

Mineração de texto aplicada à Lei de Acesso à informação - LAI

true

Rio de Janeiro, 30 de outubro de 2019

Packages for this routine

BASE DE DADOS E ANÁLISE EXPLORATÓRIA

Importação dos dados

Caminho do projeto

```
PATH = "../proj_eSIC_v10/textmining_pt/DATA/"
```

Importação ee estrutura dos dados

Tabela1: Pedidos e-SIC

- Pedidos e-SIC

Estrutura dos dados

```
glimpse(Pedidos_eSIC)
```

```
## Observations: 625
## Variables: 9
## $ Protocolo                <chr> "16853006234201716"...
## $ `Órgão Superior`         <chr> "EPE - Empresa de P...
## $ `Data de Abertura`       <dtm> 2017-08-19 20:26:4...
## $ `Prazo de Atendimento`   <dtm> 2017-09-11 23:59:5...
## $ Situação                 <chr> "Respondido", "Resp...
## $ `Descrição do Pedido`     <chr> "A Empresa de Pesqu...
## $ `Descrição da Forma de Resposta do Pedido` <chr> "Pelo sistema (com ...
## $ `Resumo da Solicitação`  <chr> "Empresa de Pesquis...
## $ `Data da Resposta`       <dtm> 2017-08-30 21:19:1...
```

Tabela2: Respostas Diretorias da EPE

- Respostas e-SIC (DIRETORIAS EPE)

Estrutura dos dados

```
glimpse(Respostas_EPE)
```

```
## Observations: 705
## Variables: 3
## $ ProtocoloPedido          <chr> "99938000045201565", "9993...
## $ DataRegistro             <dtm> 2015-07-24, 2015-07-28, 2...
## $ DiretoriaEPE_ResponsavelPelaDemanda <chr> "DGC", "DEA", "DEA", "DEE"...
```

Tabela3: Stopwords

- Stopwords

```
FILE2 = "DATA/stopwords_PT_FINAL.csv"
stopwords_pt = read.csv(paste0(PATH,FILE2), sep = ';', header = F, encoding = "UTF-8")
stopwords_pt = stopwords_pt[,-2];
cat(paste0("O nosso vetor de stopwords contém ",length(stopwords_pt), " palavras únicas"))

## O nosso vetor de stopwords contém 734 palavras únicas
## dim(stopwords_pt); class(stopwords_pt)
stopwords_pt = as.character(stopwords_pt)
stopwords_pt[1:14]

## [1] "a"          "à"          "acerca"     "acesso"     "adeus"     "agora"
## [7] "agradeço"   "agradeco"   "aí"         "ai"         "ainda"     "alem"
## [13] "além"       "algmas"
```

Tabelas4,5,6: Dicionários de variáveis e-SIC

- Dicionário > BASE DE DADOS - REAL PRO TEXTO DO TCC

Dicionário de variáveis - PEDIDOS

```
dicionario = "DATA/Dicionario-Dados-Exportacao.txt"
dic_pedidos = read.delim(paste0(PATH,dicionario), sep = "-", skip = 3, header = FALSE, nrows = 21) %>%
  select(-V1)
colnames(dic_pedidos) = c("Nome das variáveis", "Tipo e descrição da variável")
#dimnames(dic_pedidos); View(dic_pedidos)
```

Dicionário de variáveis - RECURSOS

```
dic_recursos = read.delim(paste0(PATH,dicionario), sep = "-", skip = 30, header = FALSE, nrows = 17) %>%
  select(-V1)
colnames(dic_recursos) = c("Nome das variáveis", "Tipo e descrição da variável")
#dimnames(dic_recursos); View(dic_recursos)
```

Dicionário de variáveis - SOLICITANTES

```
dicionario = "DATA/Dicionario-Dados-Exportacao.txt"
dic_solicitantes = read.delim(file = paste0(PATH,dicionario), sep = "-", skip = 53, header = FALSE, nrows = 17) %>%
  select(-V1)
colnames(dic_solicitantes) = c("Nome das variáveis", "Tipo e descrição da variável")
#dimnames(dic_solicitantes); View(dic_solicitantes)
```

Transformação e pré-processamento dos dados

Filtra, Transforma e Unifica bases

Filtro1: tabela consulta de pedidos

Filtrando apenas as variáveis de interesse do estudo na tabela de consulta de pedidos

```
LAI = Pedidos_eSIC
LAI = LAI %>% select(Protocolo, `Data de Abertura`, `Prazo de Atendimento`, `Descrição do Pedido`, `Resposta`)
```

Transformação1: renomeando colunas

Reescrevendo o nome das variáveis de ambas tabelas

```
colnames(LAI) = c("Protocolo", "DATA_REGISTRO", "DATA_PRAZOATEND", "DESCRI_PEDIDO",  
                 "RESUMO_PEDIDO", "DATA_RESPOSTA")  
LAI1 = Respostas_EPE  
colnames(LAI1) = c("Protocolo", "DATA_REGISTRO", "DIRETORIAS")  
# glimpse(LAI1)
```

Transformação2: transforma as.factor() variável DIRETORIAS

character em factor

Transformação3: cria a variável PEDIDO = RESUMO + DESCRIÇÃO

```
LAI$PEDIDO = paste(LAI$RESUMO_PEDIDO, LAI$DESCRI_PEDIDO)
```

Análise1: Quantitativo de pedidos por diretoria

Transformação3: substitui NA por OUTROS (coluna DIRETORIAS)

Primeiro conta o número de pedidos por diretoria (observações por categorias)

```
LAI1 %>%  
  group_by(DIRETORIAS) %>% count(sort = TRUE) %>%  
  kable("latex", caption = "Quantitativo de solicitações por Diretoria/EPE via e-SIC - substituição NA",  
        booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%  
  kable_styling(latex_options = c("striped", "hold_position"))
```

```
## Warning: Factor `DIRETORIAS` contains implicit NA, consider using  
## `forcats::fct_explicit_na`
```

```
## Warning: Factor `DIRETORIAS` contains implicit NA, consider using  
## `forcats::fct_explicit_na`
```

Table 1: Quantitativo de solicitações por Diretoria/EPE via e-SIC - substituição NA em OUTROS

DIRETORIAS	n
DEE	244
DEA	240
DGC	121
OUTROS	66
DPG	33
NA	1

Existe um valor NA, vamos substituí-lo como OUTROS

```
LAI1 = LAI1 %>%  
  replace_na(list(DIRETORIAS = "OUTROS"))
```

Tabela1: Quantitativo de pedidos por diretoria - sem reclassificação

- Tabela 01 número de solicitações/pedidos de informação (sem NA)

```
pedidos_diretoria = LAI1 %>%
  count(DIRETORIAS, sort = TRUE, name = "total_pedidos")
pedidos_diretoria %>%
  kable("latex", caption = "Quantitativo de solicitações por Diretoria/EPE via e-SIC - sem reclassificação",
        booktabs = T) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 2: Quantitativo de solicitações por Diretoria/EPE via e-SIC - sem reclassificação

DIRETORIAS	total_pedidos
DEE	244
DEA	240
DGC	121
OUTROS	67
DPG	33

Verificamos a existência de 4 diretorias, sendo elas: *DEA*, *DEE*, *DGC*, *DPG* e *OUTROS*. Essa última é devido a existência de informações solicitadas que não são de competência de nenhuma das cinco diretorias, daí a necessidade de uma última categoria *OUTROS* para atender essas demandas.

Fica nítida o desbalanceamento do número de pedidos por categoria. Enquanto as diretorias *DEE* e *DEA* possuem, respectivamente, 244 e 240 pedidos verifica-se uma diferença grande do número de pedido das diretorias *DGC* e *DPG* e também da categoria *OUTRAS*, onde se forem somadas possuem um total de 221 pedidos conjuntamente.

A seguir, um passo importante de reclassificação será executado devido ao número pequeno de solicitações para as diretorias *DGC* e *DPG*. Apenas uma solicitação existente no nosso banco de dados para essa diretoria. Iremos, portanto, unificar essa demanda à categoria *OUTROS*. A seguir, verificamos nas tabela 01 e 02 a distribuição de pedidos por diretoria antes e após reclassificação das mesmas.

Tabela1: Quantitativo de pedidos por diretoria - sem reclassificação

Vamos criar uma nova variável: *DIRETORIA* que é basicamente uma reclassificação da variável *DIRETORIAS*. Vamos, primeiro, armazenar um vetor com o nome das categorias de *DIRETORIAS* originais.

```
(diretorias = levels(as.factor(LAI1$DIRETORIAS)))
```

```
## [1] "DEA" "DEE" "DGC" "DPG" "OUTROS"
```

Transformação4 - Reclassificação das Diretorias

Respostas e-SIC - Reclassificação Diretorias

```
# LAI1$DIRETORIAS = as.character(LAI1$DIRETORIAS) # glimpse(LAI1)
LAI1 = LAI1 %>%
  mutate(DIRETORIA = ifelse(DIRETORIAS == "DGC", "OUTROS",
                            ifelse(DIRETORIAS == "DPG", "OUTROS", DIRETORIAS)))
(diretorias1 = levels(as.factor(LAI1$DIRETORIA)))
```

```
## [1] "1" "2" "5" "OUTROS"
```

Tabela2: Quantitativo de pedidos por diretoria - após reclassificação

- Tabela 02 número de solicitações/pedidos de informação - após reclassificação

```
pedidos_diretoria1 = LAI1 %>%
  count(DIRETORIA, sort = TRUE, name = "total_pedidos")
pedidos_diretoria1 %>%
  kable("latex", caption = "Quantitativo de solicitações por Diretoria/EPE via e-SIC - após reclassificação",
        booktabs = T) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 3: Quantitativo de solicitações por Diretoria/EPE via e-SIC - após reclassificação

DIRETORIA	total_pedidos
2	244
1	240
OUTROS	154
5	67

Temos, finalmente um maior balanceamento nas categorias da nossa variável resposta com 244, 240 e 221 pedidos que foram destinados à *DEE*, *DEA* e *OUTROS*, respectivamente. Onde *OUTROS* é a categoria formada com a união dos pedidos das diretorias *DGC*, *DPG* e *OUTROS*.

A reclassificação foi, também, uma decisão suportada por análises préveias do presente estudo. Foi avaliada a viabilidade de aplicar o estudo com as categorias originais, entretanto na fase de modelagem preditiva o desempenho do modelo do Random Forest foi muito inferior comparado ao modelo após reclassificação. Um motivo plausível para a melhoria de performance pode ser por conta do maior balanceamento entre as categorias da variável resposta **Diretoria**, em questão.

- Unificando as Bases

É necessário, agora, unificar as bases de dados pertinentes às solicitações e respostas.

Join1: União das bases em questão

```
LAI1 = LAI1 %>% select(-DATA_REGISTRO); #dim(LAI1)
DB = left_join(x = LAI, y = LAI1, by = "Protocolo") %>%
  drop_na()
#View(head(DB))
```

```
glimpse(DB)
```

```
## Observations: 624
## Variables: 9
## $ Protocolo      <chr> "16853006234201716", "18600000523201890", "234...
## $ DATA_REGISTRO <dtm> 2017-08-19 20:26:47, 2018-03-07 18:29:43, 201...
## $ DATA_PRAZOATEND <dtm> 2017-09-11 23:59:59, 2018-03-28 23:59:59, 201...
## $ DESCR_PEDIDO   <chr> "A Empresa de Pesquisa Energética (vinculada a...
## $ RESUMO_PEDIDO   <chr> "Empresa de Pesquisa Energética", "Demanda ou ...
## $ DATA_RESPOSTA <dtm> 2017-08-30 21:19:17, 2018-03-08 16:50:43, 201...
## $ PEDIDO          <chr> "Empresa de Pesquisa Energética A Empresa de P...
## $ DIRETORIAS       <fct> DGC, OUTROS, DEA, DEE, DEA, DPG, DEA, DEE, OUT...
## $ DIRETORIA        <chr> "OUTROS", "5", "1", "2", "1", "OUTROS", "1", "...
```

```
cat(paste0("Existem ", dim(DB)[1], " observações/pedidos na base de dados."))
```

```
## Existem 624 observações/pedidos na base de dados.
```

```
cat(paste0("Com registros de pedidos datados entre ", format(min(DB$DATA_REGISTRO), '%d de %B de %Y'), "
```

Com registros de pedidos datados entre 07 de Julho de 2015 a 25 de Março de 2019.
Ver Anexo 01 c/ amostra dos dados da tabela que será utilizada para manipulação daqui pra frente.

- Evolução de pedidos:

Data de registro do pedido

```
db_evolPedido = DB %>% select(Protocolo, DIRETORIAS, DIRETORIA, DATA_REGISTRO) %>%
  mutate(DIASEMANA_REGISTRO = weekdays(DB$DATA_REGISTRO),
    HORA_REGISTRO = hour(DB$DATA_REGISTRO),
    MES_REGISTRO = base::months.Date(DB$DATA_REGISTRO),
    ANO_REGISTRO = year(DB$DATA_REGISTRO))

ano_evolution = db_evolPedido %>%
  group_by(ANO_REGISTRO) %>% count()

hc2_1 <- highchart() %>%
  hc_add_series(data = ano_evolution$n,
    type = "column",
    name = "Evolução",
    showInLegend = TRUE,
    tooltip = list(valueDecimals = 2, valuePrefix = "",
      valueSuffix = " pedidos registrados",
      color = "#5F83EE", fillOpacity = 0.1) %>%
  hc_yAxis(title = list(text = "Quantitativo de pedidos registrados"),
    allowDecimals = FALSE, max = max(ano_evolution$n),
    labels = list(format = "{value}")) %>%
  hc_xAxis(title = list(text = "Ano"),
    categories = ano_evolution$ANO_REGISTRO,
    tickmarkPlacement = "on",
    opposite = FALSE) %>%
  hc_title(text = "Evolução de pedidos registrados via LAI (EPE)",
    style = list(fontWeight = "bold")) %>%
  hc_subtitle(text = paste("")) %>%
  hc_tooltip(valueDecimals = 2,
    pointFormat = "Importância: {point.y}") %>%
  #pointFormat = "Variável: {point.x} <br> Importância: {point.y}")
  hc_credits(enabled = TRUE,
    text = "Fonte: CGU, e-SIC. Elaboração: Leal, Alize; Pimenta, Ewerson.",
    style = list(fontSize = "10px")) %>%
  hc_exporting(enabled = TRUE, filename = "F6_1-importance-Pimenta")
#hc <- hc %>%
# hc_add_theme(hc_theme_darkunica())
hc2_1

ano_evolution_DIR = db_evolPedido %>%
  group_by(DIRETORIAS, ANO_REGISTRO) %>% count()

DEE = ano_evolution_DIR %>% filter(DIRETORIAS == "DEE") %>% arrange(desc(ANO_REGISTRO), .by_group = TRUE)
DEA = ano_evolution_DIR %>% filter(DIRETORIAS == "DEA") %>% arrange(desc(ANO_REGISTRO), .by_group = TRUE)
DGC = ano_evolution_DIR %>% filter(DIRETORIAS == "DGC") %>% arrange(desc(ANO_REGISTRO), .by_group = TRUE)
DPG = ano_evolution_DIR %>% filter(DIRETORIAS == "DPG") %>% arrange(desc(ANO_REGISTRO), .by_group = TRUE)
OUTROS = ano_evolution_DIR %>% filter(DIRETORIAS == "OUTROS") %>% arrange(desc(ANO_REGISTRO), .by_group = TRUE)
```

```

hc2_2 <- highchart() %>%
  hc_add_series(data = DEE$n,
    type = "line",
    name = "DEE",
    showInLegend = TRUE,
    tooltip = list(valueDecimals = 0, valuePrefix = "",
      valueSuffix = " pedidos registrados",
      color = "#5F83EE", fillOpacity = 0.1) %>%
  hc_add_series(data = DEA$n,
    type = "line",
    name = "DEA",
    showInLegend = TRUE,
    tooltip = list(valueDecimals = 0, valuePrefix = "",
      valueSuffix = " pedidos registrados",
      color = "skyblue", fillOpacity = 0.1) %>%
  hc_add_series(data = DGC$n,
    type = "line",
    name = "DGC",
    showInLegend = TRUE,
    tooltip = list(valueDecimals = 0, valuePrefix = "",
      valueSuffix = " pedidos registrados",
      color = "green", fillOpacity = 0.1) %>%
  hc_add_series(data = DPG$n,
    type = "line",
    name = "DPG",
    showInLegend = TRUE,
    tooltip = list(valueDecimals = 0, valuePrefix = "",
      valueSuffix = " pedidos registrados",
      color = "black", fillOpacity = 0.1) %>%
  hc_add_series(data = OUTROS$n,
    type = "area",
    name = "OUTROS",
    showInLegend = TRUE,
    tooltip = list(valueDecimals = 0, valuePrefix = "",
      valueSuffix = " pedidos registrados",
      color = "pink", fillOpacity = 0.5) %>%

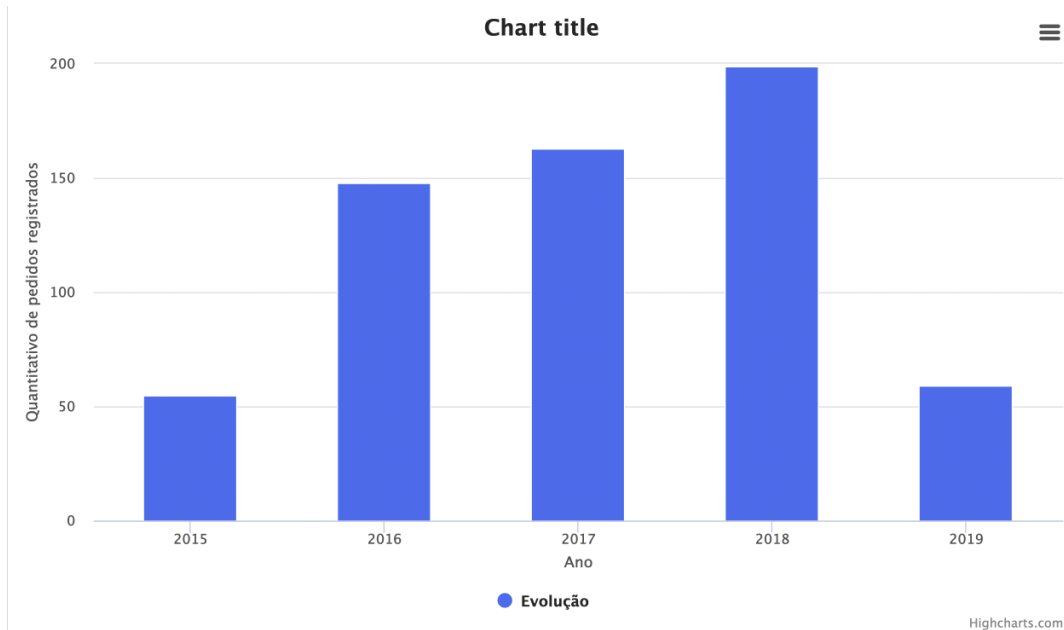
  hc_yAxis(title = list(text = "Quantitativo de pedidos registrados"),
    allowDecimals = FALSE, max = max(DEE$n, DEA$n, DGC$n, DPG$n, OUTROS$n),
    labels = list(format = "{value}"), #minorTickInterval = "auto",
    #minorGridLineDashStyle = "LongDashDotDot",
    showFirstLabel = TRUE,
    showLastLabel = TRUE) %>%
  hc_xAxis(title = list(text = "Ano"),
    categories = ano_evolution$ANO_REGISTRO,
    tickmarkPlacement = "on",
    opposite = FALSE) %>%
  hc_title(#text = "Evolução de pedidos registrados via LAI (EPE)",
    style = list(fontWeight = "bold")) %>%
  hc_subtitle(text = paste("")) %>%
  hc_tooltip(valueDecimals = 2,
    pointFormat = "Número de {point.y}") %>%
    #pointFormat = "Variável: {point.x} <br> Importância: {point.y}")

```

```

hc_credits(enabled = TRUE,
            #text = "Fonte: CGU, e-SIC. Elaboração: Leal, Alize; Pimenta, Ewerson.",
            style = list(fontSize = "10px")) %>%
hc_exporting(enabled = TRUE, filename = "F6_1-importance-Pimenta")
#hc <- hc %>%
# hc_add_theme(hc_theme_darkunica())
hc2_2

```



```

summary(DB$DATA_REGISTRO)
class()
inic = as.Date(min(DB$DATA_REGISTRO), format = "%m/%d/%y %H:%M:%S", tz = "UTC")
fim = max(DB$DATA_REGISTRO), date_format())
cat(paste0("Período de pedidos registrados que serão utilizados nessa análise ", inic, " até ", fim))

time_index_h <- seq(from = as.POSIXct(inic),
                    to = as.POSIXct(fim), by = "hour")
time_index_w <- weekdays(time_index_h)
# or
#time_index_w <- lubridate::wday(time_index_h)
library(lubridate)
date<-ymd_hms("2016-06-06 09:45:12")
wday(date)

```

Mineração de texto

Palavras por pedido

Análise2: distribuição de frequência de palavras por diretoria e algumas estatísticas descritivas

Ferramentas

Iniciamos as manipulações utilizando recursos da função `unnest_tokens()` do pacote `library(tidytext)`

que nos permite trabalhar com textos em um formato `tidy`, ou seja que coloca uma palavra por linha em uma única coluna, formando, assim, *termos/palavras* por linha. Utilizamos, também, ainda os recursos do pacote `library(dplyr)` para, posteriormente, agrupar esses termos por diretoria e calcular a frequência dos *termos*.

Verificamos que as 10 palavras mais frequentes em todos os pedidos realizados são palavras sem acréscimo contextual, pois essas não acrescentam nenhum sentido semântico como, por exemplo: preposições (de, da, do, para, em, no), conjunção (e) e artigos(o,a).

Citar o que é preposição.

Tabela3: Palavras mais frequentes

- Tabela 03 Palavras mais frequentes em todo o conjunto de solicitações

```
library(tidytext)
palavras <- DB %>%
  unnest_tokens(palavra, PEDIDO) %>%
  count(palavra, sort = TRUE) %>%
  ungroup()

palavras[0:10,] %>%
  kable("latex", caption = "Principais palavras com stopwords",
        booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 4: Principais palavras com stopwords

palavra	n
de	4.071
a	1.311
e	1.162
do	900
o	891
da	821
para	712
no	584
em	575
energia	553

Tabelas4,5,6: Palavras mais frequentes por diretoria

Cria o objeto de palavras por diretoria

```
palavras_diretoria <- DB %>%
  unnest_tokens(palavra, PEDIDO) %>%
  count(DIRETORIA, palavra, sort = TRUE) %>%
  ungroup() %>% droplevels() %>% drop_na()

palavras_diretoria$DIRETORIA = as.factor(palavras_diretoria$DIRETORIA)
```

Tabelas4: Palavras mais frequentes DEA

- Tabela 04 Palavras mais frequentes no conjunto de solicitações por diretoria

```
DEA_termo =
palavras_diretoria %>%
  filter(DIRETORIA == "DEA") %>% droplevels()

DEA_termo %>%
  top_n(n = 10) %>%
  kable("latex", caption = "Principais palavras com stopwords (DEA)",
        booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))

## Selecting by n
```

Table 5: Principais palavras com stopwords (DEA)

DIRETORIA	palavra	n
-----------	---------	---

Tabelas5: Palavras mais frequentes DEE

- Tabela 05 Palavras mais frequentes no conjunto de solicitações por diretoria

```
DEE_termo =
palavras_diretoria %>%
  filter(DIRETORIA == "DEE") %>% droplevels()

DEE_termo %>%
  top_n(n = 10) %>%
  kable("latex", caption = "Principais palavras com stopwords (DEA)",
        booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))

## Selecting by n
```

Table 6: Principais palavras com stopwords (DEA)

DIRETORIA	palavra	n
-----------	---------	---

Tabelas6: Palavras mais frequentes OUTROS

- Tabela 06 Palavras mais frequentes no conjunto de solicitações por diretoria

```
OUTROS =
palavras_diretoria %>%
  filter(DIRETORIA == "OUTROS") %>% droplevels()

OUTROS %>%
  top_n(n = 10) %>%
  kable("latex", caption = "Principais palavras com stopwords (OUTROS)",
        booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))

## Selecting by n
```

Mesmo assim, abrindo para cada uma das 3 possíveis categorias da variável **Diretoria** temos que as principais palavras não agregam nenhum valor semântico, exceto pela palavra energia que apareceu na oitava

Table 7: Principais palavras com stopwords (OUTROS)

DIRETORIA	palavra	n
OUTROS	de	1.073
OUTROS	a	384
OUTROS	e	359
OUTROS	o	261
OUTROS	da	224
OUTROS	do	207
OUTROS	para	162
OUTROS	ou	160
OUTROS	em	158
OUTROS	que	152

e nona colocação de maior frequência dos documentos de pedidos enviados à *DEA* e *DEE*, respectivamente. Isso devido ao excesso de uso de **stop words** em textos humanos.

Em passos mais adiante serão removidas essas palavras, **stop words**, e a partir da remoção o trabalho se dará apenas com palavras de sentido semântico relevante aos subjetivos solicitados às diretorias, acrescentando assim maior assertividade do modelo de classificação.

Verificamos, antes disso, o total, freq. e média de palavras por diretoria, bem como comparações 2 a 2 para cada uma das categorias. E avançamos um pouco com gráficos da contagem de frequência e a lei de **Zipf** que dá suporte as conclusões do passo anterior e, a por conseguinte, é definida a estatística de **tf_idf** (**term frequency times inverse document frequency**), uma estatística utilizada para ressaltar termos relevantes para um documento em particular.

Análise2: Comparação de freq. de palavras por diretoria

- Total de palavras por diretoria, total de pedidos por diretoria e número médio de palavras por pedido e diretoria

```
total_palavras = palavras_diretoria %>%
  group_by(DIRETORIA) %>%
  summarize(total_palavras = sum(n))

total_palavras$DIRETORIA = as.character(total_palavras$DIRETORIA)
total_palavras = left_join(x = total_palavras, y = pedidos_diretoria1,
  by = "DIRETORIA") %>%
  mutate(media_palavras_porpedidoEdiretoria = total_palavras/total_pedidos)
```

Tabelas7: Total de palavras por diretoria, total de pedidos por diretoria e número médio de palavras por pedido e diretoria

- Total de palavras por diretoria, total de pedidos por diretoria e número médio de palavras por pedido e diretoria

```
total_palavras %>%
  kable("latex", caption = "Total de palavras, total de pedidos e número médio de palavras
  por pedido e diretoria",
  booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 8: Total de palavras, total de pedidos e número médio de palavras por pedido e diretoria

DIRETORIA	total_palavras	total_pedidos	media_palavras_porpedidoEdiretoria
1	17.211	240	71,71250
2	15.536	244	63,67213
5	3.619	67	54,01493
OUTROS	13.351	154	86,69481

Temos que o número médio de palavras por pedido é parecido entre as diretorias. com médias de 55 palavras por pedido para DEE e 69,7 e 61,7, respectivamente para DEA e OUTROS.

