Passo-a-passo

Etapa 10

Prof. Pedro Feliú

Instituto de Relações Internacionais
Universidade de São Paulo

INTRODUÇÃO

Nesta aula vamos aprender a produzir e analisar estatísticas descritivas a partir de texto, uma variável qualitativa nominal ou categórica. No R, ela é representada como caractere ("chr"). Nas aulas anteriores utilizamos muitas variáveis quantitativas contínuas, realizando gráficos e análises descritivas com variáveis numéricas. Nessa aula, vamos utilizar uma base de dados apenas com variáveis qualitativas, com especial atenção para a análise quantitativa de texto. O uso de texto nas pesquisas de relações internacionais é muito comum, especialmente os discursos dos tomadores de decisão em política externa. O discurso na diplomacia é uma ação política e a sua análise permite averiguar a influência de variáveis subjetivas, como as ideias, ideologia e identidade, no comportamento dos países. Uma forma bastante útil de fazer análise de texto é por meio do uso de estatística descritiva e análise de sentimento das palavras.

Na presente aula, utilizaremos como banco de dados os discursos das autoridades de Estado de China e EUA no Conselho de Segurança da ONU (CSONU) entre 1995 e 2020. Um aspecto de grande interesse nas relações internacionais é a projeção do poder chinês no mundo e a competição com a grande potência norte-americana. Como vimos na aula anterior, o CSONU é um órgão central na governança global da segurança e ambos os países possuem assento permanente e direito ao veto. Assim, analisar o posicionamento de ambos os países por meio de discursos permite analisar a evolução da agenda de ambos os países na arena global.

Percebam que vamos nesta aula quantificar os discursos dos países onde são mobilizados valores e ideias, elementos subjetivos das relações exteriores do país. O banco de dados é denominado "discursos_CSONU_China e EUA.xls" e está disponível na pasta de base de dados do moodle. Vamos iniciar os comandos para instalar os pacotes necessários, importar os dados em Excel para o Rstudio e preparar os dados para execução dos gráficos.

PASSO 1: Importar, preparar e inspecionar os dados
Iniciamos com os pacotes requeridos[1]:

```
install.packages("knitr")
install.packages("kableExtra")
install.packages("gridExtra")
install.packages("tidytext")
install.packages("stringr")
install.packages("tidyr")
install.packages("ggplot2")
install.packages("wordcloud2")
install.packages("readxl")
install.packages("openxlsx")
install.packages("textdata")
install.packages("igraph")
```

[1] Nem todos os pacotes serão efetivamente utilizados, mas queria já indicar a vocês os principais que pelo nome depois vocês podem pesquisar e descobrir mais funções para analisar e produzir outras análises dos textos. Vai demorar um pouquinho instalar tudo.

```
install.packages("ggraph")
install.packages("ggrepel")
install.packages("tm")
install.packages("foreign")
install.packages("dplyr")
install.packages("writexl")
install.packages ("memery")
install.packages ("magick")
install.packages("circlize")
install.packages("SentimentAnalysis")
install.packages("tidyverse")
install.packages("devtools")
install.packages("widyr")
install.packages("wordcloud")

library(widyr)
library(devtools)
library(tidyverse)
library(SentimentAnalysis)
library(circlize)
library(memery)
library(magick)
library(writexl)
library(dplyr)
library(tidytext)
library(stringr)
library(tidyr)
library(foreign)
library(igraph)
library(ggraph)
library(ggrepel)
library(tm)
library(ggplot2)
library(wordcloud2)
library(readxl)
library(openxlsx)
library(textdata)
library(knitr)
library(kableExtra)
library(gridExtra)
library(wordcloud)
```

Feita a instalação e convocação dos muitos pacotes necessários, vamos importar a base de dados “discursos_CSONU_China e EUA.xls” pra o Rstudio, denominando o objeto de “discurso”:

```
discurso <-
read_excel("C:/Users/Paulo/Documents/Documents/CursoR_Apolo/Bases de dados/discursos_CSONU_China e EUA.xls")
```

Vocês podem importar os dados da outra forma que aprendemos também, com cliques no Rstudio. O comando que eu utilizei é preciso indicar o caminho das pastas do arquivo que queremos abrir. Esse é o comando para caso não estejam usando o Rstudio e apenas o R, por exemplo.

