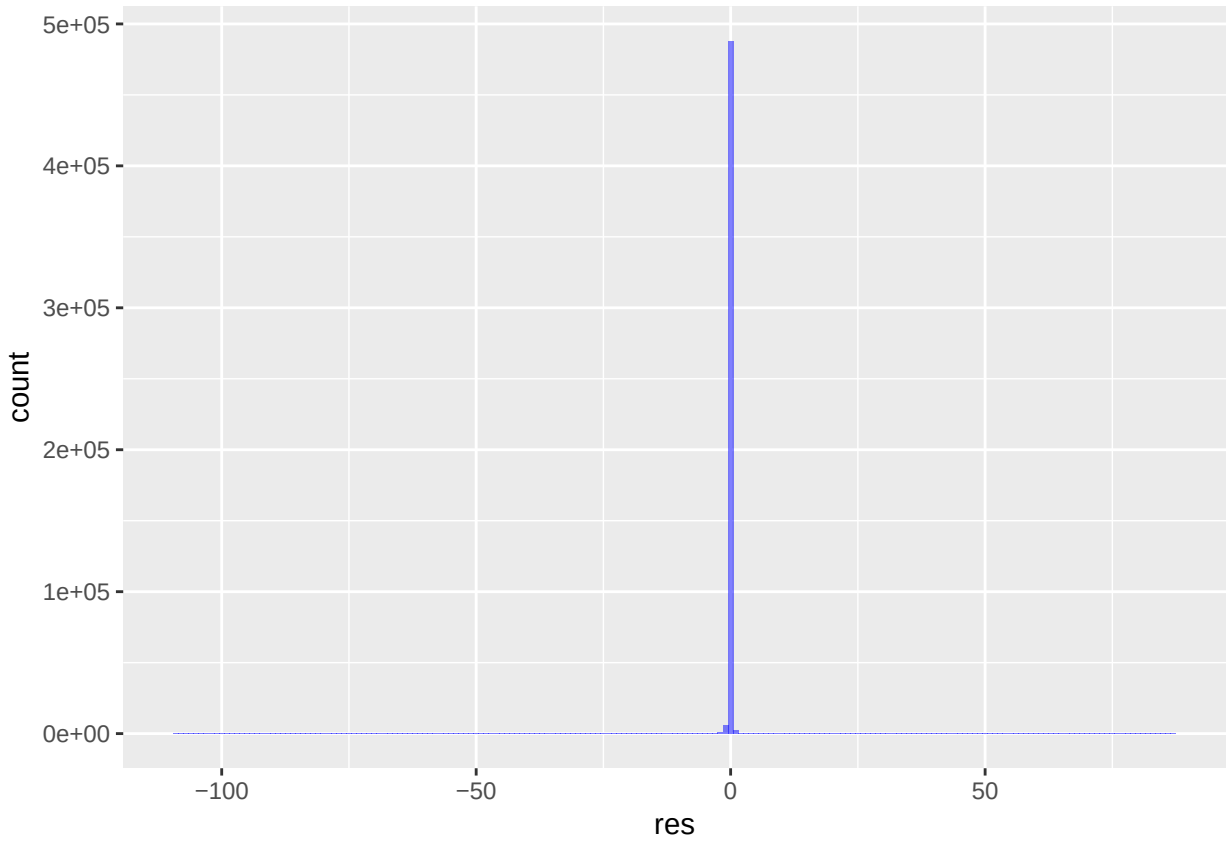
```
summary(modelo_lm_v3)
```

```
##
## Call:
## lm(formula = demanda_ajustada ~ dia + unidade_venda + valor_venda +
##     unidade_dev_next_week + valor_dev_next_week + id_rota + id_canal_venda,
##     data = dados_vendas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -108.672    0.000    0.006    0.021   86.573
##
## Coefficients:
##                           Estimate Std. Error   t value Pr(>|t|)
## (Intercept)           -1.144e-03  3.303e-03    -0.346  0.72914
## dia                   -6.652e-04  4.757e-04    -1.398  0.16197
## unidade_venda          9.950e-01  7.951e-05 12513.050  < 2e-16 ***
## valor_venda            5.860e-05  5.540e-06    10.578  < 2e-16 ***
## unidade_dev_next_week -5.280e-01  9.469e-04  -557.621  < 2e-16 ***
## valor_dev_next_week    1.798e-03  8.819e-05    20.391  < 2e-16 ***
## id_rota                5.836e-06  7.325e-07     7.967 1.63e-15 ***
## id_canal_venda         2.305e-03  7.502e-04     3.073  0.00212 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6771 on 499992 degrees of freedom
## Multiple R-squared:  0.9989, Adjusted R-squared:  0.9989
## F-statistic: 6.493e+07 on 7 and 499992 DF,  p-value: < 2.2e-16
```

Obtendo os resíduos do modelo 2 que foi criado, convertendo para um Dataframe e gerando um histograma destes residuos.

```
res <- residuals(modelo_lm_v2)
res <- as.data.frame(res)

ggplot(res, aes(res)) +
  geom_histogram(fill = 'blue',
                 alpha = 0.5,
                 binwidth = 1)
```



Tratando os valores outliers

Vou retirar do conjunto de dados valores com unidade de venda superiores a 12 e demanda superiores a 12 conforme avaliado anteriormente na analise dos quartis.

