

# Aplicação da técnica quantitativa do Grupo 4 (vespertino) - Práticas de Pesquisa em Sociologia

Giovanna Claudino de Almeida Silva

2025-05-28

Nesse documento vamos unificar as duas técnicas aplicadas nos dados obtidos pelo grupo 4, do vespertino, da disciplina Práticas de Pesquisa em Sociologia. A nossa pergunta de pesquisa é “Como o G1 enquadrrou a violência política sofrida e cometida por mulheres em cargos eletivos, levando em conta a ideologia de seus partidos?”. Nossa hipótese é que o enquadramento do G1 sobre violência política varia de acordo com o gênero e ideologia política das pessoas ocupantes de cargos no legislativo e executivo envolvidas nestes casos. Isto é, a maneira como a mídia noticia casos de violência política tende a variar a depender do partido político ao qual os envolvidos pertencem, bem como tende a variar dependendo se a violência foi praticada ou sofrida por homens ou mulheres.

Como técnicas quantitativas a serem aplicadas, escolhemos o uso de n-gramas para as notícias, e a frequência de partidos mais comuns nas categorias: mulheres agressoras, homens agressores, mulheres vítimas, e homens vítimas. Nesse arquivo não haverá uma análise das técnicas, só um registro de sua aplicação.

A técnica dos n-gramas consiste em identificar sequências de palavras que aparecem lado a lado, pois ela permite visualizar os padrões linguísticos mais recorrentes no conjunto de textos. Usamos bigramas, que são pares de palavras, para títulos e subtítulos, e trigramas, que são trios de palavras, para o corpo das matérias. Essa separação foi feita porque os títulos sintetizam o que é considerado mais relevante na notícia, enquanto o texto completo oferece elementos mais amplos do enquadramento. Essa técnica será utilizada para entender melhor os enquadramentos das notícias, mas também para fazer um levantamento dos atores mais frequentes nas notícias, e auxiliar a escolha dos casos para a análise qualitativa.

A frequência dos partidos servirá como uma descrição inicial dos dados, permitindo visualizar quais partidos estão mais associados aos episódios de violência. Esses dados quantitativos vão auxiliar na seleção de casos para a análise qualitativa, oferecendo um critério baseado na relevância dos partidos para a escolha dos casos.

**Vamos iniciar com a aplicação da técnica dos n-gramas.** Inicialmente, vamos criar um objeto com as stopwords, e uma função para criar os n-gramas e gerar os gráficos. Eu decidi usar um objeto para as stopwords, e definir ele á parte no começo do código, para facilitar a edição dele. Ou seja, foram criado os gráficos dos n-gramas algumas vezes antes da versão final, e a cada etapa eu editava esse objeto.

```
# Lista de stopwords (truncada aqui, mas mantenha completa no seu script real)
stopwords <- c(
  "a", "as", "o", "os", "um", "uma", "uns", "umas",
  "de", "da", "do", "das", "dos", "em", "na", "no", "nas", "nos", "por", "para",
  "com", "sem", "sobre", "sob", "entre", "contra", "após", "ante", "até", "desde",
  "durante", "mediante", "perante", "segundo", "conforme", "consoante",
  "e", "ou", "mas", "porém", "contudo", "todavia", "entretanto", "no entanto",
  "portanto", "logo", "pois", "porque", "como", "quando", "onde", "enquanto",
  "embora", "ainda que", "mesmo que", "caso", "se", "que", "qual", "quais",
  "eu", "tu", "ele", "ela", "nós", "vós", "eles", "elas", "me", "te", "se", "nos",
  "vos", "meu", "minha", "meus", "minhas", "teu", "tua", "teus", "tuas", "seu", "sua",
  "seus", "suas", "nosso", "nossa", "nossos", "nossas", "vosso", "vossa", "vossos",
```

```

"vossas","este", "esta", "estes", "estas", "esse", "essa", "esses", "essas", "aquele",
"aquela","aqueles", "aquelas", "isto", "isso", "aquilo", "quem", "cujo", "cuja",
"cujos", "cujas", "não", "sim", "já", "ainda", "mais", "menos", "muito", "pouco", "bem",
"mal", "aqui", "ali", "lá", "cá", "aí", "aonde", "assim", "então", "agora", "hoje", "ontem",
"amanhã", "sempre", "nunca", "jamais", "talvez", "acaso", "apenas", "só", "somente",
"também", "inclusive", "ser", "estar", "ter", "haver", "ir", "vir", "fazer", "dar",
"dizer", "ver", "saber", "poder", "querer", "dever", "ficar", "passar", "chegar",
"sair", "entrar", "voltar", "trazer", "levar", "pôr", "tirar", "deixar", "encontrar",
"é", "são", "foi", "foram", "era", "eram", "será", "serão", "seja", "sejam",
"está", "estão", "esteve", "estiveram", "estava", "estavam", "estará", "estarão",
"tem", "têm", "teve", "tiveram", "tinha", "tinham", "terá", "terão", "há", "houve",
"houveram", "havia", "havam", "haverá", "haverão", "segunda", "terça", "quarta", "quinta",
"sexta", "sábado", "domingo", "nesta", "neste", "nessa", "nesse", "dessa", "desse", "até hoje"
)

```

*# Função auxiliar para gerar bigramas ou trigramas, gráfico e salvar*

```

gerar_ngramas <- function(df, nome_base, tipo = "bigrama", cor = "gray40", arquivo_saida) {
  df_filtrado <- df %>%
    mutate(Relevância_num = as.numeric(as.character(Relevância))) %>%
    filter(!is.na(Relevância_num) & Relevância_num != 0)

  # Verificação: número de registros filtrados por relevância
  cat("[", nome_base, "] Total após filtrar por relevância: ", nrow(df_filtrado), "\n")

  if (tipo == "bigrama") {
    df_texto <- df_filtrado %>%
      filter(!is.na(titulo) & is.na(subtitulo)) %>%
      mutate(
        titulo = ifelse(is.na(titulo), "", titulo),
        subtitulo = ifelse(is.na(subtitulo), "", subtitulo),
        texto_combinado = paste(titulo, subtitulo, sep = " ")
      ) %>%
      select(texto_combinado) %>%
      filter(texto_combinado != "" & texto_combinado != " ")

    # Verificação: exemplos de textos combinados
    cat("Exemplo de texto combinado:\n")
    print(head(df_texto$texto_combinado, 3))

    ngram_coluna <- "bigram"
    ngram_n <- 2
    texto_base <- df_texto
    texto_campo <- "texto_combinado"
  } else if (tipo == "trigrama") {
    df_texto <- df_filtrado %>%
      filter(!is.na(conteudo) & conteudo != "" & str_trim(conteudo) != "") %>%
      select(conteudo)

    # Verificação: exemplos de conteúdos válidos
    cat("Exemplo de conteúdo:\n")
    print(head(df_texto$conteudo, 3))

    ngram_coluna <- "trigrama"
  }
}