PASSO 2: Gráfico Descritivo

Para personalizar os gráficos, criaremos uma lista exclusiva de cores por meio dos códigos das cores, conformando o objeto my_color que será utilizado adiante.

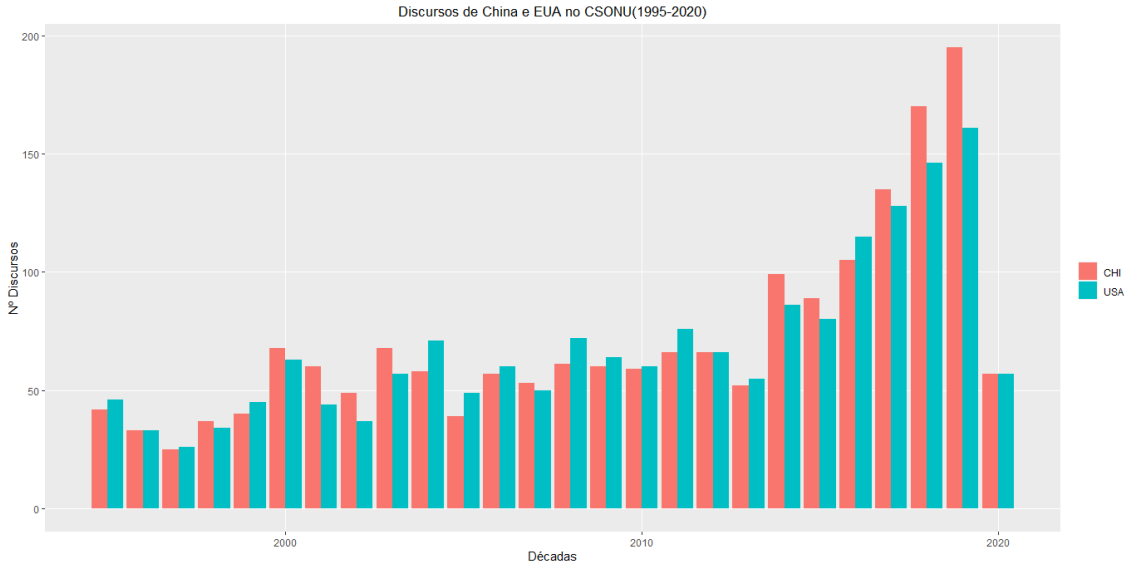
```
my_colors <- c("#E69F00", "#56B4E9", "#009E73", "#CC79A7", "#D55E00")
```

Eu quero saber quantos são os discursos que a China e os EUA proferiram por ano no CSONU. Para fazer esse gráfico descritivo, vamos criar o objeto "speeches_potencia", e desagregar por ano e potência usando função "group_by", contabilizando a frequência absoluta de discursos por ano ("number_of_texto = n()"). Segue o comando, o execute todo de uma vez no Rstudio.

```
speeches_potencia <- discurso %>%
  group_by(sigla, ano) %>%
  summarise(number_of_texto = n())
```

Agora usamos o ggplot já conhecido, mas criando para cada etapa do código, o objeto "plot". Começamos indicando o objeto, depois as variáveis que queremos (ano, number_of_texto) e incluímos com a função "fill" a sigla (CHI e USA). Depois escolhemos o estilo somando o objeto plot ao geom_bar (gráfico de barras), com a função "position" dodge, que indica as barras seriadas. Depois são comandos de tamanho do texto do título e estilo do pano de fundo do gráfico, assim como os títulos legendas. Por fim execute o comando plot:

```
plot <- ggplot(speeches_potencia, aes(ano, number_of_texto, fill=sigla))
plot <- plot + geom_bar(stat = "identity", position = 'dodge')+
  theme(plot.title = element_text(hjust = 0.5),
        legend.title = element_blank(),
        panel.grid.minor = element_blank()) +
  labs(x = "Décadas", y = "Nº Discursos") +
  ggtitle("Discursos de China e EUA no CSONU(1995-2020)")
plot
```



Podemos ver com clareza o aumento de discursos da China e dos EUA ao longo do tempo. Mais recentemente, a China consistentemente se pronuncia mais no CSONU do que os EUA, mas sempre muito equilibrado. Notem que como estamos usando pacotes específicos, como o ggplot2 que foi abordado na aula anterior. Passamos agora para a análise do texto dos discursos proferidos por autoridades chinesas e norte-americanas no órgão de segurança internacional da ONU.