Apos retirada desses outliers ficamos com 443.750 observacoes no dataset

```
dados_vendas <- dados_vendas %>%
                  filter(unidade_venda <= 12)

dados_vendas <- dados_vendas %>%
  filter(demanda_ajustada <= 12)
```

Avaliando novamente as Medias de Tendência Central das variáveis.

Agora podemos observar que os dados de media e mediana sao proximos e o desvio padrão NÃO é alto.

```
summary(dados_vendas$demanda_ajustada)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   3.731   5.000  12.000
```

```
summary(dados_vendas$unidade_venda)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   3.785   5.000  12.000
```

```
summary(dados_vendas$unidade_dev_next_week)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   0.00000   0.00000   0.00000   0.09019   0.00000 150.00000
```

```
sd(dados_vendas$unidade_venda)
```

```
## [1] 2.80017
```

```
sd(dados_vendas$unidade_dev_next_week)
```

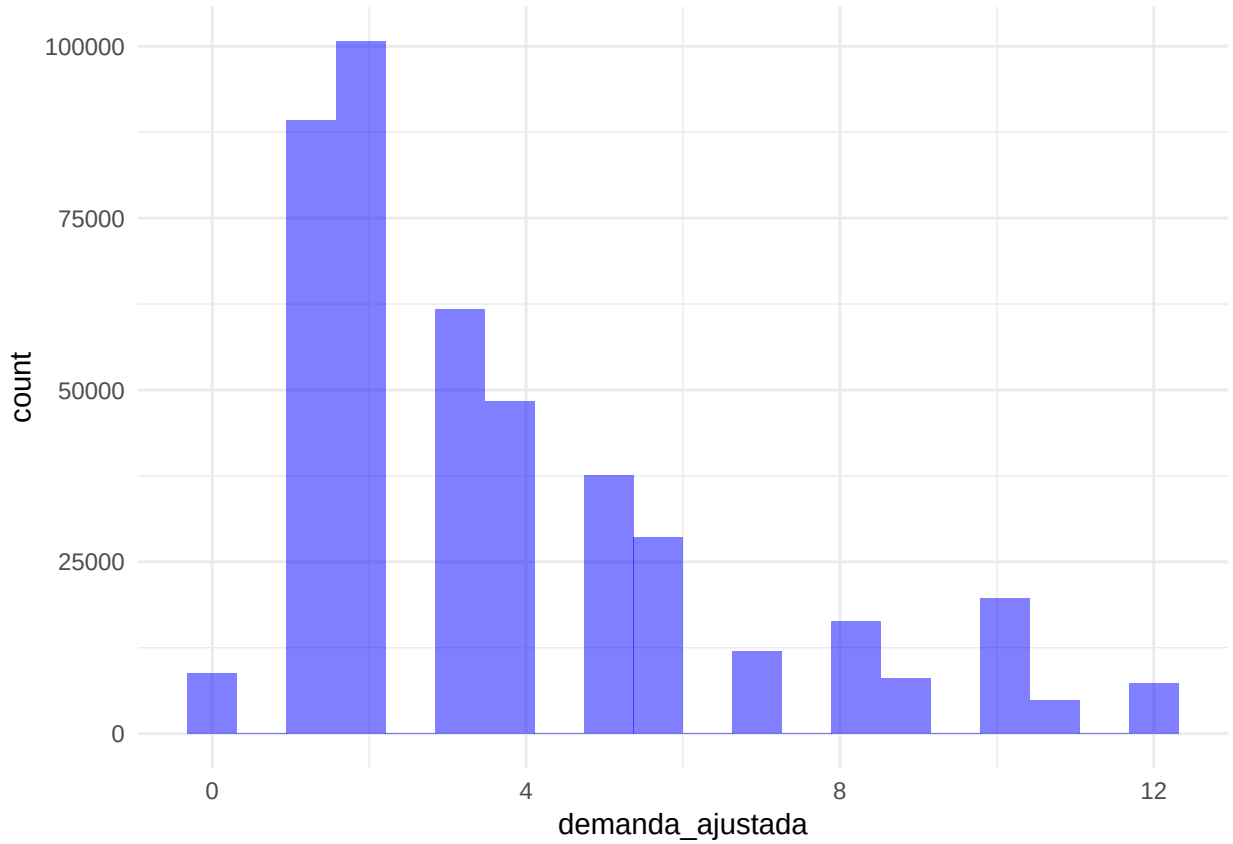
```
## [1] 1.054939
```

```
sd(dados_vendas$demanda_ajustada)
```

```
## [1] 2.816348
```