Figural1: Distribuição de frequência de termos por diretoria

- Distribuição da freq. de palavras usadas em solicitações por diretoria (histograma)

```
diretoria_palavras <- DB %>%
  unnest_tokens(palavra, PEDIDO) %>%
  count(DIRETORIA, palavra, sort = TRUE) %>%
  ungroup()

diretoria_palavras = left_join(diretoria_palavras, total_palavras, by = "DIRETORIA")

library(ggplot2)
gcomma <- function(x) format(x, big.mark = ".", decimal.mark = ",", scientific = FALSE)

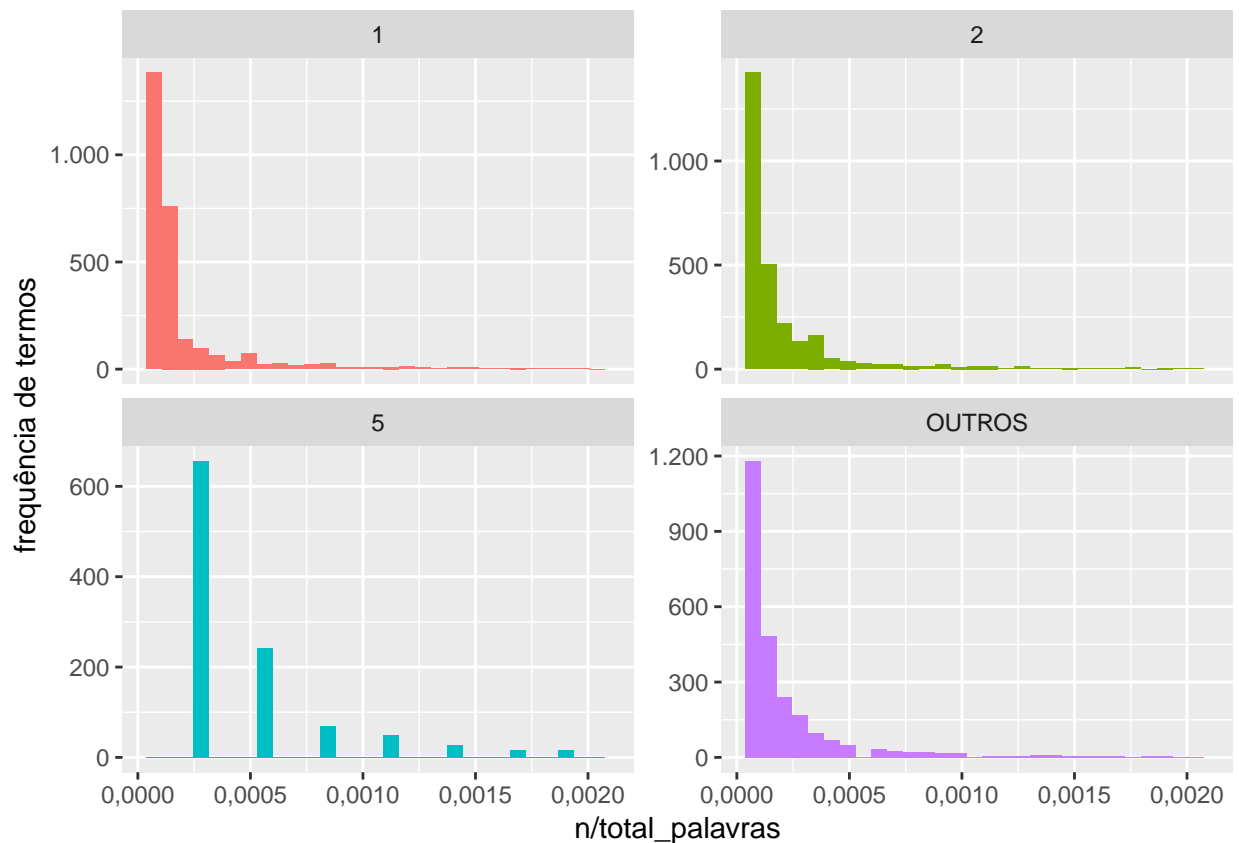
ggplot(diretoria_palavras, aes(n/total_palavras, fill = DIRETORIA)) +
  geom_histogram(show.legend = FALSE) + xlim(NA, 0.0021) +
  facet_wrap(~DIRETORIA, ncol = 2, scales = "free_y") +
  scale_y_continuous(labels=gcomma) +
  scale_x_continuous(labels=gcomma, limits = c(NA, 0.0021)) +
  labs(y = "frequência de termos")

## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 249 rows containing non-finite values (stat_bin).

## Warning: Removed 4 rows containing missing values (geom_bar).
```



Pelos histogramas fica claro que as distribuições da frequência de termos por diretoria possuem caudas mais alongadas à direita. Além disso, algumas frequências não foram evidenciadas no gráfico por questões de escala. De fato, as palavras/termos de maior recorrência nos documentos/textos são as de menor relevância em contexto semântica.

Sabemos, portanto, que queremos encontrar valor exatamente nas partes mais longas à direita das distribuições de frequência de termos, uma vez que ali se encontram as palavras de maior relevância contextual.

Logo, a seguir, usamos da definição da lei **Zipf** que afirma que a frequência que uma palavra (ou termo) aparece em um documento é inversamente proporcional ao seu ranque.

lei de Zipf's

Citar, aqui, “There are very long tails to the right for these novels (those extremely common words!) that we have not shown in these plots. These plots exhibit similar distributions for all the novels, with many words that occur rarely and fewer words that occur frequently.” pág. 31 (Silge, Robinson). Que averigua que documentos de texto tendem a ter distribuições de frequência de palavras similar, por conta das stopwords.

Ainda de acordo com os autores, “Distributions like those shown in Figure 3-1 are typical in language. In fact, those types of long-tailed distributions are so common in any given corpus of natural language (like a book, or a lot of text from a website, or spoken words) that the relationship between the frequency that a word is used and its rank has been the subject of study.” e por essa razão é a relação verificada por George Zipf da relação inversa entre freq. de palavra e ranque tiramos valor dos documentos partindo dessas premissas.

- Ranque de palavras pela lei de **Zipf**

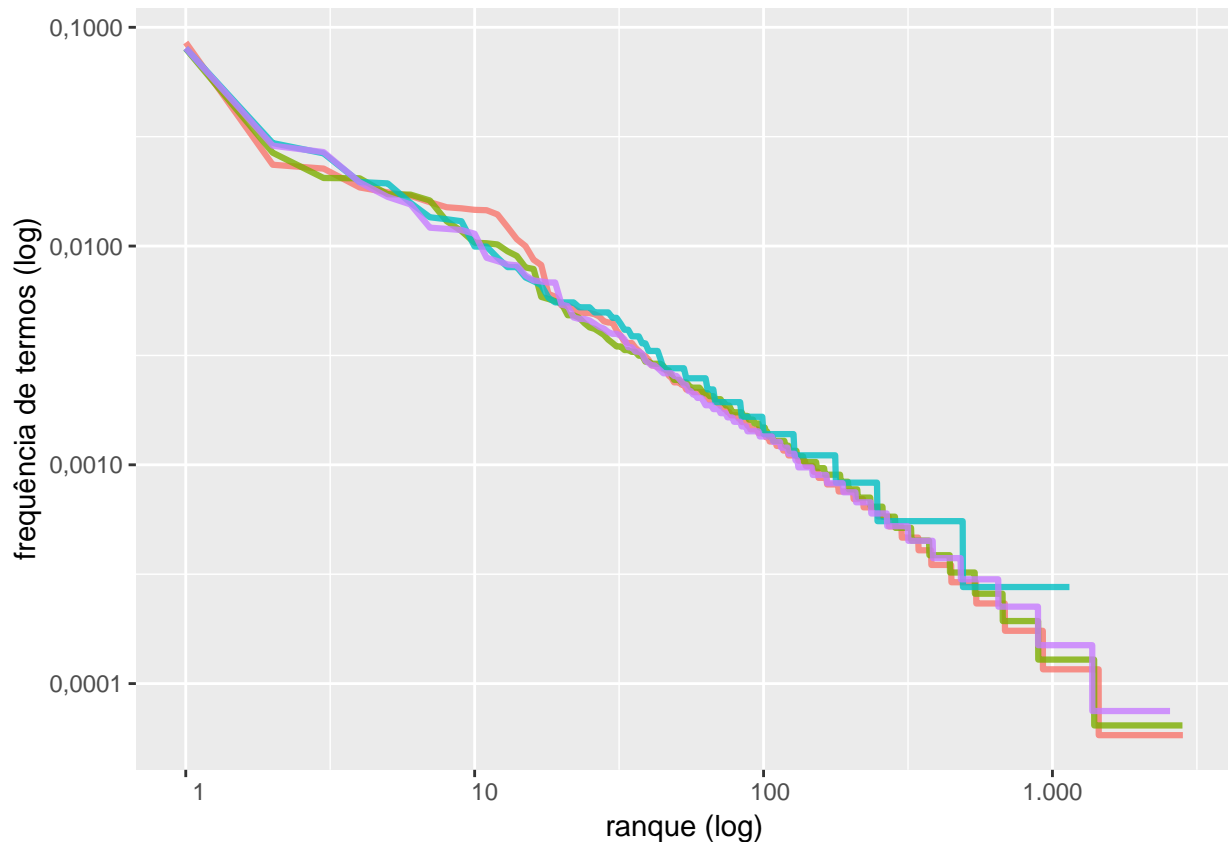
```
freq_by_rank <- diretoria_palavras %>%
  group_by(DIRETORIA) %>%
  mutate(ranque = row_number(),
         `frequência de termos` = n/total_palavras)
```

Figura1: Lei de Zipf

- Zipf's law

```
#plot1
freq_by_rank %>%
ggplot(aes(ranke, `frequência de termos`, color = DIRETORIA)) +
  geom_line(size = 1.1, alpha = 0.8, show.legend = FALSE) + scale_x_log10() +
  scale_y_log10(labels=gcomma) +
  scale_x_log10(labels=gcomma) +
  labs(y = "frequência de termos (log)", x = "ranque (log)")

## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```



Vemos que exatamente nas extremidades do gráfico tem-se uma não sobreposição de frequências por diretoria. Detalhe que o gráfico, em questão, está na escala logarítmica no eixo x (ranque) e eixo y (freq. de termos). Plotando desta forma, a relação inversamente proporcional terá uma inclinação constante e negativa.

Tendo em vista, portanto, que o gráfico referido está em coordenadas log-log e dado a semelhança de todos os documentos de texto das diferentes diretorias, afirmamos que para todas as diretorias pela Lei de **Zipf** a relação entre ranque e freq. de termos assumirá, sempre, uma inclinação negativa, ou seja,

Daí, aplicando a escala log-log temos que e podemos aplicar um ajuste a fim de encontrar um intercepto e coef. angular para traçar no gráfico anterior.

$$frequência \propto \frac{1}{ranque} \implies \log(frequência) \propto \log\left(\frac{1}{ranque}\right)$$

Reescrever e explicar a segmentação em 3 partes como uma “lei de potenciação dividida em 3 partes” e então

utilizar do seguimento do meio, onde as freq. de termos são mais semelhantes para diferentes ranques das diferentes diretorias. Fica claro pela eq.

“Notice that Figure 3-2 is in log-log coordinates. We see that all six of Jane Austen’s novels are similar to each other, and that the relationship between rank and frequency does have negative slope. It is not quite constant, though; perhaps we could view this as a broken power law with, say, three sections. Let’s see what the exponent of the power law is for the middle section of the rank range.”

```
rank_subset <- freq_by_rank %>%
  filter(ranque < 500, ranque > 50)

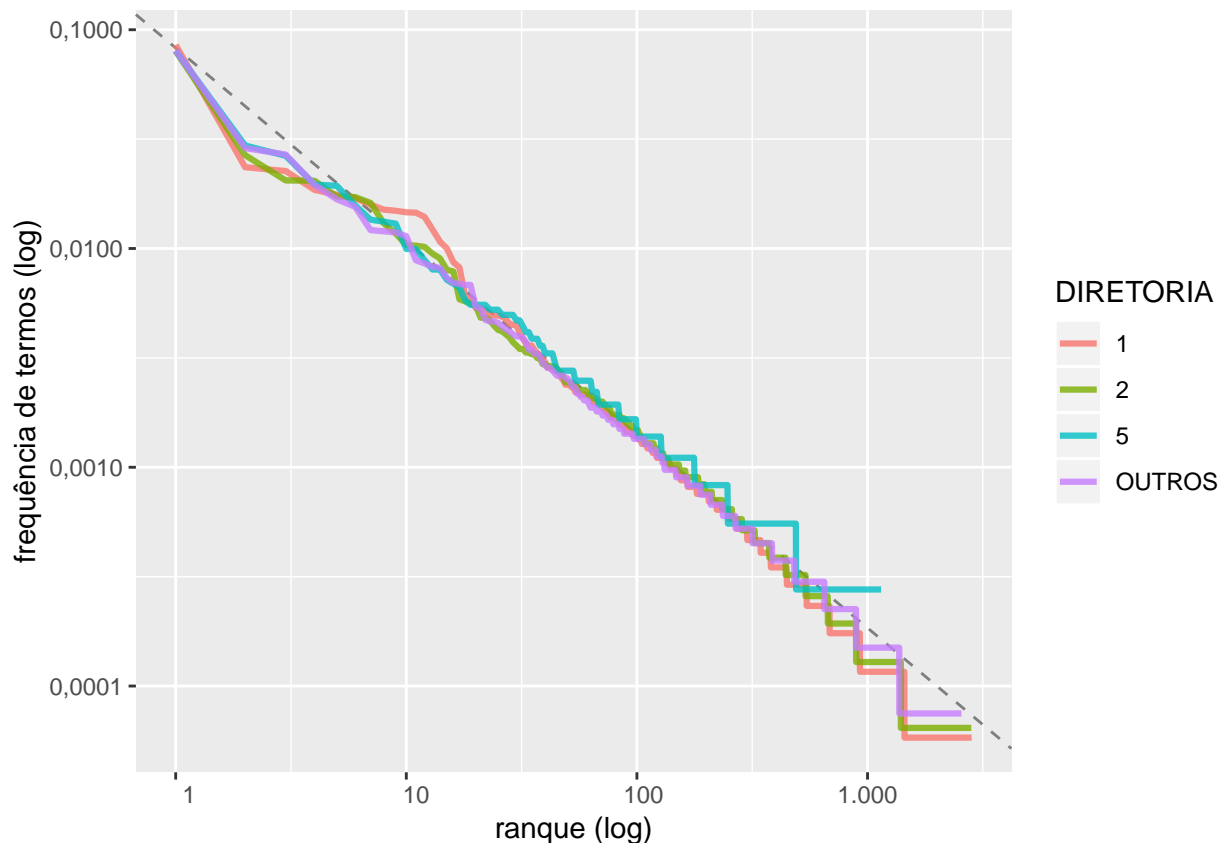
(zipf_ajusteloglog <- lm(log10(`frequência de termos`) ~ log10(ranque),
  data = rank_subset))

##
## Call:
## lm(formula = log10(`frequência de termos`) ~ log10(ranque),
##     data = rank_subset)
##
## Coefficients:
##      (Intercept)      log10(ranque)
##          -1.0832           -0.8837
```

Finalmente, traçando e sobrepondo o gráfico anterior com os valores de intercepto e coeficiente angular obtidos no ajuste do passo anterior temos a figura a seguir.

Figural: Lei de Zipf + ajuste log-log

```
freq_by_rank %>%
  ggplot(aes(ranque, `frequência de termos`, color = DIRETORIA)) +
  geom_abline(intercept = coefficients(zipf_ajusteloglog)[1], slope = coefficients(zipf_ajusteloglog)[2],
  geom_line(size = 1.1, alpha = 0.8, show.legend = TRUE) +
  scale_y_log10(labels=gcomma) +
  scale_x_log10(labels=gcomma) +
  labs(y = "frequência de termos (log)", x = "ranque (log)")
```



The Bind `tf_idf`

Fundamentar o uso da estatística `tf_idf`, bem como descrever a definição.

The idea of `tf-idf` is to find the important words for the content of each document by decreasing the weight for commonly used words and increasing the weight for words that are not used very much in a collection or corpus of documents, in this case, the group of Jane Austen's novels as a whole. Calculating `tf-idf` attempts to find the words that are important (i.e., common) in a text, but not too common. Let's do that now. The `bind_tf_idf` function in the `tidytext` package takes a tidy text dataset as input with one row per token (term), per document. One column (word here) contains the terms/tokens, one column contains the documents (book in this case), and the last necessary column contains the counts, or how many times each document contains each term (`n` in this example). We calculated a total for each book for our explorations in previous sections, but it is not necessary for the `bind_tf_idf` function; the table only needs to contain all the words in each document.

```
round_df <- function(x, digits) {
  # round all numeric variables
  # x: data frame
  # digits: number of digits to round
  numeric_columns <- sapply(x, mode) == 'numeric'
  x[numeric_columns] <- round(x[numeric_columns], digits)
  x
}
```

- Palavras mais relevantes de acordo com a estatística `tf_idf`


```

diretoria_palavras_tfidf <- diretorio_palavras %>%
  bind_tf_idf(palavra, DIRETORIA, n) %>%
  select(-total_palavras, -total_pedidos, -media_palavras_porpedidoEdiretoria) %>%
  arrange(desc(tf_idf))

#options(digits=4)
set.seed(7456)
amostra1 = sample(seq(1:dim(diretorio_palavras_tfidf)[1]), 10, replace = FALSE)
round_df(diretorio_palavras_tfidf[amostra1,],6) %>%
  kable("latex", caption = "Total de palavras, total de pedidos e número médio de palavras
por pedido e diretoria",
  booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))

```

Table 9: Total de palavras, total de pedidos e número médio de palavras por pedido e diretoria

DIRETORIA	palavra	n	tf	idf	tf_idf
1	verão	3	0,000174	1,386294	0,000242
1	contratar	1	0,000058	0,693147	0,000040
5	pública	2	0,000553	0,000000	0,000000
2	prezados	58	0,003733	0,000000	0,000000
5	apresentação	1	0,000276	0,693147	0,000192
2	coelba	1	0,000064	0,693147	0,000045
1	fazendo	9	0,000523	0,287682	0,000150
2	power	1	0,000064	1,386294	0,000089
OUTROS	encontro	1	0,000075	0,287682	0,000022
2	prezado	4	0,000257	0,287682	0,000074

A estatística faz um trabalho brilhante ao ressaltar as palavras mais relevantes dentro de cada conjunto de documentos (diretorias). As tabelas a seguir mostram as 10 palavras mais relevantes de acordo com a estatística tf_idf por diretoria

Tabela8: top 12 termos ordenados pela estatística tf_idf (DEE)

```

round_df(diretorio_palavras_tfidf,5) %>%
  filter(DIRETORIA == "DEE") %>%
  top_n(12,tf_idf) %>%
  kable("latex", caption = "Top 10 termos (DEE)",
  booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))

```

Table 10: Top 10 termos (DEE)

DIRETORIA	palavra	n	tf	idf	tf_idf
-----------	---------	---	----	-----	--------

Tabela9: top 12 termos ordenados pela estatística tf_idf (DEA)

```

round_df(diretorio_palavras_tfidf,5) %>%
  filter(DIRETORIA == "DEA") %>%
  top_n(12,tf_idf) %>%
  kable("latex", caption = "Top 10 termos (DEA)",

```

```
booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
kable_styling(latex_options = c("striped", "hold_position"))
```

Table 11: Top 10 termos (DEA)

DIRETORIA	palavra	n	tf	idf	tf_idf
-----------	---------	---	----	-----	--------

Tabela10: top 10 termos ordenados pela estatística tf_idf (OUTROS)

```
round_df(diretoria_palavras_tfidf,5) %>%
  filter(DIRETORIA == "OUTROS") %>%
  top_n(12,tf_idf) %>%
  kable("latex", caption = "Top 10 termos (DEA)",
        booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 12: Top 10 termos (DEA)

DIRETORIA	palavra	n	tf	idf	tf_idf
OUTROS	funcionários	37	0,00277	1,38629	0,00384
OUTROS	entidade	27	0,00202	1,38629	0,00280
OUTROS	contratos	53	0,00397	0,69315	0,00275
OUTROS	empregados	20	0,00150	1,38629	0,00208
OUTROS	locação	18	0,00135	1,38629	0,00187
OUTROS	salários	16	0,00120	1,38629	0,00166
OUTROS	pessoal	12	0,00090	1,38629	0,00125
OUTROS	cargos	22	0,00165	0,69315	0,00114
OUTROS	prestação	11	0,00082	1,38629	0,00114
OUTROS	cargo	20	0,00150	0,69315	0,00104
OUTROS	pagamento	10	0,00075	1,38629	0,00104
OUTROS	passagens	10	0,00075	1,38629	0,00104
OUTROS	viagem	10	0,00075	1,38629	0,00104

Ou simplesmente verificamos através de um gráfico

Figura2: Termos mais relevantes por diretoria pela estatística tf_idf

```
diretoria_palavras <- DB %>%
  unnest_tokens(palavra, PEDIDO) %>%
  count(DIRETORIA, palavra, sort = TRUE) %>%
  ungroup()
diretoria_palavras = left_join(diretoria_palavras, total_palavras, by = "DIRETORIA")
```

Figura3: Top 10 termos por diretoria (ordenados pela estatística tf_idf e com stop words e sem stemming)

```
plot_diretoria_palavras <- diretoria_palavras %>%
  bind_tf_idf(palavra, DIRETORIA, n) %>%
  arrange(desc(tf_idf)) %>%
  mutate(palavra = factor(palavra, levels = rev(unique(palavra)))) %>%
```

```

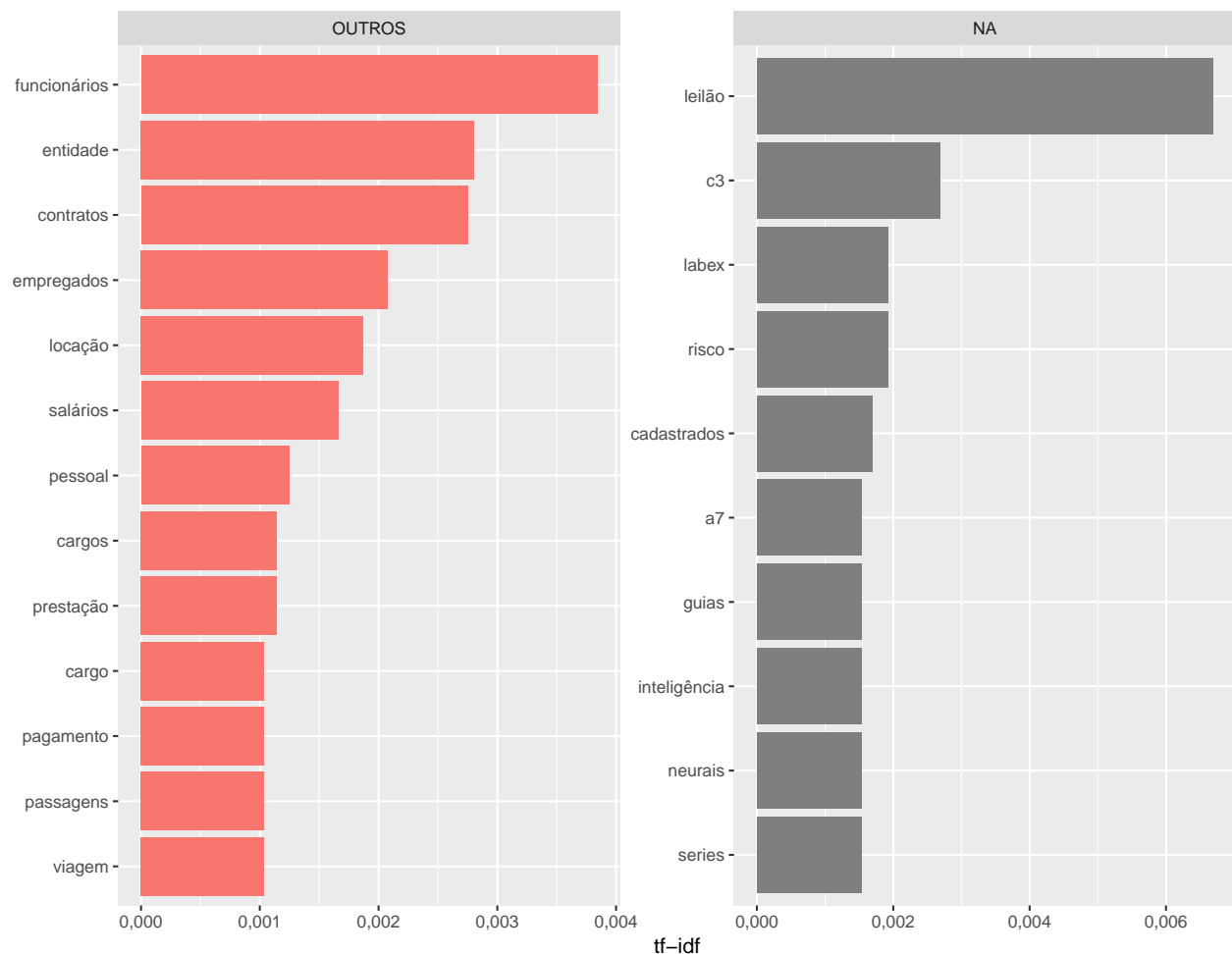
mutate(DIRETORIA = factor(DIRETORIA, levels = c("DEA", "DEE", "OUTROS")))
#View(head(plot_diretoria_palavras))
#jpeg("02_freq_palavras_dir.jpeg")
plot_diretoria_palavras %>%
  group_by(DIRETORIA) %>%
  top_n(10, tf_idf) %>%
  ungroup() %>%
  mutate(palavra = reorder(palavra, tf_idf)) %>%
  ggplot(aes(palavra, tf_idf, fill = DIRETORIA)) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "tf-idf") +
  facet_wrap(~DIRETORIA, ncol = 2, scales = "free") +
  coord_flip() +
  scale_y_continuous(labels=gcomma)

```

```

## Warning: Factor `DIRETORIA` contains implicit NA, consider using
## `forcats::fct_explicit_na`

```



```

#dev.off()

```

Filtrando um pedaço de texto

```
DB %>%
  filter(str_detect(PEDIDO, "in")) %>%
  select(PEDIDO) %>%
  head()
```

```
## # A tibble: 6 x 1
##   PEDIDO
##   <chr>
## 1 "Empresa de Pesquisa Energética A Empresa de Pesquisa Energética (vincul~
## 2 Demanda ou carga Energética total (comercial, industrial, residencial e ~
## 3 "para EPE empresa de pesquisa energética - dados sobre custos de energia~
## 4 "Dados sobre Consumo de Energia por Unidades da Federação Boa tarde,\n\n~
## 5 Manifestação no processo de licenciamento da UHE Castanheira (processo n~
## 6 "Dados sobre quantidade de energia consumida e dinheiro pago pelo consum~
```

Uma limpeza removendo palavras sem significado semântico (**stop words**) pode auxiliar o algoritmo a retornar palavras ainda mais assertivas, bem como o tratamento de **stemming**, abordados a seguir.

Colocar tudo em minúsculo

```
DB$PEDIDO1 = tolower(DB$PEDIDO)
```

Stopwords

Com o arquivo de **stop words**, vamos remover as palavras sem sentido semântico

```
mystopwords <- data_frame(palavra = stopwords_pt)
for (j in 1:dim(DB)[1]) {
  for(i in 1:dim(mystopwords)[1]){
    stopw = as.character(mystopwords[i,1])
    DB$PEDIDO1[j] = gsub(paste0("\\ ",stopw," "), " ", as.character(DB$PEDIDO1[j]))
  }
}
```

Ou simplesmente

```
mystopwords <- data_frame(palavra = stopwords_pt)

## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.

DB$PEDIDO1 <- removeWords(DB$PEDIDO1, mystopwords$palavra)
#View(head(DB))
```

Stemming

Podemos diminuir redundâncias por parte do algoritmo ensinando-o a compreender palavras que podem estar escritas de forma diferente mas que em significado semântico são semelhantes. Para isso, analisamos o radical de palavras com um mesmo prefixo mas com sufixos diferentes seja por quisistos como gênero ou plural.