```

```

ngram_n <- 3
texto_base <- df_texto
texto_campo <- "conteudo"
} else {
  stop("Tipo inválido. Use 'bigrama' ou 'trigrama'.")
}

ngramas <- texto_base %>%
  unnest_tokens(ng, .data[[texto_campo]], token = "ngrams", n = ngram_n) %>%
  separate(ng, paste0("palavra", 1:ngram_n), sep = " ") %>%
  filter(if_all(starts_with("palavra"), ~ !.x %in% stopwords & str_detect(.x, "[[:alpha:]]"))) %>%
  unite(!ngram_coluna, starts_with("palavra"), sep = " ") %>%
  count(!sym(ngram_coluna), sort = TRUE) %>%
  top_n(20, n)

# Verificação: imprimir os top ngramas
cat("\nTop 5 ", tipo, "\n")
print(head(ngramas, 5))

grafico <- ngramas %>%
  mutate(!ngram_coluna := reorder(!sym(ngram_coluna), n)) %>%
  ggplot(aes_string(x = ngram_coluna, y = "n")) +
  geom_col(fill = cor, alpha = 0.8) +
  coord_flip() +
  labs(
    title = paste("Top 20", tipo, "mais frequentes em", nome_base),
    subtitle = ifelse(tipo == "bigrama", "Título + Subtítulo", "Conteúdo das Notícias"),
    x = stringr::str_to_title(tipo),
    y = "Frequência"
  ) +
  theme_minimal()

ggsave(arquivo_saida, grafico, width = 12, height = 8, dpi = 300, bg = "white")
cat("Gráfico salvo em:", arquivo_saida, "\n\n")

return(list(dados = ngramas, grafico = grafico))
}

```

Agora, vamos importar as bases, executar a função nelas e visualizar os gráficos.

```

# Importar bases de dados (ajuste os caminhos conforme necessário)
mulheres_dados <- read_excel("C:/Users/gigic/Downloads/m_vitimas.xlsx")

```

```

## New names:
## * ' ' -> '...21'

```

```

homens_dados <- read_excel("C:/Users/gigic/Downloads/h_vitimas.xlsx")

```

```

## New names:
## * ' ' -> '...21'

```

```

# Executar análises
bigramas_mulheres <- gerar_ngramas(mulheres_dados, "mulheres vítimas", tipo = "bigrama", cor = "darkred")

## [ mulheres vítimas ] Total após filtrar por relevância: 151
## Exemplo de texto combinado:
## [1] "Vídeo mostra confusão com deputada Alê Silva em aeroporto de BH: 'A senhora está infringindo a lei'"
## [2] "Polícia Civil de Porto Alegre investiga denúncia de ameaças contra ex-deputada Manuela D'Ávila"
## [3] "Vereadora de Contagem, na Grande BH, procura polícia após ser vítima de ataques racistas Enquanto isso..."
##
## Top 5 bigrama :
## # A tibble: 5 x 2
##   bigram          n
##   <chr>         <int>
## 1 redes sociais    22
## 2 violência política 17
## 3 isa penna       14
## 4 benny briolly   10
## 5 deputada andréia 10

## Warning: 'aes_string()' was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with 'aes()'.
## i See also 'vignette("ggplot2-in-packages")' for more information.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

## Gráfico salvo em: bigramas_mulheres.png

trigramas_mulheres <- gerar_ngramas(mulheres_dados, "mulheres vítimas", tipo = "trigrama", cor = "indianred")

## [ mulheres vítimas ] Total após filtrar por relevância: 151
## Exemplo de conteúdo:
## [1] "A deputada federal Alessandra Silva (PSL-MG), conhecida como Alê Silva , filmou parte da área de trabalho de uma funcionária da Polícia Civil de Porto Alegre"
## [2] "A Polícia Civil do Rio Grande do Sul confirmou, nesta quinta-feira (3), que a ex-deputada federal Alessandra Silva (PSL-MG) foi ameaçada por uma funcionária da Polícia Civil de Porto Alegre"
## [3] "A vereadora de Contagem, na Região Metropolitana de Belo Horizonte, Moara Sabóia (PT), procurou a Polícia Civil de Porto Alegre para investigar uma ameaça recebida de uma funcionária da Polícia Civil de Porto Alegre"
##
## Top 5 trigrama :
## # A tibble: 5 x 2
##   trigram          n
##   <chr>         <int>
## 1 deputada isa penna    34
## 2 isa penna psol       19
## 3 minas gerais almg     17
## 4 deputada mônica seixas 15
## 5 deputado wellington moura 14

## Gráfico salvo em: trigramas_mulheres.png

bigramas_homens <- gerar_ngramas(homens_dados, "homens vítimas", tipo = "bigrama", cor = "navy", arqu

```

```
## [ homens vítimas ] Total após filtrar por relevância: 46
## Exemplo de texto combinado:
## [1] "Polícia Civil abre investigação sobre vídeo com ameaça a Lula Governador João Doria diz ter det
## [2] "Comitiva de Haddad é atacada com ovos em Cordeirópolis; VÍDEO Câmera de segurança flagrou momen
## [3] "PT aciona Conselho de Ética contra deputado que ameaçou Lula em vídeo com arma dentro da Alesp
##
## Top 5 bigrama :
## # A tibble: 5 x 2
##   bigram          n
##   <chr>         <int>
## 1 câmara municipal      5
## 2 vídeo mostra          5
## 3 ex vereador          4
## 4 maciel santos         4
## 5 polícia civil         4

## Gráfico salvo em: bigramas_homens.png
```

```
trigramas_homens <- gerar_ngramas(homens_dados, "homens vítimas", tipo = "trigrama", cor = "cornflower"
```