Criando um histograma para a variavel target

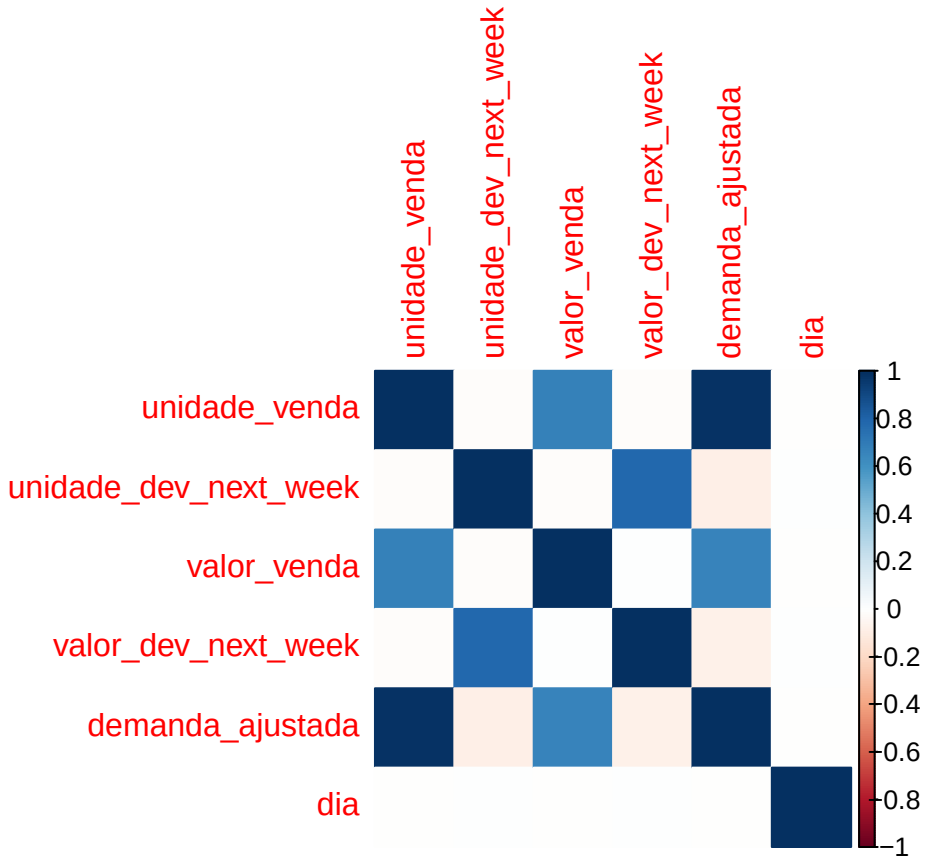
```
ggplot(dados_vendas, aes(x = demanda_ajustada)) +
  geom_histogram(bins = 20,
                 alpha = 0.5, fill = 'blue') +
  theme_minimal()
```



Analisando a correlacao das variaveis

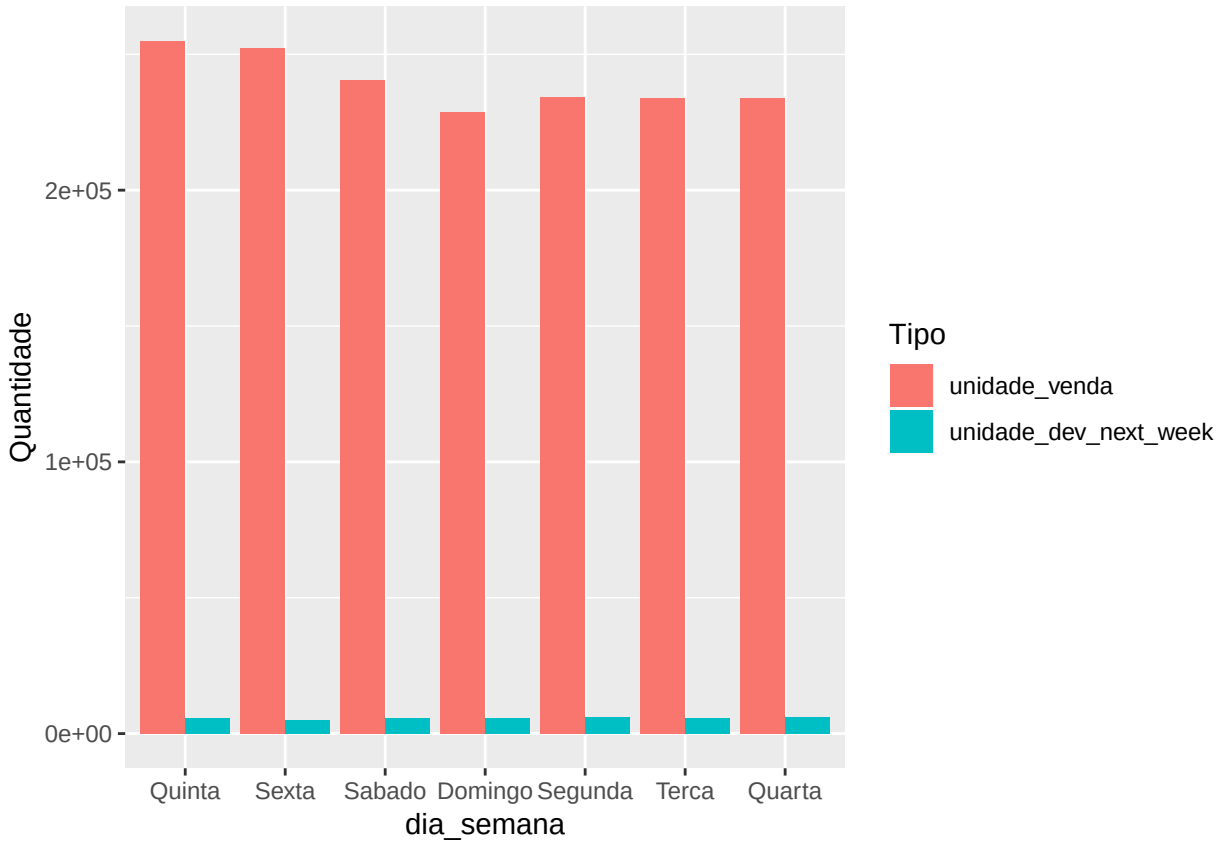
Definindo as colunas para a análise de correlação e criando um coorplot

```
cols <- c("unidade_venda", "unidade_dev_next_week", "valor_venda" ,"valor_dev_next_week","demanda_ajusta
correlacao <- cor(dados_vendas[,cols])

corrplot(correlacao, method = 'color')
```



