Acessando Dados da Web em R

Introdução ao R

Tiago Ventura | venturat@umd.edu

University of Maryland, College Park

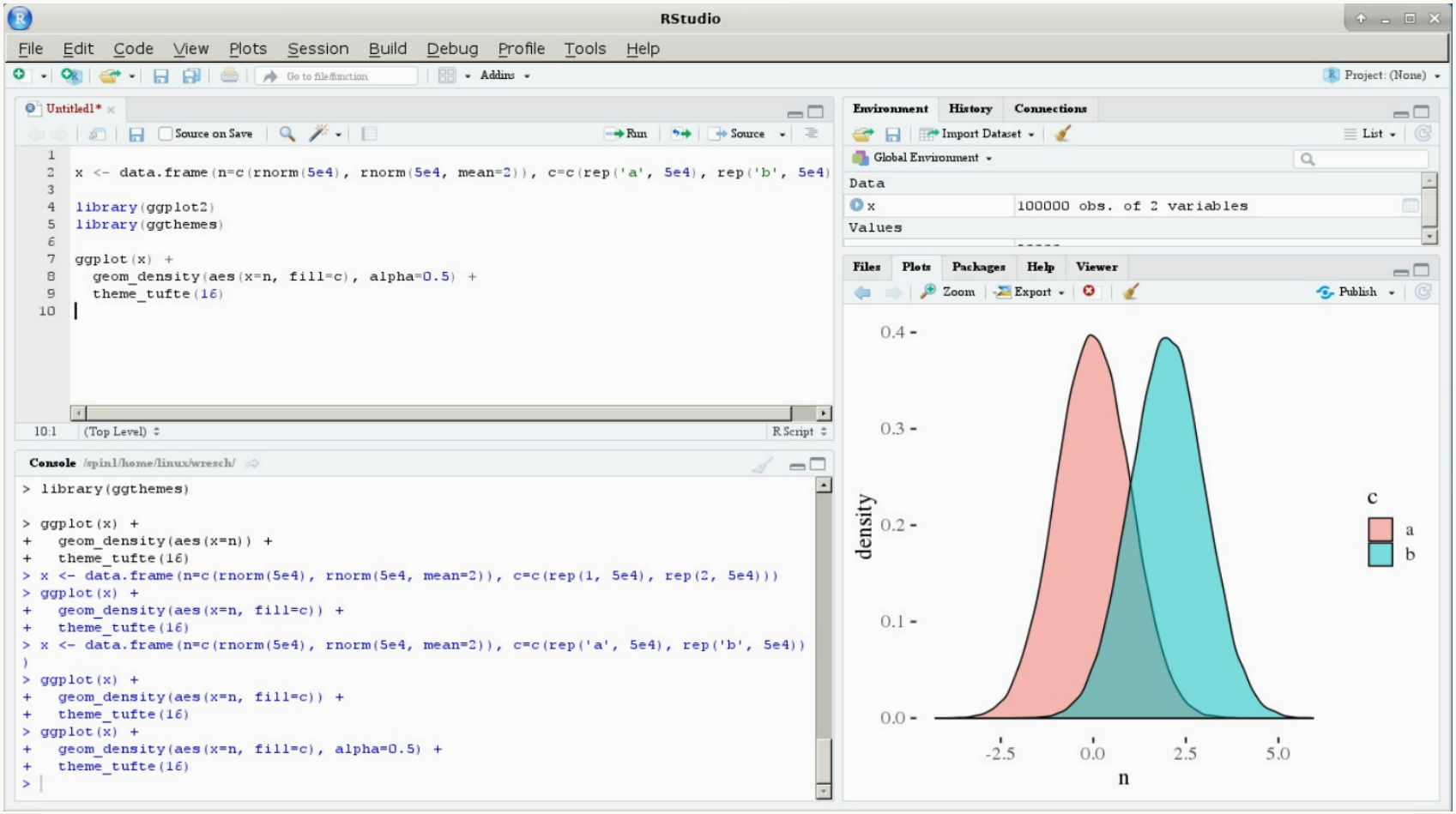
Introdução ao R

R é uma linguagem de programação de código aberto versátil, útil tanto para estatística quanto para ciência de dados.

- Gratuito
- Aberto
- Mais usado em Ciência Política, atualmente.
- Excelente comunidade de usuários.

Rstudio: O RStudio é a principal interface gráfica do usuário (GUI) e o ambiente de desenvolvimento integrado (IDE) que facilita o uso do R.