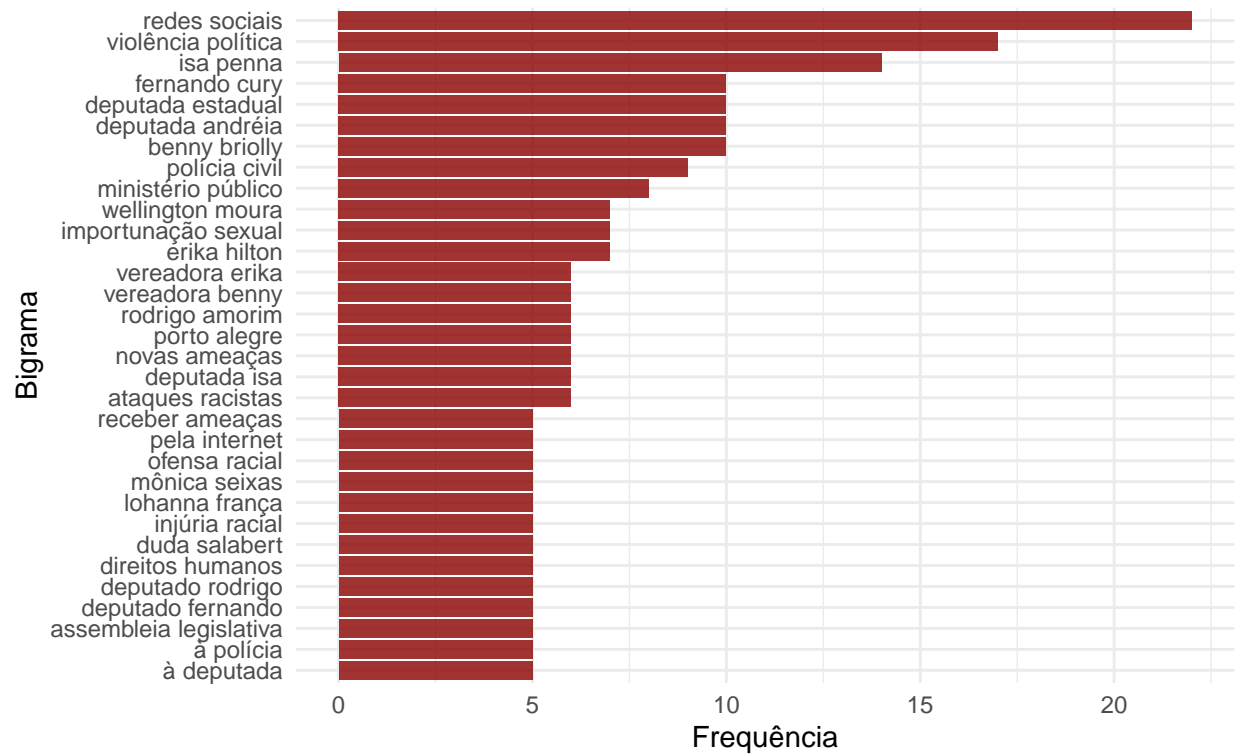
```
## [ homens vítimas ] Total após filtrar por relevância: 46
## Exemplo de conteúdo:
## [1] "O governador de São Paulo, João Doria (PSDB), determinou que a polícia investigue um homem resp
## [2] "A comitiva do candidato a governador de São Paulo Fernando Haddad (PT) foi atacada com ovos dur
## [3] "A bancada de deputados do Partido dos Trabalhadores ( PT ) na Assembleia Legislativa de São Pau
##
## Top 5 trigrama :
## # A tibble: 5 x 2
##   trigram          n
##   <chr>         <int>
## 1 roberto maciel santos      9
## 2 atirou diversas vezes     7
## 3 vereador renato freitas   7
## 4 vereador sidnei toaldo    6
## 5 aline dequi palma         5

## Gráfico salvo em: trigramas_homens.png
```

```
# Exibir gráficos (opcional)
print(bigramas_mulheres$grafico)
```

## Top 20 bigrama mais frequentes em mulheres vítimas

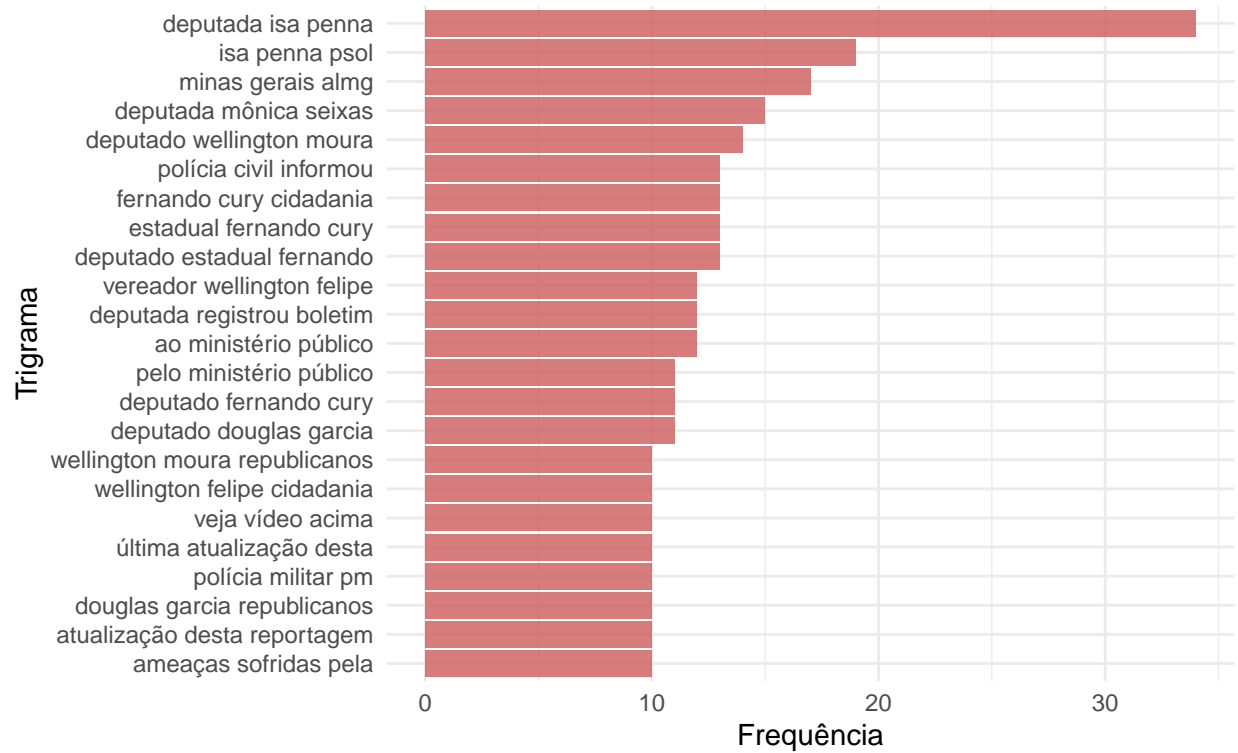
Título + Subtítulo



```
print(trigramas_mulheres$grafico)
```

## Top 20 trigrama mais frequentes em mulheres vítimas

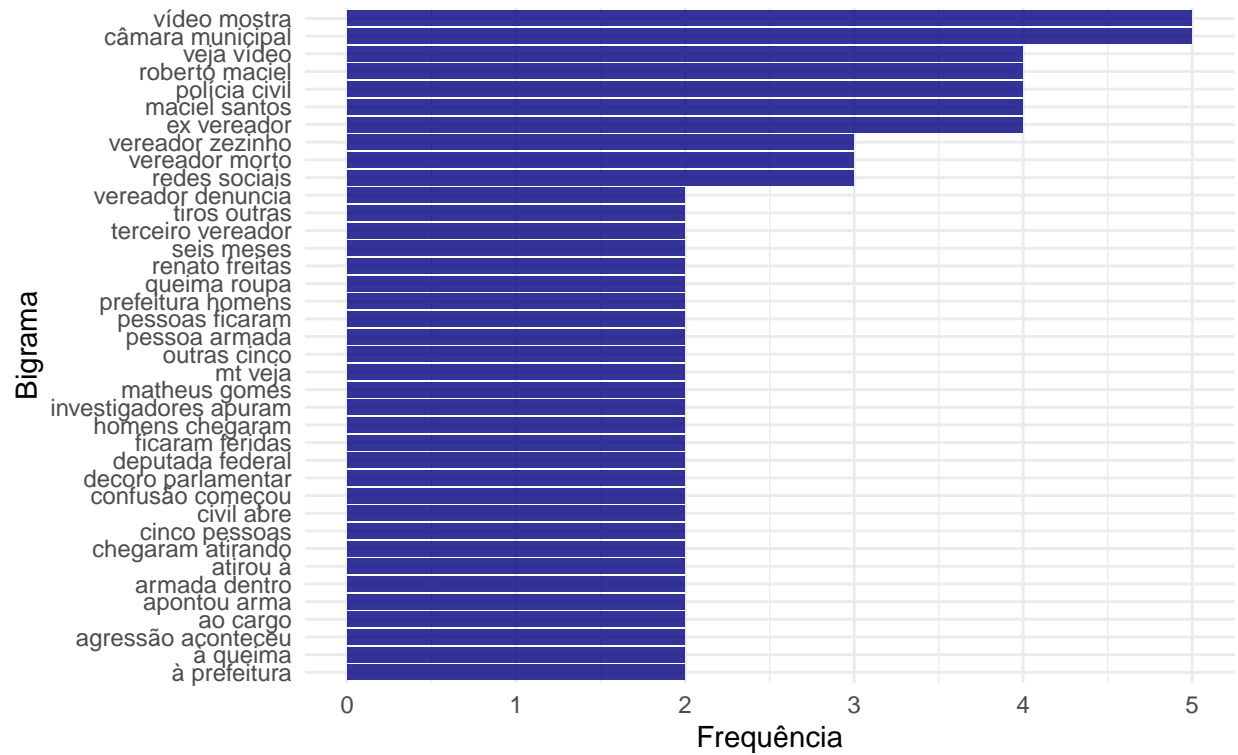
### Conteúdo das Notícias



```
print(bigramas_homens$grafico)
```