PASSO 3: Text Mining (mineração de dados de texto)

Vamos começar a análise do texto pelo primeiro ponto fundamental: a limpeza do texto. Com os comandos abaixo vamos retirar caracteres indesejados que não possuem significado semântico e podem atrapalhar a análise do texto. Usamos a coluna texto do objeto discurso para transformar tudo em minúsculo (str_to_lower()), retirar pontuações ((" *-+ *", "")) e números (removeNumbers()). Segue o comando para rodar de uma vez:

```
discurso$texto <- discurso$texto %>%
  str_to_lower() %>%
  str_replace_all(" *-+ *", "") %>%
  str_replace_all("[[:punct:]]", " ") %>%
  removeNumbers() %>%
  trimws()
```

Para continuar a limpeza, vamos utilizar um dicionário em inglês de palavras prontas chamadas stopwords. São palavras utilizadas como conector ou não possuem valor semântico para a análise de sentimento que iremos realizar a seguir. Em português, alguns exemplos seriam: onde, assim, ainda, seguido, entre outras. Segue o comando abaixo para retirar essas stopwords. O primeiro comando exibe as palavras que serão retiradas. Em português basta substituir o "en" do comando abaixo por "pt", caso seu texto esteja em português (há um pacote para análise de sentimento em português: lexiconPT).

```
stopwords(kind = "en")
discurso$texto <- discurso$texto %>%
removeWords(words = stopwords(kind = "en"))
```

Removidas as stopwords, vamos criar o objeto "texto_un" que realiza a Tokenização do nosso banco de dados. Essa tokenização pega do texto de cada fala das autoridades no objeto "discursos" e transforma em uma nova matriz de dados em que cada palavra dos textos vira uma linha no novo objeto "texto_un". Isso é necessário para realizar a análise de sentimento. Notem que o novo objeto "texto_un" possui 1150575 de linhas ou palavras tokenizadas. Segue o comando:

```
texto_un <- discurso %>%
  unnest_tokens(output = "words", input = texto)
texto_un
```

PASSO 4: Análise de Sentimento

Iniciamos agora a análise de sentimento. Temos o banco de dados no formato adequado já, com cada palavra na linha da matriz de dados como vocês puderam notar anteriormente. Precisaremos de mais pacotes no R, aqueles específicos sobre sentimentos das palavras. Basicamente, esses pacotes e comandos de sentimento importam para o R dicionários de palavras da língua inglesa previamente classificadas como negativas, neutras ou positivas. Um trabalho prévio realizado por linguistas, compreendendo enormes listas de palavras que buscam cobrir o máximo de palavras de determinada língua. Para o português, como citado anteriormente, há pacotes, mas em menor abundância e qualidade quando comparado ao inglês. Utilize o comando para obter o dicionário AFINN, criando objeto de mesmo nome:

```
AFINN <- get_sentiments("afinn")
```

O pacote tidytext inclui um conjunto de dados chamado sentimentos que fornece vários léxicos distintos. Esses léxicos são dicionários de palavras com uma categoria ou valor de sentimento atribuído. O tidytext fornece três léxicos de uso geral:

affin: atribui palavras com uma pontuação que varia entre -5 e 5, com pontuações negativas indicando sentimentos negativos e pontuações positivas indicando sentimentos positivos

Bing: atribui palavras a categorias positivas e negativas

nrc: atribui palavras a uma ou mais das dez categorias a seguir: positivo, negativo, raiva, antecipação, nojo, medo, alegria, tristeza, surpresa e confiança