Navegando no R Studio.



Instalando um pacote no R

O R é uma linguagem funcional. Pacotes apenas agregam diversas funções em um único tema.

Há pacotes básicos no R. E há pacotes criados por desenvolvedores. Tudo gratuito.

Os passos para instalar e ativar um pacote são os seguintes.

1. Instalar o pacote – somente uma vez.
2. Ativar o pacote
3. Divertir-se com os pacotes

```r
# Instalando um pacote.
install.packages("ggplot2")
install.packages("tidyverse")
```

```r
# Activando the package
library("ggplot2")
library("tidyverse")
```

```
## -- Attaching packages ----------------------- tidyverse 1.2

## v tibble  2.1.3     v purrr   0.3.2
## v tidyr   0.8.3     v dplyr   0.8.1
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0

## -- Conflicts -------------------------- tidyverse_conflicts
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Comandos Básicos em R

Atribuindo um objeto

```
# Objeto numérico
x <- 3

# objeto de texto

my_name <- "Tiago"

# Onde estão?
ls()
```

```
## [1] "my_name" "x"
```

```r
# Remova um objeto
rm(x)

# Checando de novo
ls()
```

```
## [1] "my_name"
```

```r
# Você pode reescrevê-los.

my_name <- "Tiago Da Silva Ventura"

my_name
```

```
## [1] "Tiago Da Silva Ventura"
```

Estrutura dos Dados.

Um objeto pode ter diferente estruturas. Vamos vê-las rapidamente.

- `vector`
- `matrix`
- `data.frame`
- `list`
- `array`

Vector

```
# vetor de números
X <- c(1, 2.3, 4, 5, 6.78, 6:10)
X
```

```
##  [1]  1.00  2.30  4.00  5.00  6.78  6.00  7.00  8.00  9.00 10
```

```
# Class
class(X)
```

```
## [1] "numeric"
```

Data Frame

O tipo de dados mais útil para análise de dados. É como uma planilha de excel no seu ambiente R.

```
# Coercing
d <- as.data.frame(X)

# Create a data frame

data <- data.frame(name="Tiago", last_name="ventura", school="UM

data
```

```
##    name last_name school age
## 1 Tiago   ventura    UMD  30
```

Matrix

```
# Coerce to a matrix
as.matrix(X)
```

```
##         [,1]
##  [1,]  1.00
##  [2,]  2.30
##  [3,]  4.00
##  [4,]  5.00
##  [5,]  6.78
##  [6,]  6.00
##  [7,]  7.00
##  [8,]  8.00
##  [9,]  9.00
## [10,] 10.00
```

List

```
# coerce to a list

as.list(X)
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 2.3
##
## [[3]]
## [1] 4
##
## [[4]]
## [1] 5
##
## [[5]]
## [1] 6.78
```

Working Directory

Definir sua estação de trabalho é um passo que sempre causa muita dor de cabeça à iniciantes em R.

O `R` não sabe intuitivamente onde estão seus dados. Se os dados estiverem em uma pasta especial chamada "pesquisa super secreta", temos que dizer ao `R` como chegar lá.

Toda vez que o `R` é inicializado, ele olha para o mesmo lugar, a menos que seja solicitado a ir para outro lugar.

Qual meu diretório atual?

```
getwd () # Obter o diretório de trabalho atual
```

```
## [1] "C:/Users/Tiago Ventura/Dropbox/webscraping_workshop_ufpa
```

Definindo um novo diretório.

```
## [1] "C:/Users/Tiago Ventura/Dropbox/webscraping_workshop_ufpa
## [1] "d"       "data"    "list"    "my_name" "X"
```

```
# adicione onde você quer o R olhando.

setwd("C:/Users/Tiago Ventura/Dropbox/
      webscraping_workshop_ufpa/html/dia_00")

getwd()

# Veja o que existe no seu diretorio.

ls()
```

Manipulação de Dados com o Tidyverse

Tidyverse

O tidyverse nada mais é do que um conjunto de pacotes R construídos para nos ajudar a fazer ciência de dados.

- **dplyr**: para manipulação de dados
- **ggplot2**: para visualização de dados
- **tidyr**: para modelagem e gerenciamento de dados
- **purrr**: para otimizar seu código e para programação funcional
- **readr**: para abrir e organizar os dados

Por que devo usar os pacotes tidyverse?

As principais vantagens:

- Mais fácil de executar análise de dados em comparação com a base R
- Aumente legíbilidade do código.
- Integra bem com uma série de pacotes úteis

Instalação

```
# Install from CRAN
install.packages("tidyverse")
```

```
library(tidyverse)
```

O pipe

Todos os pacotes no tidyverse dependem do uso da função pipe %>% do pacote `magritt`.

O objetivo é evitar como o R lê códigos de dentro para fora.

A lógica com pipe: objeto -> depois as funções -> uma sequência de funções.