## Top 20 bigrama mais frequentes em homens vítimas

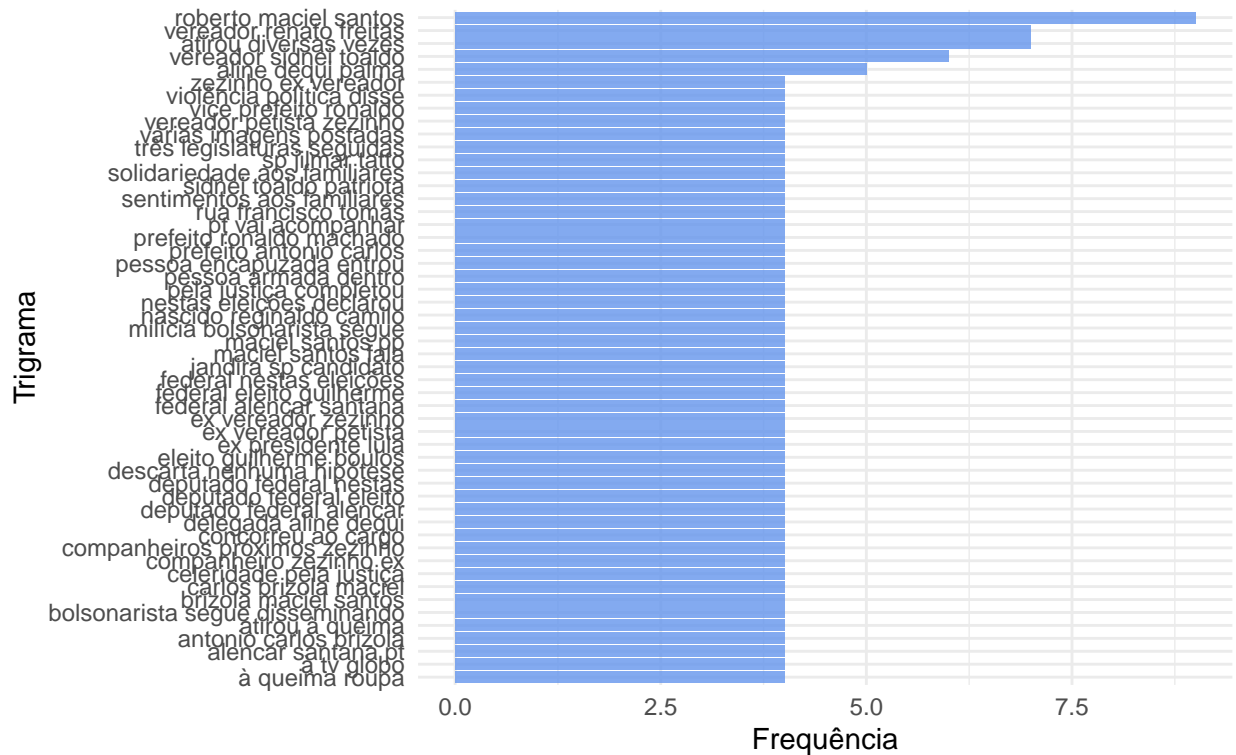
Título + Subtítulo



```
print(trigramas_homens$grafico)
```



## Top 20 trígama mais frequentes em homens vítimas Conteúdo das Notícias



Agora, vamos aplicar a frequência de partidos por categorias (mulheres agressoras, mulheres vítimas, homens agressores, homens vítimas). Inicialmente, vamos juntar as duas bases que usamos na primeira técnica e conferir se a junção ocorreu corretamente.

```
# Juntar os dados
dados_geral <- rbind(homens_dados, mulheres_dados)

# Verificação:
# Verificar número de linhas e colunas após união
print(dim(dados_geral))
```

```
## [1] 198 21
```

```
# Visualizar primeiros registros
head(dados_geral)
```

```
## # A tibble: 6 x 21
##   titulo  subtítulo autor conteúdo regiao data link busca partidos_mencionados
##   <chr>   <chr>    <chr> <chr>   <chr> <chr> <chr> <chr> <chr>
## 1 Políci Governad Por ~ "O gove SP - ~ 15/0~ http~ viol~ PSDB, PT
## 2 Comiti Câmara d Por ~ "A comi SP - ~ 08/0~ http~ viol~ PT
## 3 PT aci Em vídeo Por ~ "A banc SP - ~ 06/0~ http~ viol~ PP, PT
## 4 VÍDEO:~ Ainda nã Por ~ "O pref AM - ~ 02/0~ http~ viol~ UNIÃO
## 5 Corpo ~ Político Por ~ "O corp SP - ~ 29/1~ http~ viol~ PSDB, PSOL, PT
## 6 Políci Zezinho ~ Por ~ "A Polí SP - ~ 29/1~ http~ viol~ PSDB, PSOL, PT
```

```
## # i 12 more variables: Relevância <dbl>, 'Cargo Agressor(a)' <chr>,
## #   'Cargo Agredido(a)' <chr>, 'Partido Agressor' <chr>,
## #   'Partido Agredido' <chr>, 'Nome Agressor(a)' <chr>,
## #   'Nome Agredido(a)' <chr>, 'Tipo de Violência' <chr>,
## #   'Gênero Agressor' <chr>, 'Gênero Agredido' <chr>, Observação <chr>,
## #   ...21 <chr>
```

Agora, vamos aplicar uma função que padronize e limpe nossa base de dados. Por exemplo, se em uma linha foi digitado “Homem” e outra “H”, ou foi digitado “Psol” e outra “PSOL”, será criado novas colunas padronizando isso. Também vamos padronizar os NAs e outras variáveis como “não consta na notícia”, ou “\_”, ou “não se aplica”, em uma variável chamada “Não se aplica ou não encontrado”. Vamos aplicar essa função a base de dados, e verificar como ela ficou depois dessa alteração.