Exemplos:

leilão \propto leilões estado \propto estados região \propto regiões

Usando o pacote **ptstem**

```
library(ptstem)
temp_stem1 = proc.time()
stemming1 = ptstem(DB$PEDID01)
tempo_stem1 = proc.time() - temp_stem1
```

- Frequência de palavras por diretoria do stemming 1

```
diretoria_palavras_stem1 <- DB %>%
  mutate(PEDID01 = stemming1) %>%
  unnest_tokens(palavra, PEDID01) %>%
  count(palavra, sort = TRUE) %>%
  ungroup()
```

```
cat(paste0("Utilizando o algoritmo de stemming do pacote 'ptstem' o número de palavras chaves sem stemming reduziu"))
```

```
## Utilizando o algoritmo de stemming do pacote 'ptstem' o número de palavras chaves sem stemming reduziu
```

Usando o pacote rslp

```
library(rslp)
temp_stem2 = proc.time()
stemming2 = rslp(DB$PEDID01)
tempo_stem2 = proc.time() - temp_stem2
```

- Frequência de palavras por diretoria do stemming 2

```
diretoria_palavras_stem2 <- DB %>%
  mutate(PEDID01 = stemming2) %>%
  unnest_tokens(palavra, PEDID01) %>%
  count(palavra, sort = TRUE) %>%
  ungroup()
```

```
cat(paste0("Utilizando o algoritmo de stemming do pacote 'rslp' o número de palavras chaves sem stemming reduziu"))
```

```
## Utilizando o algoritmo de stemming do pacote 'rslp' o número de palavras chaves sem stemming reduziu
```

Uma redução considerável no número de termos ocorreu ao usar o algoritmo `ptstem`, cerca de 61% de redução de termos versus 36% utilizando o algoritmo `rslp`, ou seja, o algoritmo `ptstem` foi mais eficiente na tarefa de agrupar os semelhantes (termos únicos).

Vale ressaltar, também, o tempo de processamento que ambos os algoritmos requerem.

```
cat(paste0("O tempo de processamento do stemming rslp( ) foi de ",round(tempo_stem1[3],2), " segundos decorridos"))
```

```
## O tempo de processamento do stemming rslp( ) foi de 10.51 segundos decorridos.
```

```
remove(tempo_stem1)
```

```
cat(paste0("O tempo de processamento do stemming ptstem( ) foi de ",round(tempo_stem2[3],2), " segundos decorridos"))
```

```
## O tempo de processamento do stemming ptstem( ) foi de 0.82 segundos decorridos.
```

```
remove(tempo_stem2)
```

O tempo decorrido para processamento do algoritmo do `ptstem` foi de aproximadamente 12 segundos versus 1 segundo decorrido para o processamento do algoritmo do `rslp`. Logo, o `rslp` é quase 12 vezes mais eficiente em termos de tempo de processamento. Além disso, o `rslp` remove acentuações e caracteres como “ç”. Isso irá nos ajudar mais a frente quando utilizarmos os principais termos como variáveis binárias e preditoras do modelo de classificação.

Entretanto, o algoritmo mais lento, `ptstem`, foi mais interessante em termos de redução do número de termos únicos, cerca de 25% menos termos únicos em relação ao outro algoritmo. Além disso, por se tratar de uma base de dados relativamente pequena, 625 pedidos, e pouco mais de 4 mil termos únicos em todo o conjunto de texto, além disso vamos utilizar de um alto poder de processamento da máquina no referido estudo. Optamos, portanto, por utilizar ambos algoritmos. Vamos, primeiro, aplicar o removedor de sufixos da língua portuguesa `rsnp` seguido do `ptstem`.

Comparação do texto original c/ os 2 algoritmos e o final implementados após diferentes **stemmings**

```
DB$PEDIDO[227]
```

```
## [1] "Destino dos honorários sucumbências Prezados, boa tarde. Desejo obter informações acerca da des"
```

```
#stemming1[227]
```

```
#stemming2[227]
```

```
DB$PEDIDO1[227]
```

```
## [1] "destino honorários sucumbências , . desejo obter destinação dada honorários sucumbência"
```

```
DB$PEDIDO[350]
```

```
## [1] "SOLICITAÇÃO DE NOTAS TÉCNICAS Prezados(as) Senhores(as),\n\nsou o Engº Eletricista Lidinei Serg"
```

```
#stemming1[350]
```

```
#stemming2[350]
```

```
DB$PEDIDO1[350]
```

```
## [1] " notas técnicas () senhores(),\n\n engº eletricista lidinei sergio mesquita neri (ex-colabora"
```

```
DB$PEDIDO[615]
```

```
## [1] "Gás do pré-sal Com fundamento na Lei 12.527/2011 (Lei de Acesso a\n\nInformações Públicas) venho a"
```

```
#stemming1[615]
```

```
#stemming2[615]
```

```
DB$PEDIDO1[615]
```

```
## [1] "gás pré-sal fundamento lei 12.527/2011 (lei \n\n públicas) requerer , 20 dias corridos"
```

```
DB$PEDIDO[617]
```

```
## [1] "Questionamento sobre dados do PIB apresentados em Planos Decenais de Expansão Energética (PDEs)"
```

```
#stemming1[617]
```

```
#stemming2[617]
```

```
DB$PEDIDO1[617]
```

```
## [1] "questionamento pib apresentados planos decenais expansão energética (pdes) publicados ."
```

```
DB$PEDIDO[619]
```

```
## [1] "Dados distribuição de energia- UF Amapá Verifiquei que no Plano Decenal de Expansão de Energia"
```

```
#stemming1[619]
```

```
#stemming2[619]
```

```
DB$PEDIDO1[619]
```

```
## [1] " distribuição - uf amapá verifiquei plano decenal expansão (2006/ 2015) tabela -25 - \n"
```

Cria, antes, uma variável `PEDIDO1` que repete os passos feitos aos termos quanto ao stemming so que no texto fonte.

```
DB$PEDIDO1 = tolower(DB$PEDIDO)
```

```
mystopwords <- data_frame(palavra = stopwords_pt)
```

```

DB$PEDIDO1 <- removeWords(DB$PEDIDO1, mystopwords$palavra) # Remove Stop Words
DB$PEDIDO1 <- removePunctuation(DB$PEDIDO1) # Remove Punctuation

rm_accent <- function(str,pattern="all") {
  if(!is.character(str))
    str <- as.character(str)

  pattern <- unique(pattern)

  if(any(pattern=="ç"))
    pattern[pattern=="ç"] <- "ς"

  symbols <- c(
    acute = "áéíóúÁÉÍÓÚýÝ",
    grave = "àèìòùÀÈÌÒÙ",
    circumflex = "âêîôûÂÊÎÔÛ",
    tilde = "ãõÃÕñÑ",
    umlaut = "äëïöüÄËÏÖÜÿ",
    cedil = "çς"
  )

  nudeSymbols <- c(
    acute = "aeiouAEIOUyY",
    grave = "aeiouAEIOU",
    circumflex = "aeiouAEIOU",
    tilde = "aoAOnN",
    umlaut = "aeiouAEIOUy",
    cedil = "cC"
  )

  accentTypes <- c("`","^","^","~","~","ç")

  if(any(c("all","al","a","todos","t","to","tod","todo")%in%pattern)) # opcao retirar todos
    return(chartr(paste(symbols, collapse=""), paste(nudeSymbols, collapse=""), str))

  for(i in which(accentTypes%in%pattern))
    str <- chartr(symbols[i],nudeSymbols[i], str)

  return(str)
}

DB$PEDIDO1 <- rm_accent(DB$PEDIDO1) # Remove accent patterns
#View(head(DB))
#View(DB$PEDIDO1)

#View(head(DB))
### CARACTERES
DB$PEDIDO1 = gsub("-", " ",DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:.]", "",DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:,:]", "",DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:':]", " ",DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:!:] ", "",DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:?:]", "",DB$PEDIDO1)

```

```

#DB$PEDIDO1 = gsub("[:-]", "_", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:_]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:__]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:;]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:&]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:/:]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:(:]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:(:)]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:%]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:°]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:°]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:ª]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\d+", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[0-9]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:\\n\\t:]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:\\t:]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:\\n:]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("[:$]", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\s+", " ", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\\"", " ", DB$PEDIDO1)

### STEMMINGS
#DB$PEDIDO1[143]
#DB$PEDIDOz = rslp(DB$PEDIDO1)
#DB$PEDIDOz = ptstem(DB$PEDIDO1, complete = FALSE)
DB$PEDIDO1 = ptstem(rslp(DB$PEDIDO1), complete = FALSE)
DB$PEDIDO1 = gsub("\\s+", " ", DB$PEDIDO1)
#teste1 = ptstem(rslp(DB$PEDIDO1), complete = FALSE)
#DB$PEDIDOz[143]
#DB$PEDIDOz[537]

## REMOVE STOP WORDS novamente
mystopwords <- data_frame(palavra = stopwords_pt)
DB$PEDIDO1 <- removeWords(DB$PEDIDO1, mystopwords$palavra)

### PALAVRAS
#DB$PEDIDO1 = gsub("\\b(Leiloes)", "leilao", DB$PEDIDO1)
#DB$PEDIDO1 = gsub("\\b(Leiloar)", "leilao", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(leiloes)", "leilao", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(leiloar)", "leilao", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(leiloes)", "leilao", DB$PEDIDO1)
#DB$PEDIDO1 = gsub("\\b(Energetica)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(energetica)", "eletrica", DB$PEDIDO1)
#DB$PEDIDO1 = gsub("\\b(Eletricas)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(eletricas)", "eletrica", DB$PEDIDO1)
#DB$PEDIDO1 = gsub("\\b(Eletricos)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(eletricos)", "eletrica", DB$PEDIDO1)
#DB$PEDIDO1 = gsub("\\b(Eletrico)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(eletrico)", "eletrica", DB$PEDIDO1)
#DB$PEDIDO1 = gsub("\\b(Eletricidade)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(eletricidade)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(energetica)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 = gsub("\\b(energeticas)", "eletrica", DB$PEDIDO1)

```



```

DB$PEDIDO1 =gsub("\\b(energetico)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(energeticos)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(energia)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(energias)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(energy)", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(energies)", "eletrica", DB$PEDIDO1)
x#DB$PEDIDO1 =gsub("\\b(Termoeletricas)", "termoeletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(termoeletricas)", "termoeletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Termeletrica)", "termoeletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(termeletrica)", "termoeletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Termeletricas)", "termoeletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(termeletricas)", "termoeletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Hidreletricas)", "hidreletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(hidreletricas)", "hidreletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Hidroeletricas)", "hidreletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(hidroeletricas)", "hidreletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Hidroeletricos)", "hidreletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(hidroeletricos)", "hidreletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Hidroeletrica)", "hidreletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(hidroeletrica)", "hidreletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Administracao)", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(administracao)", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Administrativo)", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(administrativo)", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Administrativos)", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(administrativos)", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Administrativa)", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(administrativa)", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Administrativas)", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(administrativas)", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Consumo)", "consumo", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Consumidores)", "consumo", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(consumidores)", "consumo", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Consumidor)", "consumo", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(consumidor)", "consumo", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(Consumir)", "consumo", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\b(consumir)", "consumo", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\b(http)>", "", DB$PEDIDO1)
#View(DB$PEDIDO1)

```

```

DB$PEDIDO1 =gsub("\\ leiloes ", "leilao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ leiloar ", "leilao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ leiloes ", "leilao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Energetica ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energetica ", "eletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Eletricas ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ eletricas ", "eletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Eletricos ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ eletricos ", "eletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Eletrico ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ eletrico ", "eletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Eletricidade ", "eletrica", DB$PEDIDO1)

```

```

DB$PEDIDO1 =gsub("\\ eletricidade ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energetica ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energeticas ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energetico ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energeticos ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energia ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energias ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energy ", "eletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ energies ", "eletrica", DB$PEDIDO1)
x#DB$PEDIDO1 =gsub("\\ Termoeletricas ", "termoeletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ termoeletricas ", "termoeletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Termeletrica ", "termoeletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ termeletrica ", "termoeletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Termeletricas ", "termoeletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ termoeletricas ", "termoeletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Hidreletricas ", "hidreletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ hidreletricas ", "hidreletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Hidroeletricas ", "hidreletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ hidroeletricas ", "hidreletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Hidroeletricos ", "hidreletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ hidroeletricos ", "hidreletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Hidroeletrica ", "hidreletrica", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ hidroeletrica ", "hidreletrica", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Administracao ", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ administracao ", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Administrativo ", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ administrativo ", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Administrativos ", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ administrativos ", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Administrativa ", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ administrativa ", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Administrativas ", "administracao", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ administrativas ", "administracao", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Consumo ", "consumo", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Consumidores ", "consumo", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ consumidores ", "consumo", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Consumidor ", "consumo", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ consumidor ", "consumo", DB$PEDIDO1)
#DB$PEDIDO1 =gsub("\\ Consumir ", "consumo", DB$PEDIDO1)
DB$PEDIDO1 =gsub("\\ consumir ", "consumo", DB$PEDIDO1)

```

Frequência de palavras por diretoria

```

diretoria_palavras_stem3 <- DB %>%
  unnest_tokens(palavra, PEDIDO1) %>%
  count(DIRETORIA, palavra, sort = TRUE) %>%
  ungroup()

```

Stopwords

Com o arquivo de **stop words** previamente inserido vamos, primeiramente, transforma-lo em um data_frame a fim de futuramente utilizá-lo para extrair do texto palavras em comum.

Freq. de palavras sem stopwords por diretoria

```
mystopwords <- data_frame(palavra = stopwords_pt)
diretoria_palavras_noSTOP <- anti_join(diretoria_palavras_stem3, mystopwords,
                                       by = "palavra")
```

Filtrando um pedaço de texto

```
DB %>%
  filter(str_detect(PEDIDO01, "leiloes")) %>%
  select(PEDIDO01) %>%
  head()
```

Comparação do texto original c/ os 2 algoritmos e o final implementados após diferentes **stemmings**

```
DB$PEDIDO[227]
```

```
## [1] "Destino dos honorários sucumbências Prezados, boa tarde. Desejo obter informações acerca da des
ptstem(DB$PEDIDO[227])
```

```
## [1] "Destino dos honorários sucumbências Prezados, boa tarde. Desejo obter informações acerca da des
rslp(DB$PEDIDO[227])
```

```
## [1] "Destino dos honorarios sucumbencias Prezados, boa tarde. Desejo obter informacoes acerca da des
DB$PEDIDO01[227]
```

```
## [1] "destin honora sucumbenc desej obt destinaca dad honora sucumbenc ambit empr repart advog carr in
DB$PEDIDO[350]
```

```
## [1] "SOLICITAÇÃO DE NOTAS TÉCNICAS Prezados(as) Senhores(as),\n\nsou o Engº Eletricista Lidinei Serg
ptstem(DB$PEDIDO[350])
```

```
## [1] "SOLICITAÇÃO DE NOTAS TÉCNICAS Prezados(as) Senhores(as),\n\nsou o Engº Eletricista Lidinei Serg
rslp(DB$PEDIDO[350])
```

```
## [1] "SOLICITACAO DE NOTAS TECNICAS Prezados(as) Senhores(as),\n\nsou o Engº Eletricista Lidinei Serg
DB$PEDIDO01[350]
```

```
## [1] " not tecn senh eng eletric lidin sergi mesquit ner excolabor eletrobr eletronucl gentil inform p
DB$PEDIDO[617]
```

```
## [1] "Questionamento sobre dados do PIB apresentados em Planos Decenais de Expansão Energética (PDEs)
DB$PEDIDO01[617]
```

```
## [1] "question pib apresent plan decen expansa energ pde public explicaco val tax cresc pib nacion ap
```

Figura4: Termos mais relevantes por diretoria pela estatística **tf_idf**, após stemming porém ainda com stop words

Vamos, agora, plotar as quinze palavras mais relevantes de acordo com a estatística **tf_idf**, por diretoria

```
plot_diretoria_palavras_stem <- diretoria_palavras_stem3 %>%
  bind_tf_idf(palavra, DIRETORIA, n) %>%
  arrange(desc(tf_idf)) %>%
```

```

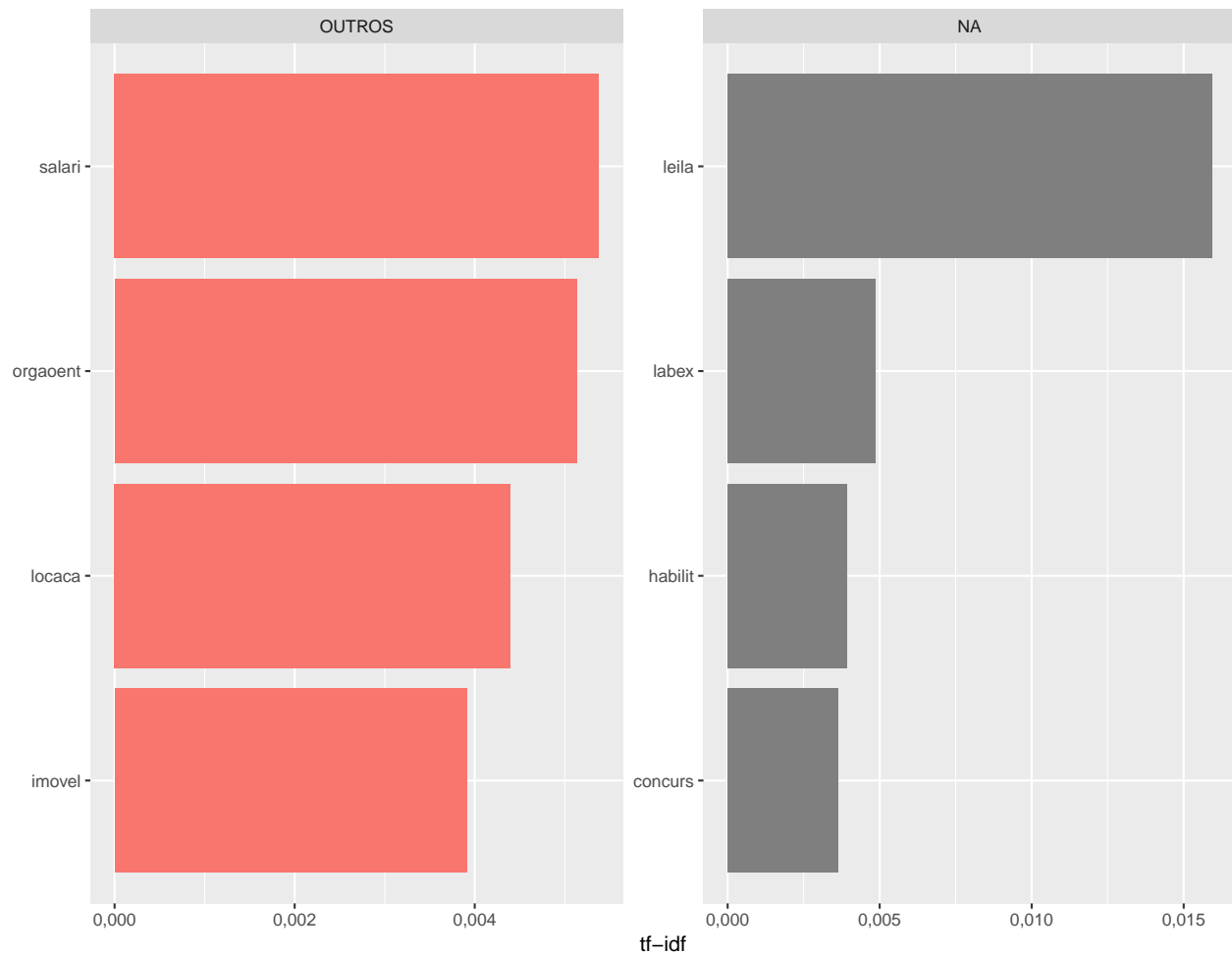
mutate(palavra = factor(palavra, levels = rev(unique(palavra)))) %>%
mutate(DIRETORIA = factor(DIRETORIA, levels = c("DEA", "DEE", "OUTROS")))
#View(head(plot_diretoria_palavras))
#jpeg("02_freq_palavras_dir.jpeg")
plot_diretoria_palavras_stem %>%
  group_by(DIRETORIA) %>%
  top_n(4, tf_idf) %>%
  ungroup() %>%
  mutate(palavra = reorder(palavra, tf_idf)) %>%
  ggplot(aes(palavra, tf_idf, fill = DIRETORIA)) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "tf-idf") +
  facet_wrap(~DIRETORIA, ncol = 2, scales = "free") +
  coord_flip() +
  scale_y_continuous(labels=gcomma)

```

```

## Warning: Factor `DIRETORIA` contains implicit NA, consider using
## `forcats::fct_explicit_na`

```



```

#dev.off()

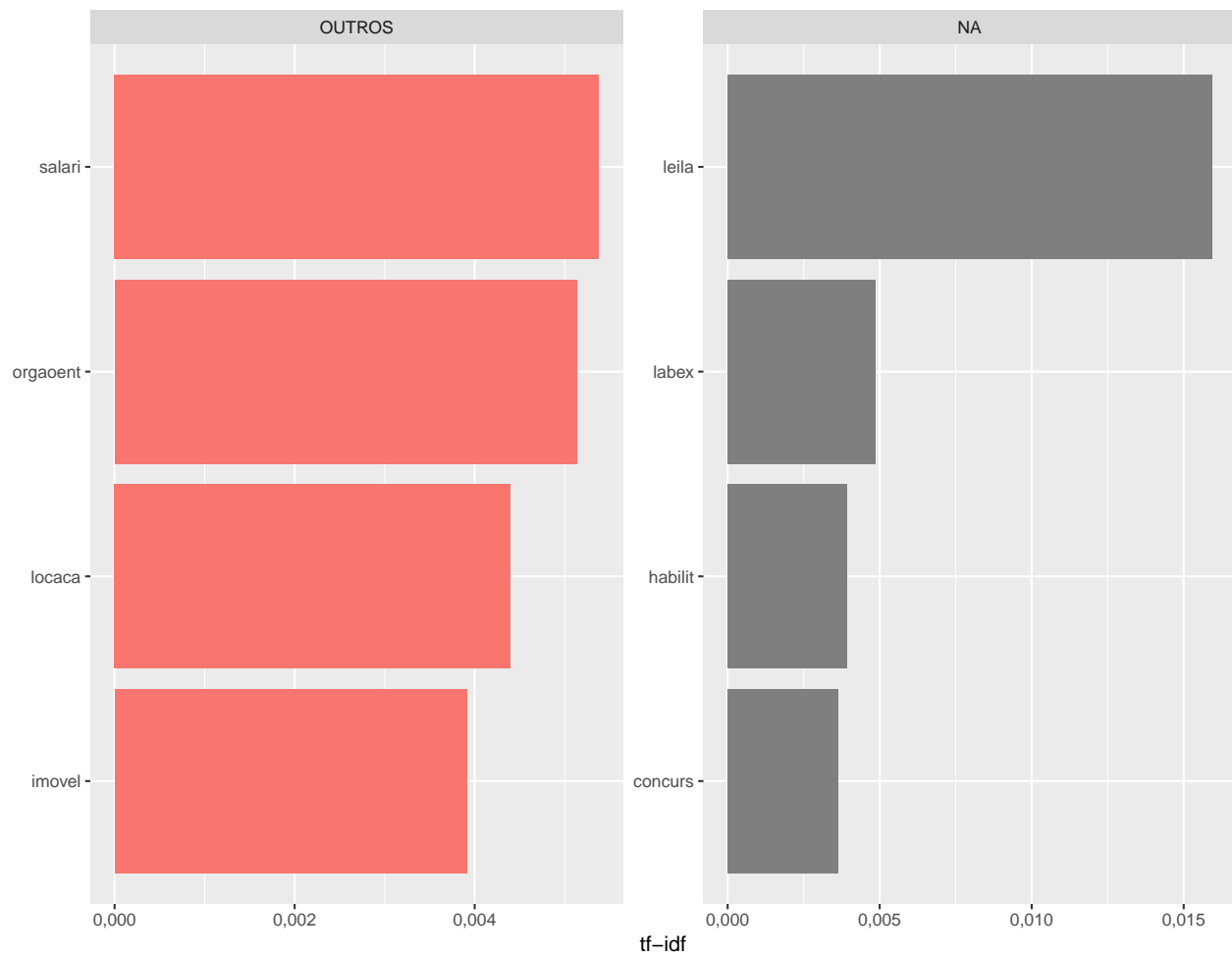
```

Figura5: Termos mais relevantes por diretoria pela estatística `tf_idf`, após stemming e sem stop words

Sim, a remoção de **stop words** não alterou em nada a ordem das 4 palavras mais relevantes de acordo com a estatística.

```
#diretoria_palavras_noSTOP_noSTOP
plot_diretoria_palavras_noSTOP <- diretoria_palavras_noSTOP %>%
  bind_tf_idf(palavra, DIRETORIA, n) %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(palavra, levels = rev(unique(palavra)))) %>%
  mutate(DIRETORIA = factor(DIRETORIA, levels = c("DEA", "DEE", "OUTROS")))
#plot_diretoria_palavras_noSTOP
#windows.options(width=10, height=10)
#jpeg("03_freq_palavras_dir_nostop.jpeg")
plot_diretoria_palavras_noSTOP %>%
  group_by(DIRETORIA) %>%
  top_n(4, tf_idf) %>%
  ungroup() %>%
  mutate(palavra = reorder(palavra, tf_idf)) %>%
  ggplot(aes(palavra, tf_idf, fill = DIRETORIA)) +
  geom_col(show.legend = FALSE) +
  labs(x = NULL, y = "tf-idf") +
  facet_wrap(~DIRETORIA, ncol = 2, scales = "free") +
  coord_flip() +
  scale_y_continuous(labels=gcomma)
```

```
## Warning: Factor `DIRETORIA` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```



```
#dev.off()
```

Wordcloud2 - DEE

```
set.seed(6423)
plot_diretoria_palavras_noSTOP %>%
  filter(DIRETORIA == "DEE") %>%
  select(word = palavra, freq = tf_idf) %>%
  mutate(word = as.factor(word)) %>%
  #top_n(150, freq) %>%
  as.data.frame() %>%
  wordcloud2(shuffle = TRUE, color = "random-dark", shape = "circle", size = 1.10)
```



Wordcloud2 - DEA

```
set.seed(6423)
plot_diretoria_palavras_noSTOP %>%
  filter(DIRETORIA == "DEA") %>%
  select(word = palavra, freq = tf_idf) %>%
  mutate(word = as.factor(word)) %>%
  #top_n(150, freq) %>%
  as.data.frame() %>%
  wordcloud2(shuffle = TRUE, color = "random-dark", shape = "circle", size = .25)
```




```
set.seed(6423)
plot_diretoria_palavras_noSTOP %>%
  filter(DIRETORIA == "OUTROS") %>%
  select(word = palavra, freq = tf_idf) %>%
  mutate(word = as.factor(word)) %>%
  #top_n(150, freq) %>%
  as.data.frame() %>%
  wordcloud2(shuffle = TRUE, color = "random-dark", shape = "circle", size = 0.35)
```



Vamos agora comparar a frequência de palavras entre diretorias. Antes disso, vamos criar documentos de texto no formato tidy separadamente para cada uma das 3 categorias: *DEA*, *DEE* e *OUTROS*.