Gerando alguns graficos para insights dos gestores

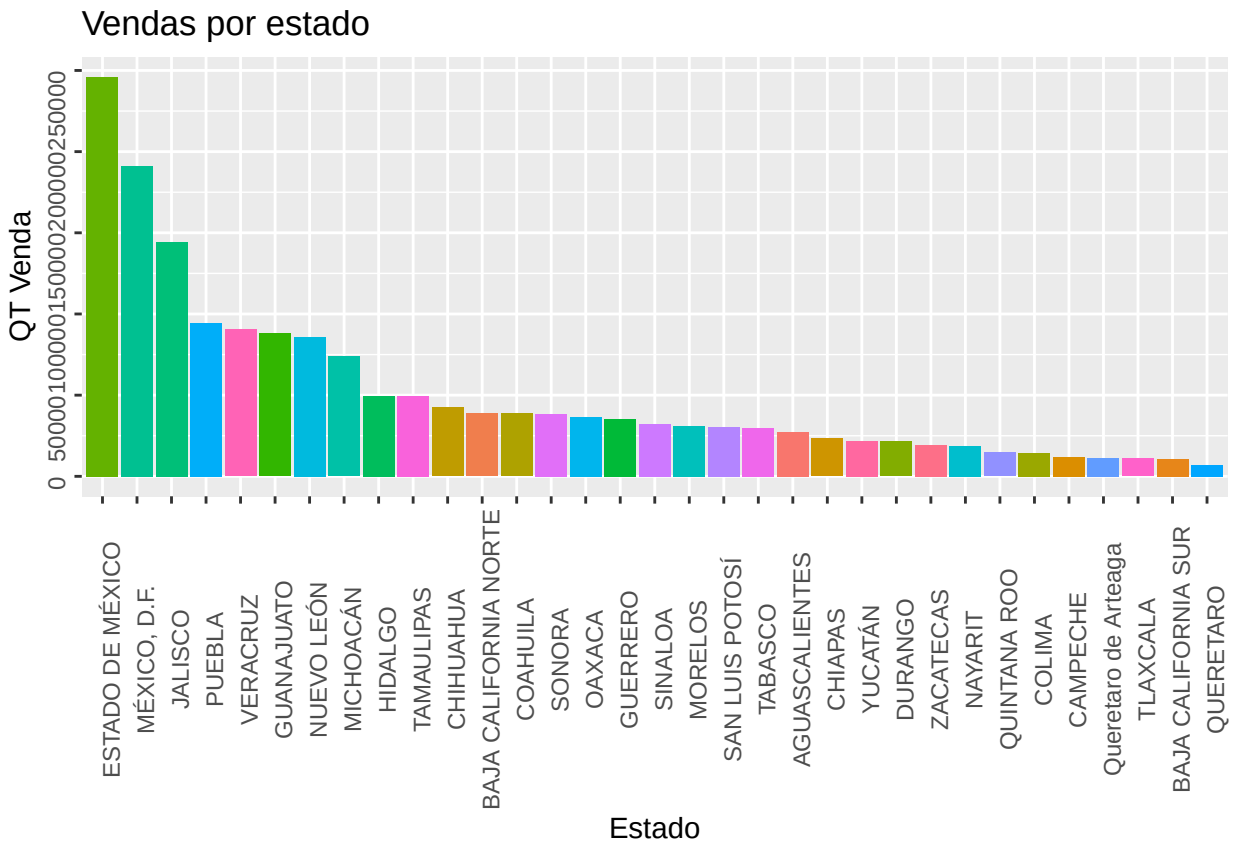
```
dados_vendas %>%
  group_by(dia_semana) %>%
  summarise(unidade_venda = sum(unidade_venda),
            unidade_dev_next_week = sum(unidade_dev_next_week))%>%
  melt(id = c("dia_semana")) %>%
  ggplot(aes(x = dia_semana, y = value, fill = variable)) +
  geom_bar(stat = "identity", position="dodge") +
  ylab("Quantidade") +
  labs(fill = "Tipo")
```