#### *#Função de limpeza e padronização*

```
limpar_dados <- function(df) {
  df %>%
    mutate(
      # Padronizar gênero do agressor
      genero_agressor_pad = case_when(
        is.na(`Gênero Agressor`) | `Gênero Agressor` == "-" ~ "Não se aplica ou não encontrado",
        str_detect(`Gênero Agressor`, "(?i)H") & str_detect(`Gênero Agressor`, "(?i)M") ~ "Ambos",
        str_detect(`Gênero Agressor`, "(?i)H") ~ "Homem",
        str_detect(`Gênero Agressor`, "(?i)M") ~ "Mulher",
        TRUE ~ "Não se aplica ou não encontrado"
      ),

      # Padronizar gênero do agredido
      genero_agredido_pad = case_when(
        is.na(`Gênero Agredido`) | `Gênero Agredido` == "-" ~ "Não se aplica ou não encontrado",
        str_detect(`Gênero Agredido`, "(?i)H") & str_detect(`Gênero Agredido`, "(?i)M") ~ "Ambos",
        str_detect(`Gênero Agredido`, "(?i)H") ~ "Homem",
        str_detect(`Gênero Agredido`, "(?i)M") ~ "Mulher",
        TRUE ~ "Não se aplica ou não encontrado"
      ),

      # Padronizar partidos do agressor
      partido_agressor_pad = tolower(`Partido Agressor`),
      partido_agressor_pad = case_when(
        is.na(partido_agressor_pad) | partido_agressor_pad %in% c("-", "não consta na notícia") ~ "não consta na notícia",
        partido_agressor_pad %in% c("progressistas") ~ "progressistas",
        partido_agressor_pad %in% c("cidadania sp", "cidadania") ~ "cidadania",
        partido_agressor_pad %in% c("republicanos", "republicanos ") ~ "republicanos",
        partido_agressor_pad %in% c("psol/republicanos") ~ "psol e republicanos",
        partido_agressor_pad %in% c("psol/") ~ "psol",
        TRUE ~ partido_agressor_pad
      ),

      # Padronizar partidos do agredido
      partido_agredido_pad = tolower(`Partido Agredido`),
      partido_agredido_pad = case_when(
        is.na(partido_agredido_pad) | partido_agredido_pad %in% c("-", "não consta na notícia") ~ "não consta na notícia",
        partido_agredido_pad %in% c("progressistas") ~ "progressistas",
        partido_agredido_pad %in% c("cidadania sp", "cidadania") ~ "cidadania",
```

```

partido_agredido_pad %in% c("republicanos", "republicanos ") ~ "republicanos",
partido_agredido_pad %in% c("psol e pcdob", "psol e pcdob") ~ "psol e pcdob",
partido_agredido_pad %in% c("pt, pcdob e psol") ~ "pt e pcdob e psol",
partido_agredido_pad %in% c("psol/") ~ "psol",
TRUE ~ partido_agredido_pad
),

# Uniformizar separadores
partido_agressor_pad = str_replace_all(partido_agressor_pad, ",", " e "),
partido_agredido_pad = str_replace_all(partido_agredido_pad, ",", " e ")
)
}

dados_geral <- limpar_dados(dados_geral)

# Verificação:
# Verificar se as colunas padronizadas foram criadas corretamente
table(dados_geral$genero_agressor_pad)

```

```

##
##               Ambos               Homem
##               5               106
##               Mulher Não se aplica ou não encontrado
##               17               70

```

```
table(dados_geral$partido_agressor_pad)
```

```

##
##               cidadania               mdb
##               9               8
## não se aplica ou não encontrado               novo
##               123               2
##               patriota               pdt
##               1               1
##               pl               podemos
##               3               2
##               pp               progressistas
##               2               3
##               prtb               psb
##               1               2
##               psc               psd
##               5               6
##               psdb               psl
##               3               4
##               psol               psol e republicanos
##               6               2
##               pt               ptb
##               1               5
##               republicanos               sem partido
##               7               1
##               união
##               1

```

Agora precisamos ajustar linhas que tem mais de um partido em uma categoria, por exemplo, se houve um agressor do PSOL e um do Cidadania, e aí na cédula está escrito “Psol/Cidadania”. Temos que duplicar essas linha, mas sem correr o risco de ter combinação cruzada, caso os partidos também estejam duplicados nas duas colunas de partidos (a referente a partidos dos agressores, e a referente a partido dos agredidos). Vamos fazer uma função para isso, e vamos verificar se a expansão das linhas ocorreu corretamente.

```
#Função para expandir os partidos
expandir_partidos <- function(df, coluna_partido, prefixo) {
  df %>%
    separate_rows(!sym(coluna_partido), sep = " e |/,") %>%
    mutate(!sym(coluna_partido) := str_trim(!sym(coluna_partido))) %>%
    group_by(across(-!sym(coluna_partido))) %>%
    mutate(linha_id = paste0(prefixo, "_", row_number())) %>%
    ungroup()
}

# Criar dados separados para análise:
dados_para_agressores <- expandir_partidos(dados_geral, "partido_agressor_pad", "agr")
dados_para_agredidos <- expandir_partidos(dados_geral, "partido_agredido_pad", "agd")

# Verificação:
# Conferir se a expansão duplicou apenas a coluna correta
head(dados_para_agressores)
```

```
## # A tibble: 6 x 26
##   titulo subtitulo autor conteudo regioao data link busca partidos_mencionados
##   <chr>   <chr>   <chr> <chr>   <chr> <chr> <chr> <chr> <chr>
## 1 Políci Governad Por ~ "O gove SP - ~ 15/0~ http~ viol~ PSDB, PT
## 2 Comiti Câmara d Por ~ "A comi SP - ~ 08/0~ http~ viol~ PT
## 3 PT aci Em vídeo Por ~ "A banc SP - ~ 06/0~ http~ viol~ PP, PT
## 4 VÍDEO:~ Ainda nã Por ~ "O pref AM - ~ 02/0~ http~ viol~ UNIÃO
## 5 Corpo ~ Político Por ~ "O corp SP - ~ 29/1~ http~ viol~ PSDB, PSOL, PT
## 6 Políci Zezinho ~ Por ~ "A Polí SP - ~ 29/1~ http~ viol~ PSDB, PSOL, PT
## # i 17 more variables: Relevância <dbl>, 'Cargo Agressor(a)' <chr>,
## # 'Cargo Agredido(a)' <chr>, 'Partido Agressor' <chr>,
## # 'Partido Agredido' <chr>, 'Nome Agressor(a)' <chr>,
## # 'Nome Agredido(a)' <chr>, 'Tipo de Violência' <chr>,
## # 'Gênero Agressor' <chr>, 'Gênero Agredido' <chr>, Observação <chr>,
## # ...21 <chr>, genero_agressor_pad <chr>, genero_agredido_pad <chr>,
## # partido_agressor_pad <chr>, partido_agredido_pad <chr>, linha_id <chr>
```

```
table(dados_para_agressores$partido_agressor_pad)
```

```
##
##               cidadania                mdb
##                9                8
## não se aplica ou não encontrado          novo
##               123                2
##               patriota                pdt
##                1                1
##                pl                podemos
##                3                2
##                pp                progressistas
```

```
##          2          3
##      prt      psb
##          1          2
##      psc      psd
##          5          6
##      psdb     psl
##          3          4
##      psol      pt
##          8          1
##      ptb      republicanos
##          5          9
##      sem partido      união
##          1          1
```

```
table(dados_para_agressores$genero_agressor_pad)
```

```
##
##      Ambos      Homem
##          7      106
##      Mulher Não se aplica ou não encontrado
##          17      70
```