```
{r child = '032_textminingpart2.Rmd'}
```

Preparação e partição de dados

Vamos, portanto, contar o número de termos únicos dentro de cada diretoria.

32


```
key_DIR = plot_diretoria_palavras_noSTOP %>%
  group_by(DIRETORIA) %>%
  count(DIRETORIA)

## Warning: Factor `DIRETORIA` contains implicit NA, consider using
## `forcats::fct_explicit_na`

## Warning: Factor `DIRETORIA` contains implicit NA, consider using
## `forcats::fct_explicit_na`

## Warning: Factor `DIRETORIA` contains implicit NA, consider using
## `forcats::fct_explicit_na`

key_DIR %>%
  kable("latex", caption = "Número de termos por diretoria",
        booktabs = T, format.args = list(decimal.mark = ',', big.mark = ".")) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 13: Número de termos por diretoria

DIRETORIA	n
OUTROS	1.400
NA	3.728

Vamos, agora, selecionar as $n = 250$ palavras mais importantes de cada uma das 4 diretorias e da categoria 'OUTROS'. Para isso vamos, primeiro, separar os documentos em 3 documentos distintos, um para cada diretoria.

```
n=25
#n=500
#n=500

termos_dir_DEE =
plot_diretoria_palavras_noSTOP %>%
filter(DIRETORIA == "DEE")
termos_DEE = termos_dir_DEE %>%
  top_n(n, tf_idf)

termos_dir_DEA =
plot_diretoria_palavras_noSTOP %>%
filter(DIRETORIA == "DEA")
termos_DEA = termos_dir_DEA %>%
  top_n(n, tf_idf)

termos_dir_OUTROS =
plot_diretoria_palavras_noSTOP %>%
filter(DIRETORIA == "OUTROS")
termos_OUTROS = termos_dir_OUTROS %>%
  top_n(n*3, tf_idf)

termos_dir = bind_rows(mutate(termos_DEE, DIRETORIA = "DEE"),
                       mutate(termos_DEA, DIRETORIA = "DEA"),
                       mutate(termos_OUTROS, DIRETORIA = "OUTROS")) %>%
  select(palavra) %>%
```

```

unique()

gg <- termos_dir$palavra
gg <- unique(gg)
fe <- matrix(data = 0, nrow = length(DB$PEDID01), ncol = length(gg))
fe <- data.frame(fe); colnames(fe) <- gg
i=j=0
for(i in 1:length(DB$Protocolo)){
  for(j in 1:length(gg)){
    g <- grepl(gg[j], DB$PEDID01[i])
    if(g == TRUE){
      fe[i, j] <- 1
    }
  }
}

#sum(rowSums(fe))
dim(fe)

## [1] 624 99

#colSums(fe)
cat(paste0("Existem ", dim(fe)[2], " termos/palavras-chaves únicas na matriz em questão."))

## Existem 99 termos/palavras-chaves únicas na matriz em questão.

NumTermos = as_tibble(rbind(apply(fe,2,sum)))
NumTermos = gather(NumTermos, key = "termo", value = "Num_Pedidos")
NumTermos = NumTermos[order(NumTermos$Num_Pedidos, decreasing = TRUE), ]
#View(colSums(fe))
#View(colnames(fe))

Vamos excluir alguns termos com pouca frequência (abaixo de 8) (abaixo do terceiro quartil)

summary(colSums(fe))

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00   3.00   5.00  11.77   9.00  310.00

#removing unnecessary terms
exclui_termos <- as.character(c())
cbind(Termos = colnames(fe), Freq_Termos = colSums(fe))

##      Termos      Freq_Termos
## salari    "salari"      "10"
## orgaoent  "orgaoent"    "8"
## locaca    "locaca"      "10"
## imovel    "imovel"      "4"
## pessoal   "pessoal"     "9"
## carr      "carr"        "6"
## prestaca  "prestaca"    "8"
## empreg    "empreg"     "18"
## it        "it"         "310"
## gasodut   "gasodut"     "7"
## reajust   "reajust"     "3"
## func      "func"       "41"
## control   "control"     "5"

```

## gratificaca	"gratificaca"	"3"
## seleca	"seleca"	"3"
## anal	"anal"	"60"
## audi	"audi"	"7"
## contrataco	"contrataco"	"5"
## direit	"direit"	"6"
## jurid	"jurid"	"5"
## licitaco	"licitaco"	"4"
## loc	"loc"	"45"
## ocup	"ocup"	"8"
## ouvid	"ouvid"	"3"
## sed	"sed"	"11"
## segur	"segur"	"7"
## concurs	"concurs"	"24"
## esport	"esport"	"4"
## licitaca	"licitaca"	"9"
## quadr	"quadr"	"14"
## vag	"vag"	"12"
## veicul	"veicul"	"8"
## adit	"adit"	"6"
## cronolog	"cronolog"	"2"
## demissa	"demissa"	"4"
## desp	"desp"	"11"
## exclus	"exclus"	"3"
## fiscalizaca	"fiscalizaca"	"2"
## instruca	"instruca"	"3"
## organizaca	"organizaca"	"3"
## quantit	"quantit"	"4"
## regr	"regr"	"5"
## retir	"retir"	"3"
## terceir	"terceir"	"9"
## terceirizaca	"terceirizaca"	"3"
## patrocini	"patrocini"	"4"
## projeca	"projeca"	"10"
## administr	"administr"	"28"
## prec	"prec"	"91"
## objet	"objet"	"9"
## aluguel	"aluguel"	"2"
## conarq	"conarq"	"1"
## confianc	"confianc"	"2"
## estagia	"estagia"	"2"
## etanol	"etanol"	"3"
## funco	"funco"	"5"
## ministr	"ministr"	"31"
## pad	"pad"	"13"
## quaisqu	"quaisqu"	"4"
## visual	"visual"	"4"
## pass	"pass"	"17"
## aloc	"aloc"	"6"
## firm	"firm"	"19"
## fix	"fix"	"8"
## incen	"incen"	"6"
## priv	"priv"	"7"
## simil	"simil"	"9"

```
## gesta      "gesta"      "16"
## acuc       "acuc"       "3"
## comunicaca "comunicaca" "7"
## fiscal    "fiscal"     "9"
## parec     "parec"      "15"
## apostil   "apostil"    "4"
## aven      "aven"       "2"
## cole      "cole"       "13"
## complement "complement" "3"
## conform   "conform"    "3"
## constituc "constituc"  "4"
## continu   "continu"    "5"
## convenco  "convenco"   "2"
## corporat  "corporat"   "2"
## curricul  "curricul"   "3"
## desinfecca "desinfecca" "2"
## empresar  "empresar"   "2"
## ergonom   "ergonom"    "2"
## exploraca "exploraca"  "3"
## farmac    "farmac"     "1"
## folh      "folh"       "3"
## instaur   "instaur"    "4"
## logis     "logis"      "2"
## mao       "mao"        "4"
## natalin   "natalin"    "1"
## salar     "salar"      "10"
## sediment  "sediment"   "3"
## seleco    "seleco"     "1"
## temporal  "temporal"   "1"
## tesour    "tesour"     "3"
## vig       "vig"        "12"
## word      "word"       "4"
```

```
z=0
for (k in 1:dim(fe)[2]) {
  if (colSums(fe)[k] <= 7) {
    exclui_termos[z] <- colnames(fe)[k]
    z = z+1
  }
}
```

```
#length(exclui_termos) # [1] 409 [1] 2118
```

```
cat(paste0("Existem ", length(exclui_termos), " termos com freq. menor ou igual a 7. Logo, se remov
```

```
## Existem 63 termos com freq. menor ou igual a 7. Logo, se removermos estes o número de variáveis (term
```

```
fe <- fe %>% select(-exclui_termos)
```

```
cat(paste0("Existem, agora, ", dim(fe)[2], " termos/palavras-chaves únicas."))
```

```
## Existem, agora, 36 termos/palavras-chaves únicas.
```

Critério de escolha dos termos, se a frequência for maior ou igual a 10

IMPLEMENTAR

```
highchart() %>%
  hc_add_series(data = NumTermos$Num_Pedidos,
    type = "bar",
    name = "# de pedidos",
    showInLegend = FALSE,
    tooltip = list(valueDecimals = 0, valuePrefix = "", valueSuffix = ""), color="blue") %>%
  hc_yAxis(title = list(text = "Quantitativo de pedidos"),
    allowDecimals = TRUE, max = (max(NumTermos$Num_Pedidos)+103),
    labels = list(format = "{value}")) %>%
  hc_xAxis(title = list(text = "Termo"),
    categories = NumTermos$termo,
    tickmarkPlacement = "on",
    opposite = FALSE) %>%
  hc_title(text = "Quantitativo de pedidos por termo (sem exclusividade)",
    style = list(fontWeight = "bold")) %>%
  hc_subtitle(text = paste("")) %>%
  hc_tooltip(valueDecimals = 2,
    pointFormat = "{point.y} pedidos") %>%
  #pointFormat = "Variável: {point.x} <br> Missing: {point.y}"
  hc_credits(enabled = TRUE,
    text = "Fonte: CGU, e-SIC (2019). Elaboração: Ewerson Pimenta.",
    style = list(fontSize = "10px")) %>%
  hc_exporting(enabled = TRUE, filename = "F3-filmes-genero-Pimenta")
```

```
db_modelo0 = as_tibble(cbind(select(DB,Protocolo, DATA_REGISTRO, DIRETORIAS, DIRETORIA),fe))
db_modelo = as_tibble(cbind(select(DB,DIRETORIA),fe))
```

```
# __Porcentagem de ZEROS por variável__
```

```
zeros <- (colSums(fe==0)/nrow(fe)*100); var <- names(fe)
db_zero <- data.frame(var,zeros); rownames(db_zero) <- NULL
db_zero <- db_zero[order(db_zero$zeros, decreasing = TRUE), ]
```

```
hc4_1 <- highchart() %>%
  hc_add_series(data = db_zero$zeros,
    type = "bar",
    name = "Porcentagem de zeros",
    showInLegend = FALSE,
    tooltip = list(valueDecimals = 2, valuePrefix = "", valueSuffix = " %"), color="pink") %>%
  hc_yAxis(title = list(text = "Porcentagem de zero"),
    allowDecimals = TRUE, max = 100,
    labels = list(format = "{value}%")) %>%
  hc_xAxis(categories = db_zero$var,
    tickmarkPlacement = "on",
    opposite = FALSE) %>%
  hc_title(text = "Porcentagem de zeros por variável",
    style = list(fontWeight = "bold")) %>%
  hc_subtitle(text = paste("")) %>%
  hc_tooltip(valueDecimals = 2,
    pointFormat = "Zeros: {point.y}") %>%
  #pointFormat = "Variável: {point.x} <br> Missing: {point.y}"
```

```

    hc_credits(enabled = TRUE,
              text = "Fonte: IMDB/KAGGLE. Elaboração: Ewerson Pimenta.",
              style = list(fontSize = "10px")) %>%
  hc_exporting(enabled = TRUE, filename = "Fig00-Pimenta")
#hc <- hc %>%
# hc_add_theme(hc_theme_darkunica())
hc4_1; remove(hc4_1, var, zeros)

```

Modelos de classificação

Partição dos dados

Particionando a base de dados em Treino e Teste, esses dois (Treino e Teste) também terão armazenados as diretorias que foram responsáveis por cada pedido via amostragem probabilística dos dados originais separadamente das bases de Treino e Teste.

```

#db_modelo = as_tibble(cbind(select(DB,DIRETORIA),fe))
#getwd()
#setwd("/Users/ewersonpimenta/Desktop/ESIC_TCC/TCC_v2.1/RMARKDOWN/WEB_APP/")
#write.csv(db_modelo0, file = "db_modelo_rf_v21.csv", row.names = FALSE)
#db_modelo = read.csv("db_modelo_rf_v10.csv",header = T); dim(db_modelo)
#db_modelo = db_modelo %>% select(-r,-venc)
#write.csv(db_modelo, file = "db_modelo_rf_v11.csv", row.names = FALSE)

```

Para amostragem aleatória simples

```

set.seed(098798) # 756446 ou 75452 (OOB_erro: 35,63% ACC: 65,44%) # 2967 (OOB_erro: 34,15% ACC: 64,98%)
intrain <- createDataPartition(y = db_modelo$DIRETORIA, p = 0.65, list = FALSE)
training <- db_modelo[intrain,]
testing <- db_modelo[-intrain,]

```

Modelagem 1 - Random Forest (RF)

Random Forest (RF) - Metodologia

- Descrição**
1. Random Forest foi desenvolvido para agregar árvores de decisão (modelo de classificação);
 2. Pode ser usado para modelo de classificação (p/ var. resposta categórica) ou regressão (no caso de haver variável resposta contínua);
 3. Evita *overfitting*;
 4. Permite trabalhar com um largo número de características de um conjunto de dados;
 5. Auxilia na seleção de variáveis baseada em um algoritmo que calcula a importância por variável (assim, tendo conhecimento de quais variáveis são mais importantes, podemos usar essa informação para outros modelos de classificação);
 6. User-friendly: apenas 2 parâmetros livres:

- Trees - *ntrees*, default 500 (Nº de árvores);
- Variáveis selecionadas via amostragem aleatória candidatas à cada “split” (quebra da árvore) - *mtry*, default \sqrt{p} p/ classificação e $\frac{p}{3}$ p/ regressão (p: nº de features/variáveis);

Passo-a-Passo

É realizado em 3 passos:

1. Desenha as amostras via bootstrap do número de árvores *ntrees*;

2. Para cada amostra via bootstrap, cresce o número de árvores “un-puned” para a escolha da melhor quebra da árvore baseado na amostra aleatória do valor predito de mtry a cada nó da árvore;
- 3. Faz classificação de novos valores usando a maioria de votos p/ classificação e usa a média p/ regressão baseada nas amostras de ntrees.

Random Forest - Aplicação e Resultados

Inicialmente utilizaremos o pacote `randomForest` que implementa o algoritmo de Random Forest de Breiman (baseado na clusterização de Breiman, originalmente codificada em Fortran) que tem por finalidade classificar e/ou criar regressão. Além disso, pode ser usado em um modelo não supervisionado para avaliar proximidades entre pontos.

Estamos usando, a partir daqui, a base de treino.

```
#library(randomForest)
#library(rpart)
#library(rpart.plot)
#rf <- randomForest(proximity = T, ntree = 38, do.trace = T, WR~., data=training)
set.seed(9984512)
# Training with classification tree
rf <- rpart(DIRETORIA ~ ., data=training, method="class", xval = 4, )
print(rf, digits = 3)
```

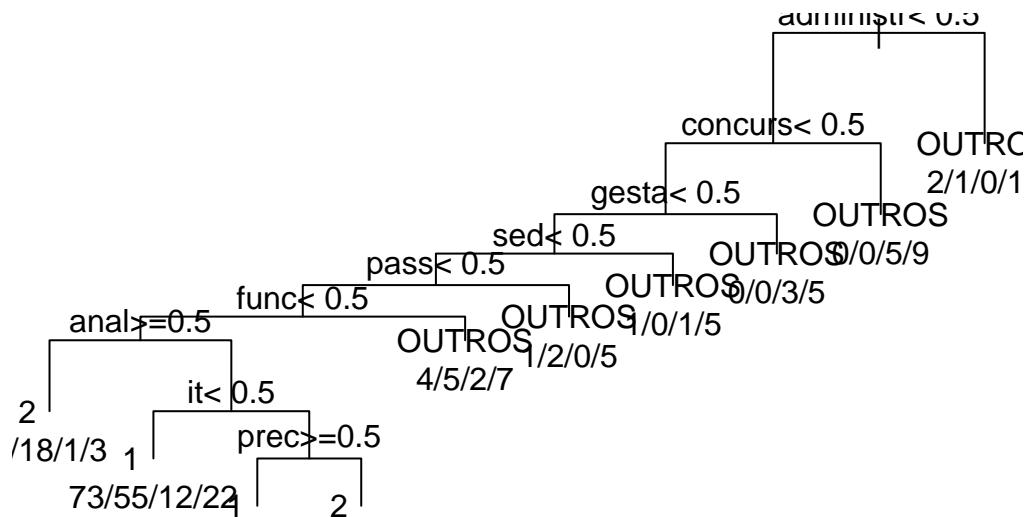
```
## n= 408
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
##      1) root 408 268 1 (0.3431 0.3358 0.0907 0.2304)
##      2) administr< 0.5 389 251 1 (0.3548 0.3496 0.0951 0.2005)
##      4) concurs< 0.5 375 237 1 (0.3680 0.3627 0.0853 0.1840)
##      8) gesta< 0.5 367 229 1 (0.3760 0.3706 0.0790 0.1744)
##     16) sed< 0.5 360 223 1 (0.3806 0.3778 0.0778 0.1639)
##     32) pass< 0.5 352 216 1 (0.3864 0.3807 0.0795 0.1534)
##     64) func< 0.5 334 202 1 (0.3952 0.3862 0.0778 0.1407)
##    128) anal>=0.5 31 13 2 (0.2903 0.5806 0.0323 0.0968) *
##    129) anal< 0.5 303 180 1 (0.4059 0.3663 0.0825 0.1452)
##    258) it< 0.5 162 89 1 (0.4506 0.3395 0.0741 0.1358) *
##    259) it>=0.5 141 85 2 (0.3546 0.3972 0.0922 0.1560)
##    518) prec>=0.5 21 10 1 (0.5238 0.2381 0.1429 0.0952) *
##    519) prec< 0.5 120 69 2 (0.3250 0.4250 0.0833 0.1667) *
##    65) func>=0.5 18 11 OUTROS (0.2222 0.2778 0.1111 0.3889) *
##    33) pass>=0.5 8 3 OUTROS (0.1250 0.2500 0.0000 0.6250) *
##    17) sed>=0.5 7 2 OUTROS (0.1429 0.0000 0.1429 0.7143) *
##     9) gesta>=0.5 8 3 OUTROS (0.0000 0.0000 0.3750 0.6250) *
##     5) concurs>=0.5 14 5 OUTROS (0.0000 0.0000 0.3571 0.6429) *
##     3) administr>=0.5 19 3 OUTROS (0.1053 0.0526 0.0000 0.8421) *
```

```
attributes(rf)
```

```
## $names
## [1] "frame"          "where"          "call"
## [4] "terms"          "cptable"        "method"
## [7] "parms"          "control"        "functions"
```

```
## [10] "numresp"          "splits"            "variable.importance"
## [13] "y"                "ordered"
##
## $xlevels
## named list()
##
## $ylevels
## [1] "1"      "2"      "5"      "OUTROS"
##
## $class
## [1] "rpart"
```

```
plot(rf)
text(rf, use.n = TRUE)
```



```
# Predict the testing set with the trained model
predictions <- predict(rf, testing, type = "class")

# Accuracy and other metrics
confusionMatrix(predictions, as.factor(testing$DIRETORIA))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  5 OUTROS
##      1      44 42  8     19
##      2      22 28  6     13
##      5       0  0  0       0
##     OUTROS   8  3  5     18
##
## Overall Statistics
##
##           Accuracy : 0.4167
##           95% CI : (0.3502, 0.4855)
##      No Information Rate : 0.3426
##      P-Value [Acc > NIR] : 0.01401
##
##           Kappa : 0.1376
##
```



```
## McNemar's Test P-Value : 2.78e-06
##
## Statistics by Class:
##
##           Class: 1 Class: 2 Class: 5 Class: OUTROS
## Sensitivity      0.5946   0.3836   0.00000   0.36000
## Specificity      0.5141   0.7133   1.00000   0.90361
## Pos Pred Value   0.3894   0.4058   NaN       0.52941
## Neg Pred Value   0.7087   0.6939   0.91204   0.82418
## Prevalence       0.3426   0.3380   0.08796   0.23148
## Detection Rate   0.2037   0.1296   0.00000   0.08333
## Detection Prevalence 0.5231  0.3194  0.00000   0.15741
## Balanced Accuracy 0.5543   0.5484   0.50000   0.63181
```

Olhando as 6 primeiras observações real X predito

```
p1 <- predict(rf,training)
head(p1)
```

```
##           1           2           5      OUTROS
## 1 0.4506173 0.33950617 0.07407407 0.1358025
## 2 0.5238095 0.23809524 0.14285714 0.0952381
## 3 0.3250000 0.42500000 0.08333333 0.1666667
## 4 0.5238095 0.23809524 0.14285714 0.0952381
## 5 0.4506173 0.33950617 0.07407407 0.1358025
## 6 0.1052632 0.05263158 0.00000000 0.8421053
```

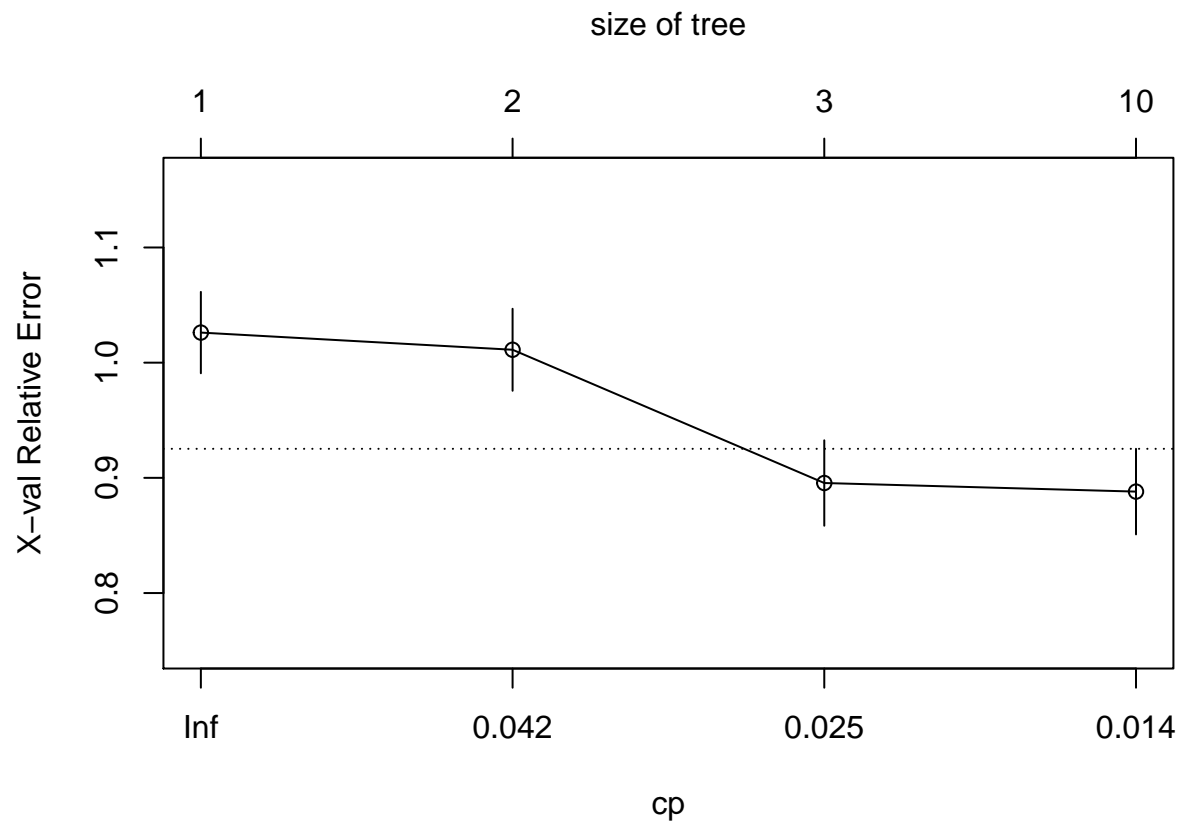
```
head(training$DIRETORIA)
```

```
## [1] "5"      "1"      "2"      "1"      "OUTROS" "1"
```

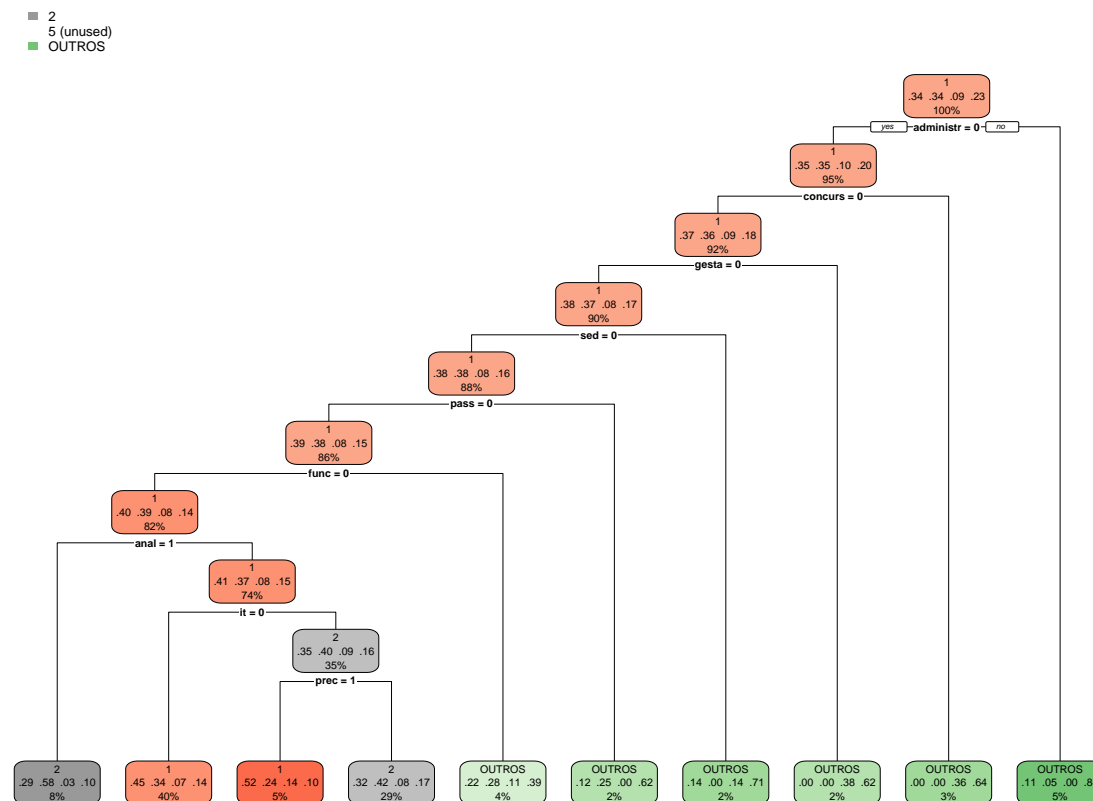
Selecionando uma árvore

```
rp <- rpart::rpart(formula = DIRETORIA~.,data=training)
```