```r
# Exemplo 1

# R

round(exp(diff(log(runif(100, 0,1)))), 1)
```

```r
# Com pipe

runif(100, 0, 1) %>%
  log() %>%
  diff() %>%
  exp() %>%
  round(.,1) # or round(1)
```

Dplyr

O dplyr é o pacote tidyverse mais famoso do Tidyverse.

É usado para gerenciamento de dados.

As funções no Dplyr fazem o que seus nomes descrevem.

Acessando dados. Bonus: electionsBR

```
#install.packages("electionsBR")
library(electionsBR)
```

```
##
## To cite electionsBR in publications, use: citation('elections
```

```
## To learn more, visit: http://electionsbr.com
```

```
# Importando dados
d  <- party_mun_zone_fed(2014, uf = "PA")
```

```
## Processing the data...
```

```
## Done.
```

```
glimpse(d)
```

```
## Observations: 11,961
## Variables: 22
## $ DATA_GERACAO          <chr> "17/05/2018", "17/05/2018", "17/0
## $ HORA_GERACAO          <drtn> 04:15:39, 04:15:39, 04:15:39, 04
## $ ANO_ELEICAO           <dbl> 2014, 2014, 2014, 2014, 2014, 201
## $ NUM_TURNO             <dbl> 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1,
## $ DESCRICAO_ELEICAO     <chr> "Eleições Gerais 2014", "Eleições
## $ SIGLA_UF              <chr> "PA", "PA", "PA", "PA", "PA", "PA
## $ SIGLA_UE              <chr> "PA", "PA", "PA", "PA", "PA", "PA
## $ CODIGO_MUNICIPIO      <chr> "04057", "04340", "04316", "05274
## $ NOME_MUNICIPIO        <chr> "AFUÁ", "AURORA DO PARÁ", "BONITO
## $ NUMERO_ZONA           <dbl> 16, 49, 11, 2, 23, 57, 57, 62, 86
## $ CODIGO_CARGO          <dbl> 7, 7, 3, 3, 3, 5, 6, 6, 6, 6, 6,
## $ DESCRICAO_CARGO       <chr> "Deputado Estadual", "Deputado Es
## $ TIPO_LEGENDA          <chr> "C", "C", "C", "C", "C", "P", "C"
## $ NOME_COLIGACAO        <chr> "AQUI O PARÁ TEM CHANCE", "PDT, P
## $ COMPOSICAO_LEGENDA    <chr> "PTC / PT do B", "PDT / PPL / PTN
## $ SIGLA_PARTIDO         <chr> "PTC", "PPL", "PSDB", "PSDB", "PS
```

Funções do DplyR

- **select()**: select colunas
- **mutate()**: cria novas variáveis e altera existentes
- **filter()**: filtra o banco de dados
- **summarize()**: sumariza os dados
- **group_by()**: agrupa e faz análise de acordo com as variáveis.
- **slice()**: seleciona linhas específicas

Some others

- **count()**: conta dos dados por subgroup.
- **arrange()**: ordena o banco de dados por colunas
- **distinct()**: elimina repetições
- **n()**: conta quantas observações há em dados agrupados.
- **sample_n()**: Selecion N amostras do seu banco de dadosl
- **glimpse()**: Fornece um sumário dos seus dados. quickly preview the data
- **top_n()**: Seleciona por linhas de acordo com o rank das variáveis.

Select

```
# Dplyr

d %>%
   select(CODIGO_CARGO, NOME_COLIGACAO,
          SIGLA_PARTIDO, NOME_MUNICIPIO,
          QTDE_VOTOS_NOMINAIS,  QTDE_VOTOS_LEGENDA)
```

Remover Coluna

```
d %>% select(-CODIGO_CARGO)
```

Filter.

