Mini Projeto - Data Science Academy

André Campos da Silva

7 de Novembro, 2020

Projeto - Analise Ocorrência Zica Virus

Realizar uma analise exploratória das ocorrências do Zica virus em determinadas datas.

Coletando os dados

```
# Carrego os pacotes necessários para o projeto

#install.packages('tidyverse')
#install.packages("plyr")
#install.packages("plotly")
#install.packages('sf')
#install.packages("geobr")


library('tidyverse')
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

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

```
library("plyr")
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library('plotly')
```

```
##
## Attaching package: 'plotly'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, mutate, rename, summarise
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library('sf')
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
```

```
library('geobr')
library('ggthemes')
```

```
# Carrego os dados que serão usados para a análise.
arquivos <- list.files('F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos', full.names = TRUE)
arquivos
```

```
## [1] "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos/Epidemiological_Bulletin-2016-04-02.csv"
## [2] "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos/Epidemiological_Bulletin-2016-04-23.csv"
## [3] "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos/Epidemiological_Bulletin-2016-04-30.csv"
## [4] "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos/Epidemiological_Bulletin-2016-05-07.csv"
## [5] "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos/Epidemiological_Bulletin-2016-05-14.csv"
## [6] "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos/Epidemiological_Bulletin-2016-05-21.csv"
## [7] "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos/Epidemiological_Bulletin-2016-05-28.csv"
## [8] "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Resolução/Arquivos/Epidemiological_Bulletin-2016-06-11.csv"
```

```
class(arquivos)
```

```
## [1] "character"
```

```
# Uso o lapply pra colocar todos os arquivos em uma lista.
df_list <- lapply(arquivos, read_csv)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   report_date = col_date(format = ""),
##   location = col_character(),
##   location_type = col_character(),
##   data_field = col_character(),
##   data_field_code = col_character(),
##   time_period = col_logical(),
##   time_period_type = col_logical(),
##   value = col_double(),
##   unit = col_character()
## )
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   report_date = col_date(format = ""),
##   location = col_character(),
##   location_type = col_character(),
##   data_field = col_character(),
##   data_field_code = col_character(),
##   time_period = col_logical(),
##   time_period_type = col_logical(),
##   value = col_double(),
##   unit = col_character()
## )
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   report_date = col_date(format = ""),
##   location = col_character(),
##   location_type = col_character(),
##   data_field = col_character(),
##   data_field_code = col_character(),
##   time_period = col_logical(),
##   time_period_type = col_logical(),
##   value = col_double(),
##   unit = col_character()
## )
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   report_date = col_date(format = ""),
##   location = col_character(),
##   location_type = col_character(),
##   data_field = col_character(),
##   data_field_code = col_character(),
##   time_period = col_logical(),
##   time_period_type = col_logical(),
##   value = col_double(),
```

```
##   unit = col_character()
## )
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   report_date = col_date(format = ""),
##   location = col_character(),
##   location_type = col_character(),
##   data_field = col_character(),
##   data_field_code = col_character(),
##   time_period = col_logical(),
##   time_period_type = col_logical(),
##   value = col_double(),
##   unit = col_character()
## )
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   report_date = col_date(format = ""),
##   location = col_character(),
##   location_type = col_character(),
##   data_field = col_character(),
##   data_field_code = col_character(),
##   time_period = col_logical(),
##   time_period_type = col_logical(),
##   value = col_double(),
##   unit = col_character()
## )
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   report_date = col_date(format = ""),
##   location = col_character(),
##   location_type = col_character(),
##   data_field = col_character(),
##   data_field_code = col_character(),
##   time_period = col_logical(),
##   time_period_type = col_logical(),
##   value = col_double(),
##   unit = col_character()
## )
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   report_date = col_date(format = ""),
##   location = col_character(),
##   location_type = col_character(),
##   data_field = col_character(),
##   data_field_code = col_character(),
##   time_period = col_logical(),
```

```
##   time_period_type = col_logical(),
##   value = col_double(),
##   unit = col_character()
## )
```

```
# Uso a função do.call, para trazer toda lista para um só DF.
 df_base <- do.call(rbind,df_list)
 df <- do.call(rbind,df_list)
```

Tratamento dos dados

```
# Retiro do  dataset as colunas que não são necessarias para a analise.
 df$data_field_code <- NULL
 df$data_field <- NULL
 df$time_period <- NULL
 df$time_period_type <- NULL
```

```
 # Aqui tenho o dataset que precisamos tratar, temos que jogar a região para uma nova coluna ass
im
 # como o pais, a ideia é deixar uma coluna pra pais, região e estado.
 head(df)
```

```
## # A tibble: 6 x 5
##   report_date location        location_type value unit
##   <date>      <chr>           <chr>         <dbl> <chr>
## 1 2016-04-02  Norte           region         6295 cases
## 2 2016-04-02  Brazil-Rondonia state           618 cases
## 3 2016-04-02  Brazil-Acre     state           375 cases
## 4 2016-04-02  Brazil-Amazonas state          1520 cases
## 5 2016-04-02  Brazil-Roraima  state            44 cases
## 6 2016-04-02  Brazil-Para     state           771 cases
```