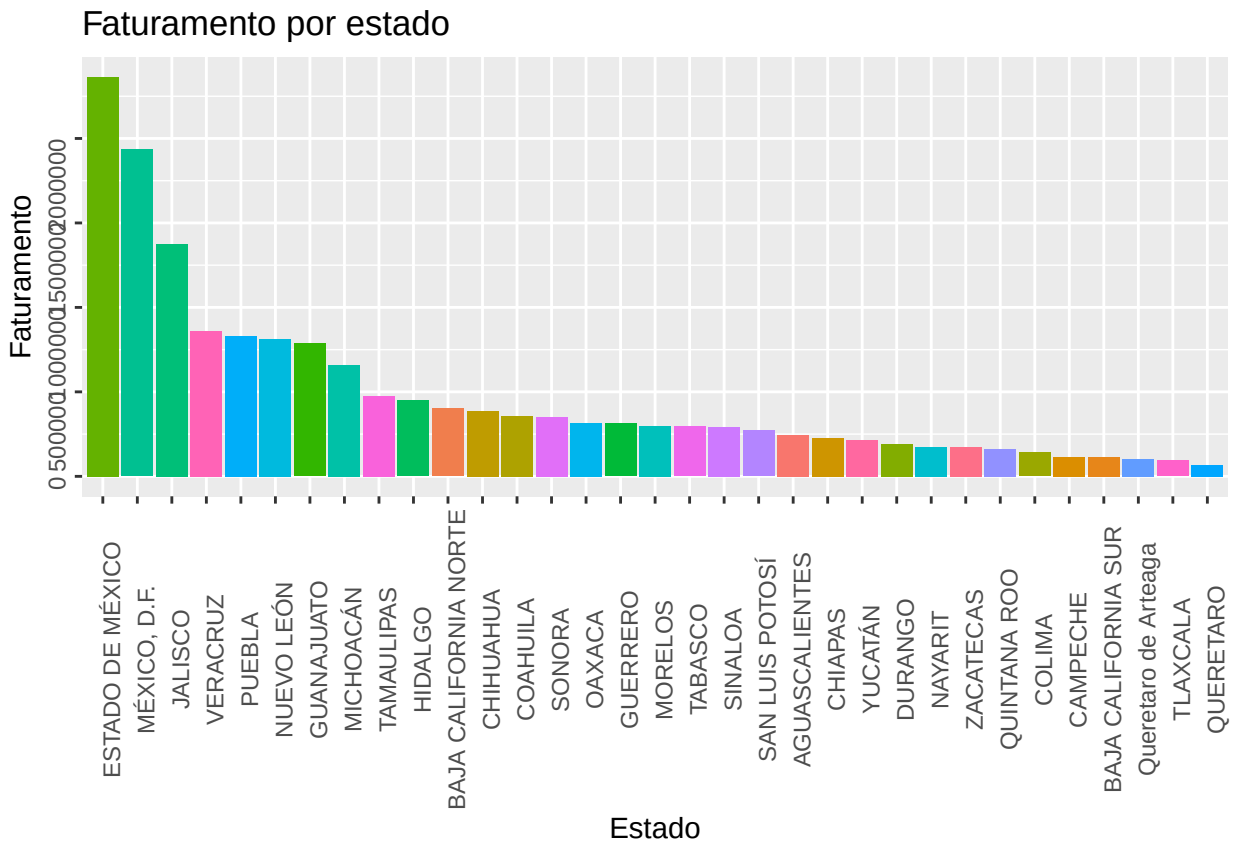

```
  ggtitle("Quantidade Vendida x Devolvida")
```

```
## $title
## [1] "Quantidade Vendida x Devolvida"
##
## attr(,"class")
## [1] "labels"
```

```
  dados_vendas %>%
  inner_join(dados_cidades, by = 'loja') %>%
  group_by(estado) %>%
  select(estado, loja, unidade_venda) %>%
  summarise(unidade_venda = sum(unidade_venda))%>%
  ggplot(aes(x = reorder(estado, -unidade_venda), y = unidade_venda, fill = estado)) +
  geom_bar(stat = "identity", position="dodge") +
  guides(fill = FALSE) +
  xlab("Estado") +
  ylab("QT Venda") +
  ggtitle("Vendas por estado") +
  theme(axis.text = element_text(angle = 90))
```




```
dados_vendas %>%
inner_join(dados_cidades, by = 'loja') %>%
group_by(estado) %>%
select(estado, loja, valor_venda) %>%
summarise(valor_venda = sum(valor_venda))%>%
ggplot(aes(x = reorder(estado, -valor_venda), y = valor_venda, fill = estado)) +
geom_bar(stat = "identity", position="dodge") +
guides(fill = FALSE) +
xlab("Estado") +
ylab("Faturamento") +
ggtitle("Faturamento por estado") +
theme(axis.text = element_text(angle = 90))
```



4 - Criando e avaliando os modelos preditivos

Criando dados de treino e de teste (70% e 30% respectivamente)

```
amostra <- sample.split(dados_vendas$demanda_ajustada, SplitRatio = 0.70)
treino = subset(dados_vendas, amostra == TRUE)
teste = subset(dados_vendas, amostra == FALSE)
```

Modelo LM Em torno de 98% de acuracia

```
modelo_lm <- lm(demanda_ajustada ~ dia + unidade_venda + valor_venda + unidade_dev_next_week + valor_dev

summary(modelo_lm)
```