```
# Somente o PT

d %>%
  filter(SIGLA_PARTIDO == "PT")


# Duas condicoes

d %>%
  filter(SIGLA_PARTIDO == "PT" ,
         NOME_MUNICIPIO=="BELÉM")
```

Mutate: Criar novas variáveis

```
d <- d %>% filter(CODIGO_CARGO==6)

d %>%
  mutate(razaovotos=QTDE_VOTOS_LEGENDA/
      (QTDE_VOTOS_NOMINAIS+QTDE_VOTOS_LEGENDA)) %>%
  select(razaovotos, SIGLA_PARTIDO, NOME_MUNICIPIO)
```

Arrange: para ordernar

```
# What was the largest difference in gols?

d %>%
  mutate(razaovotos=QTDE_VOTOS_LEGENDA/
           (QTDE_VOTOS_NOMINAIS+QTDE_VOTOS_LEGENDA)) %>%
  select(razaovotos, SIGLA_PARTIDO,
         NOME_MUNICIPIO) %>%
  arrange(desc(razaovotos))
```

Combinando Algumas Operações

```
d_new <- d %>%

  # Repetimos o que fizemos
  mutate(razaovotos=QTDE_VOTOS_LEGENDA/
           (QTDE_VOTOS_NOMINAIS+QTDE_VOTOS_LEGENDA)) %>%

  select(razaovotos, SIGLA_PARTIDO,
         NOME_MUNICIPIO) %>%

  filter(razaovotos!=1)  %>%

  # Vamos filtrar pelos 5 maiores partidos

  filter(SIGLA_PARTIDO %in% c("PT", "PSDB",
                              "PSB", "DEM", "PP"))
```

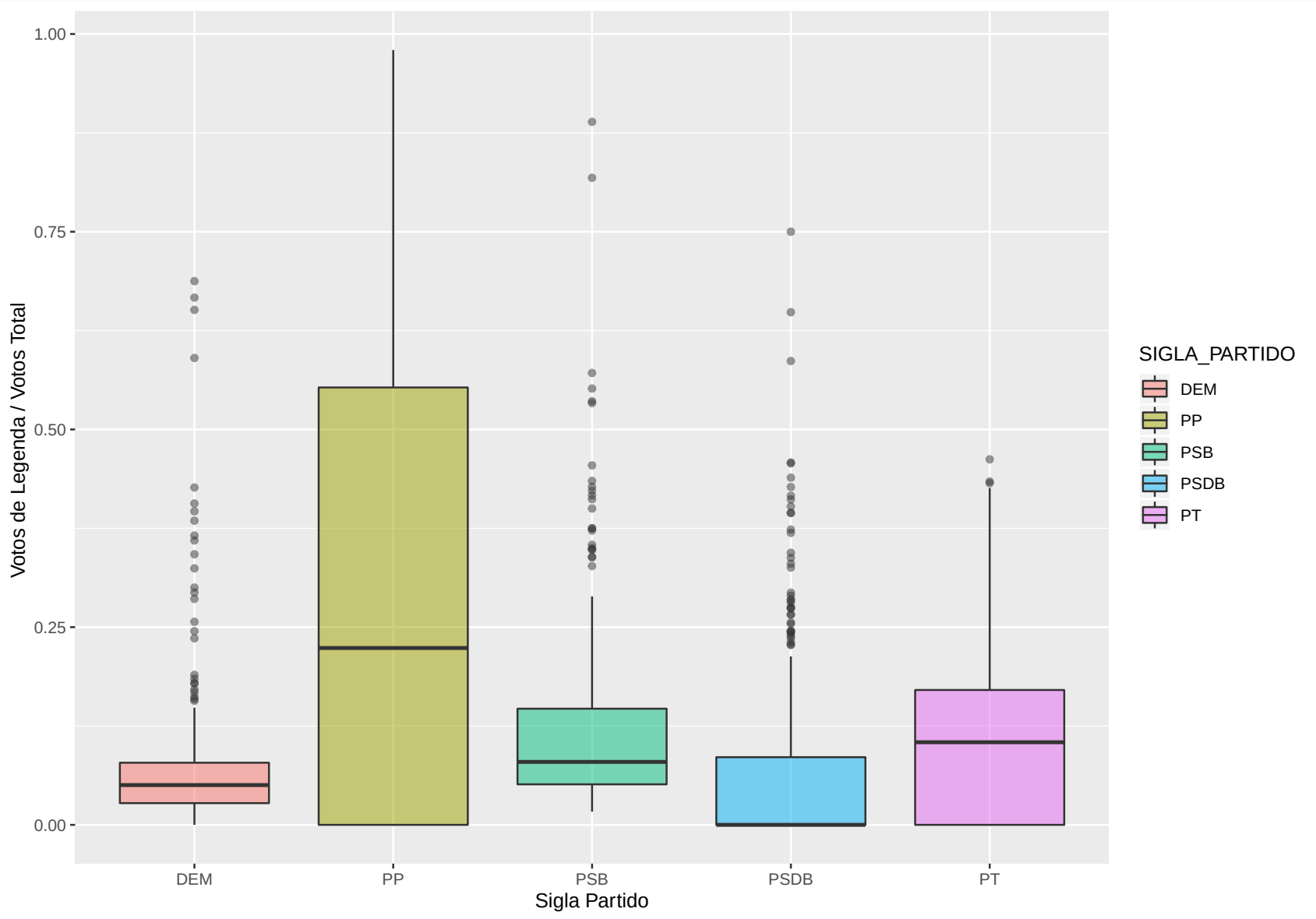
```
## # A tibble: 5 x 3
##   razaovotos SIGLA_PARTIDO NOME_MUNICIPIO
##        <dbl> <chr>         <chr>
## 1     0      PSDB          BONITO
## 2     0      PSDB          SANTA CRUZ DO ARARI
## 3     0      PSDB          NOVA IPIXUNA
## 4     0.879  PP            PIÇARRA
## 5     0.0170 DEM           TUCUMÃ
```