Vamos usar nessa aula "afinn" agora e depois o "nrc". Começamos com o AFINN criado anteriormente. Vamos utilizar a função inner_join para juntar os sentimentos do dicionário AFINN com as palavras do nosso objeto "texto_un". Vamos usar as colunas de palavras do AFINN (word) e a coluna com os sentimentos (value). A junção é feita

no words (a coluna palavras do objeto texto_un) e word (a coluna de palavras do AFINN):

```
tb_sen <- inner_join(texto_un,
                     AFINN[, c("word", "value")],
                     by = c("words" = "word"))
```

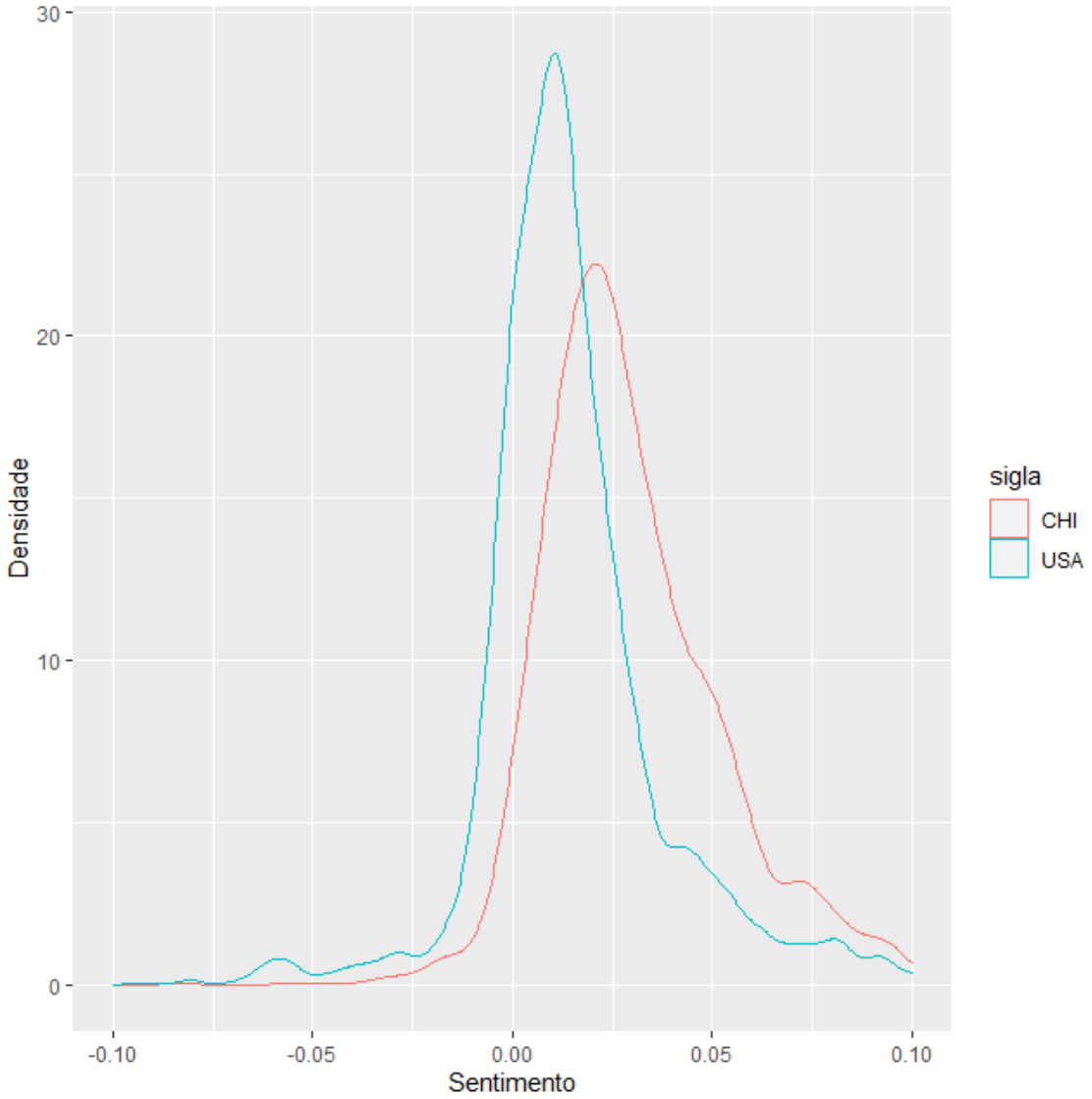
Criamos o objeto "tb_sen" com os sentimentos de cada palavra, na nova coluna "value". Agora vamos fazer uma operação para agregar essas medidas de sentimento de cada palavra:

```
tb <- tb_sen %>%
  group_by(id, sigla, ano) %>%
  summarise(media = mean(value),
            n = n(),
            sentiment = media/n)
```