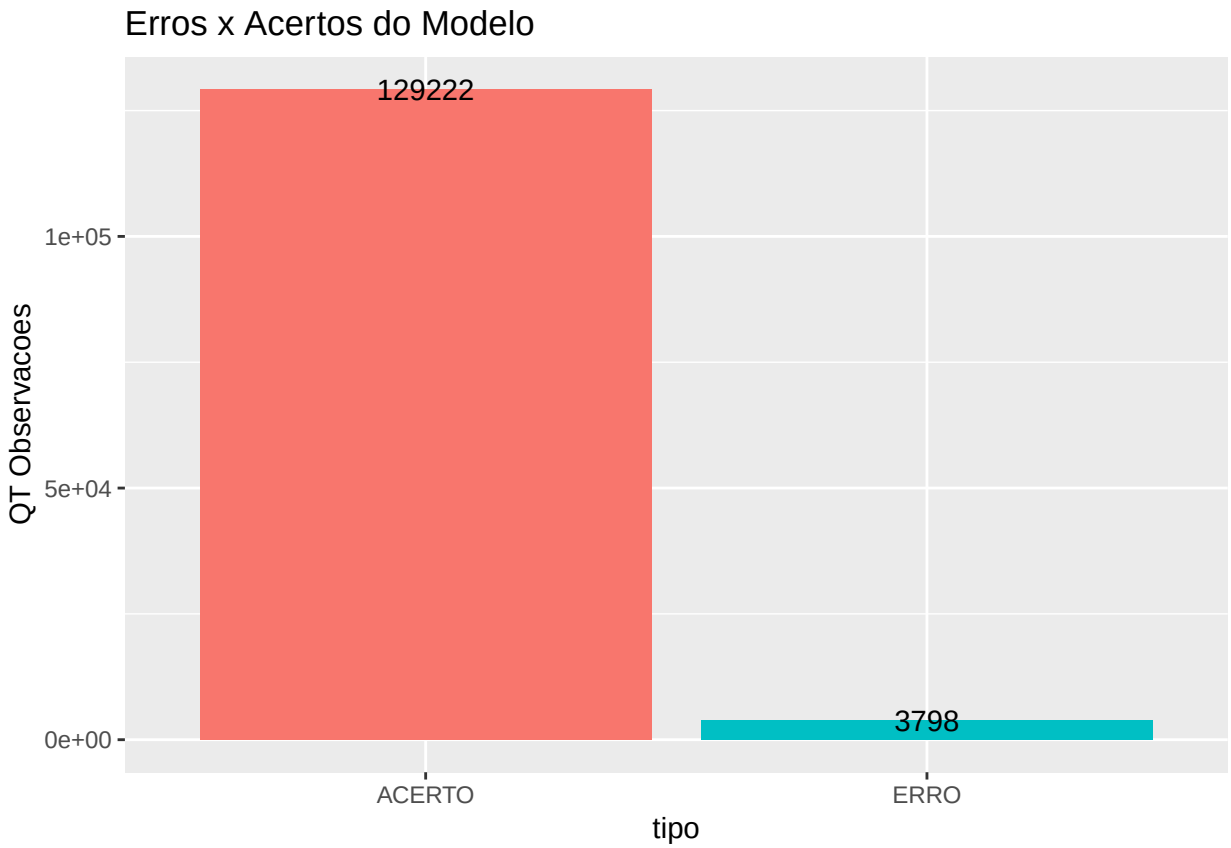
```
##
## Call:
## lm(formula = demanda_ajustada ~ dia + unidade_venda + valor_venda +
##     unidade_dev_next_week + valor_dev_next_week, data = dados_vendas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6424  0.0227  0.0302  0.0429 29.1407
##
## Coefficients:
##                         Estimate Std. Error  t value Pr(>|t|)
```

```
## (Intercept)           -3.785e-03  1.758e-03   -2.154   0.0313 *
## dia                   -1.524e-03  2.559e-04   -5.955  2.6e-09 ***
## unidade_venda          9.959e-01  2.493e-04 3995.420  < 2e-16 ***
## valor_venda           -1.933e-04  2.092e-05   -9.241  < 2e-16 ***
## unidade_dev_next_week -2.348e-01  7.914e-04 -296.727  < 2e-16 ***
## valor_dev_next_week    2.793e-03  6.261e-05   44.609  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3436 on 443395 degrees of freedom
## Multiple R-squared:  0.9851, Adjusted R-squared:  0.9851
## F-statistic: 5.871e+06 on 5 and 443395 DF,  p-value: < 2.2e-16
```

```
previsao_lm <- predict(modelo_lm, teste)
```

Visualizando os valores previstos e observados, tratando valores negativos e gerando um gráfico parademonstrar a quantidade de erros e acertos do modelo.

```
resultados <- cbind(teste$demanda_ajustada,round(previsao_lm))
colnames(resultados) <- c('Real','Previsto')
resultados <- as.data.frame(resultados)

trata_zero <- function(x){
  if  (x < 0){
    return(0)
  }else{
    return(x)
  }
}

resultados$Previsto <- sapply(resultados$Previsto, trata_zero)

func <- function(x , y){
  if(x == y){
    "ACERTO"
  }else{
    "ERRO"
  }}

resultados$tipo = mapply(func,resultados$Real,resultados$Previsto)

resultados %>%
  group_by(tipo) %>%
  summarise(total = n()) %>%
  ggplot(aes(x = tipo, y = total, fill = tipo)) +
  geom_bar(stat = "identity", position="dodge") +
  geom_text(aes(label = total)) +
  guides(fill = FALSE) +
  ylab("QT Observacoes") +
  ggtitle("Erros x Acertos do Modelo")
```