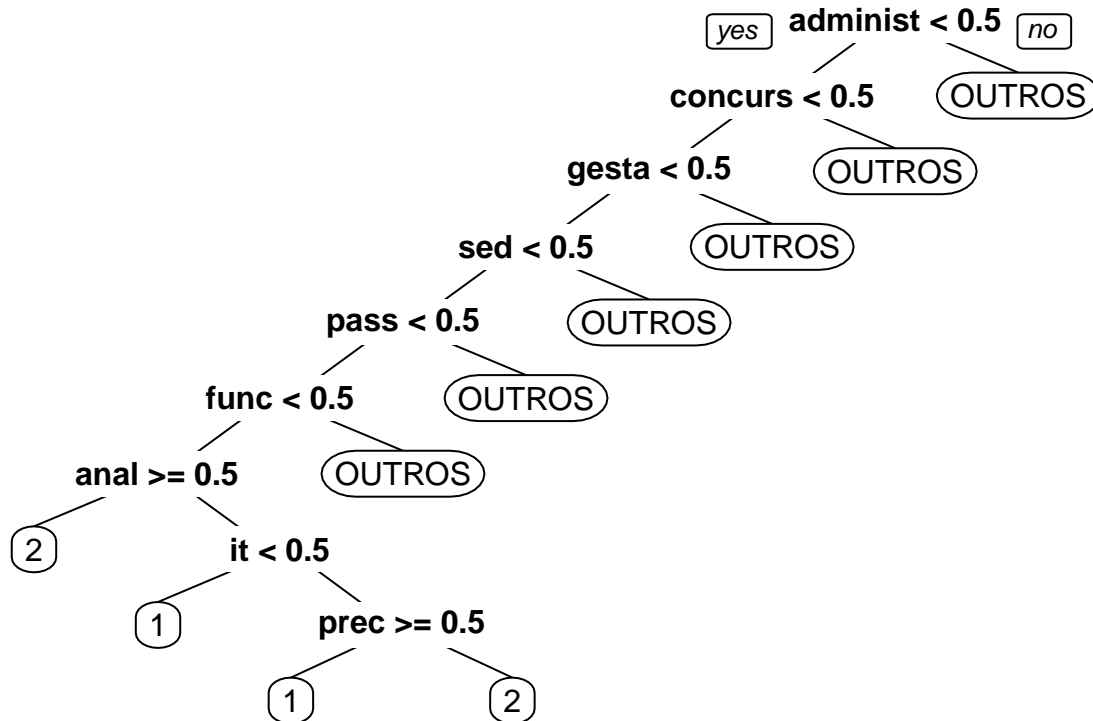
```
rpart::plotcp(rf)
```



`rpart.plot(rf)`



```
rpart.plot.version1(rf)
```



Outra forma de escrever o modelo é usando a função `randomForest()`

```
set.seed(09986755)
rf1 <- randomForest(as.factor(DIRETORIA) ~ ., data=training,
                    importance = TRUE,
                    proximity = TRUE)
rf1
```

```
##
```

```
## Call:
```

```
## randomForest(formula = as.factor(DIRETORIA) ~ ., data = training, importance = TRUE, proximity
```

```
## Type of random forest: classification
```

```
## Number of trees: 500
```

```
## No. of variables tried at each split: 6
```

```
##
```

```
## OOB estimate of error rate: 51.96%
```

```
## Confusion matrix:
```

```
##      1  2  5 OUTROS class.error
## 1      82 50  0      8  0.4142857
## 2      64 68  0      5  0.5036496
## 5      16 11  2      8  0.9459459
## OUTROS 25 20  5     44  0.5319149
```

```
# Predict the testing set with the trained model
```

```
predictions1 <- predict(rf1, testing, type = "class")
```

```
# Accuracy and other metrics
```

```
confusionMatrix(predictions1, as.factor(testing$DIRETORIA))
```

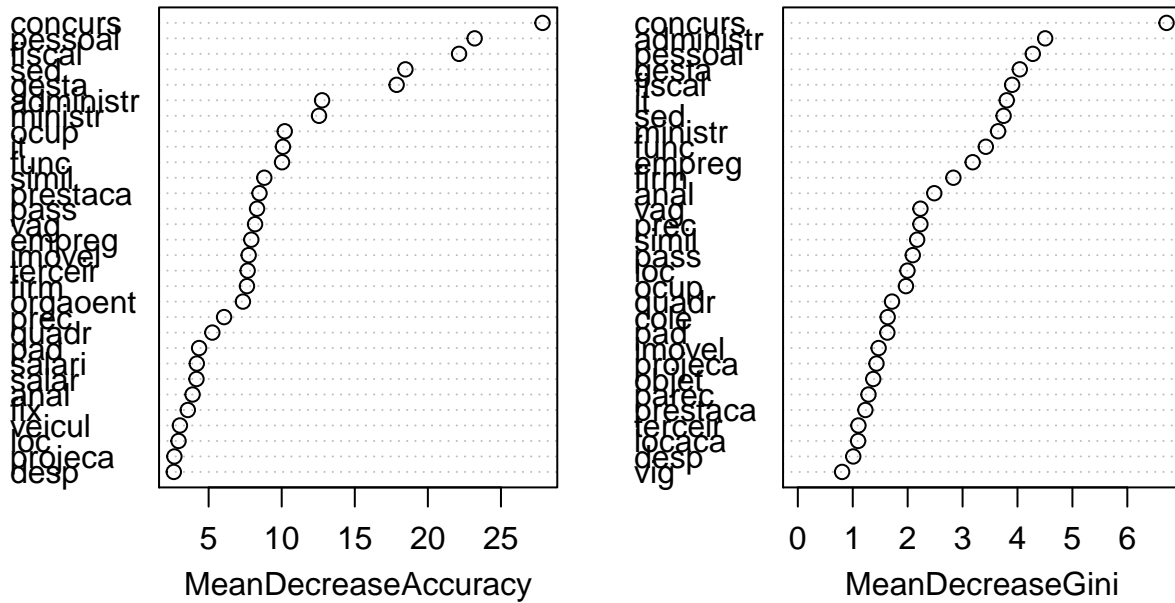
```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction  1  2  5 OUTROS
##      1      45 40  8    20
##      2      25 33  6     9
##      5       2  0  0     1
##      OUTROS  2  0  5    20
##
## Overall Statistics
##
##           Accuracy : 0.4537
##           95% CI : (0.386, 0.5227)
##      No Information Rate : 0.3426
##      P-Value [Acc > NIR] : 0.0004699
##
##           Kappa : 0.1923
##
## McNemar's Test P-Value : 5.827e-07
##
## Statistics by Class:
##
##           Class: 1 Class: 2 Class: 5 Class: OUTROS
## Sensitivity      0.6081  0.4521  0.00000  0.40000
## Specificity      0.5211  0.7203  0.98477  0.95783
## Pos Pred Value   0.3982  0.4521  0.00000  0.74074
## Neg Pred Value   0.7184  0.7203  0.91080  0.84127
## Prevalence       0.3426  0.3380  0.08796  0.23148
## Detection Rate   0.2083  0.1528  0.00000  0.09259
## Detection Prevalence 0.5231  0.3380  0.01389  0.12500
## Balanced Accuracy 0.5646  0.5862  0.49239  0.67892
```

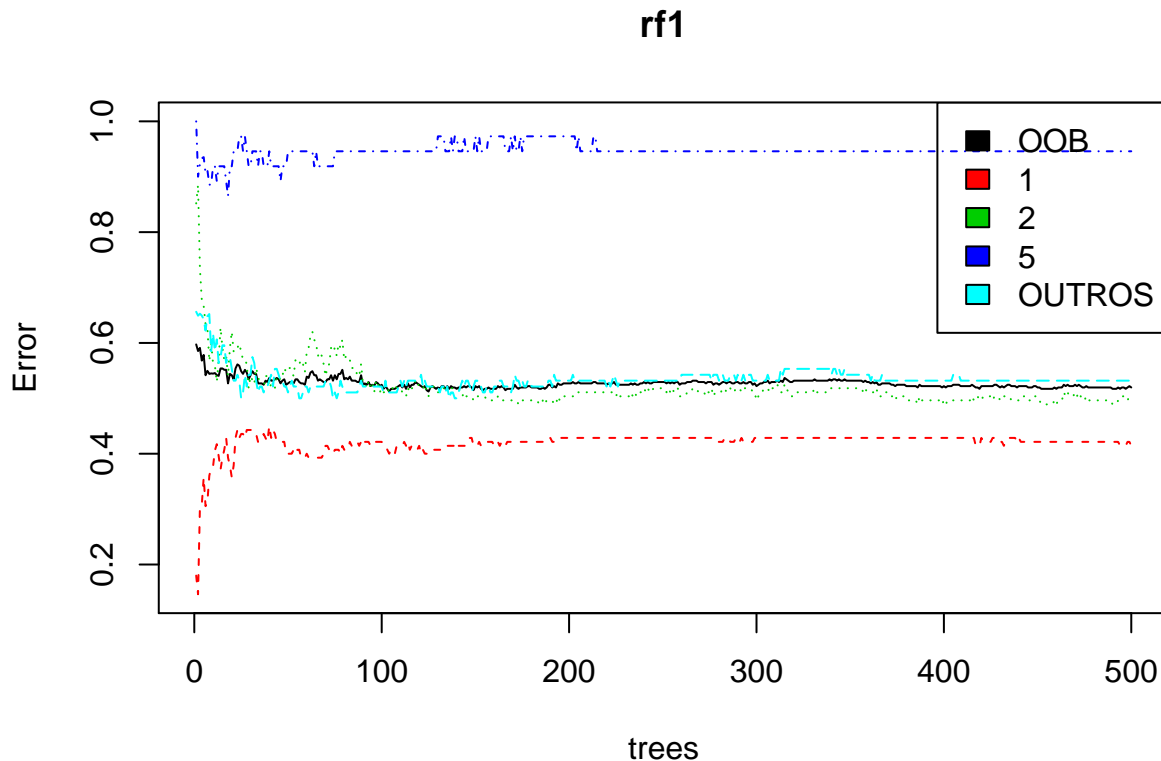
Importância de variáveis

```
RF_importance = randomForest::importance(rf1)[order(randomForest::importance(rf1)[,1], decreasing = TRUE)]
randomForest::varImpPlot(rf1)
```

rf1



```
plot(rf1)
legend('topright', colnames(rf1$err.rate), col=1:5, fill=1:5)
```



A partir de $n = 420$ árvores a taxa do erro **OOB** (Out of Bag) tende a estabilizar.

Tuning do modelo

Fixando, então, $n = 420$ árvores

Aparentemente $mtry = 26$ parece ser um bom palpite para o segundo parâmetro do random forest, uma vez que esse retornou menor taxa de erro **OOB**, 27,03%. Entretanto esse erro ainda é muito alto. Vamos reescrever o modelo com os parâmetros tunados.

```
set.seed(09986755)
rf2 <- randomForest(as.factor(DIRETORIA) ~ ., data=training,
                    ntree = 420,
                    mtry = 26,
                    importance = TRUE,
                    proximity = TRUE)
rf2
```

```
##
## Call:
## randomForest(formula = as.factor(DIRETORIA) ~ ., data = training,      ntree = 420, mtry = 26, impo
##               Type of random forest: classification
##               Number of trees: 420
## No. of variables tried at each split: 26
##
##       OOB estimate of  error rate: 53.92%
## Confusion matrix:
##      1  2  5 OUTROS class.error
## 1    79  54  1      6  0.4357143
## 2    66  68  0      3  0.5036496
## 5    15  12  5      5  0.8648649
```

```

## OUTROS 27 23 8      36    0.6170213
# Predict the testing set with the trained model
predictions2 <- predict(rf2, testing, type = "class")

# Accuracy and other metrics
(rf2_CONFUSIONM = confusionMatrix(predictions2, as.factor(testing$DIRETORIA)))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  5 OUTROS
##      1      38 35  8      18
##      2      32 36  7      12
##      5       2  0  2       1
##      OUTROS  2  2  2      19
##
## Overall Statistics
##
##           Accuracy : 0.4398
##           95% CI : (0.3725, 0.5088)
##      No Information Rate : 0.3426
##      P-Value [Acc > NIR] : 0.001911
##
##           Kappa : 0.1738
##
##      McNemar's Test P-Value : 2.523e-05
##
## Statistics by Class:
##
##           Class: 1 Class: 2 Class: 5 Class: OUTROS
## Sensitivity      0.5135   0.4932 0.105263      0.38000
## Specificity      0.5704   0.6434 0.984772      0.96386
## Pos Pred Value   0.3838   0.4138 0.400000      0.76000
## Neg Pred Value   0.6923   0.7132 0.919431      0.83770
## Prevalence       0.3426   0.3380 0.087963      0.23148
## Detection Rate   0.1759   0.1667 0.009259      0.08796
## Detection Prevalence 0.4583 0.4028 0.023148      0.11574
## Balanced Accuracy 0.5420   0.5683 0.545017      0.67193

p2 <- predict(rf2,training)
as.character(head(p2))

## [1] "1" "1" "2" "1" "1" "1"

head(training$DIRETORIA)

## [1] "5"      "1"      "2"      "1"      "OUTROS" "1"

(DEA_erroCLASS = sum(rf2_CONFUSIONM$table[1,2:3])/ sum(rf2_CONFUSIONM$table[1,]))

## [1] 0.4343434

(DEE_erroCLASS = sum(rf2_CONFUSIONM$table[2,c(1,3)])/ sum(rf2_CONFUSIONM$table[2,]))

## [1] 0.4482759

```

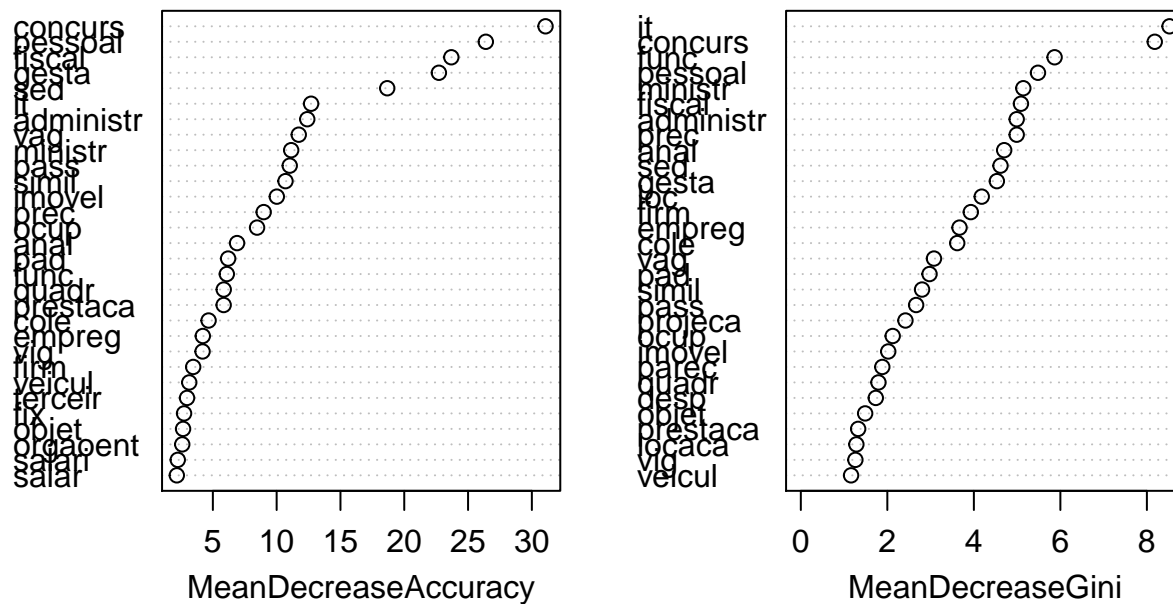
```
(OUTROS_erroCLASS = sum(rf2_CONFUSIONM$table[3,1:2]) / sum(rf2_CONFUSIONM$table[3,]))
```

```
## [1] 0.4
```

Acurácia de aproximadamente 72% na base de teste. E as taxas de erro de classificação foram 30%, 44% e 45% para *DEA*, *DEE* e *OUTROS*, respectivamente. Houve um melhor desempenho na classificação do modelo para a categoria *DEA*

```
RF_importance = randomForest::importance(rf2)[order(randomForest::importance(rf2)[,1], decreasing = TRUE)]
randomForest::varImpPlot(rf2)
```

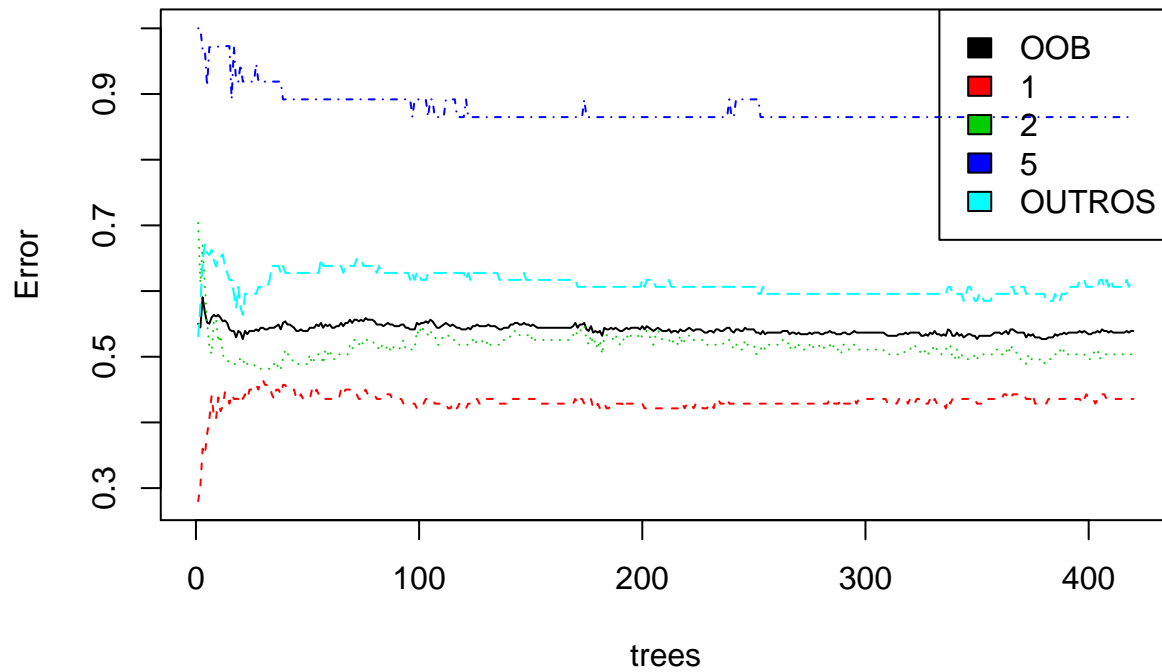
rf2



Taxa de Erro Random Forest

```
plot(rf2, main = "Taxa de erro OOB - Out of Bag")
legend('topright', colnames(rf2$err.rate), col=1:5, fill=1:5)
```

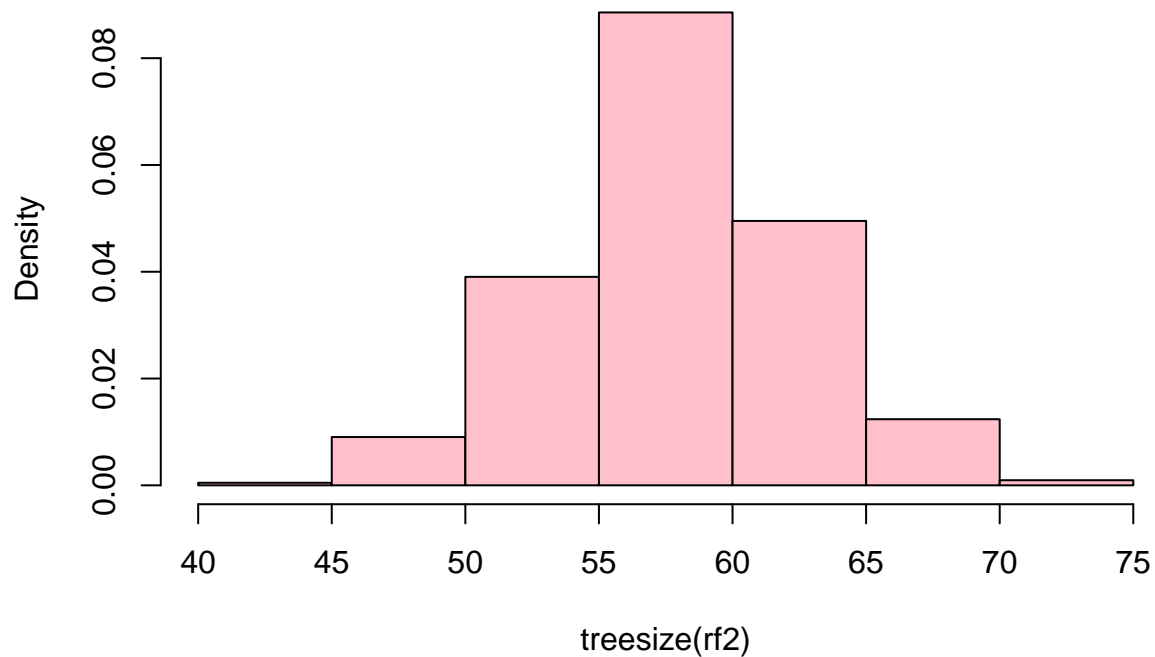

Taxa de erro OOB – Out of Bag



Histograma do Número de nós por árvore

```
hist(treesize(rf2), probability = T,  
     main = "Distribuição do nº de nós por árvore",  
     col = "pink")
```

Distribuição do nº de nós por árvore



Vamos excluir as variáveis que não retornaram valor de importância para o algoritmo do random forest.

```
RF_importance = randomForest::importance(rf2)[order(randomForest::importance(rf2)[,1], decreasing = TRUE),]
```

```
RF1 = data.frame(variables = rownames(RF_importance), importance = RF_importance[,4])
```

```
RF1 = RF1[order(RF1$importance, decreasing = TRUE),]
```

```
rownames(RF1) <- NULL
```

```
#summary(RF1)
```

```
RF2 = RF1[1:20,]
```

```
#library("highcharter")
```

```
hc6_1 <- highchart() %>%
```

```
  hc_add_series(data = RF2$importance,
```

```
                type = "bar",
```

```
                name = "Importância",
```

```
                showInLegend = FALSE,
```

```
                tooltip = list(valueDecimals = 2, valuePrefix = "", valueSuffix = "")) %>%
```

```
  hc_yAxis(title = list(text = "Importância"),
```

```
            allowDecimals = TRUE, max = 12,
```

```
            labels = list(format = "{value}")) %>%
```

```
  hc_xAxis(title = list(text = "Fatores"),
```

```
            categories = RF2$variables,
```

```
            tickmarkPlacement = "on",
```

```
            opposite = FALSE) %>%
```

```
  hc_title(text = "Importância por fator - Random Forest",
```

```
            style = list(fontWeight = "bold")) %>%
```

```
  hc_subtitle(text = paste("")) %>%
```

```
    hc_tooltip(valueDecimals = 2,
```

```
               pointFormat = "Importância: {point.y}") %>%
```

```
               #pointFormat = "Variável: {point.x} <br> Importância: {point.y}")
```

```
    hc_credits(enabled = TRUE,
```

```
               text = "Fonte: CGU, e-SIC. Elaboração: Leal, Alize; Pimenta, Ewerson.",
```

```
               style = list(fontSize = "10px")) %>%
```

```
    hc_exporting(enabled = TRUE, filename = "F6_1-importance-Pimenta")
```

```
#hc <- hc %>%
```

```
# hc_add_theme(hc_theme_darkunica())
```

```
hc6_1
```

Vamos excluir todas as variáveis que retornaram importância menor ou igual a zero.

```
variaveis_sem_importancia = RF1 %>% filter(as.character(importance) <= 0)
```

```
#summary(variaveis_sem_importancia)
```

```
variaveis_sem_importancia = as.character(variaveis_sem_importancia$variables)
```

```
training1 = training %>% select(-(variaveis_sem_importancia))
```

```
testing1 = testing %>% select(-(variaveis_sem_importancia))
```

```
db_modelo1 = db_modelo %>% select(-(variaveis_sem_importancia))
```

```
set.seed(756446) #2967
```

```
rf3 <- randomForest(as.factor(DIRETORIA) ~ ., data=training1,
```

```
                    ntree = 420,
```

```
                    mtry = 26,
```

```
                    importance = TRUE,
```

```
                    proximity = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within
```

```

## valid range
rf3

##
## Call:
## randomForest(formula = as.factor(DIRETORIA) ~ ., data = training1,      ntree = 420, mtry = 26, imp
##           Type of random forest: classification
##           Number of trees: 420
## No. of variables tried at each split: 23
##
##           OOB estimate of  error rate: 53.43%
## Confusion matrix:
##           1  2  5 OUTROS class.error
## 1           76 57  0         7  0.4571429
## 2           63 74  0         0  0.4598540
## 5           13 14  3         7  0.9189189
## OUTROS      26 21 10        37  0.6063830

# Predict the testing set with the trained model
predictions3 <- predict(rf3, testing1[,-1], type = "class")
#predict(rf3, testing1[,-1], type = "prob")

# Accuracy and other metrics
confusionMatrix(predictions3, as.factor(testing1$DIRETORIA))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2  5 OUTROS
##      1           44 35  8      17
##      2           25 38  6      13
##      5              2  0  3       1
##      OUTROS       3  0  2      19
##
## Overall Statistics
##
##           Accuracy : 0.4815
##           95% CI : (0.4132, 0.5503)
##      No Information Rate : 0.3426
##      P-Value [Acc > NIR] : 1.772e-05
##
##           Kappa : 0.2359
##
##      McNemar's Test P-Value : 5.631e-06
##
## Statistics by Class:
##
##           Class: 1 Class: 2 Class: 5 Class: OUTROS
## Sensitivity           0.5946   0.5205  0.15789      0.38000
## Specificity           0.5775   0.6923  0.98477      0.96988
## Pos Pred Value        0.4231   0.4634  0.50000      0.79167
## Neg Pred Value        0.7321   0.7388  0.92381      0.83854
## Prevalence            0.3426   0.3380  0.08796      0.23148
## Detection Rate        0.2037   0.1759  0.01389      0.08796
## Detection Prevalence  0.4815   0.3796  0.02778      0.11111