```
 glimpse(df)
```

```
## Rows: 264
## Columns: 5
## $ report_date   <date> 2016-04-02, 2016-04-02, 2016-04-02, 2016-04-02, 2016...
## $ location      <chr> "Norte", "Brazil-Rondonia", "Brazil-Acre", "Brazil-Am...
## $ location_type <chr> "region", "state", "state", "state", "state", "state"...
## $ value         <dbl> 6295, 618, 375, 1520, 44, 771, 74, 2893, 30286, 1202,...
## $ unit          <chr> "cases", "cases", "cases", "cases", "cases", "cases",...
```

```
 # Tiro o nome Brazil antes de cada estado;
 df$location <- gsub('Brazil-', '',df$location)
 # Adciono a variavél Região que vou usar na programação que vou criar para
 # atribuir a região a cada estado em uma coluna.
 df$Region <- NA
```

```
 # Crio os vetores com os nomes dos estados de cada região para usar na formula.
 Norte <- c('Acre', 'Amazonas', 'Roraima', 'Para', 'Amapa', 'Tocantins', 'Rondonia')
 Nordeste <- c('Maranhao','Piaui','Ceara', 'Rio_Grande_do_Norte','Paraiba',
               'Pernambuco','Alagoas', 'Sergipe','Bahia')
 Sudeste <- c('Minas_Gerais','Espirito_Santo','Rio_de_Janeiro','Sao_Paulo')

 Sul <- c('Parana', 'Santa_Catarina', 'Rio_Grande_do_Sul')
 Centro_Oeste <- c('Mato_Grosso_do_Sul', 'Mato_Grosso','Goias','Distrito_Federal')
```

```
 # Programação que usa os vetores de região criados acima para atribuir os valores certos
 # nomeando cada Região na linha correta na variável região.

for (i in 1:length(df$location)){
   if (df$location[i] %in% Norte){
        df$Region[i] = 'Norte'

   }else if
      (df$location[i] %in% Nordeste){
        df$Region[i] = 'Nordeste'

   }else if
      (df$location[i] %in% Sudeste){
        df$Region[i] = 'Sudeste'

   }else if
      (df$location[i] %in% Sul){
        df$Region[i] = 'Sul'

   }else if
      (df$location[i] %in% Centro_Oeste){
        df$Region[i] = 'Centro_Oeste'
   }
}

head(df)
```

```
## # A tibble: 6 x 6
##   report_date location location_type value unit  Region
##   <date>      <chr>    <chr>         <dbl> <chr> <chr>
## 1 2016-04-02  Norte    region         6295 cases <NA>
## 2 2016-04-02  Rondonia state           618 cases Norte
## 3 2016-04-02  Acre     state           375 cases Norte
## 4 2016-04-02  Amazonas state          1520 cases Norte
## 5 2016-04-02  Roraima  state            44 cases Norte
## 6 2016-04-02  Para     state           771 cases Norte
```

```
# Crio um outro data frame para pegar os valores totais de cada região por data que ficaram
# com NA na variavel Region, pois eu vou retirar eles do data frame, pois esse somatório eu poss
o pegar
# depois com o pacote dplyr summarizando, mas estou salvando para comparar para verificar se ouv
e algum erro.
dfNulos <- subset(df, is.na(Region))
```

```
# Faço uma copia do DF que tratei a variavel Region, tirando os valores nulos que salvei acima,
# ficando com o data set quase da forma esperada, depois eu criou uma variavel pais colocando Br
asil
# Só para constar mesmo, embora não seja necessario pois são regiões apenas do Brasil.
df2 <- df[!is.na(df$Region),]
```

```
# Tiro as variveis location_type e unit que não são relevantes mais, e ja add uma variável,
# Country como passei acima, e no final faço o segundo select para acertar as colunas nas posiçõ
es
# que acho mais interessante.
df2 <- df2 %>%
        select(report_date, Region, location, value)%>%
        mutate(Country = 'Brazil')%>%
        select(report_date,Country, Region, location, value)

# Como é um data set do Brasil eu vou renomear as variáveis para os nomes PT.
colunas <- c('Data_reportada', 'Pais', 'Regiao', 'Estado', 'Qtd_Casos')
names(df2) <- colunas
View(df2)
glimpse(df2)
```

```
## Rows: 216
## Columns: 5
## $ Data_reportada <date> 2016-04-02, 2016-04-02, 2016-04-02, 2016-04-02, 201...
## $ Pais           <chr> "Brazil", "Brazil", "Brazil", "Brazil", "Brazil", "B...
## $ Regiao         <chr> "Norte", "Norte", "Norte", "Norte", "Norte", "Norte"...
## $ Estado         <chr> "Rondonia", "Acre", "Amazonas", "Roraima", "Para", "...
## $ Qtd_Casos      <dbl> 618, 375, 1520, 44, 771, 74, 2893, 1202, 7, 156, 640...
```