O novo objeto "tb" agrupa as palavras por id (que é o número de identificação de cada discurso proferido no CSONU, juntando as palavras usadas no mesmo discurso e o sentimento delas. Também mantemos a sigla, USA ou CHI, e o ano. Usamos a função "summarise" para obter as médias dos sentimentos das palavras agregadas por discurso. Basicamente desagregamos no passo anterior as palavras para obter o valor de cada uma segundo o dicionário AFINN e agora, com esses valores, vamos agregar novamente pela média. Poderíamos usar outro operador também, como soma ou mediana, por exemplo.

Agora já podemos fazer o primeiro gráfico com os valores dos sentimentos médios por discurso proferido pelas autoridades chinesas e norte-americanas no CSONU. Iniciamos com um gráfico chamado densidade de Kernell:

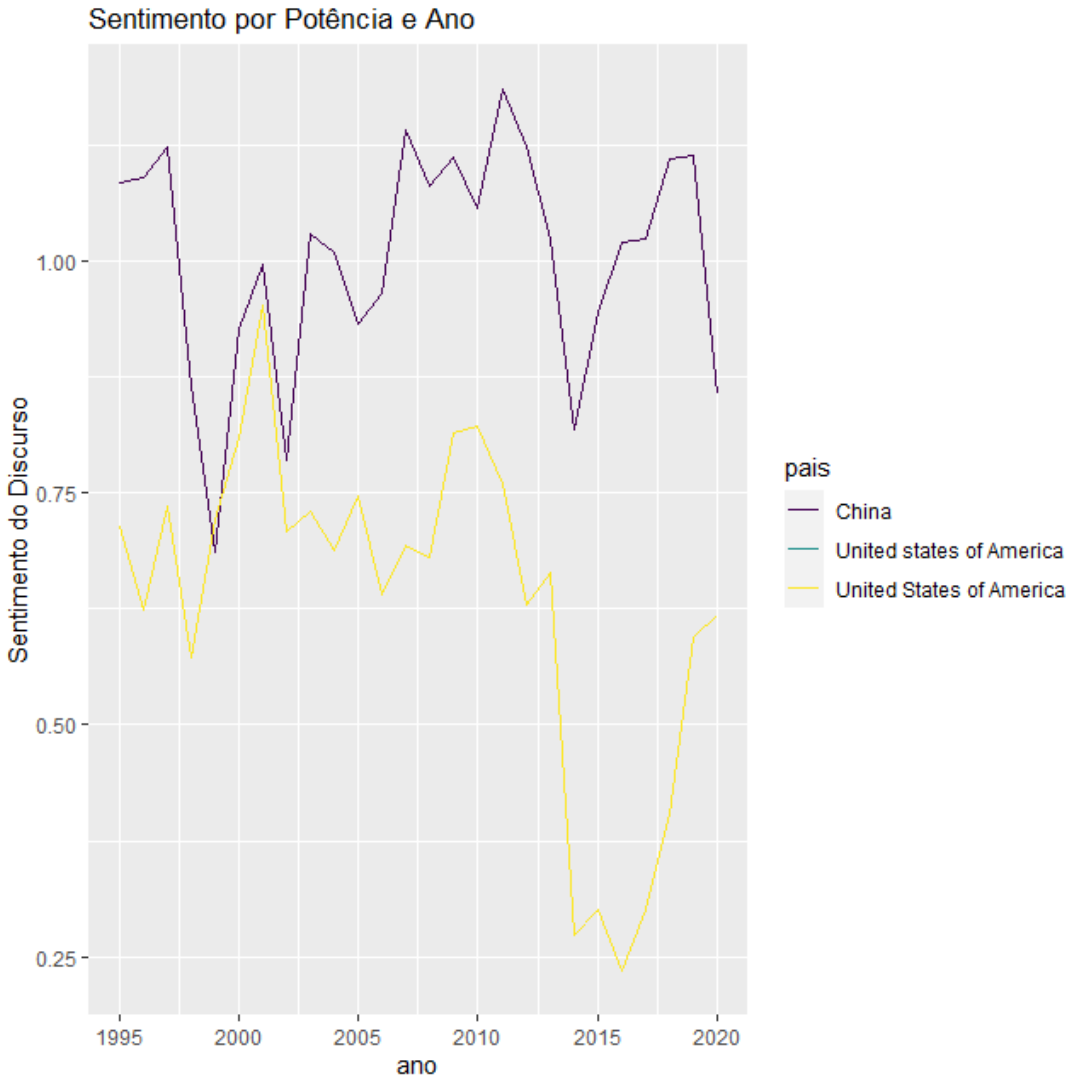
```
ggplot(tb, aes(sentiment, colour = sigla)) +
  geom_density() +
  xlim(-0.1, 0.1)+
  labs(x = "Sentimento", y = "Densidade")
```