```r
# Grafico
ggplot(d_new, aes(y=razaovotos,
                  x=SIGLA_PARTIDO,
                  fill=SIGLA_PARTIDO)) +

  geom_boxplot(alpha=.5) +

  xlab("Sigla Partido") +

  ylab("Votos de Legenda / Votos Total ")
```


1.00
0.75
0.50
0.25
0.00
Votos de Legenda / Votos Total
DEM
PP
PSB
PSDB
PT
Sigla Partido
SIGLA_PARTIDO
DEM
PP
PSB
PSDB
PT

Group by + sumarize : Agregar por grupos e calcular valores

- Use group_by para agregar
- Use summarize para calcular algo do seu interesse
- Use ungroup para desagrupar

Summarize: Transforma várias linhas em uma.

Exemplos de operações dentro de summarize

- min(x) - minimo de x.
- max(x) - máximo de x.
- mean(x) - média de x.
- median(x) - mediana de x.
- quantile(x, p) - quantile de x.
- sd(x) - desvio padrão de x.
- var(x) - variancia de x.

```r
# Principal Munícipio dos 5 maiores partidos

d %>%

 group_by(SIGLA_PARTIDO, NOME_MUNICIPIO) %>%

 summarise(total_votos=sum(QTDE_VOTOS_NOMINAIS,
                           na.rm = TRUE)) %>%

 filter(SIGLA_PARTIDO%in%c("DEM", "PT", "PSDB",
                           "PP", "PSB")) %>%

  top_n(1)
```

```
## Selecting by total_votos
```

```
## # A tibble: 5 x 3
## # Groups:   SIGLA_PARTIDO [5]
##   SIGLA_PARTIDO NOME_MUNICIPIO total_votos
##   <chr>         <chr>                <dbl>
## 1 DEM           CASTANHAL            74499
## 2 PP            BELÉM               331714
## 3 PSB           BELÉM                41606
## 4 PSDB          BELÉM              1052084
## 5 PT            BELÉM               261118
```

Count: Contar casos agrupados

```
# Número de Zonais que os partidos receberam votos

d %>%

  # exclue zero

  filter(QTDE_VOTOS_NOMINAIS>0) %>%

  # Contando

  count(SIGLA_PARTIDO)
```

```
## # A tibble: 30 x 2
##    SIGLA_PARTIDO     n
##    <chr>         <int>
##  1 DEM             316
##  2 PC do B         299
##  3 PDT             316
##  4 PEN             285
##  5 PHS             237
##  6 PMDB            632
##  7 PMN             278
##  8 PP              471
##  9 PPL             228
## 10 PPS             311
## # ... with 20 more rows
```

Conclusão

Nessa curta introdução, nós tocamos somente na superficie da manipulação de dados em R

Apender uma linguaguem de programação é mais do que ler um livro.

Exige muita, muita repetição.

No caso do R, há diversas formas de fazer a mesma operação. Encontre o que funciona para você.