```
dim(df2)
```

```
## [1] 216   5
```

```
# Salvo em um arquivo o dataset ja tratado.
# write_csv(df2, "F:/Cursos/Formação-Ciêntista-de-Dados-DSA/Big-Data-Analytics-com-R-e-Microsoft
-Azure-Machine-Learning/19.Mini-Projeto-Ocorrencia-de-ZicaVirus-em-Grafico-Interativo/Minha_Reso
lução/Arquivos/ZicaVirus_Analyse_Tratado.csv")
```

Analise Exploratória

```
# Total de casos agrupando por data e região.
df2%>%
ddply(.(Data_reportada, Regiao),
      summarize,
     Media_Casos = sum(Qtd_Casos))
```

```
##    Data_reportada       Regiao Media_Casos
## 1      2016-04-02 Centro_Oeste       17504
## 2      2016-04-02     Nordeste       30286
## 3      2016-04-02        Norte        6295
## 4      2016-04-02      Sudeste       35505
## 5      2016-04-02          Sul        1797
## 6      2016-04-23 Centro_Oeste       20101
## 7      2016-04-23     Nordeste       43000
## 8      2016-04-23        Norte        8545
## 9      2016-04-23      Sudeste       46318
## 10     2016-04-23          Sul        2197
## 11     2016-04-30 Centro_Oeste       21364
## 12     2016-04-30     Nordeste       47709
## 13     2016-04-30        Norte        8379
## 14     2016-04-30      Sudeste       48027
## 15     2016-04-30          Sul        2343
## 16     2016-05-07 Centro_Oeste       21756
## 17     2016-05-07     Nordeste       51065
## 18     2016-05-07        Norte        8053
## 19     2016-05-07      Sudeste       54803
## 20     2016-05-07          Sul        2431
## 21     2016-05-14 Centro_Oeste       21756
## 22     2016-05-14     Nordeste       51065
## 23     2016-05-14        Norte        8053
## 24     2016-05-14      Sudeste       54803
## 25     2016-05-14          Sul        2431
## 26     2016-05-21 Centro_Oeste       22508
## 27     2016-05-21     Nordeste       54165
## 28     2016-05-21        Norte        8432
## 29     2016-05-21      Sudeste       61309
## 30     2016-05-21          Sul        2491
## 31     2016-05-28 Centro_Oeste       24683
## 32     2016-05-28     Nordeste       59745
## 33     2016-05-28        Norte        9022
## 34     2016-05-28      Sudeste       65328
## 35     2016-05-28          Sul        2463
## 36     2016-06-11 Centro_Oeste       25246
## 37     2016-06-11     Nordeste       61829
## 38     2016-06-11        Norte       10645
## 39     2016-06-11      Sudeste       65820
## 40     2016-06-11          Sul        2392
```

```
# Total de casos por região.
df2%>%
  select(Regiao,Data_reportada,Qtd_Casos)%>%
        group_by(Regiao)%>%
  filter(Data_reportada == '2016-06-11')%>%
  summarise(Total = sum(Qtd_Casos))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 5 x 2
##   Regiao       Total
##   <chr>        <dbl>
## 1 Centro_Oeste 25246
## 2 Nordeste     61829
## 3 Norte        10645
## 4 Sudeste      65820
## 5 Sul           2392
```

```
# Total de casos agrupando por região e estado.
df2%>%
  select(Regiao,Data_reportada,Qtd_Casos,Estado)%>%
  group_by(Regiao,Estado)%>%
  filter(Data_reportada == '2016-06-11')%>%
  summarise(Total = sum(Qtd_Casos))
```

```
## `summarise()` regrouping output by 'Regiao' (override with `.groups` argument)
```

```
## # A tibble: 27 x 3
## # Groups:   Regiao [5]
##    Regiao       Estado             Total
##    <chr>        <chr>              <dbl>
##  1 Centro_Oeste Distrito_Federal     367
##  2 Centro_Oeste Goias               4132
##  3 Centro_Oeste Mato_Grosso        19985
##  4 Centro_Oeste Mato_Grosso_do_Sul   762
##  5 Nordeste     Alagoas             3847
##  6 Nordeste     Bahia              46427
##  7 Nordeste     Ceara               2358
##  8 Nordeste     Maranhao            2840
##  9 Nordeste     Paraiba             2889
## 10 Nordeste     Pernambuco           394
## # ... with 17 more rows
```

```
# Quantidade de casos por data.
df2%>%
  ddply(.(Data_reportada),
        summarize,
        Casos = sum(Qtd_Casos))
```