```

```
## Balanced Accuracy      0.5860   0.6064   0.57133      0.67494
```

```
p3 <- predict(rf3,training1)
as.character(head(p3))
```

```
## [1] "1" "2" "2" "2" "1" "1"
```

```
head(training1$DIRETORIA)
```

```
## [1] "5"      "1"      "2"      "1"      "OUTROS" "1"
```

```
set.seed(756446)
```

```
rf4 <- randomForest(as.factor(DIRETORIA) ~ ., data=db_modelo,
                    ntree = 420,
                    mtry = 45,
                    importance = TRUE,
                    proximity = TRUE)
```

```
## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within
## valid range
```

```
rf4
```

```
##
```

```
## Call:
```

```
## randomForest(formula = as.factor(DIRETORIA) ~ ., data = db_modelo,      ntree = 420, mtry = 45, imp
```

```
##           Type of random forest: classification
```

```
##           Number of trees: 420
```

```
## No. of variables tried at each split: 36
```

```
##
```

```
##           OOB estimate of  error rate: 53.04%
```

```
## Confusion matrix:
```

```
##           1   2  5 OUTROS class.error
```

```
## 1          129  72  4          9   0.3971963
```

```
## 2          104 101  0          5   0.5190476
```

```
## 5           26  17  7          6   0.8750000
```

```
## OUTROS      44  37  7          56   0.6111111
```

```
# Predict the testing set with the trained model
```

```
predictions4 <- predict(rf4, testing, type = "class")
```

```
# Accuracy and other metrics
```

```
confusionMatrix(predictions4, as.factor(testing$DIRETORIA))
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction  1  2  5 OUTROS
```

```
##      1      51 33  8      12
```

```
##      2      22 40  6       9
```

```
##      5       0  0  5       1
```

```
## OUTROS     1  0  0      28
```

```
##
```

```
## Overall Statistics
```

```
##
```

```
##           Accuracy : 0.5741
```

```
##           95% CI : (0.5052, 0.6409)
```

```
## No Information Rate : 0.3426
```

```
##      P-Value [Acc > NIR] : 3.080e-12
##
##              Kappa : 0.3746
##
## McNemar's Test P-Value : 3.435e-06
##
## Statistics by Class:
##
##              Class: 1 Class: 2 Class: 5 Class: OUTROS
## Sensitivity      0.6892   0.5479   0.26316   0.5600
## Specificity      0.6268   0.7413   0.99492   0.9940
## Pos Pred Value   0.4904   0.5195   0.83333   0.9655
## Neg Pred Value   0.7946   0.7626   0.93333   0.8824
## Prevalence       0.3426   0.3380   0.08796   0.2315
## Detection Rate   0.2361   0.1852   0.02315   0.1296
## Detection Prevalence 0.4815   0.3565   0.02778   0.1343
## Balanced Accuracy 0.6580   0.6446   0.62904   0.7770
```

```
p4 <- predict(rf4,training)
as.character(head(p4))
```

```
## [1] "1" "1" "2" "1" "1" "1"
```

```
head(training$DIRETORIA)
```

```
## [1] "5"      "1"      "2"      "1"      "OUTROS" "1"
```

Comparacao do poder de predicao dos modelos treinados e propostos

```
as.character(p1[1:15])
```

```
## [1] "0.450617283950617" "0.523809523809524" "0.325"
## [4] "0.523809523809524" "0.450617283950617" "0.105263157894737"
## [7] "0.450617283950617" "0.523809523809524" "0.450617283950617"
## [10] "0.450617283950617" "0.105263157894737" "0.325"
## [13] "0.325"              "0.325"              "0.450617283950617"
```

```
as.character(rf$y[1:15])
```

```
## [1] "3" "1" "2" "1" "4" "1" "3" "1" "2" "1" "1" "1" "1" "4" "4"
```

```
as.character(p2[1:15])
```

```
## [1] "1"      "1"      "2"      "1"      "1"      "1"      "1"
## [8] "1"      "1"      "1"      "1"      "2"      "2"      "OUTROS"
## [15] "1"
```

```
as.character(p4[1:15])
```

```
## [1] "1"      "1"      "2"      "1"      "1"      "1"      "1"
## [8] "1"      "1"      "1"      "1"      "2"      "2"      "OUTROS"
## [15] "1"
```

```
training$DIRETORIA[1:15]
```

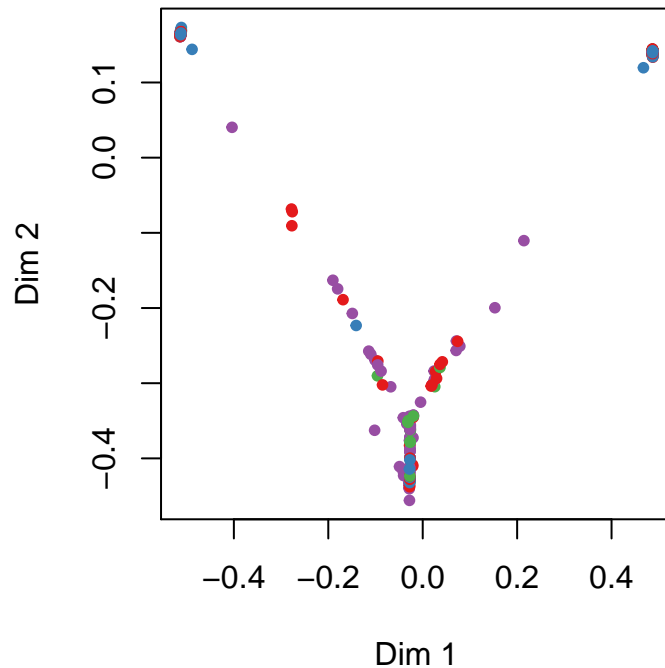
```
## [1] "5"      "1"      "2"      "1"      "OUTROS" "1"      "5"
## [8] "1"      "2"      "1"      "1"      "1"      "1"      "OUTROS"
## [15] "OUTROS"
```

```

#edit(MDSplot)
fig.align="center"
training1$DIRETORIA = as.factor(training1$DIRETORIA)
testing1$DIRETORIA = as.factor(testing1$DIRETORIA)
db_modelo$DIRETORIA = as.factor(db_modelo$DIRETORIA)

(MDIM_treino = MDSplot(rf3, training1$DIRETORIA, pch=20))

```



```

## $points
##          Dim 1      Dim 2
## 1    0.48706199 0.13832372
## 2   -0.51365242 0.16158370
## 3   -0.02750134 -0.42840656
## 4   -0.51421632 0.16082709
## 5    0.48679248 0.14277997
## 6   -0.02131776 -0.40854075
## 7    0.48693772 0.14015690
## 8   -0.51327875 0.16319477
## 9    0.48705865 0.14047510
## 10   0.48701012 0.13644992
## 11  -0.02117122 -0.41037581
## 12  -0.51238672 0.16591077
## 13  -0.51271311 0.16600004
## 14  -0.10991013 -0.26153938
## 15   0.01947657 -0.30229950
## 16   0.48667965 0.14432116
## 17   0.48686064 0.14103881
## 18  -0.51238717 0.16834268
## 19  -0.02699864 -0.39933997
## 20   0.04154658 -0.27156286
## 21   0.48713386 0.14038301
## 22   0.02551741 -0.30464243

```

## 23	0.48718274	0.13896670
## 24	-0.02648562	-0.37242162
## 25	0.48675624	0.13981756
## 26	0.48709071	0.14046916
## 27	0.48715317	0.13510545
## 28	-0.51313117	0.16543880
## 29	-0.51276292	0.16514584
## 30	0.02682511	-0.28482271
## 31	-0.02757584	-0.43194056
## 32	-0.51272403	0.16316354
## 33	0.48690474	0.14113806
## 34	-0.02865857	-0.43591082
## 35	-0.51303738	0.16424189
## 36	-0.51248334	0.16749880
## 37	-0.51215508	0.16600790
## 38	-0.10183763	-0.36258945
## 39	0.48702333	0.13658432
## 40	0.48693958	0.13649018
## 41	-0.51237553	0.16691228
## 42	-0.51234092	0.16680098
## 43	-0.02699718	-0.39929196
## 44	-0.02726683	-0.39099294
## 45	-0.02730537	-0.41886757
## 46	-0.51219454	0.16611299
## 47	-0.02850523	-0.44039193
## 48	0.48717604	0.13995521
## 49	0.02412469	-0.28381847
## 50	0.48681814	0.14171034
## 51	-0.51338052	0.16472197
## 52	0.48688923	0.13921130
## 53	0.48699855	0.14329725
## 54	-0.51225865	0.16581270
## 55	0.48712351	0.13739097
## 56	0.48690213	0.14373172
## 57	-0.51294988	0.16102077
## 58	-0.02723839	-0.39134711
## 59	0.48697233	0.14004293
## 60	-0.02695162	-0.38182752
## 61	-0.02815733	-0.34394660
## 62	-0.51216657	0.16749710
## 63	-0.02636481	-0.36192195
## 64	0.48711134	0.14052867
## 65	-0.51293719	0.16657409
## 66	-0.51261710	0.16646151
## 67	-0.51286110	0.16721306
## 68	-0.51261819	0.16534741
## 69	0.48689809	0.14003513
## 70	-0.51301522	0.16502153
## 71	0.48681471	0.13509451
## 72	0.48665569	0.14257094
## 73	-0.51285631	0.16478926
## 74	0.48689714	0.14304658
## 75	-0.51278494	0.16647662
## 76	-0.51296779	0.16372824

```

## 77 0.48670241 0.14252392
## 78 0.48693608 0.14052451
## 79 -0.51129826 0.16881451
## 80 0.02437334 -0.29657388
## 81 0.48686531 0.14342124
## 82 -0.02677348 -0.37428525
## 83 0.48742713 0.13940288
## 84 0.48719106 0.13718388
## 85 -0.51166458 0.16799148
## 86 -0.01981900 -0.34443188
## 87 0.48728385 0.13744844
## 88 0.48689565 0.13746035
## 89 -0.51305557 0.16272527
## 90 0.48678616 0.13935915
## 91 -0.02698745 -0.39889988
## 92 -0.27634708 -0.07189875
## 93 0.48706546 0.14057651
## 94 -0.02803162 -0.37616517
## 95 -0.51236972 0.16700094
## 96 -0.51248995 0.16771292
## 97 0.48668713 0.14096192
## 98 -0.51218448 0.16823943
## 99 -0.51247077 0.16662417
## 100 0.46752282 0.11960607
## 101 -0.51371125 0.16555960
## 102 0.48683060 0.14260857
## 103 -0.51351012 0.16167892
## 104 -0.51283980 0.16433681
## 105 0.48692979 0.14111019
## 106 0.48682207 0.14188806
## 107 -0.51198205 0.16808075
## 108 -0.51152081 0.16823935
## 109 -0.02764897 -0.41701336
## 110 0.48693682 0.14202035
## 111 0.48731521 0.13856962
## 112 -0.02677750 -0.37115267
## 113 -0.51363921 0.16334882
## 114 0.48677564 0.13951098
## 115 -0.51361237 0.16212182
## 116 -0.02658209 -0.35632541
## 117 0.01799079 -0.30383719
## 118 -0.51272649 0.16564816
## 119 0.48683120 0.13878201
## 120 -0.51267492 0.16587556
## 121 -0.02742694 -0.42470876
## 122 -0.51392212 0.16213507
## 123 -0.02879347 -0.43308084
## 124 -0.51241930 0.16410973
## 125 0.48675023 0.14353522
## 126 -0.02753120 -0.42989795
## 127 0.48706838 0.14217114
## 128 0.48712265 0.14099518
## 129 0.48710370 0.13943988
## 130 0.48685602 0.14219728

```


131 0.48733089 0.13590943
132 -0.14098885 -0.22321685
133 0.48716762 0.13835724
134 0.07084709 -0.25643887
135 0.48677467 0.13654052
136 -0.51252582 0.16673436
137 0.48688807 0.13876374
138 -0.51259265 0.16755862
139 0.48713394 0.13804966
140 0.48689335 0.14119695
141 -0.02679312 -0.38985238
142 -0.51175094 0.16676012
143 -0.02613236 -0.35592516
144 0.48705429 0.14231249
145 -0.04107436 -0.41519562
146 -0.02717412 -0.41263674
147 0.48686004 0.13979715
148 0.48699291 0.14004156
149 -0.02886886 -0.42313365
150 0.02501296 -0.29475983
151 -0.51162990 0.17031526
152 0.48702195 0.14039900
153 -0.51274420 0.16436318
154 0.48722826 0.13801171
155 0.01968680 -0.30226424
156 -0.02632681 -0.34668667
157 -0.02691977 -0.38775423
158 -0.02033183 -0.34543060
159 -0.51261479 0.16550819
160 -0.02865173 -0.42796055
161 -0.09585502 -0.29015171
162 0.48745620 0.13741749
163 0.48737790 0.13555559
164 -0.02863077 -0.43218836
165 0.48697541 0.13870282
166 -0.04916135 -0.41089847
167 -0.02639562 -0.36298240
168 -0.14903641 -0.20720847
169 -0.51305531 0.16514226
170 -0.02847864 -0.43882395
171 -0.51249729 0.16467793
172 -0.02879449 -0.43201723
173 -0.06775527 -0.30478065
174 0.48702931 0.14067225
175 -0.51275935 0.16719808
176 -0.51352657 0.16129565
177 -0.02779315 -0.38284137
178 -0.02807487 -0.45583784
179 0.07136452 -0.24386469
180 0.48684386 0.13918389
181 0.48715704 0.13719982
182 -0.11434129 -0.25734655
183 -0.10089361 -0.26946633
184 -0.51212758 0.16576037

185 0.48686902 0.14014498
186 -0.02744236 -0.42535800
187 -0.51261937 0.16571058
188 0.48703868 0.13783354
189 -0.51331720 0.16348032
190 -0.51312792 0.16765108
191 -0.01962682 -0.34283921
192 0.48728429 0.13859410
193 -0.51247188 0.16625739
194 -0.51304480 0.16390961
195 -0.02766244 -0.43581360
196 0.48710408 0.14173396
197 -0.02701554 -0.40008268
198 -0.18020933 -0.17451139
199 -0.01988743 -0.34229844
200 0.48670003 0.14136218
201 0.48683441 0.13922578
202 -0.51226991 0.16962762
203 -0.51325596 0.16629632
204 -0.51203049 0.16768635
205 0.48711727 0.14079768
206 0.48712167 0.13741070
207 0.48716227 0.13763595
208 0.48710292 0.13684606
209 -0.40387636 0.04037033
210 -0.51266389 0.16541010
211 -0.51281341 0.16555359
212 0.48677849 0.14145723
213 0.48684726 0.13547234
214 -0.51200852 0.16593581
215 0.48682317 0.13805461
216 -0.02755464 -0.43102921
217 0.48736030 0.13359581
218 -0.51266232 0.16714897
219 -0.02652428 -0.37109718
220 -0.04090641 -0.41806977
221 -0.02699519 -0.39921547
222 0.48729329 0.13858438
223 -0.51251745 0.16645630
224 0.48683906 0.14268678
225 -0.02755466 -0.43088461
226 0.02716698 -0.28448030
227 -0.51376070 0.16132913
228 0.48731155 0.13780587
229 0.48684166 0.14294897
230 0.48707262 0.14009725
231 -0.51347531 0.16423083
232 -0.51363359 0.16350513
233 0.48708318 0.13805878
234 0.48716188 0.13863956
235 -0.02770025 -0.43770354
236 0.48677469 0.14311460
237 0.03658463 -0.27908817
238 -0.51289085 0.16455586

239 0.48726335 0.13824146
240 0.48728391 0.13978108
241 -0.51240042 0.16728700
242 -0.51255067 0.16803133
243 0.48661658 0.14327695
244 -0.27794539 -0.06834729
245 -0.02670451 -0.39963091
246 -0.04042645 -0.42286147
247 -0.51308664 0.16613406
248 -0.51244523 0.16753746
249 0.48721608 0.13665123
250 -0.51288278 0.16522315
251 0.48698186 0.14368084
252 0.48668045 0.14265837
253 -0.51141931 0.16694052
254 -0.00453419 -0.32520622
255 -0.51267141 0.16733883
256 0.48661624 0.14016976
257 -0.51310350 0.16410470
258 0.07864277 -0.25073087
259 0.48691045 0.14393707
260 0.48734348 0.13952489
261 -0.51225584 0.16716570
262 -0.51289831 0.16586784
263 -0.51258181 0.16742951
264 -0.08511403 -0.30224045
265 -0.02757866 -0.43211185
266 0.48697287 0.13631054
267 0.48681327 0.14154556
268 0.48698168 0.14379076
269 -0.51305795 0.16412834
270 0.48707777 0.14042342
271 -0.02748428 -0.42754660
272 -0.02647677 -0.37189129
273 0.48687838 0.14291487
274 0.48696233 0.14082511
275 -0.51273349 0.16961432
276 -0.02699344 -0.39913739
277 -0.02730852 -0.41914205
278 -0.51245495 0.16489852
279 -0.51277281 0.16394382
280 0.48742279 0.13835302
281 -0.27716911 -0.09055552
282 -0.51245236 0.16676575
283 0.48679742 0.14373217
284 -0.02679408 -0.38712427
285 -0.51220282 0.16528078
286 -0.51230287 0.16663843
287 -0.02054016 -0.37254261
288 0.48684735 0.14108828
289 -0.51236937 0.16343541
290 -0.51317466 0.16501285
291 -0.04112937 -0.34588925
292 -0.02624756 -0.36120923

293 0.48687318 0.14058910
294 -0.02699516 -0.39922389
295 0.48684550 0.14314310
296 0.48694506 0.13592962
297 -0.02629063 -0.36303073
298 -0.51231005 0.16629853
299 -0.02662651 -0.37852043
300 0.48688854 0.14094432
301 -0.02805030 -0.41113155
302 -0.51264168 0.16687987
303 -0.51135055 0.17030416
304 0.48694605 0.14209908
305 0.48744217 0.13773746
306 -0.02744578 -0.42550302
307 0.48683741 0.14298847
308 -0.19025785 -0.16304372
309 0.48702921 0.13942301
310 -0.02746720 -0.42684704
311 0.48725663 0.13868637
312 -0.51243092 0.16680334
313 0.03659801 -0.27495730
314 -0.51265628 0.16573157
315 0.48675151 0.14032994
316 -0.51252227 0.16784669
317 -0.01965163 -0.34370677
318 0.48724528 0.14029093
319 -0.51168456 0.16922867
320 -0.51300243 0.16404932
321 -0.51206755 0.16795136
322 0.48724421 0.14200713
323 -0.03376552 -0.35394892
324 0.48683046 0.13894574
325 -0.51243970 0.16823839
326 0.48707520 0.13666069
327 -0.51247426 0.16856061
328 0.48675715 0.14407659
329 -0.51261620 0.16400547
330 -0.51255017 0.16624650
331 -0.51247536 0.16841587
332 -0.51301859 0.16387049
333 -0.02700314 -0.39956895
334 -0.51309135 0.16386227
335 0.48702901 0.13527142
336 -0.51204534 0.16920909
337 0.02937912 -0.29359522
338 -0.51241516 0.16826474
339 -0.02700374 -0.39955138
340 -0.51290371 0.16321342
341 -0.02748788 -0.42785094
342 0.48726679 0.13646004
343 -0.51282513 0.16644322
344 0.48712265 0.13797419
345 -0.48852491 0.14403628
346 -0.51258047 0.16689961

347 -0.51282179 0.16427595
348 -0.51220253 0.16461256
349 0.48708182 0.13810691
350 0.48678353 0.14102360
351 -0.02624421 -0.36100305
352 0.48697290 0.14221124
353 -0.02743446 -0.42505979
354 0.21438274 -0.11038038
355 0.48658705 0.13946612
356 -0.51258418 0.16887355
357 0.48682528 0.14276050
358 0.48722649 0.13412936
359 -0.51285970 0.16586034
360 -0.09505898 -0.27060771
361 0.48695566 0.14063531
362 -0.51125007 0.17314715
363 -0.51328656 0.16411544
364 -0.02660096 -0.37747199
365 -0.51265053 0.16379868
366 -0.02706361 -0.36177415
367 0.48661595 0.14463254
368 -0.02739921 -0.42327784
369 0.48714593 0.13669515
370 -0.02719841 -0.40200509
371 0.48675316 0.13764749
372 0.48693916 0.13743158
373 0.48715096 0.13702714
374 -0.16878086 -0.18892277
375 -0.02619131 -0.35859311
376 -0.51365279 0.16134746
377 0.48678569 0.14105265
378 0.48691964 0.14040297
379 -0.08816316 -0.28388893
380 -0.51294478 0.16359337
381 0.15290156 -0.19961962
382 -0.51246314 0.16724572
383 -0.09538355 -0.27566479
384 0.48688223 0.13889805
385 -0.51316047 0.16457519
386 0.48712275 0.13856442
387 -0.02739635 -0.42336184
388 -0.51121711 0.16894316
389 0.48725629 0.13753840
390 -0.02810609 -0.41421775
391 0.48704760 0.14004000
392 -0.51271540 0.16433242
393 -0.51278317 0.16355149
394 -0.51223688 0.16778839
395 0.48718956 0.13989236
396 -0.03058124 -0.34978519
397 -0.03125394 -0.35207884
398 0.48718975 0.13584490
399 -0.51191851 0.16934609
400 -0.51209990 0.16761729

```

## 401 0.48717265 0.13736091
## 402 0.07393553 -0.24431152
## 403 -0.51183214 0.16739802
## 404 0.48744910 0.13535988
## 405 -0.51211312 0.16581973
## 406 0.48695569 0.13833650
## 407 0.48702684 0.13967578
## 408 0.48700728 0.14215552
##
## $eig
## [1] 7.154072e+01 2.260444e+01 6.142944e+00 5.612111e+00 5.356335e+00
## [6] 4.228210e+00 3.284671e+00 2.922426e+00 2.778434e+00 2.681691e+00
## [11] 2.620385e+00 2.510296e+00 2.467015e+00 2.191005e+00 2.078947e+00
## [16] 1.917182e+00 1.846059e+00 1.792800e+00 1.158628e+00 1.089363e+00
## [21] 1.009891e+00 9.371735e-01 7.206314e-01 5.370035e-01 5.189383e-01
## [26] 4.860600e-01 4.656067e-01 4.281153e-01 4.173279e-01 4.025849e-01
## [31] 3.732467e-01 3.633766e-01 3.491361e-01 3.312614e-01 3.178951e-01
## [36] 3.075031e-01 3.000037e-01 2.946116e-01 2.412091e-01 2.327315e-01
## [41] 2.188737e-01 2.129291e-01 2.051127e-01 1.931579e-01 1.914391e-01
## [46] 1.820200e-01 1.776070e-01 1.684189e-01 1.597512e-01 1.536129e-01
## [51] 1.441120e-01 1.415717e-01 1.328320e-01 1.230696e-01 1.181261e-01
## [56] 1.114740e-01 1.070471e-01 1.032831e-01 1.008210e-01 9.480377e-02
## [61] 8.729672e-02 8.323205e-02 8.137579e-02 7.295963e-02 6.890010e-02
## [66] 6.741029e-02 6.549907e-02 6.195604e-02 5.916309e-02 5.706993e-02
## [71] 5.591420e-02 5.325271e-02 4.949706e-02 4.795811e-02 4.547968e-02
## [76] 4.126039e-02 3.862348e-02 3.738781e-02 3.580720e-02 3.210857e-02
## [81] 3.044963e-02 2.509894e-02 2.246196e-02 2.104936e-02 1.778965e-02
## [86] 1.391910e-02 1.264778e-02 1.204355e-02 1.131545e-02 9.546429e-03
## [91] 7.541357e-03 5.573194e-03 2.725189e-03 2.064650e-03 6.078363e-04
## [96] 4.138645e-15 2.634612e-15 1.175688e-15 6.330920e-16 5.890808e-16
## [101] 5.555422e-16 5.402404e-16 5.172120e-16 4.907629e-16 4.615918e-16
## [106] 3.442972e-16 3.417240e-16 2.987823e-16 2.921588e-16 2.908981e-16
## [111] 2.622128e-16 2.611105e-16 2.364166e-16 1.913130e-16 1.790187e-16
## [116] 1.768161e-16 1.640063e-16 1.598318e-16 1.503600e-16 1.485370e-16
## [121] 1.372126e-16 1.332630e-16 1.312929e-16 1.299645e-16 1.134943e-16
## [126] 1.105201e-16 1.095396e-16 1.095218e-16 1.049813e-16 9.747550e-17
## [131] 8.969762e-17 8.102185e-17 7.953630e-17 6.845039e-17 6.039355e-17
## [136] 5.537779e-17 5.250938e-17 4.949924e-17 4.869363e-17 4.850911e-17
## [141] 4.301308e-17 3.954418e-17 3.949639e-17 3.671258e-17 3.628826e-17
## [146] 3.570248e-17 3.564546e-17 3.521152e-17 3.488612e-17 3.450431e-17
## [151] 2.988645e-17 2.671491e-17 2.645389e-17 2.613598e-17 2.484533e-17
## [156] 2.403978e-17 2.401784e-17 2.379152e-17 2.362956e-17 2.359985e-17
## [161] 2.327335e-17 2.236586e-17 2.154002e-17 2.149519e-17 2.028517e-17
## [166] 2.007724e-17 1.977315e-17 1.840639e-17 1.828335e-17 1.773784e-17
## [171] 1.631547e-17 1.628754e-17 1.614066e-17 1.602094e-17 1.596290e-17
## [176] 1.588612e-17 1.578621e-17 1.570502e-17 1.545980e-17 1.435951e-17
## [181] 1.397178e-17 1.387791e-17 1.383539e-17 1.360247e-17 1.358778e-17
## [186] 1.327493e-17 1.289235e-17 1.202899e-17 1.068537e-17 1.060518e-17
## [191] 8.424352e-18 8.386574e-18 8.060100e-18 7.920085e-18 7.090664e-18
## [196] 6.782495e-18 6.749501e-18 6.537259e-18 6.375203e-18 5.896398e-18
## [201] 5.699499e-18 5.495396e-18 5.149090e-18 4.894049e-18 4.728270e-18
## [206] 4.648356e-18 4.642101e-18 4.489704e-18 4.250791e-18 4.056572e-18
## [211] 3.416414e-18 3.160013e-18 3.102442e-18 3.002409e-18 2.857604e-18
## [216] 2.639762e-18 1.707803e-18 1.611843e-18 1.227324e-18 1.129270e-18

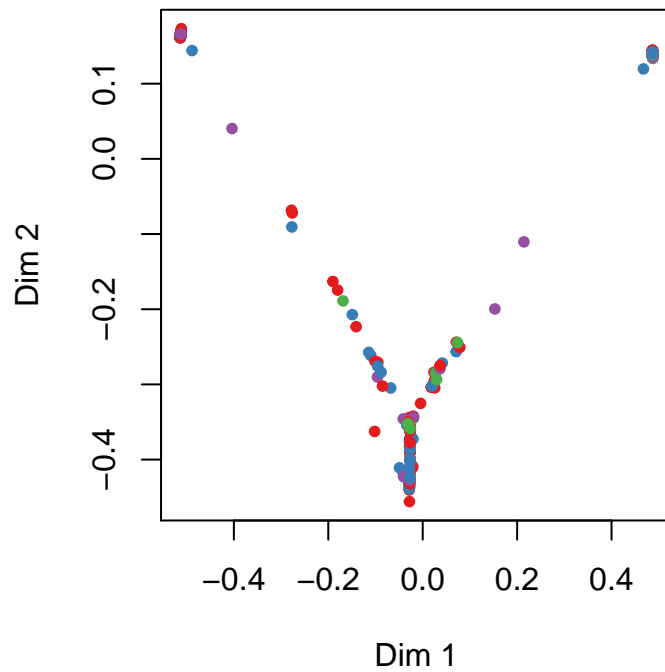
```

```

## [221] 9.446178e-19 8.847936e-19 7.569917e-19 6.290560e-19 5.945541e-19
## [226] 5.343330e-19 4.243650e-19 -9.359642e-19 -1.226441e-18 -2.190531e-18
## [231] -2.278966e-18 -2.340581e-18 -2.570205e-18 -2.655616e-18 -2.946622e-18
## [236] -4.910198e-18 -4.920260e-18 -4.940601e-18 -5.118634e-18 -5.390240e-18
## [241] -5.766580e-18 -5.819451e-18 -6.115810e-18 -7.519902e-18 -7.760665e-18
## [246] -8.236694e-18 -8.447660e-18 -8.452402e-18 -8.884159e-18 -8.978156e-18
## [251] -9.614729e-18 -9.624937e-18 -9.695860e-18 -1.041532e-17 -1.159739e-17
## [256] -1.186978e-17 -1.212565e-17 -1.218597e-17 -1.229858e-17 -1.274439e-17
## [261] -1.416147e-17 -1.465689e-17 -1.476018e-17 -1.541887e-17 -1.740426e-17
## [266] -1.773165e-17 -1.776412e-17 -1.835683e-17 -1.876019e-17 -1.909112e-17
## [271] -1.965292e-17 -1.988369e-17 -1.996469e-17 -2.003620e-17 -2.043187e-17
## [276] -2.044032e-17 -2.067933e-17 -2.133117e-17 -2.156873e-17 -2.186353e-17
## [281] -2.301569e-17 -2.504849e-17 -2.506474e-17 -2.565723e-17 -2.574737e-17
## [286] -2.653549e-17 -2.704155e-17 -2.956625e-17 -3.242186e-17 -3.294802e-17
## [291] -3.442858e-17 -3.507684e-17 -3.546032e-17 -3.547032e-17 -3.634026e-17
## [296] -3.859902e-17 -3.894960e-17 -4.421730e-17 -4.687827e-17 -4.808756e-17
## [301] -5.038506e-17 -5.419574e-17 -5.423806e-17 -5.994962e-17 -6.871194e-17
## [306] -8.465594e-17 -1.041368e-16 -1.185283e-16 -1.193142e-16 -1.328431e-16
## [311] -1.360447e-16 -1.860684e-16 -1.980117e-16 -2.004490e-16 -2.195498e-16
## [316] -2.204128e-16 -2.574140e-16 -3.302231e-16 -3.419604e-16 -3.422964e-16
## [321] -3.660549e-16 -5.583986e-16 -5.773028e-16 -6.745385e-16 -1.227583e-15
## [326] -1.352065e-15 -2.753607e-15 -3.571514e-15 -3.206919e-05 -8.623957e-04
## [331] -1.252205e-03 -1.545425e-03 -1.849052e-03 -2.163410e-03 -5.078032e-03
## [336] -5.698310e-03 -7.468272e-03 -8.654237e-03 -8.960363e-03 -1.093129e-02
## [341] -1.156955e-02 -1.289780e-02 -1.392421e-02 -1.759915e-02 -1.794018e-02
## [346] -2.103592e-02 -2.547492e-02 -2.662051e-02 -2.929132e-02 -3.447477e-02
## [351] -3.711743e-02 -3.803192e-02 -3.994176e-02 -4.228982e-02 -4.542149e-02
## [356] -4.775701e-02 -4.939157e-02 -5.116449e-02 -5.546252e-02 -5.848408e-02
## [361] -5.908713e-02 -6.361706e-02 -6.521525e-02 -6.789260e-02 -7.036070e-02
## [366] -7.460343e-02 -7.675212e-02 -7.811232e-02 -7.929642e-02 -8.494227e-02
## [371] -8.533985e-02 -8.748983e-02 -9.005717e-02 -9.519660e-02 -1.023627e-01
## [376] -1.055915e-01 -1.110482e-01 -1.140106e-01 -1.209387e-01 -1.239211e-01
## [381] -1.344748e-01 -1.390099e-01 -1.458075e-01 -1.608968e-01 -1.631702e-01
## [386] -1.691948e-01 -1.771840e-01 -1.832890e-01 -1.986827e-01 -2.060067e-01
## [391] -2.149887e-01 -2.215848e-01 -2.288375e-01 -2.351471e-01 -2.394344e-01
## [396] -2.547218e-01 -2.749835e-01 -2.761848e-01 -2.912567e-01 -3.011581e-01
## [401] -3.464090e-01 -3.643249e-01 -3.815793e-01 -3.987337e-01 -4.147181e-01
## [406] -4.326975e-01 -4.482354e-01 -5.330642e-01
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.5529590 0.5875442