Vamos fazer uma função para gerar os gráficos e tabelas, criar as bases e aplicar a função a elas.

```
# Função auxiliar para gerar tabela e gráfico
gerar_tabela_grafico <- function(df, coluna_partido, cor, titulo, arquivo) {
  tabela <- as.data.frame(table(df[[coluna_partido]]))
  colnames(tabela) <- c("Partido", "Frequencia")
  tabela$Porcentagem <- prop.table(tabela$Frequencia) * 100

  grafico <- ggplot(tabela, aes(x = reorder(Partido, Frequencia), y = Frequencia)) +
    geom_bar(stat = "identity", fill = cor) +
    geom_text(aes(label = paste0(round(Porcentagem, 1), "%")), hjust = -0.1, size = 3) +
    coord_flip() +
    labs(x = "Partido", y = "Frequência", title = titulo) +
    theme_minimal() +
    ylim(0, max(tabela$Frequencia) * 1.15)

  ggsave(arquivo, plot = grafico, width = 12, height = 8, dpi = 300, bg = "white")

  return(list(tabela = tabela, grafico = grafico))
}

# Filtrar e gerar outputs para os quatro casos:

# Mulheres agressoras
dados_agressoras_mulheres <- dados_para_agressores %>% filter(genero_agressor_pad %in% c("Mulher", "Ambos"))
saida_am <- gerar_tabela_grafico(dados_agressoras_mulheres, "partido_agressor_pad", "firebrick", "Partidos de agressoras mulheres")

# Mulheres vítimas
dados_vitimas_mulheres <- dados_para_agredidos %>% filter(genero_agredido_pad %in% c("Mulher", "Ambos"))
saida_vm <- gerar_tabela_grafico(dados_vitimas_mulheres, "partido_agredido_pad", "salmon", "Partidos de vítimas mulheres")
```

```

# Homens agressores
dados_agressores_homens <- dados_para_agressores %>% filter(genero_agressor_pad %in% c("Homem", "Ambos"))
saida_ah <- gerar_tabela_grafico(dados_agressores_homens, "partido_agressor_pad", "dodgerblue", "Partidos d

# Homens vítimas
dados_vitimas_homens <- dados_para_agredidos %>% filter(genero_agredido_pad %in% c("Homem", "Ambos"))
saida_vh <- gerar_tabela_grafico(dados_vitimas_homens, "partido_agredido_pad", "steelblue", "Partidos d

# VERIFICAÇÃO DOS DADOS FINAIS:
# Ver os dados tabulados antes do gráfico
print(saida_am$tabela)

```

```

##                Partido Freqüencia Porcentagem
## 1                mdb             1    4.166667
## 2 não se aplica ou não encontrado      7    29.166667
## 3                novo             2    8.333333
## 4                pl              2    8.333333
## 5      progressistas             2    8.333333
## 6                psd             2    8.333333
## 7                psl             2    8.333333
## 8                psol            3   12.500000
## 9      republicanos             2    8.333333
## 10               união            1    4.166667

```

```
print(saida_vm$tabela)
```

```

##                Partido Freqüencia Porcentagem
## 1                cdn             1    0.6289308
## 2      cidadania             2    1.2578616
## 3                dem             3    1.8867925
## 4                mdb             2    1.2578616
## 5 não se aplica ou não encontrado     25   15.7232704
## 6                novo             2    1.2578616
## 7      partido verde             1    0.6289308
## 8      patriota             3    1.8867925
## 9      patriota (atual prd)          1    0.6289308
## 10               pcdob             7    4.4025157
## 11               pdt              4    2.5157233
## 12               pl              1    0.6289308
## 13      podemos             1    0.6289308
## 14               psd             3    1.8867925
## 15               psol            50   31.4465409
## 16               pt             38   23.8993711
## 17               ptb             3    1.8867925
## 18               pv              4    2.5157233
## 19               rede             1    0.6289308
## 20      republicanos             6    3.7735849
## 21               união            1    0.6289308

```

```
print(saida_ah$tabela)
```

```

##                Partido Freqüencia Porcentagem

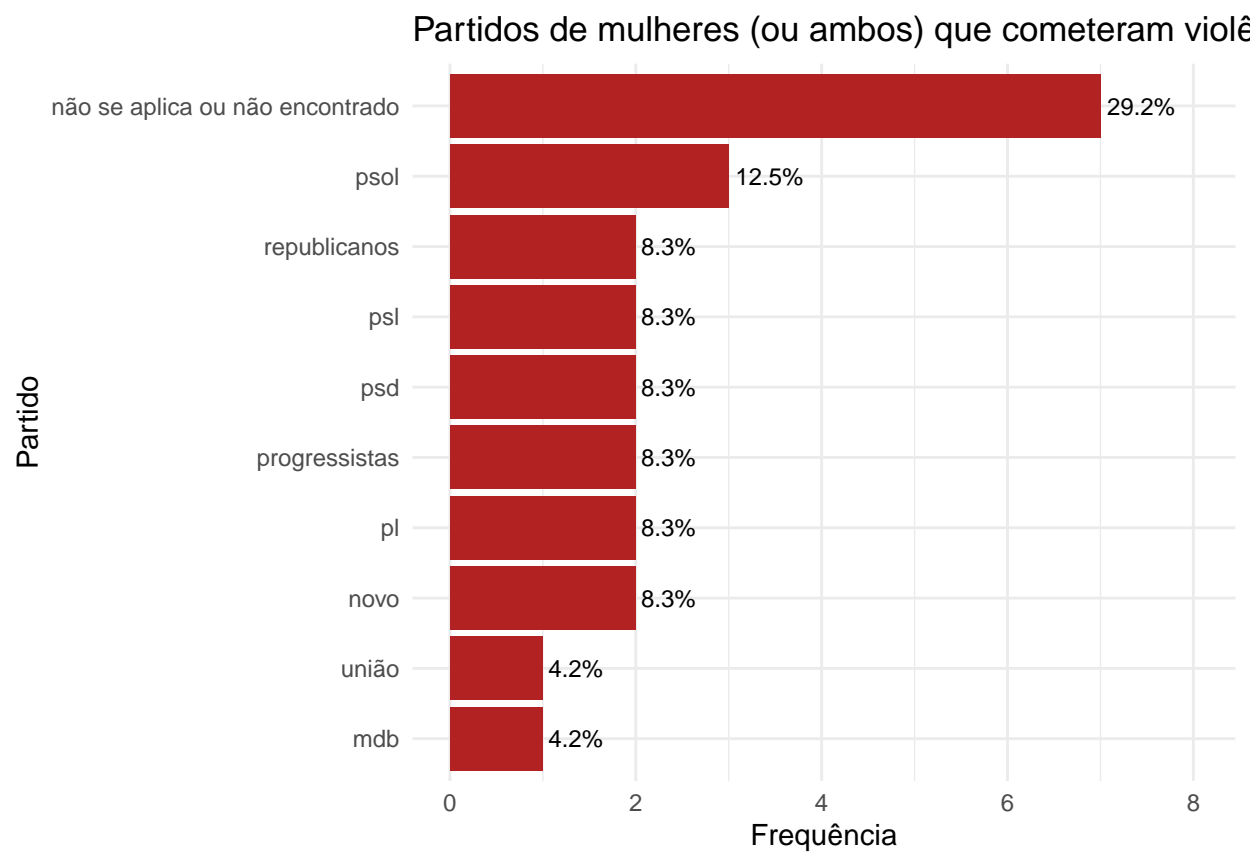
```

```
## 1      cidadania      9  7.9646018
## 2      mdb          7  6.1946903
## 3 não se aplica ou não encontrado 50 44.2477876
## 4      patriota      1  0.8849558
## 5      pdt          1  0.8849558
## 6      pl           1  0.8849558
## 7      podemos      2  1.7699115
## 8      pp           2  1.7699115
## 9      progressistas 1  0.8849558
## 10     prt          1  0.8849558
## 11     psb          2  1.7699115
## 12     psc          5  4.4247788
## 13     psd          4  3.5398230
## 14     psdb         3  2.6548673
## 15     psl          2  1.7699115
## 16     psol         7  6.1946903
## 17     ptb          5  4.4247788
## 18     republicanos  9  7.9646018
## 19     sem partido   1  0.8849558
```

```
print(saida_vh$tabela)
```