```
##   Data_reportada  Casos
## 1     2016-04-02  91387
## 2     2016-04-23 120161
## 3     2016-04-30 127822
## 4     2016-05-07 138108
## 5     2016-05-14 138108
## 6     2016-05-21 148905
## 7     2016-05-28 161241
## 8     2016-06-11 165932
```

```
# Quantidade de casos por Estado
df2%>%
  select(Estado,Data_reportada,Qtd_Casos)%>%
  group_by(Estado)%>%
  filter(Data_reportada == '2016-06-11')%>%
  summarise(Total = (Qtd_Casos))
```

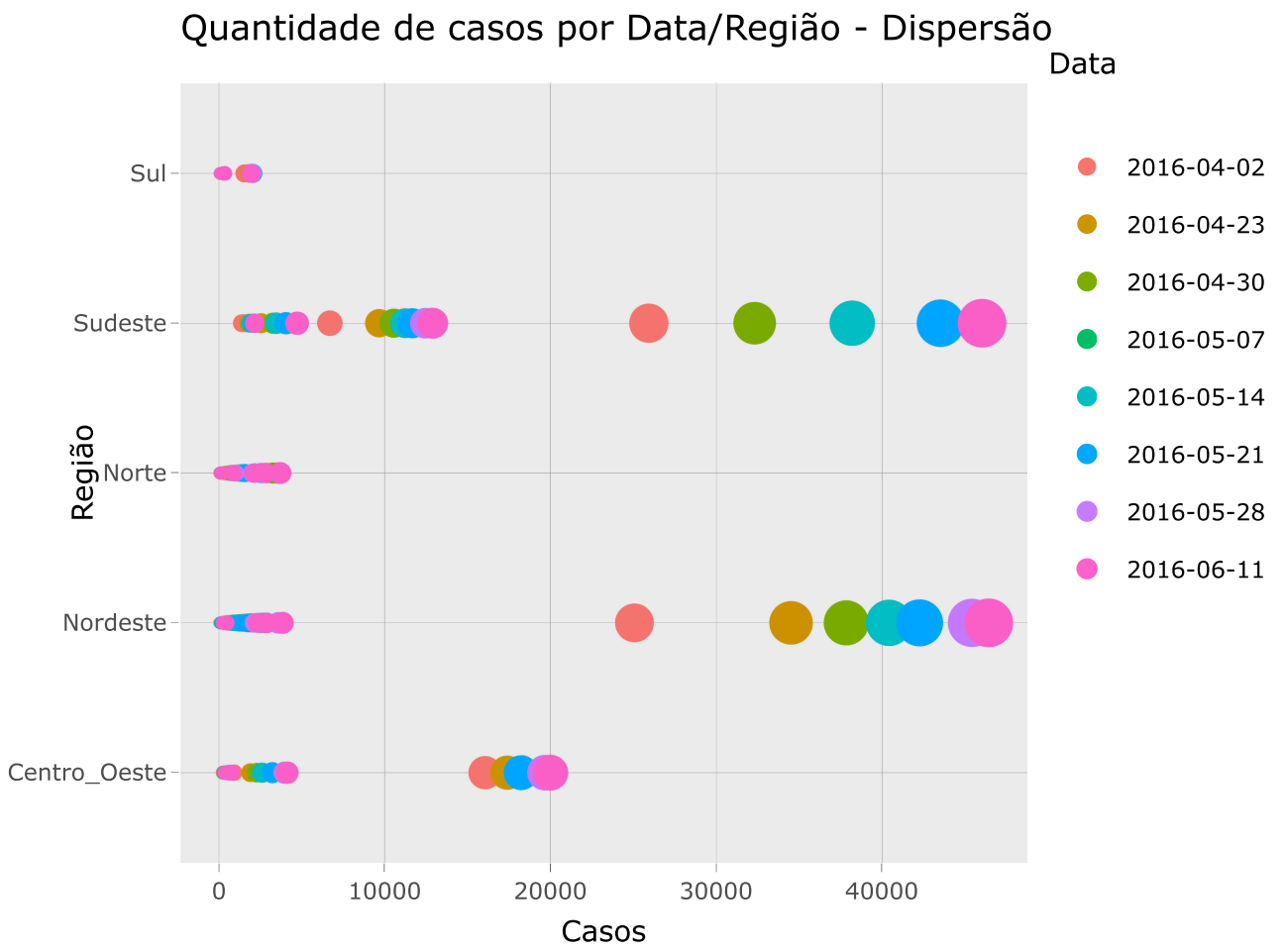
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 27 x 2
##    Estado           Total
##    <chr>            <dbl>
##  1 Acre               846
##  2 Alagoas           3847
##  3 Amapa              189
##  4 Amazonas          3713
##  5 Bahia            46427
##  6 Ceara             2358
##  7 Distrito_Federal   367
##  8 Espirito_Santo    2166
##  9 Goias             4132
## 10 Maranhao          2840
## # ... with 17 more rows
```