```

```
(MDIM_teste = MDSplot(rf3, testing1$DIRETORIA, pch=20))
```



```
## $points
##      Dim 1      Dim 2
## 1  0.48706199 0.13832372
## 2 -0.51365242 0.16158370
## 3 -0.02750134 -0.42840656
## 4 -0.51421632 0.16082709
## 5  0.48679248 0.14277997
## 6 -0.02131776 -0.40854075
## 7  0.48693772 0.14015690
## 8 -0.51327875 0.16319477
## 9  0.48705865 0.14047510
## 10 0.48701012 0.13644992
## 11 -0.02117122 -0.41037581
## 12 -0.51238672 0.16591077
## 13 -0.51271311 0.16600004
## 14 -0.10991013 -0.26153938
## 15  0.01947657 -0.30229950
## 16  0.48667965 0.14432116
## 17  0.48686064 0.14103881
## 18 -0.51238717 0.16834268
## 19 -0.02699864 -0.39933997
## 20  0.04154658 -0.27156286
## 21  0.48713386 0.14038301
## 22  0.02551741 -0.30464243
## 23  0.48718274 0.13896670
## 24 -0.02648562 -0.37242162
## 25  0.48675624 0.13981756
## 26  0.48709071 0.14046916
## 27  0.48715317 0.13510545
## 28 -0.51313117 0.16543880
## 29 -0.51276292 0.16514584
## 30  0.02682511 -0.28482271
```


31 -0.02757584 -0.43194056
32 -0.51272403 0.16316354
33 0.48690474 0.14113806
34 -0.02865857 -0.43591082
35 -0.51303738 0.16424189
36 -0.51248334 0.16749880
37 -0.51215508 0.16600790
38 -0.10183763 -0.36258945
39 0.48702333 0.13658432
40 0.48693958 0.13649018
41 -0.51237553 0.16691228
42 -0.51234092 0.16680098
43 -0.02699718 -0.39929196
44 -0.02726683 -0.39099294
45 -0.02730537 -0.41886757
46 -0.51219454 0.16611299
47 -0.02850523 -0.44039193
48 0.48717604 0.13995521
49 0.02412469 -0.28381847
50 0.48681814 0.14171034
51 -0.51338052 0.16472197
52 0.48688923 0.13921130
53 0.48699855 0.14329725
54 -0.51225865 0.16581270
55 0.48712351 0.13739097
56 0.48690213 0.14373172
57 -0.51294988 0.16102077
58 -0.02723839 -0.39134711
59 0.48697233 0.14004293
60 -0.02695162 -0.38182752
61 -0.02815733 -0.34394660
62 -0.51216657 0.16749710
63 -0.02636481 -0.36192195
64 0.48711134 0.14052867
65 -0.51293719 0.16657409
66 -0.51261710 0.16646151
67 -0.51286110 0.16721306
68 -0.51261819 0.16534741
69 0.48689809 0.14003513
70 -0.51301522 0.16502153
71 0.48681471 0.13509451
72 0.48665569 0.14257094
73 -0.51285631 0.16478926
74 0.48689714 0.14304658
75 -0.51278494 0.16647662
76 -0.51296779 0.16372824
77 0.48670241 0.14252392
78 0.48693608 0.14052451
79 -0.51129826 0.16881451
80 0.02437334 -0.29657388
81 0.48686531 0.14342124
82 -0.02677348 -0.37428525
83 0.48742713 0.13940288
84 0.48719106 0.13718388

85 -0.51166458 0.16799148
86 -0.01981900 -0.34443188
87 0.48728385 0.13744844
88 0.48689565 0.13746035
89 -0.51305557 0.16272527
90 0.48678616 0.13935915
91 -0.02698745 -0.39889988
92 -0.27634708 -0.07189875
93 0.48706546 0.14057651
94 -0.02803162 -0.37616517
95 -0.51236972 0.16700094
96 -0.51248995 0.16771292
97 0.48668713 0.14096192
98 -0.51218448 0.16823943
99 -0.51247077 0.16662417
100 0.46752282 0.11960607
101 -0.51371125 0.16555960
102 0.48683060 0.14260857
103 -0.51351012 0.16167892
104 -0.51283980 0.16433681
105 0.48692979 0.14111019
106 0.48682207 0.14188806
107 -0.51198205 0.16808075
108 -0.51152081 0.16823935
109 -0.02764897 -0.41701336
110 0.48693682 0.14202035
111 0.48731521 0.13856962
112 -0.02677750 -0.37115267
113 -0.51363921 0.16334882
114 0.48677564 0.13951098
115 -0.51361237 0.16212182
116 -0.02658209 -0.35632541
117 0.01799079 -0.30383719
118 -0.51272649 0.16564816
119 0.48683120 0.13878201
120 -0.51267492 0.16587556
121 -0.02742694 -0.42470876
122 -0.51392212 0.16213507
123 -0.02879347 -0.43308084
124 -0.51241930 0.16410973
125 0.48675023 0.14353522
126 -0.02753120 -0.42989795
127 0.48706838 0.14217114
128 0.48712265 0.14099518
129 0.48710370 0.13943988
130 0.48685602 0.14219728
131 0.48733089 0.13590943
132 -0.14098885 -0.22321685
133 0.48716762 0.13835724
134 0.07084709 -0.25643887
135 0.48677467 0.13654052
136 -0.51252582 0.16673436
137 0.48688807 0.13876374
138 -0.51259265 0.16755862

139 0.48713394 0.13804966
140 0.48689335 0.14119695
141 -0.02679312 -0.38985238
142 -0.51175094 0.16676012
143 -0.02613236 -0.35592516
144 0.48705429 0.14231249
145 -0.04107436 -0.41519562
146 -0.02717412 -0.41263674
147 0.48686004 0.13979715
148 0.48699291 0.14004156
149 -0.02886886 -0.42313365
150 0.02501296 -0.29475983
151 -0.51162990 0.17031526
152 0.48702195 0.14039900
153 -0.51274420 0.16436318
154 0.48722826 0.13801171
155 0.01968680 -0.30226424
156 -0.02632681 -0.34668667
157 -0.02691977 -0.38775423
158 -0.02033183 -0.34543060
159 -0.51261479 0.16550819
160 -0.02865173 -0.42796055
161 -0.09585502 -0.29015171
162 0.48745620 0.13741749
163 0.48737790 0.13555559
164 -0.02863077 -0.43218836
165 0.48697541 0.13870282
166 -0.04916135 -0.41089847
167 -0.02639562 -0.36298240
168 -0.14903641 -0.20720847
169 -0.51305531 0.16514226
170 -0.02847864 -0.43882395
171 -0.51249729 0.16467793
172 -0.02879449 -0.43201723
173 -0.06775527 -0.30478065
174 0.48702931 0.14067225
175 -0.51275935 0.16719808
176 -0.51352657 0.16129565
177 -0.02779315 -0.38284137
178 -0.02807487 -0.45583784
179 0.07136452 -0.24386469
180 0.48684386 0.13918389
181 0.48715704 0.13719982
182 -0.11434129 -0.25734655
183 -0.10089361 -0.26946633
184 -0.51212758 0.16576037
185 0.48686902 0.14014498
186 -0.02744236 -0.42535800
187 -0.51261937 0.16571058
188 0.48703868 0.13783354
189 -0.51331720 0.16348032
190 -0.51312792 0.16765108
191 -0.01962682 -0.34283921
192 0.48728429 0.13859410

```

## 193 -0.51247188 0.16625739
## 194 -0.51304480 0.16390961
## 195 -0.02766244 -0.43581360
## 196 0.48710408 0.14173396
## 197 -0.02701554 -0.40008268
## 198 -0.18020933 -0.17451139
## 199 -0.01988743 -0.34229844
## 200 0.48670003 0.14136218
## 201 0.48683441 0.13922578
## 202 -0.51226991 0.16962762
## 203 -0.51325596 0.16629632
## 204 -0.51203049 0.16768635
## 205 0.48711727 0.14079768
## 206 0.48712167 0.13741070
## 207 0.48716227 0.13763595
## 208 0.48710292 0.13684606
## 209 -0.40387636 0.04037033
## 210 -0.51266389 0.16541010
## 211 -0.51281341 0.16555359
## 212 0.48677849 0.14145723
## 213 0.48684726 0.13547234
## 214 -0.51200852 0.16593581
## 215 0.48682317 0.13805461
## 216 -0.02755464 -0.43102921
## 217 0.48736030 0.13359581
## 218 -0.51266232 0.16714897
## 219 -0.02652428 -0.37109718
## 220 -0.04090641 -0.41806977
## 221 -0.02699519 -0.39921547
## 222 0.48729329 0.13858438
## 223 -0.51251745 0.16645630
## 224 0.48683906 0.14268678
## 225 -0.02755466 -0.43088461
## 226 0.02716698 -0.28448030
## 227 -0.51376070 0.16132913
## 228 0.48731155 0.13780587
## 229 0.48684166 0.14294897
## 230 0.48707262 0.14009725
## 231 -0.51347531 0.16423083
## 232 -0.51363359 0.16350513
## 233 0.48708318 0.13805878
## 234 0.48716188 0.13863956
## 235 -0.02770025 -0.43770354
## 236 0.48677469 0.14311460
## 237 0.03658463 -0.27908817
## 238 -0.51289085 0.16455586
## 239 0.48726335 0.13824146
## 240 0.48728391 0.13978108
## 241 -0.51240042 0.16728700
## 242 -0.51255067 0.16803133
## 243 0.48661658 0.14327695
## 244 -0.27794539 -0.06834729
## 245 -0.02670451 -0.39963091
## 246 -0.04042645 -0.42286147

```

```

## 247 -0.51308664 0.16613406
## 248 -0.51244523 0.16753746
## 249 0.48721608 0.13665123
## 250 -0.51288278 0.16522315
## 251 0.48698186 0.14368084
## 252 0.48668045 0.14265837
## 253 -0.51141931 0.16694052
## 254 -0.00453419 -0.32520622
## 255 -0.51267141 0.16733883
## 256 0.48661624 0.14016976
## 257 -0.51310350 0.16410470
## 258 0.07864277 -0.25073087
## 259 0.48691045 0.14393707
## 260 0.48734348 0.13952489
## 261 -0.51225584 0.16716570
## 262 -0.51289831 0.16586784
## 263 -0.51258181 0.16742951
## 264 -0.08511403 -0.30224045
## 265 -0.02757866 -0.43211185
## 266 0.48697287 0.13631054
## 267 0.48681327 0.14154556
## 268 0.48698168 0.14379076
## 269 -0.51305795 0.16412834
## 270 0.48707777 0.14042342
## 271 -0.02748428 -0.42754660
## 272 -0.02647677 -0.37189129
## 273 0.48687838 0.14291487
## 274 0.48696233 0.14082511
## 275 -0.51273349 0.16961432
## 276 -0.02699344 -0.39913739
## 277 -0.02730852 -0.41914205
## 278 -0.51245495 0.16489852
## 279 -0.51277281 0.16394382
## 280 0.48742279 0.13835302
## 281 -0.27716911 -0.09055552
## 282 -0.51245236 0.16676575
## 283 0.48679742 0.14373217
## 284 -0.02679408 -0.38712427
## 285 -0.51220282 0.16528078
## 286 -0.51230287 0.16663843
## 287 -0.02054016 -0.37254261
## 288 0.48684735 0.14108828
## 289 -0.51236937 0.16343541
## 290 -0.51317466 0.16501285
## 291 -0.04112937 -0.34588925
## 292 -0.02624756 -0.36120923
## 293 0.48687318 0.14058910
## 294 -0.02699516 -0.39922389
## 295 0.48684550 0.14314310
## 296 0.48694506 0.13592962
## 297 -0.02629063 -0.36303073
## 298 -0.51231005 0.16629853
## 299 -0.02662651 -0.37852043
## 300 0.48688854 0.14094432

```

301 -0.02805030 -0.41113155
302 -0.51264168 0.16687987
303 -0.51135055 0.17030416
304 0.48694605 0.14209908
305 0.48744217 0.13773746
306 -0.02744578 -0.42550302
307 0.48683741 0.14298847
308 -0.19025785 -0.16304372
309 0.48702921 0.13942301
310 -0.02746720 -0.42684704
311 0.48725663 0.13868637
312 -0.51243092 0.16680334
313 0.03659801 -0.27495730
314 -0.51265628 0.16573157
315 0.48675151 0.14032994
316 -0.51252227 0.16784669
317 -0.01965163 -0.34370677
318 0.48724528 0.14029093
319 -0.51168456 0.16922867
320 -0.51300243 0.16404932
321 -0.51206755 0.16795136
322 0.48724421 0.14200713
323 -0.03376552 -0.35394892
324 0.48683046 0.13894574
325 -0.51243970 0.16823839
326 0.48707520 0.13666069
327 -0.51247426 0.16856061
328 0.48675715 0.14407659
329 -0.51261620 0.16400547
330 -0.51255017 0.16624650
331 -0.51247536 0.16841587
332 -0.51301859 0.16387049
333 -0.02700314 -0.39956895
334 -0.51309135 0.16386227
335 0.48702901 0.13527142
336 -0.51204534 0.16920909
337 0.02937912 -0.29359522
338 -0.51241516 0.16826474
339 -0.02700374 -0.39955138
340 -0.51290371 0.16321342
341 -0.02748788 -0.42785094
342 0.48726679 0.13646004
343 -0.51282513 0.16644322
344 0.48712265 0.13797419
345 -0.48852491 0.14403628
346 -0.51258047 0.16689961
347 -0.51282179 0.16427595
348 -0.51220253 0.16461256
349 0.48708182 0.13810691
350 0.48678353 0.14102360
351 -0.02624421 -0.36100305
352 0.48697290 0.14221124
353 -0.02743446 -0.42505979
354 0.21438274 -0.11038038

355 0.48658705 0.13946612
356 -0.51258418 0.16887355
357 0.48682528 0.14276050
358 0.48722649 0.13412936
359 -0.51285970 0.16586034
360 -0.09505898 -0.27060771
361 0.48695566 0.14063531
362 -0.51125007 0.17314715
363 -0.51328656 0.16411544
364 -0.02660096 -0.37747199
365 -0.51265053 0.16379868
366 -0.02706361 -0.36177415
367 0.48661595 0.14463254
368 -0.02739921 -0.42327784
369 0.48714593 0.13669515
370 -0.02719841 -0.40200509
371 0.48675316 0.13764749
372 0.48693916 0.13743158
373 0.48715096 0.13702714
374 -0.16878086 -0.18892277
375 -0.02619131 -0.35859311
376 -0.51365279 0.16134746
377 0.48678569 0.14105265
378 0.48691964 0.14040297
379 -0.08816316 -0.28388893
380 -0.51294478 0.16359337
381 0.15290156 -0.19961962
382 -0.51246314 0.16724572
383 -0.09538355 -0.27566479
384 0.48688223 0.13889805
385 -0.51316047 0.16457519
386 0.48712275 0.13856442
387 -0.02739635 -0.42336184
388 -0.51121711 0.16894316
389 0.48725629 0.13753840
390 -0.02810609 -0.41421775
391 0.48704760 0.14004000
392 -0.51271540 0.16433242
393 -0.51278317 0.16355149
394 -0.51223688 0.16778839
395 0.48718956 0.13989236
396 -0.03058124 -0.34978519
397 -0.03125394 -0.35207884
398 0.48718975 0.13584490
399 -0.51191851 0.16934609
400 -0.51209990 0.16761729
401 0.48717265 0.13736091
402 0.07393553 -0.24431152
403 -0.51183214 0.16739802
404 0.48744910 0.13535988
405 -0.51211312 0.16581973
406 0.48695569 0.13833650
407 0.48702684 0.13967578
408 0.48700728 0.14215552

```

##
## $eig
## [1] 7.154072e+01 2.260444e+01 6.142944e+00 5.612111e+00 5.356335e+00
## [6] 4.228210e+00 3.284671e+00 2.922426e+00 2.778434e+00 2.681691e+00
## [11] 2.620385e+00 2.510296e+00 2.467015e+00 2.191005e+00 2.078947e+00
## [16] 1.917182e+00 1.846059e+00 1.792800e+00 1.158628e+00 1.089363e+00
## [21] 1.009891e+00 9.371735e-01 7.206314e-01 5.370035e-01 5.189383e-01
## [26] 4.860600e-01 4.656067e-01 4.281153e-01 4.173279e-01 4.025849e-01
## [31] 3.732467e-01 3.633766e-01 3.491361e-01 3.312614e-01 3.178951e-01
## [36] 3.075031e-01 3.000037e-01 2.946116e-01 2.412091e-01 2.327315e-01
## [41] 2.188737e-01 2.129291e-01 2.051127e-01 1.931579e-01 1.914391e-01
## [46] 1.820200e-01 1.776070e-01 1.684189e-01 1.597512e-01 1.536129e-01
## [51] 1.441120e-01 1.415717e-01 1.328320e-01 1.230696e-01 1.181261e-01
## [56] 1.114740e-01 1.070471e-01 1.032831e-01 1.008210e-01 9.480377e-02
## [61] 8.729672e-02 8.323205e-02 8.137579e-02 7.295963e-02 6.890010e-02
## [66] 6.741029e-02 6.549907e-02 6.195604e-02 5.916309e-02 5.706993e-02
## [71] 5.591420e-02 5.325271e-02 4.949706e-02 4.795811e-02 4.547968e-02
## [76] 4.126039e-02 3.862348e-02 3.738781e-02 3.580720e-02 3.210857e-02
## [81] 3.044963e-02 2.509894e-02 2.246196e-02 2.104936e-02 1.778965e-02
## [86] 1.391910e-02 1.264778e-02 1.204355e-02 1.131545e-02 9.546429e-03
## [91] 7.541357e-03 5.573194e-03 2.725189e-03 2.064650e-03 6.078363e-04
## [96] 4.138645e-15 2.634612e-15 1.175688e-15 6.330920e-16 5.890808e-16
## [101] 5.555422e-16 5.402404e-16 5.172120e-16 4.907629e-16 4.615918e-16
## [106] 3.442972e-16 3.417240e-16 2.987823e-16 2.921588e-16 2.908981e-16
## [111] 2.622128e-16 2.611105e-16 2.364166e-16 1.913130e-16 1.790187e-16
## [116] 1.768161e-16 1.640063e-16 1.598318e-16 1.503600e-16 1.485370e-16
## [121] 1.372126e-16 1.332630e-16 1.312929e-16 1.299645e-16 1.134943e-16
## [126] 1.105201e-16 1.095396e-16 1.095218e-16 1.049813e-16 9.747550e-17
## [131] 8.969762e-17 8.102185e-17 7.953630e-17 6.845039e-17 6.039355e-17
## [136] 5.537779e-17 5.250938e-17 4.949924e-17 4.869363e-17 4.850911e-17
## [141] 4.301308e-17 3.954418e-17 3.949639e-17 3.671258e-17 3.628826e-17
## [146] 3.570248e-17 3.564546e-17 3.521152e-17 3.488612e-17 3.450431e-17
## [151] 2.988645e-17 2.671491e-17 2.645389e-17 2.613598e-17 2.484533e-17
## [156] 2.403978e-17 2.401784e-17 2.379152e-17 2.362956e-17 2.359985e-17
## [161] 2.327335e-17 2.236586e-17 2.154002e-17 2.149519e-17 2.028517e-17
## [166] 2.007724e-17 1.977315e-17 1.840639e-17 1.828335e-17 1.773784e-17
## [171] 1.631547e-17 1.628754e-17 1.614066e-17 1.602094e-17 1.596290e-17
## [176] 1.588612e-17 1.578621e-17 1.570502e-17 1.545980e-17 1.435951e-17
## [181] 1.397178e-17 1.387791e-17 1.383539e-17 1.360247e-17 1.358778e-17
## [186] 1.327493e-17 1.289235e-17 1.202899e-17 1.068537e-17 1.060518e-17
## [191] 8.424352e-18 8.386574e-18 8.060100e-18 7.920085e-18 7.090664e-18
## [196] 6.782495e-18 6.749501e-18 6.537259e-18 6.375203e-18 5.896398e-18
## [201] 5.699499e-18 5.495396e-18 5.149090e-18 4.894049e-18 4.728270e-18
## [206] 4.648356e-18 4.642101e-18 4.489704e-18 4.250791e-18 4.056572e-18
## [211] 3.416414e-18 3.160013e-18 3.102442e-18 3.002409e-18 2.857604e-18
## [216] 2.639762e-18 1.707803e-18 1.611843e-18 1.227324e-18 1.129270e-18
## [221] 9.446178e-19 8.847936e-19 7.569917e-19 6.290560e-19 5.945541e-19
## [226] 5.343330e-19 4.243650e-19 -9.359642e-19 -1.226441e-18 -2.190531e-18
## [231] -2.278966e-18 -2.340581e-18 -2.570205e-18 -2.655616e-18 -2.946622e-18
## [236] -4.910198e-18 -4.920260e-18 -4.940601e-18 -5.118634e-18 -5.390240e-18
## [241] -5.766580e-18 -5.819451e-18 -6.115810e-18 -7.519902e-18 -7.760665e-18
## [246] -8.236694e-18 -8.447660e-18 -8.452402e-18 -8.884159e-18 -8.978156e-18
## [251] -9.614729e-18 -9.624937e-18 -9.695860e-18 -1.041532e-17 -1.159739e-17
## [256] -1.186978e-17 -1.212565e-17 -1.218597e-17 -1.229858e-17 -1.274439e-17

```

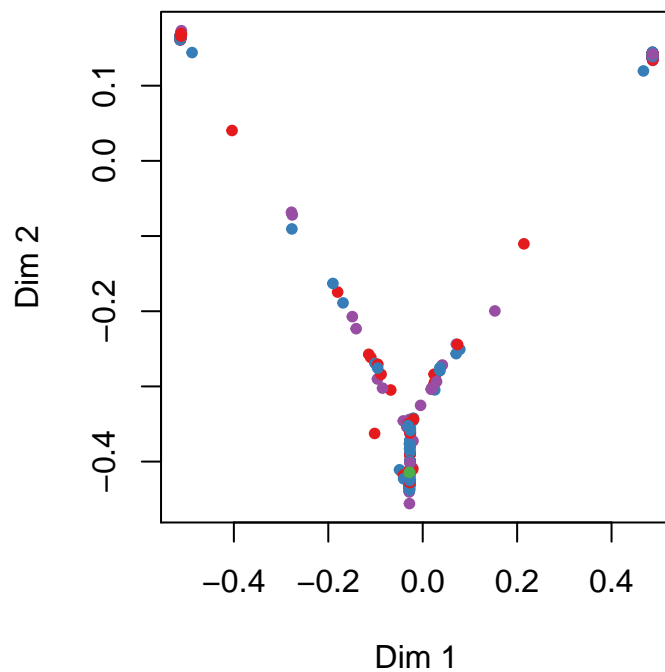


```

## [261] -1.416147e-17 -1.465689e-17 -1.476018e-17 -1.541887e-17 -1.740426e-17
## [266] -1.773165e-17 -1.776412e-17 -1.835683e-17 -1.876019e-17 -1.909112e-17
## [271] -1.965292e-17 -1.988369e-17 -1.996469e-17 -2.003620e-17 -2.043187e-17
## [276] -2.044032e-17 -2.067933e-17 -2.133117e-17 -2.156873e-17 -2.186353e-17
## [281] -2.301569e-17 -2.504849e-17 -2.506474e-17 -2.565723e-17 -2.574737e-17
## [286] -2.653549e-17 -2.704155e-17 -2.956625e-17 -3.242186e-17 -3.294802e-17
## [291] -3.442858e-17 -3.507684e-17 -3.546032e-17 -3.547032e-17 -3.634026e-17
## [296] -3.859902e-17 -3.894960e-17 -4.421730e-17 -4.687827e-17 -4.808756e-17
## [301] -5.038506e-17 -5.419574e-17 -5.423806e-17 -5.994962e-17 -6.871194e-17
## [306] -8.465594e-17 -1.041368e-16 -1.185283e-16 -1.193142e-16 -1.328431e-16
## [311] -1.360447e-16 -1.860684e-16 -1.980117e-16 -2.004490e-16 -2.195498e-16
## [316] -2.204128e-16 -2.574140e-16 -3.302231e-16 -3.419604e-16 -3.422964e-16
## [321] -3.660549e-16 -5.583986e-16 -5.773028e-16 -6.745385e-16 -1.227583e-15
## [326] -1.352065e-15 -2.753607e-15 -3.571514e-15 -3.206919e-05 -8.623957e-04
## [331] -1.252205e-03 -1.545425e-03 -1.849052e-03 -2.163410e-03 -5.078032e-03
## [336] -5.698310e-03 -7.468272e-03 -8.654237e-03 -8.960363e-03 -1.093129e-02
## [341] -1.156955e-02 -1.289780e-02 -1.392421e-02 -1.759915e-02 -1.794018e-02
## [346] -2.103592e-02 -2.547492e-02 -2.662051e-02 -2.929132e-02 -3.447477e-02
## [351] -3.711743e-02 -3.803192e-02 -3.994176e-02 -4.228982e-02 -4.542149e-02
## [356] -4.775701e-02 -4.939157e-02 -5.116449e-02 -5.546252e-02 -5.848408e-02
## [361] -5.908713e-02 -6.361706e-02 -6.521525e-02 -6.789260e-02 -7.036070e-02
## [366] -7.460343e-02 -7.675212e-02 -7.811232e-02 -7.929642e-02 -8.494227e-02
## [371] -8.533985e-02 -8.748983e-02 -9.005717e-02 -9.519660e-02 -1.023627e-01
## [376] -1.055915e-01 -1.110482e-01 -1.140106e-01 -1.209387e-01 -1.239211e-01
## [381] -1.344748e-01 -1.390099e-01 -1.458075e-01 -1.608968e-01 -1.631702e-01
## [386] -1.691948e-01 -1.771840e-01 -1.832890e-01 -1.986827e-01 -2.060067e-01
## [391] -2.149887e-01 -2.215848e-01 -2.288375e-01 -2.351471e-01 -2.394344e-01
## [396] -2.547218e-01 -2.749835e-01 -2.761848e-01 -2.912567e-01 -3.011581e-01
## [401] -3.464090e-01 -3.643249e-01 -3.815793e-01 -3.987337e-01 -4.147181e-01
## [406] -4.326975e-01 -4.482354e-01 -5.330642e-01
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.5529590 0.5875442