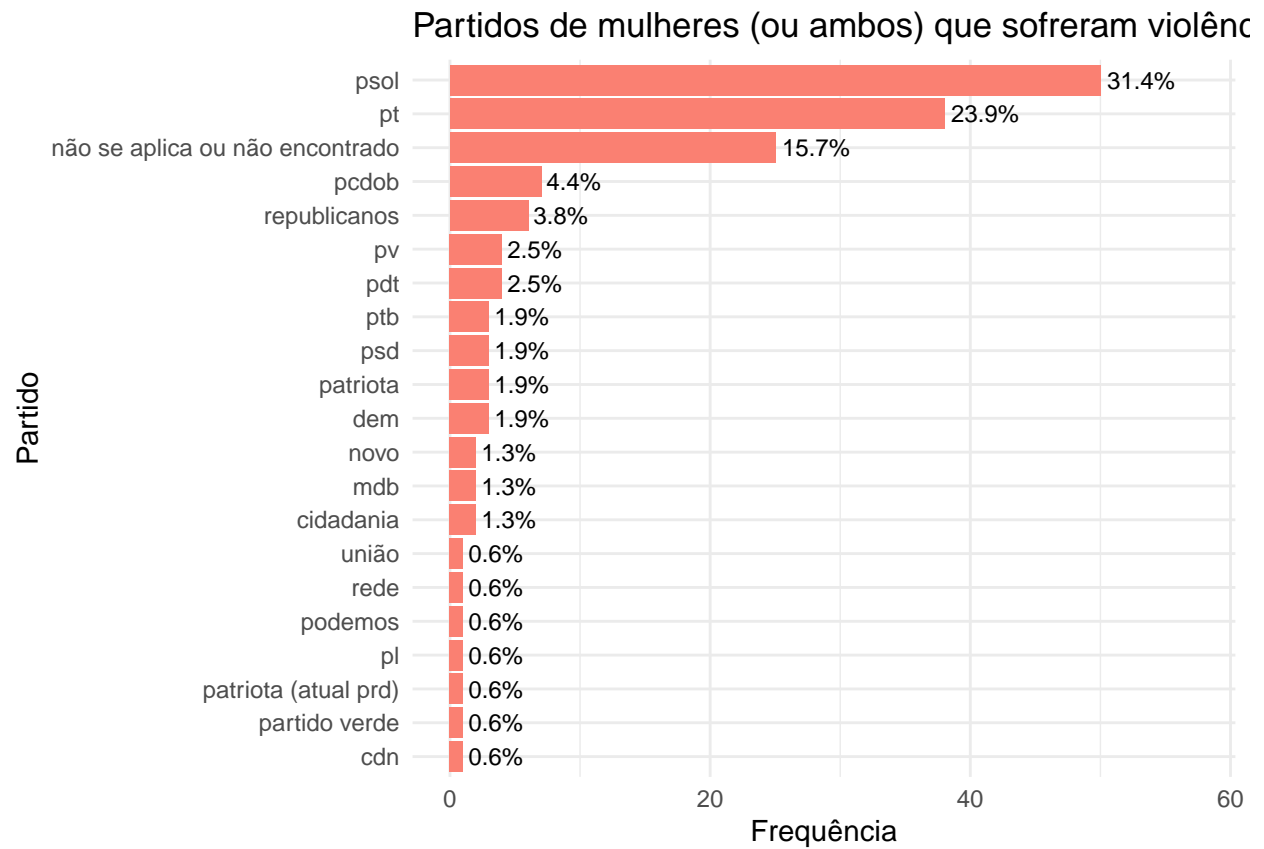
```
##      Partido Frequencia Porcentagem
## 1 não se aplica ou não encontrado 12      24
## 2      pcdob          1         2
## 3      pl            1         2
## 4      podemos       1         2
## 5      pp            7        14
## 6      psd           1         2
## 7      psdb          3         6
## 8      psl           1         2
## 9      psol          2         4
## 10     pt           11        22
## 11     ptb           1         2
## 12     republicanos  4         8
## 13     solidariedade 3         6
## 14     união         2         4
```

```
# Visualizar gráficos manualmente (se quiser no console interativo)
print(saida_am$grafico)
```