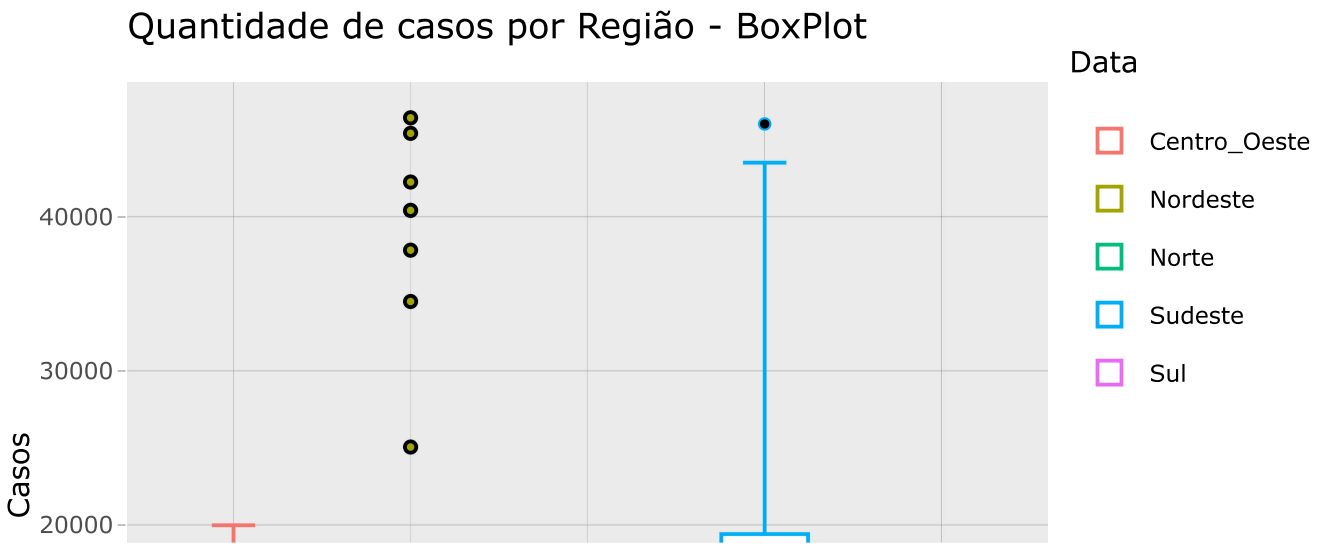
```
# Grafico de Dispersão da quantidade de casos separando por Data e Região
Caso_Regiao <- df2 %>%
  ggplot(aes( x =Qtd_Casos, y = Regiao,size=Qtd_Casos )) +
  geom_point(aes(color = as.factor(Data_reportada),
                 text = paste0(
                         "Data: ",as.factor(Data_reportada),"\n",
                         "Quantidade de casos: ",Qtd_Casos,"\n",
                         "Região: ",Regiao)))+
  labs(x ='Casos', y ="Região", title = 'Quantidade de casos por Data/Região - Dispersão ',
       color = 'Data',
       size = '')
```

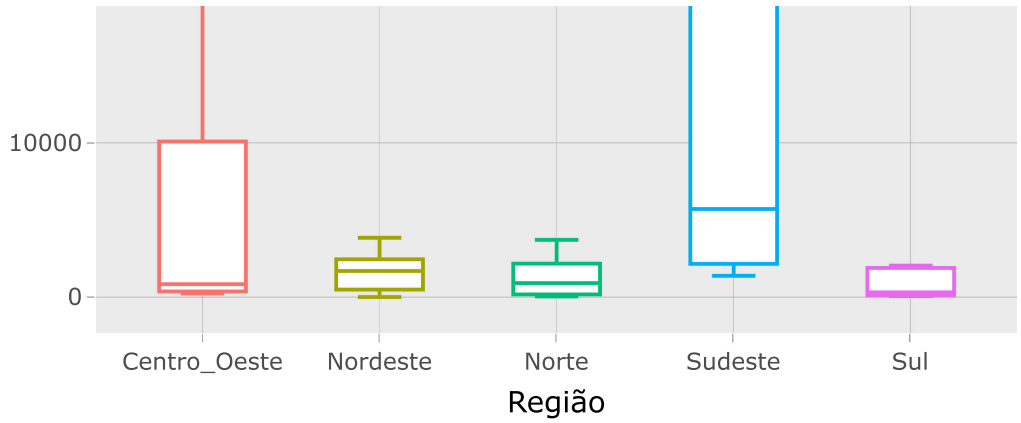
```
## Warning: Ignoring unknown aesthetics: text
```

```
ggplotly(Caso_Regiao,tooltip = "text")
```




```
# Grafigo de caixa com a quantidade total de casos por região e suas métricas.
Caso_regiao2 <- df2 %>%
        ggplot(aes( y =Qtd_Casos, x = Regiao ,color = as.factor(Regiao)))+
        geom_boxplot()+
        labs(y ='Casos', x ="Região", title = 'Quantidade de casos por Região - BoxPlot',
             color = 'Data')
ggplotly(Caso_regiao2)
```