```

```
(MDIM_teste = MDSplot(rf3, db_modelo$DIRETORIA, pch=20))
```



```
## $points
##      Dim 1      Dim 2
## 1  0.48706199 0.13832372
## 2 -0.51365242 0.16158370
## 3 -0.02750134 -0.42840656
## 4 -0.51421632 0.16082709
## 5  0.48679248 0.14277997
## 6 -0.02131776 -0.40854075
## 7  0.48693772 0.14015690
## 8 -0.51327875 0.16319477
## 9  0.48705865 0.14047510
## 10 0.48701012 0.13644992
## 11 -0.02117122 -0.41037581
## 12 -0.51238672 0.16591077
## 13 -0.51271311 0.16600004
## 14 -0.10991013 -0.26153938
## 15  0.01947657 -0.30229950
## 16  0.48667965 0.14432116
## 17  0.48686064 0.14103881
## 18 -0.51238717 0.16834268
## 19 -0.02699864 -0.39933997
## 20  0.04154658 -0.27156286
## 21  0.48713386 0.14038301
## 22  0.02551741 -0.30464243
## 23  0.48718274 0.13896670
## 24 -0.02648562 -0.37242162
## 25  0.48675624 0.13981756
## 26  0.48709071 0.14046916
## 27  0.48715317 0.13510545
## 28 -0.51313117 0.16543880
## 29 -0.51276292 0.16514584
## 30  0.02682511 -0.28482271
```

```

## 31 -0.02757584 -0.43194056
## 32 -0.51272403 0.16316354
## 33 0.48690474 0.14113806
## 34 -0.02865857 -0.43591082
## 35 -0.51303738 0.16424189
## 36 -0.51248334 0.16749880
## 37 -0.51215508 0.16600790
## 38 -0.10183763 -0.36258945
## 39 0.48702333 0.13658432
## 40 0.48693958 0.13649018
## 41 -0.51237553 0.16691228
## 42 -0.51234092 0.16680098
## 43 -0.02699718 -0.39929196
## 44 -0.02726683 -0.39099294
## 45 -0.02730537 -0.41886757
## 46 -0.51219454 0.16611299
## 47 -0.02850523 -0.44039193
## 48 0.48717604 0.13995521
## 49 0.02412469 -0.28381847
## 50 0.48681814 0.14171034
## 51 -0.51338052 0.16472197
## 52 0.48688923 0.13921130
## 53 0.48699855 0.14329725
## 54 -0.51225865 0.16581270
## 55 0.48712351 0.13739097
## 56 0.48690213 0.14373172
## 57 -0.51294988 0.16102077
## 58 -0.02723839 -0.39134711
## 59 0.48697233 0.14004293
## 60 -0.02695162 -0.38182752
## 61 -0.02815733 -0.34394660
## 62 -0.51216657 0.16749710
## 63 -0.02636481 -0.36192195
## 64 0.48711134 0.14052867
## 65 -0.51293719 0.16657409
## 66 -0.51261710 0.16646151
## 67 -0.51286110 0.16721306
## 68 -0.51261819 0.16534741
## 69 0.48689809 0.14003513
## 70 -0.51301522 0.16502153
## 71 0.48681471 0.13509451
## 72 0.48665569 0.14257094
## 73 -0.51285631 0.16478926
## 74 0.48689714 0.14304658
## 75 -0.51278494 0.16647662
## 76 -0.51296779 0.16372824
## 77 0.48670241 0.14252392
## 78 0.48693608 0.14052451
## 79 -0.51129826 0.16881451
## 80 0.02437334 -0.29657388
## 81 0.48686531 0.14342124
## 82 -0.02677348 -0.37428525
## 83 0.48742713 0.13940288
## 84 0.48719106 0.13718388

```

85 -0.51166458 0.16799148
86 -0.01981900 -0.34443188
87 0.48728385 0.13744844
88 0.48689565 0.13746035
89 -0.51305557 0.16272527
90 0.48678616 0.13935915
91 -0.02698745 -0.39889988
92 -0.27634708 -0.07189875
93 0.48706546 0.14057651
94 -0.02803162 -0.37616517
95 -0.51236972 0.16700094
96 -0.51248995 0.16771292
97 0.48668713 0.14096192
98 -0.51218448 0.16823943
99 -0.51247077 0.16662417
100 0.46752282 0.11960607
101 -0.51371125 0.16555960
102 0.48683060 0.14260857
103 -0.51351012 0.16167892
104 -0.51283980 0.16433681
105 0.48692979 0.14111019
106 0.48682207 0.14188806
107 -0.51198205 0.16808075
108 -0.51152081 0.16823935
109 -0.02764897 -0.41701336
110 0.48693682 0.14202035
111 0.48731521 0.13856962
112 -0.02677750 -0.37115267
113 -0.51363921 0.16334882
114 0.48677564 0.13951098
115 -0.51361237 0.16212182
116 -0.02658209 -0.35632541
117 0.01799079 -0.30383719
118 -0.51272649 0.16564816
119 0.48683120 0.13878201
120 -0.51267492 0.16587556
121 -0.02742694 -0.42470876
122 -0.51392212 0.16213507
123 -0.02879347 -0.43308084
124 -0.51241930 0.16410973
125 0.48675023 0.14353522
126 -0.02753120 -0.42989795
127 0.48706838 0.14217114
128 0.48712265 0.14099518
129 0.48710370 0.13943988
130 0.48685602 0.14219728
131 0.48733089 0.13590943
132 -0.14098885 -0.22321685
133 0.48716762 0.13835724
134 0.07084709 -0.25643887
135 0.48677467 0.13654052
136 -0.51252582 0.16673436
137 0.48688807 0.13876374
138 -0.51259265 0.16755862

139 0.48713394 0.13804966
140 0.48689335 0.14119695
141 -0.02679312 -0.38985238
142 -0.51175094 0.16676012
143 -0.02613236 -0.35592516
144 0.48705429 0.14231249
145 -0.04107436 -0.41519562
146 -0.02717412 -0.41263674
147 0.48686004 0.13979715
148 0.48699291 0.14004156
149 -0.02886886 -0.42313365
150 0.02501296 -0.29475983
151 -0.51162990 0.17031526
152 0.48702195 0.14039900
153 -0.51274420 0.16436318
154 0.48722826 0.13801171
155 0.01968680 -0.30226424
156 -0.02632681 -0.34668667
157 -0.02691977 -0.38775423
158 -0.02033183 -0.34543060
159 -0.51261479 0.16550819
160 -0.02865173 -0.42796055
161 -0.09585502 -0.29015171
162 0.48745620 0.13741749
163 0.48737790 0.13555559
164 -0.02863077 -0.43218836
165 0.48697541 0.13870282
166 -0.04916135 -0.41089847
167 -0.02639562 -0.36298240
168 -0.14903641 -0.20720847
169 -0.51305531 0.16514226
170 -0.02847864 -0.43882395
171 -0.51249729 0.16467793
172 -0.02879449 -0.43201723
173 -0.06775527 -0.30478065
174 0.48702931 0.14067225
175 -0.51275935 0.16719808
176 -0.51352657 0.16129565
177 -0.02779315 -0.38284137
178 -0.02807487 -0.45583784
179 0.07136452 -0.24386469
180 0.48684386 0.13918389
181 0.48715704 0.13719982
182 -0.11434129 -0.25734655
183 -0.10089361 -0.26946633
184 -0.51212758 0.16576037
185 0.48686902 0.14014498
186 -0.02744236 -0.42535800
187 -0.51261937 0.16571058
188 0.48703868 0.13783354
189 -0.51331720 0.16348032
190 -0.51312792 0.16765108
191 -0.01962682 -0.34283921
192 0.48728429 0.13859410

193 -0.51247188 0.16625739
194 -0.51304480 0.16390961
195 -0.02766244 -0.43581360
196 0.48710408 0.14173396
197 -0.02701554 -0.40008268
198 -0.18020933 -0.17451139
199 -0.01988743 -0.34229844
200 0.48670003 0.14136218
201 0.48683441 0.13922578
202 -0.51226991 0.16962762
203 -0.51325596 0.16629632
204 -0.51203049 0.16768635
205 0.48711727 0.14079768
206 0.48712167 0.13741070
207 0.48716227 0.13763595
208 0.48710292 0.13684606
209 -0.40387636 0.04037033
210 -0.51266389 0.16541010
211 -0.51281341 0.16555359
212 0.48677849 0.14145723
213 0.48684726 0.13547234
214 -0.51200852 0.16593581
215 0.48682317 0.13805461
216 -0.02755464 -0.43102921
217 0.48736030 0.13359581
218 -0.51266232 0.16714897
219 -0.02652428 -0.37109718
220 -0.04090641 -0.41806977
221 -0.02699519 -0.39921547
222 0.48729329 0.13858438
223 -0.51251745 0.16645630
224 0.48683906 0.14268678
225 -0.02755466 -0.43088461
226 0.02716698 -0.28448030
227 -0.51376070 0.16132913
228 0.48731155 0.13780587
229 0.48684166 0.14294897
230 0.48707262 0.14009725
231 -0.51347531 0.16423083
232 -0.51363359 0.16350513
233 0.48708318 0.13805878
234 0.48716188 0.13863956
235 -0.02770025 -0.43770354
236 0.48677469 0.14311460
237 0.03658463 -0.27908817
238 -0.51289085 0.16455586
239 0.48726335 0.13824146
240 0.48728391 0.13978108
241 -0.51240042 0.16728700
242 -0.51255067 0.16803133
243 0.48661658 0.14327695
244 -0.27794539 -0.06834729
245 -0.02670451 -0.39963091
246 -0.04042645 -0.42286147

247 -0.51308664 0.16613406
248 -0.51244523 0.16753746
249 0.48721608 0.13665123
250 -0.51288278 0.16522315
251 0.48698186 0.14368084
252 0.48668045 0.14265837
253 -0.51141931 0.16694052
254 -0.00453419 -0.32520622
255 -0.51267141 0.16733883
256 0.48661624 0.14016976
257 -0.51310350 0.16410470
258 0.07864277 -0.25073087
259 0.48691045 0.14393707
260 0.48734348 0.13952489
261 -0.51225584 0.16716570
262 -0.51289831 0.16586784
263 -0.51258181 0.16742951
264 -0.08511403 -0.30224045
265 -0.02757866 -0.43211185
266 0.48697287 0.13631054
267 0.48681327 0.14154556
268 0.48698168 0.14379076
269 -0.51305795 0.16412834
270 0.48707777 0.14042342
271 -0.02748428 -0.42754660
272 -0.02647677 -0.37189129
273 0.48687838 0.14291487
274 0.48696233 0.14082511
275 -0.51273349 0.16961432
276 -0.02699344 -0.39913739
277 -0.02730852 -0.41914205
278 -0.51245495 0.16489852
279 -0.51277281 0.16394382
280 0.48742279 0.13835302
281 -0.27716911 -0.09055552
282 -0.51245236 0.16676575
283 0.48679742 0.14373217
284 -0.02679408 -0.38712427
285 -0.51220282 0.16528078
286 -0.51230287 0.16663843
287 -0.02054016 -0.37254261
288 0.48684735 0.14108828
289 -0.51236937 0.16343541
290 -0.51317466 0.16501285
291 -0.04112937 -0.34588925
292 -0.02624756 -0.36120923
293 0.48687318 0.14058910
294 -0.02699516 -0.39922389
295 0.48684550 0.14314310
296 0.48694506 0.13592962
297 -0.02629063 -0.36303073
298 -0.51231005 0.16629853
299 -0.02662651 -0.37852043
300 0.48688854 0.14094432

301 -0.02805030 -0.41113155
302 -0.51264168 0.16687987
303 -0.51135055 0.17030416
304 0.48694605 0.14209908
305 0.48744217 0.13773746
306 -0.02744578 -0.42550302
307 0.48683741 0.14298847
308 -0.19025785 -0.16304372
309 0.48702921 0.13942301
310 -0.02746720 -0.42684704
311 0.48725663 0.13868637
312 -0.51243092 0.16680334
313 0.03659801 -0.27495730
314 -0.51265628 0.16573157
315 0.48675151 0.14032994
316 -0.51252227 0.16784669
317 -0.01965163 -0.34370677
318 0.48724528 0.14029093
319 -0.51168456 0.16922867
320 -0.51300243 0.16404932
321 -0.51206755 0.16795136
322 0.48724421 0.14200713
323 -0.03376552 -0.35394892
324 0.48683046 0.13894574
325 -0.51243970 0.16823839
326 0.48707520 0.13666069
327 -0.51247426 0.16856061
328 0.48675715 0.14407659
329 -0.51261620 0.16400547
330 -0.51255017 0.16624650
331 -0.51247536 0.16841587
332 -0.51301859 0.16387049
333 -0.02700314 -0.39956895
334 -0.51309135 0.16386227
335 0.48702901 0.13527142
336 -0.51204534 0.16920909
337 0.02937912 -0.29359522
338 -0.51241516 0.16826474
339 -0.02700374 -0.39955138
340 -0.51290371 0.16321342
341 -0.02748788 -0.42785094
342 0.48726679 0.13646004
343 -0.51282513 0.16644322
344 0.48712265 0.13797419
345 -0.48852491 0.14403628
346 -0.51258047 0.16689961
347 -0.51282179 0.16427595
348 -0.51220253 0.16461256
349 0.48708182 0.13810691
350 0.48678353 0.14102360
351 -0.02624421 -0.36100305
352 0.48697290 0.14221124
353 -0.02743446 -0.42505979
354 0.21438274 -0.11038038

355 0.48658705 0.13946612
356 -0.51258418 0.16887355
357 0.48682528 0.14276050
358 0.48722649 0.13412936
359 -0.51285970 0.16586034
360 -0.09505898 -0.27060771
361 0.48695566 0.14063531
362 -0.51125007 0.17314715
363 -0.51328656 0.16411544
364 -0.02660096 -0.37747199
365 -0.51265053 0.16379868
366 -0.02706361 -0.36177415
367 0.48661595 0.14463254
368 -0.02739921 -0.42327784
369 0.48714593 0.13669515
370 -0.02719841 -0.40200509
371 0.48675316 0.13764749
372 0.48693916 0.13743158
373 0.48715096 0.13702714
374 -0.16878086 -0.18892277
375 -0.02619131 -0.35859311
376 -0.51365279 0.16134746
377 0.48678569 0.14105265
378 0.48691964 0.14040297
379 -0.08816316 -0.28388893
380 -0.51294478 0.16359337
381 0.15290156 -0.19961962
382 -0.51246314 0.16724572
383 -0.09538355 -0.27566479
384 0.48688223 0.13889805
385 -0.51316047 0.16457519
386 0.48712275 0.13856442
387 -0.02739635 -0.42336184
388 -0.51121711 0.16894316
389 0.48725629 0.13753840
390 -0.02810609 -0.41421775
391 0.48704760 0.14004000
392 -0.51271540 0.16433242
393 -0.51278317 0.16355149
394 -0.51223688 0.16778839
395 0.48718956 0.13989236
396 -0.03058124 -0.34978519
397 -0.03125394 -0.35207884
398 0.48718975 0.13584490
399 -0.51191851 0.16934609
400 -0.51209990 0.16761729
401 0.48717265 0.13736091
402 0.07393553 -0.24431152
403 -0.51183214 0.16739802
404 0.48744910 0.13535988
405 -0.51211312 0.16581973
406 0.48695569 0.13833650
407 0.48702684 0.13967578
408 0.48700728 0.14215552

```

##
## $eig
## [1] 7.154072e+01 2.260444e+01 6.142944e+00 5.612111e+00 5.356335e+00
## [6] 4.228210e+00 3.284671e+00 2.922426e+00 2.778434e+00 2.681691e+00
## [11] 2.620385e+00 2.510296e+00 2.467015e+00 2.191005e+00 2.078947e+00
## [16] 1.917182e+00 1.846059e+00 1.792800e+00 1.158628e+00 1.089363e+00
## [21] 1.009891e+00 9.371735e-01 7.206314e-01 5.370035e-01 5.189383e-01
## [26] 4.860600e-01 4.656067e-01 4.281153e-01 4.173279e-01 4.025849e-01
## [31] 3.732467e-01 3.633766e-01 3.491361e-01 3.312614e-01 3.178951e-01
## [36] 3.075031e-01 3.000037e-01 2.946116e-01 2.412091e-01 2.327315e-01
## [41] 2.188737e-01 2.129291e-01 2.051127e-01 1.931579e-01 1.914391e-01
## [46] 1.820200e-01 1.776070e-01 1.684189e-01 1.597512e-01 1.536129e-01
## [51] 1.441120e-01 1.415717e-01 1.328320e-01 1.230696e-01 1.181261e-01
## [56] 1.114740e-01 1.070471e-01 1.032831e-01 1.008210e-01 9.480377e-02
## [61] 8.729672e-02 8.323205e-02 8.137579e-02 7.295963e-02 6.890010e-02
## [66] 6.741029e-02 6.549907e-02 6.195604e-02 5.916309e-02 5.706993e-02
## [71] 5.591420e-02 5.325271e-02 4.949706e-02 4.795811e-02 4.547968e-02
## [76] 4.126039e-02 3.862348e-02 3.738781e-02 3.580720e-02 3.210857e-02
## [81] 3.044963e-02 2.509894e-02 2.246196e-02 2.104936e-02 1.778965e-02
## [86] 1.391910e-02 1.264778e-02 1.204355e-02 1.131545e-02 9.546429e-03
## [91] 7.541357e-03 5.573194e-03 2.725189e-03 2.064650e-03 6.078363e-04
## [96] 4.138645e-15 2.634612e-15 1.175688e-15 6.330920e-16 5.890808e-16
## [101] 5.555422e-16 5.402404e-16 5.172120e-16 4.907629e-16 4.615918e-16
## [106] 3.442972e-16 3.417240e-16 2.987823e-16 2.921588e-16 2.908981e-16
## [111] 2.622128e-16 2.611105e-16 2.364166e-16 1.913130e-16 1.790187e-16
## [116] 1.768161e-16 1.640063e-16 1.598318e-16 1.503600e-16 1.485370e-16
## [121] 1.372126e-16 1.332630e-16 1.312929e-16 1.299645e-16 1.134943e-16
## [126] 1.105201e-16 1.095396e-16 1.095218e-16 1.049813e-16 9.747550e-17
## [131] 8.969762e-17 8.102185e-17 7.953630e-17 6.845039e-17 6.039355e-17
## [136] 5.537779e-17 5.250938e-17 4.949924e-17 4.869363e-17 4.850911e-17
## [141] 4.301308e-17 3.954418e-17 3.949639e-17 3.671258e-17 3.628826e-17
## [146] 3.570248e-17 3.564546e-17 3.521152e-17 3.488612e-17 3.450431e-17
## [151] 2.988645e-17 2.671491e-17 2.645389e-17 2.613598e-17 2.484533e-17
## [156] 2.403978e-17 2.401784e-17 2.379152e-17 2.362956e-17 2.359985e-17
## [161] 2.327335e-17 2.236586e-17 2.154002e-17 2.149519e-17 2.028517e-17
## [166] 2.007724e-17 1.977315e-17 1.840639e-17 1.828335e-17 1.773784e-17
## [171] 1.631547e-17 1.628754e-17 1.614066e-17 1.602094e-17 1.596290e-17
## [176] 1.588612e-17 1.578621e-17 1.570502e-17 1.545980e-17 1.435951e-17
## [181] 1.397178e-17 1.387791e-17 1.383539e-17 1.360247e-17 1.358778e-17
## [186] 1.327493e-17 1.289235e-17 1.202899e-17 1.068537e-17 1.060518e-17
## [191] 8.424352e-18 8.386574e-18 8.060100e-18 7.920085e-18 7.090664e-18
## [196] 6.782495e-18 6.749501e-18 6.537259e-18 6.375203e-18 5.896398e-18
## [201] 5.699499e-18 5.495396e-18 5.149090e-18 4.894049e-18 4.728270e-18
## [206] 4.648356e-18 4.642101e-18 4.489704e-18 4.250791e-18 4.056572e-18
## [211] 3.416414e-18 3.160013e-18 3.102442e-18 3.002409e-18 2.857604e-18
## [216] 2.639762e-18 1.707803e-18 1.611843e-18 1.227324e-18 1.129270e-18
## [221] 9.446178e-19 8.847936e-19 7.569917e-19 6.290560e-19 5.945541e-19
## [226] 5.343330e-19 4.243650e-19 -9.359642e-19 -1.226441e-18 -2.190531e-18
## [231] -2.278966e-18 -2.340581e-18 -2.570205e-18 -2.655616e-18 -2.946622e-18
## [236] -4.910198e-18 -4.920260e-18 -4.940601e-18 -5.118634e-18 -5.390240e-18
## [241] -5.766580e-18 -5.819451e-18 -6.115810e-18 -7.519902e-18 -7.760665e-18
## [246] -8.236694e-18 -8.447660e-18 -8.452402e-18 -8.884159e-18 -8.978156e-18
## [251] -9.614729e-18 -9.624937e-18 -9.695860e-18 -1.041532e-17 -1.159739e-17
## [256] -1.186978e-17 -1.212565e-17 -1.218597e-17 -1.229858e-17 -1.274439e-17

```

```

## [261] -1.416147e-17 -1.465689e-17 -1.476018e-17 -1.541887e-17 -1.740426e-17
## [266] -1.773165e-17 -1.776412e-17 -1.835683e-17 -1.876019e-17 -1.909112e-17
## [271] -1.965292e-17 -1.988369e-17 -1.996469e-17 -2.003620e-17 -2.043187e-17
## [276] -2.044032e-17 -2.067933e-17 -2.133117e-17 -2.156873e-17 -2.186353e-17
## [281] -2.301569e-17 -2.504849e-17 -2.506474e-17 -2.565723e-17 -2.574737e-17
## [286] -2.653549e-17 -2.704155e-17 -2.956625e-17 -3.242186e-17 -3.294802e-17
## [291] -3.442858e-17 -3.507684e-17 -3.546032e-17 -3.547032e-17 -3.634026e-17
## [296] -3.859902e-17 -3.894960e-17 -4.421730e-17 -4.687827e-17 -4.808756e-17
## [301] -5.038506e-17 -5.419574e-17 -5.423806e-17 -5.994962e-17 -6.871194e-17
## [306] -8.465594e-17 -1.041368e-16 -1.185283e-16 -1.193142e-16 -1.328431e-16
## [311] -1.360447e-16 -1.860684e-16 -1.980117e-16 -2.004490e-16 -2.195498e-16
## [316] -2.204128e-16 -2.574140e-16 -3.302231e-16 -3.419604e-16 -3.422964e-16
## [321] -3.660549e-16 -5.583986e-16 -5.773028e-16 -6.745385e-16 -1.227583e-15
## [326] -1.352065e-15 -2.753607e-15 -3.571514e-15 -3.206919e-05 -8.623957e-04
## [331] -1.252205e-03 -1.545425e-03 -1.849052e-03 -2.163410e-03 -5.078032e-03
## [336] -5.698310e-03 -7.468272e-03 -8.654237e-03 -8.960363e-03 -1.093129e-02
## [341] -1.156955e-02 -1.289780e-02 -1.392421e-02 -1.759915e-02 -1.794018e-02
## [346] -2.103592e-02 -2.547492e-02 -2.662051e-02 -2.929132e-02 -3.447477e-02
## [351] -3.711743e-02 -3.803192e-02 -3.994176e-02 -4.228982e-02 -4.542149e-02
## [356] -4.775701e-02 -4.939157e-02 -5.116449e-02 -5.546252e-02 -5.848408e-02
## [361] -5.908713e-02 -6.361706e-02 -6.521525e-02 -6.789260e-02 -7.036070e-02
## [366] -7.460343e-02 -7.675212e-02 -7.811232e-02 -7.929642e-02 -8.494227e-02
## [371] -8.533985e-02 -8.748983e-02 -9.005717e-02 -9.519660e-02 -1.023627e-01
## [376] -1.055915e-01 -1.110482e-01 -1.140106e-01 -1.209387e-01 -1.239211e-01
## [381] -1.344748e-01 -1.390099e-01 -1.458075e-01 -1.608968e-01 -1.631702e-01
## [386] -1.691948e-01 -1.771840e-01 -1.832890e-01 -1.986827e-01 -2.060067e-01
## [391] -2.149887e-01 -2.215848e-01 -2.288375e-01 -2.351471e-01 -2.394344e-01
## [396] -2.547218e-01 -2.749835e-01 -2.761848e-01 -2.912567e-01 -3.011581e-01
## [401] -3.464090e-01 -3.643249e-01 -3.815793e-01 -3.987337e-01 -4.147181e-01
## [406] -4.326975e-01 -4.482354e-01 -5.330642e-01
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.5529590 0.5875442
sum(MDIM_treino$eig[1:2]); # sum(MDIM_teste$eig[1:2]) # is the same

## [1] 94.14515
DIR = db_modelo$DIRETORIA
x = db_modelo[, -1]; colnames(x) <- c(DIR)
x = x[, 1:300]
hchart(princomp(x, cor = FALSE))

```