Modelo de Rede Neural com 100% de acerto

Como esse modelo é muito pesado, iremos utilizar somente 30 registros

```
dados_vendas <- dados_vendas %>%
  sample_n(size = 30000)
```

Obtendo os valores minimos e maximos do dataset e normalizando as variaveis preditoras

```
maxs <- apply(dados_vendas[, cols], 2, max)
mins <- apply(dados_vendas[, cols], 2, min)

dados_normalizados <- dados_vendas

dados_normalizados[, cols] <- as.data.frame(scale(dados_vendas[, cols], center = mins, scale = maxs - mi
```

Gerando dados de treino e teste

```
amostra <- sample.split(dados_normalizados$demanda_ajustada, SplitRatio = 0.70)
treino = subset(dados_normalizados, amostra == TRUE)
teste = subset(dados_normalizados, amostra == FALSE)
```

Criando uma formula e o modelo da rede neural

```
formula = "demanda_ajustada ~ dia + unidade_venda + valor_venda + unidade_dev_next_week + valor_dev_nex

modelo_rede_neural <- neuralnet(formula, data = treino, hidden = c(5,3), linear.output = TRUE)
```

Fazendo as previsoes com os dados de teste

```
previsoes_rn <- compute(modelo_rede_neural, teste)
```

Convertendo os dados normalizados para numeros normais

```
previsoes_rn <- previsoes_rn$net.result * (max(dados_vendas$demanda_ajustada) - min(dados_vendas$demanda
dados_teste_convertidos <- (teste$demanda_ajustada) * (max(dados_vendas$demanda_ajustada) - min(dados_v
```

Visualizando os dados previstos e reais e tratando valores zerados

```
resultados <- cbind(dados_teste_convertidos,round(previsoes_rn))
colnames(resultados) <- c('Real','Previsto')
resultados <- as.data.frame(resultados)

resultados$Previsto <- sapply(resultados$Previsto, trata_zero)
```

Calculando a Taxa de Acuracia e Taxa de Erro

```
MSE.nn <- sum((dados_teste_convertidos - previsoes_rn)^2)/nrow(teste)
MSE.nn
```

```
## [1] 0.0001316882
```

```
error.df <- data.frame(dados_teste_convertidos, previsoes_rn)
head(error.df)
```

```
##    dados_teste_convertidos previsoes_rn
## 7                        1   0.99697274
## 16                       5   5.00785442
## 19                       1   0.99846960
## 23                       3   2.99345176
## 24                       6   6.00221997
## 26                       0   0.04292651
```

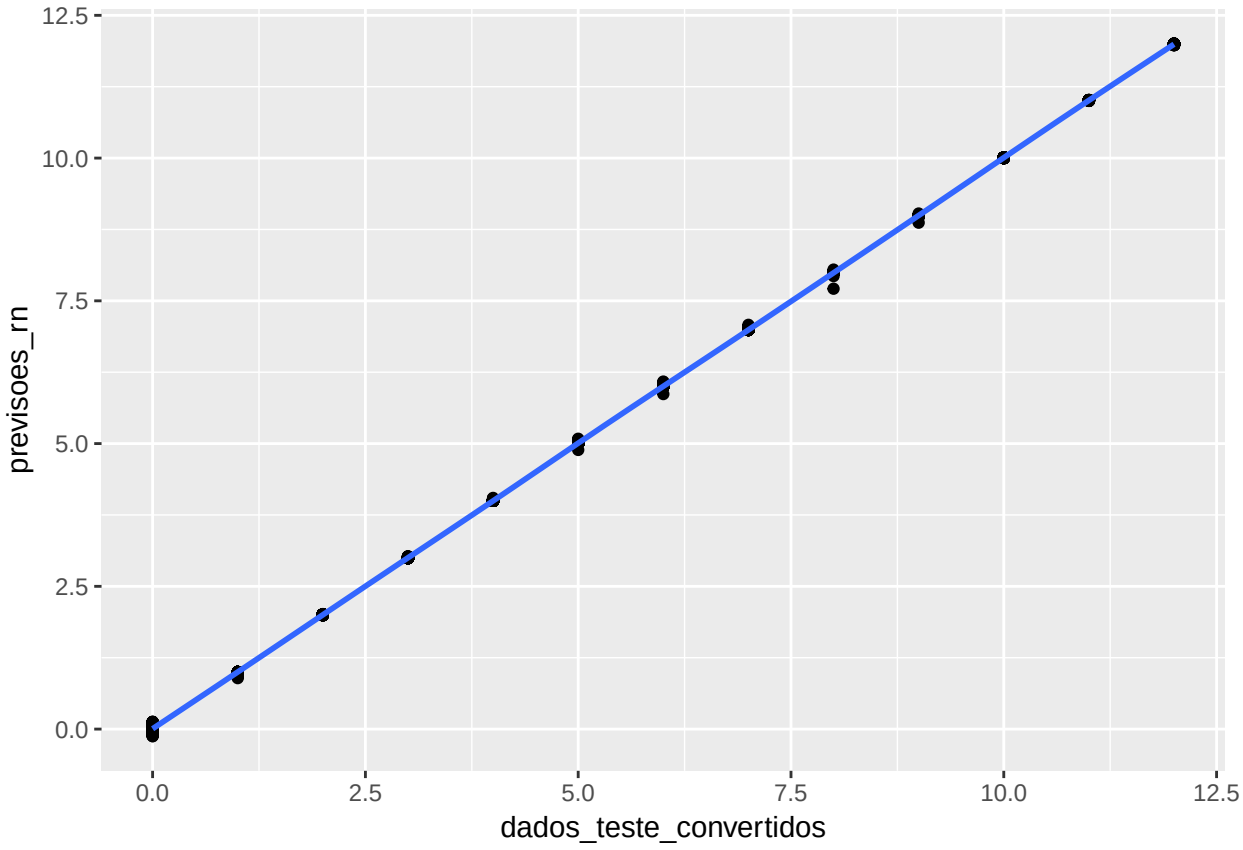
```
SSE = sum((resultados$Previsto - resultados$Real)^2)
SST = sum((mean(dados_vendas$demanda_ajustada) - resultados$Real)^2)
```

```
R2 = 1 - (SSE/SST)
R2
```