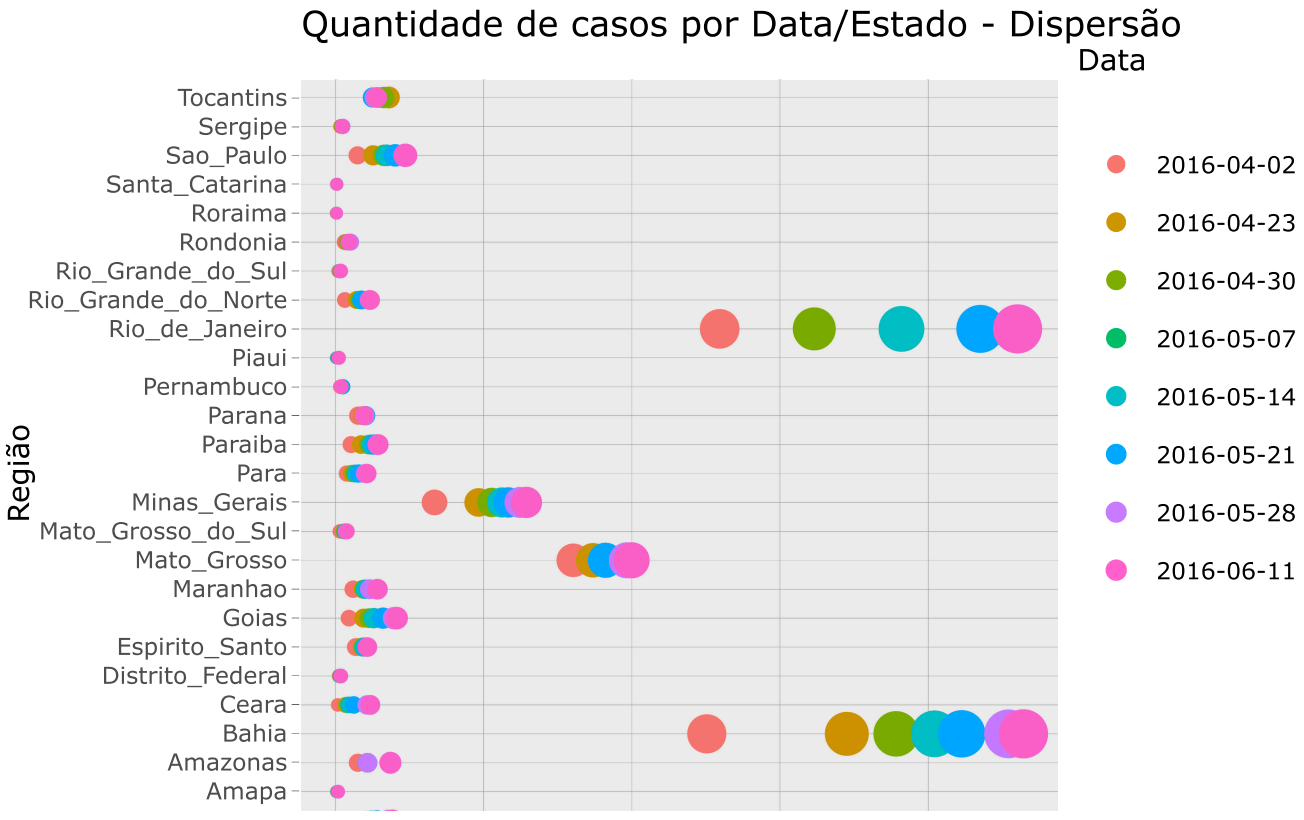


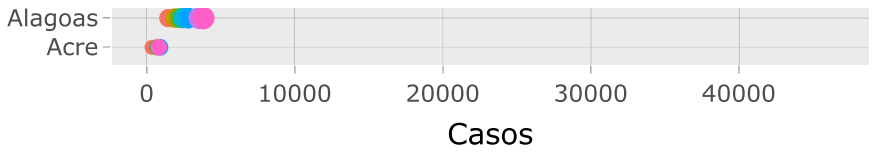



```
# Grafico de Dispersão da quantidade de casos separando por Data e Estado.
Caso_Estado <- df2 %>%
        ggplot(aes( x =Qtd_Casos, y = Estado,size=Qtd_Casos )) +
        geom_point(aes(color = as.factor(Data_reportada),
                       text = paste0(
                               "Data: ",as.factor(Data_reportada),"\n",
                               "Quantidade de casos: ",Qtd_Casos,"\n",
                               "Estado: ",Estado)))+
        labs(x ='Casos', y ="Região", title = 'Quantidade de casos por Data/Estado - Dispersão '
,
             color = 'Data',
             size = '')
```

```
## Warning: Ignoring unknown aesthetics: text
```

```
ggplotly(Caso_Estado,tooltip = "text")
```