Esse tipo de gráfico é muito útil para saber como se comporta a curva dos dados. No nosso caso, os sentimentos tanto de China quanto EUA tendem a ser medianos e neutros. Muito esperado em um ambiente diplomático. Contudo, percebemos que a maior densidade da curva chinesa em vermelho é um pouco mais positiva do que a dos EUA, mas muito parecidas e próximas. Vamos agora avaliar a evolução no tempo dos sentimentos dos discursos das autoridades no CSONU:

```
tb2 <- tb_sen %>%
  group_by(pais, ano) %>%
  summarise(media = mean(value),
            n = n(),
            sentiment = media)
```

```
tb2 %>%
  ggplot( aes(x=ano, y=sentiment, group=pais, color=pais)) +
  geom_line() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle("Sentimento por Potência e Ano") +
  ylab("Sentimento do Discurso")
```



Para produzir o gráfico acima criamos um novo objeto “tb2”, desta vez utilizamos no agrupamento (group_by) apenas pais e ano. Em seguida, a mesma função para extrair a média do sentimento das palavras daquele discurso. Notem que há um erro no nome dos Estados Unidos e aparece um caso em que falta um espaço. Eu deixei para mostrar como qualquer caractere diferente é lido como outra categoria. Criado o “tb2”, usamos os comandos conhecidos do ggplot para gerar o gráfico de linhas agrupado por país, usando a variável “sentiment” no eixo y e “ano” no eixo x.

PASSO 5: Nuvem de Palavras

Vamos agora utilizar a análise de sentimento para criar uma nuvem de palavras de China e EUA em um mesmo gráfico. Vamos começar criando o objeto "tb_words" filtrando e retirando as palavras com sentimento zero. Assim, podemos limpar a nuvem de palavras por sentimento, apenas reunindo os positivos e negativos (aqui tiramos um pouco do viés neutro diplomático). Também realizamos a contagem das palavras coma função count, que conta o número de palavras repetidas e seu respectivo valor de sentimento, nas variáveis "words" e "value". Seguem os comandos:

```
tb_words4 <- tb_sen %>%
  count(words, value, sigla, sort = TRUE) %>%
  filter(value != 0)
```