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

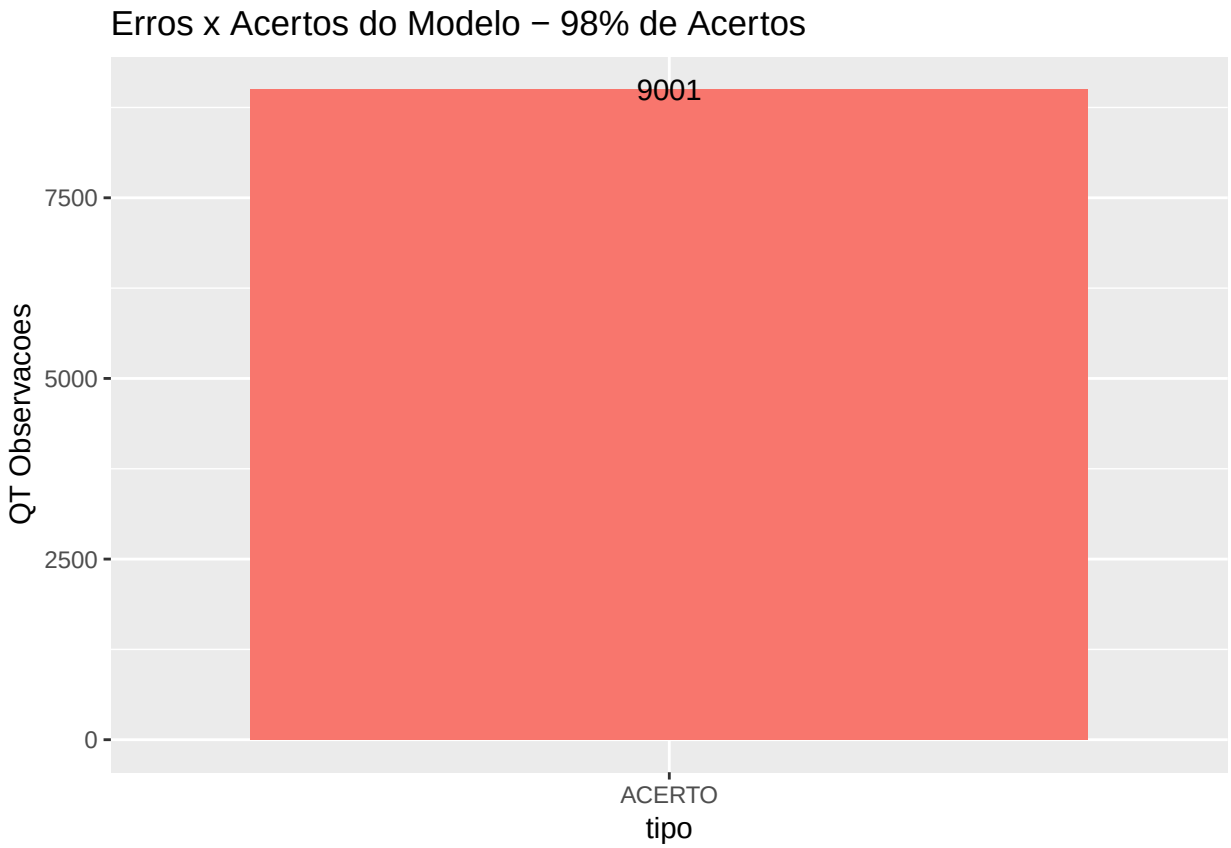
```
library(ggplot2)
ggplot(error.df, aes(x = dados_teste_convertidos,y = previsoes_rn)) +
  geom_point() + stat_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Exibindo Gráfico com Erros x Acertos

```
resultados$tipo = mapply(func,resultados$Real,resultados$Previsto)

resultados %>%
  group_by(tipo) %>%
  summarise(total = n()) %>%
  ggplot(aes(x = tipo, y = total, fill = tipo)) +
  geom_bar(stat = "identity", position="dodge") +
  geom_text(aes(label = total)) +
  guides(fill = FALSE) +
  ylab("QT Observacoes") +
  ggtitle("Erros x Acertos do Modelo - 98% de Acertos")
```


Erros x Acertos do Modelo − 98% de Acertos
9001
7500
5000
2500
0
QT Observacoes
ACERTO
tipo