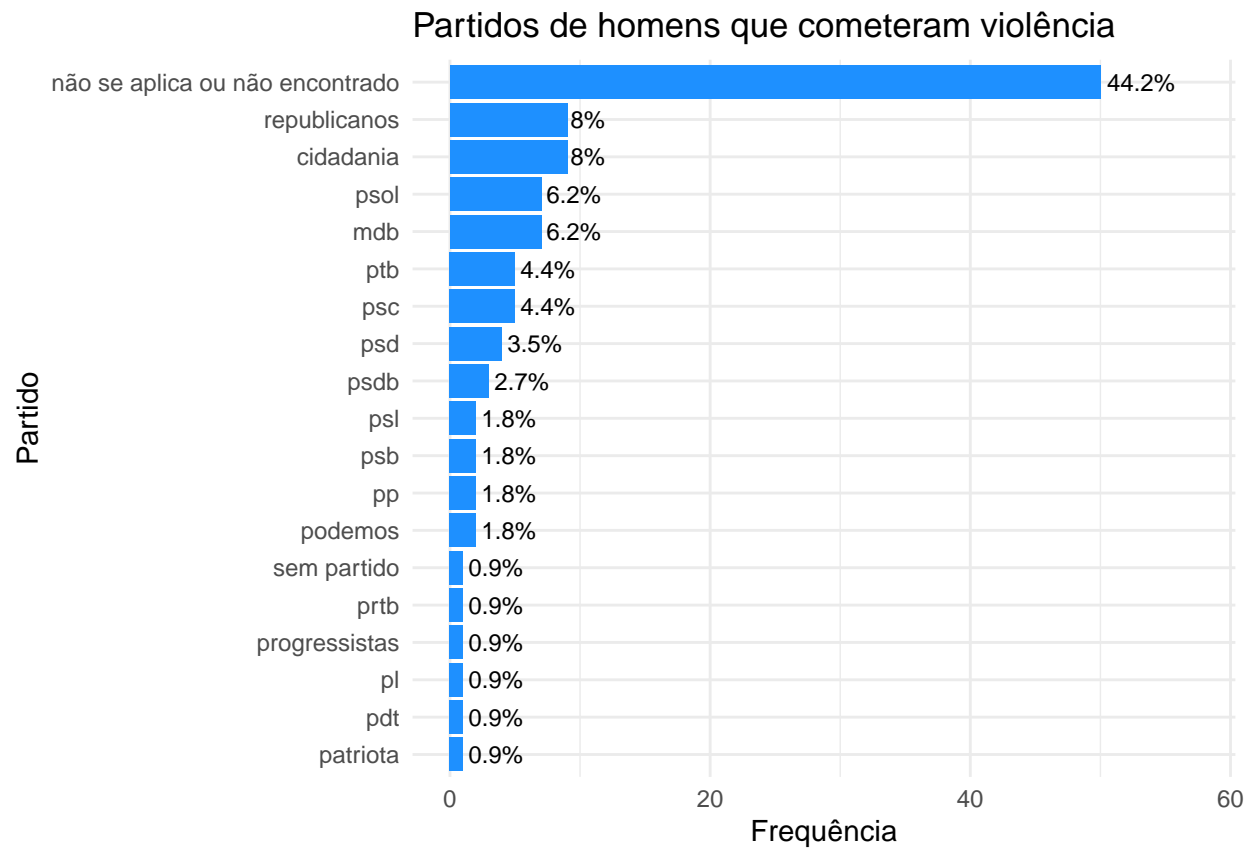


```
print(saida_vm$grafico)
```

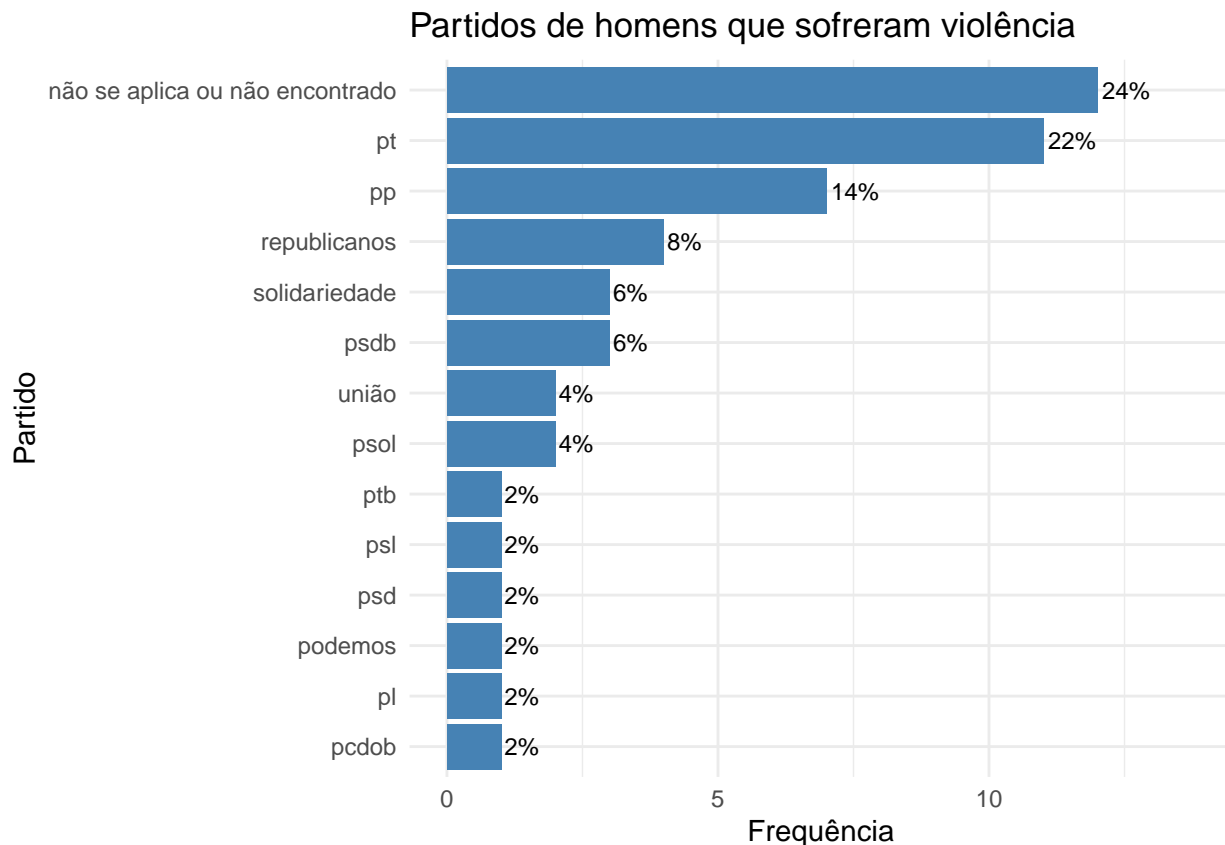




```
print(saida_ah$grafico)
```



```
print(saida_vh$grafico)
```



Por último, vamos verificar o percentual da categoria “Ambos” em cada base de dados. Esse foi um pedido dos outros integrantes do grupo. Alguns casos de violência possuem duas pessoas, uma de cada gênero, na categoria de agressores. Como dividimos a análise por gênero e posição da pessoa na violência, se tornou relevante entender o quão mesclados os gêneros estão em algumas dessas categorias.

*#Verificação da categoria "Ambos" em cada uma das tabelas*

*#Função para verificar*

```
calcular_freq_e_pct_ambos <- function(df, coluna_genero) {
  total <- nrow(df)
  freq_ambos <- df %>% filter(!sym(coluna_genero) == "Ambos") %>% nrow()
  pct_ambos <- (freq_ambos / total) * 100
  tibble(
    Frequencia = freq_ambos,
    Total = total,
    Percentual = pct_ambos
  )
}
```

*# Aplicando para as 4 bases:*

```
freq_pct_agressoras_mulheres <- calcular_freq_e_pct_ambos(dados_agressoras_mulheres, "genero_agressor_pad")
freq_pct_vitimas_mulheres <- calcular_freq_e_pct_ambos(dados_vitimas_mulheres, "genero_agredido_pad")
freq_pct_agressores_homens <- calcular_freq_e_pct_ambos(dados_agressores_homens, "genero_agressor_pad")
freq_pct_vitimas_homens <- calcular_freq_e_pct_ambos(dados_vitimas_homens, "genero_agredido_pad")
```

*# Mostrar resultados*

```
freq_pct_agressoras_mulheres
```

```
## # A tibble: 1 x 3
##   Frequencia Total Percentual
##       <int> <int>       <dbl>
## 1         7    24       29.2
```

```
freq_pct_vitimas_mulheres
```

```
## # A tibble: 1 x 3
##   Frequencia Total Percentual
##       <int> <int>       <dbl>
## 1         8   159       5.03
```

```
freq_pct_agressores_homens
```

```
## # A tibble: 1 x 3
##   Frequencia Total Percentual
##       <int> <int>       <dbl>
## 1         7   113       6.19
```

```
freq_pct_vitimas_homens
```

```
## # A tibble: 1 x 3
##   Frequencia Total Percentual
##       <int> <int>       <dbl>
## 1         8    50       16
```