Criado o "tb_words", vamos criar o objeto "tb_cloud" separando nas colunas desse novo dataframe os valores do sentimento e a quantidade que a palavra foi utilizada nos discursos de China e EUA. Para isso, na função ""key", inserimos variável "sigla" que identifica China (CHI) e Estados Unidos (USA). Na função "value" usamos a variável "n", que basicamente conta o número de palvras criado anteriormente. Segue o comando:

```
tb_cloud <- tb_words4 %>%
  spread(key = "sigla", value = "n", fill = 0) %>%
  rename("China" = "CHI", "Estados Unidos" = "USA")
tb_cloud
```

Feito isso, precisamos ainda mais uma transformação, criando o objeto "tb3", uma matriz de dados apenas com as colunas EUA e China e a quantidade de cada palavra:

```
tb3 <- as.data.frame(tb_cloud[, c("China", "Estados Unidos")])
rownames(tb3) <- tb_cloud$words
head(tb3)
```

Por fim, rodamos o comando "comparison.cloud" para fazer uma nuvem comparativa do objeto tb3 criado indicando as cores vermelho e azul e um máximo de 500 palavras:

```
comparison.cloud(tb3,
                 colors = c("red", "blue"),
                 max.words = min(nrow(tb), 500))
```

PASSO 6: Sentimento com o NRC

Vamos agora utilizar o léxico nrc, que categoriza as palavras em positivo, negativo, raiva, antecipação, nojo, medo, alegria, tristeza, surpresa e confiança para gerar um gráfico indicando os sentimentos predominantes nos discursos:

NRC <- get_sentiments("nrc")

Criamos o objeto NRC com o dicionário de sentimento. Em seguida vamos juntar ele como objeto criado anteriormente "texto_un", da mesma forma como fizemos com o AFINN, criando outro objeto "discurso_nrc":

```
discurso_nrc <- inner_join(texto_un,
                           NRC[, c("word", "sentiment")],
                           by = c("words" = "word"))
```

Criado o objeto "discurso_nrc", vamos agora desenvolver um novo objeto, chamado "decade_mood", que basicamente filtra pela sigla (CHI ou USA) todos os sentimentos do pacote e a soma desses sentimentos para as potências. Assim, usamos as variáveis "sigla" e "sentiment" para gerar um objeto que nos dê a quantidade de repetição do sentimento nas palavras usadas nos discursos.

```
decade_mood <-  discurso_nrc %>%
  filter(sigla != "NA" & !sentiment %in% c("positive", "negative")) %>%
  count(sentiment, sigla) %>%
  group_by(sigla, sentiment) %>%
  summarise(sentiment_sum = sum(n)) %>%
  ungroup()
```

Percebam que utilizamos vários comandos já conhecidos dos passos anteriores, como o filter, count, group_by e summarise. Podemos agora definir os parâmetros para realizar um gráfico circular com esses sentimentos todos e suas frequências por potência.
Iniciamos definindo o parâmetro das cores. Lembrem que apenas afora vamos usar o objeto "my_colors" criado anteriormente. Aqui defininos as cores 1 para China e 2 para USA, depois vocês podem mudar. Também definimos cada sentimento do pacote como cinza ("grey").

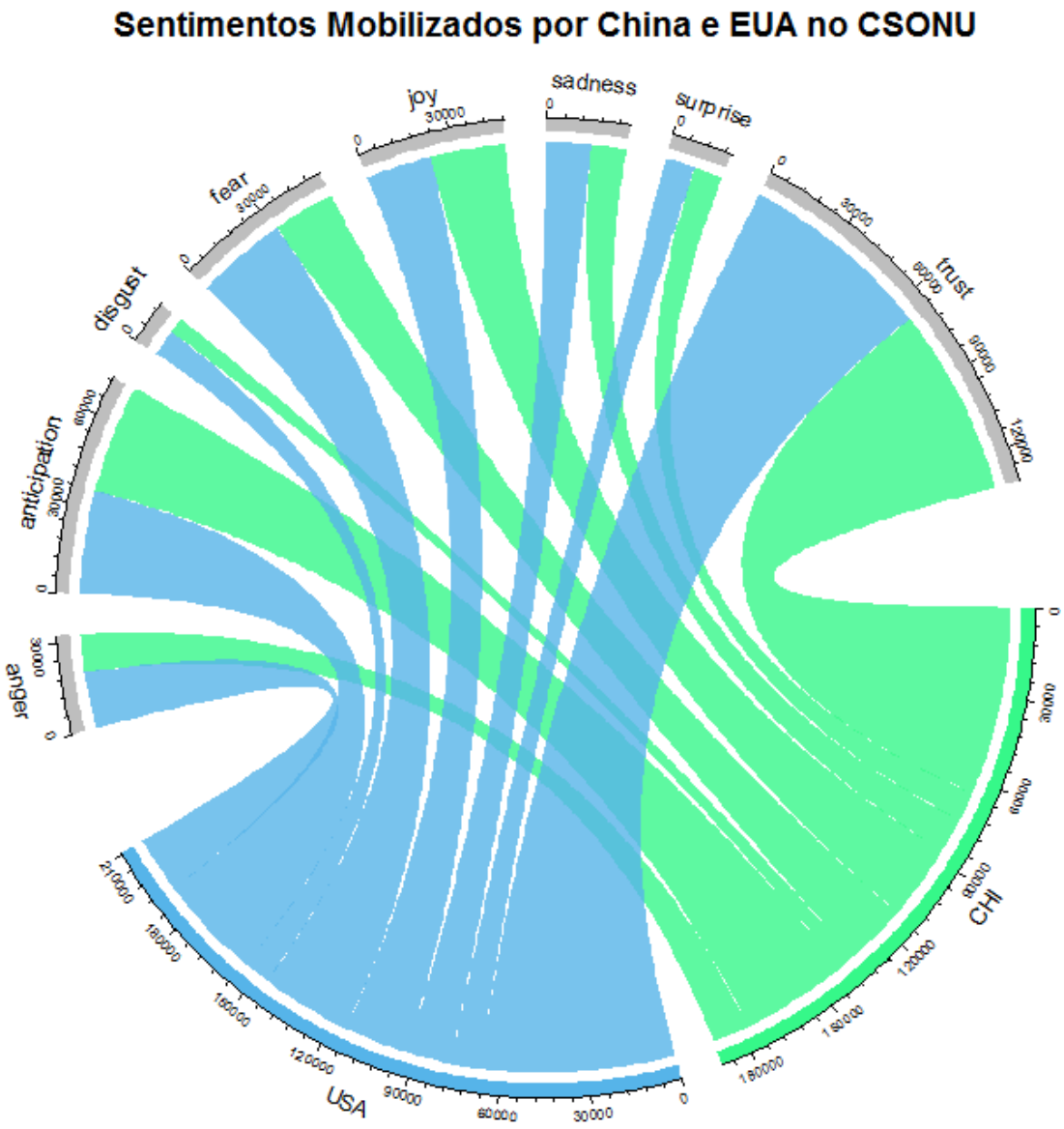
```
grid.col = c("China" = my_colors[1], "USA" = my_colors[2], "anger" = "grey", "anticipation" = "grey", "disgust" = "grey", "fear" = "grey", "joy" = "grey", "sadness" = "grey", "surprise" = "grey", "trust" = "grey")
```

Agora criamos a interface para plotar o gráfico:

```
circos.clear()
```

Estamos prontos para convocar o gráfico. Usamos a função circos.par e o objeto decade_mood. Definimos os tamanhos das partes do círculo para China (1) e Estados Unidos (2) e inserimos na função chordDiagram o objeto, as cores criadas e o nível transparência que queremos. Por fim, adicionamos um título para o gráfico:

```
circos.par(gap.after = c(rep(5, length(unique(decade_mood[[1]])) - 1), 15,
                         rep(5, length(unique(decade_mood[[2]])) - 1), 15))
chordDiagram(decade_mood,grid.col = grid.col, transparency = .2)
title("Sentimentos Mobilizados por China e EUA no CSONU")
```



Finalizamos essa aula que introduziu vocês à utilização de estatística descritiva a partir de discursos políticos e análise de sentimento utilizando léxicos de dicionários disponíveis no R. Outra opção mais complexa é realizar análise de sentimento por meio de inteligência artificial, chegamos lá um dia...