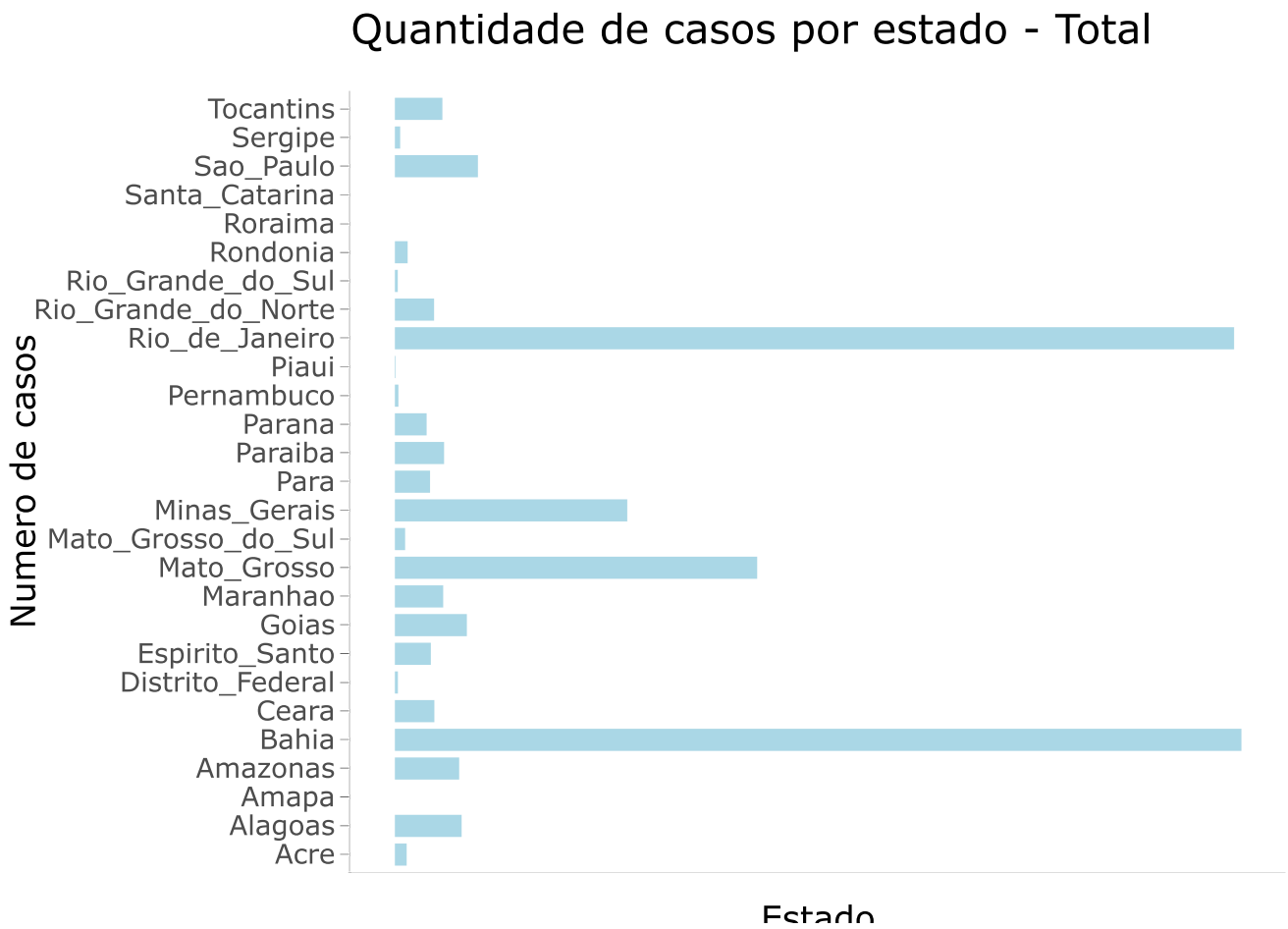





```
# Grafico de barras com a Quantidade de casos por Estado.
Caso_estado2 <- df2 %>%
  select(Estado,Data_reportada,Qtd_Casos)%>%
  group_by(Estado)%>%
  filter(Data_reportada == '2016-06-11')%>%
  summarise(Total = sum(Qtd_Casos)) %>%
  ggplot(aes(x = Total, y = Estado,  text = paste0(
    "Casos: ",Total, "\n",
    "Estado: ", Estado
  )))+
  geom_bar(stat = "identity",color = "white", fill = "lightblue")+
  theme_classic(base_size = 13) +
  labs(title = 'Quantidade de casos por estado - Total',
       x = 'Estado', y = 'Numero de casos')+
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

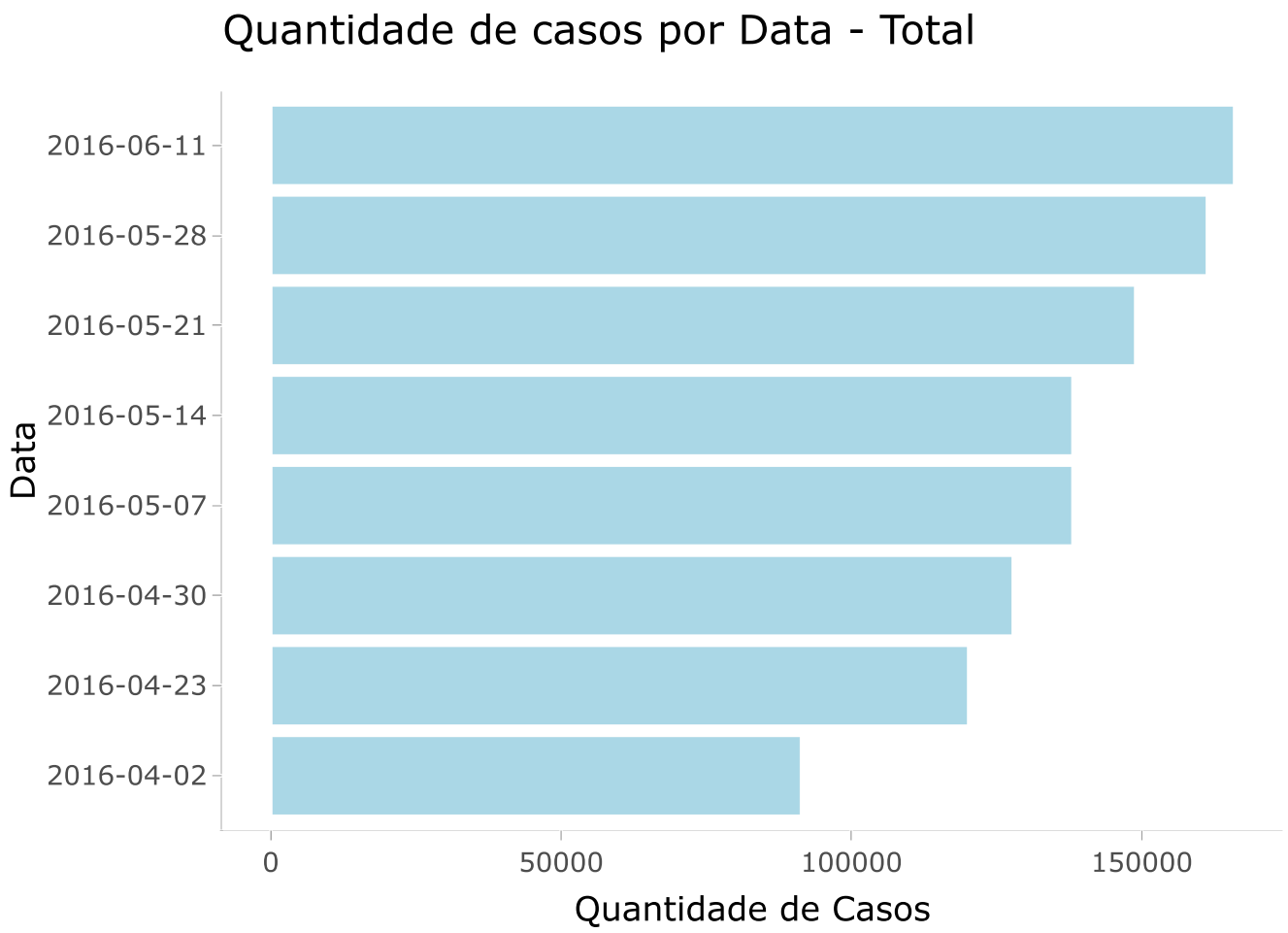
```
ggplotly(Caso_estado2,tooltip = "text")
```



Estado

```
# Quantidade total de casos por data reportada.
Data_casos <- df2%>%
  ddply(.(Data_reportada),
        summarize,
        Casos = sum(Qtd_Casos))%>%
  ggplot(aes(x = Casos, y = as.factor(Data_reportada
             ),
              text = paste0(
    "Casos: ",Casos, "\n",
    "Data: ", Data_reportada
  )))+
  geom_bar(stat = "identity",color = "white", fill = "lightblue")+
  theme_classic(base_size = 13) +
  labs(title = 'Quantidade de casos por Data - Total',
       x = 'Quantidade de Casos', y = 'Data')

ggplotly(Data_casos,tooltip = "text")
```



```
# Faço um facet wrap com graficos de barra com total de casos por região
# em cada data.
Facet_data <- ggplot(df2, aes(x = Qtd_Casos,y = Regiao, group = Regiao,
                fill = Regiao))+
  geom_bar(stat = "identity")+
  theme_classic(base_size = 9) +
  labs(title = 'Quantidade de casos por região - data',
       x = 'Quantidade de Casos', y = 'Região') +
  facet_wrap(~Data_reportada)

Facet_data
```


```
# Mapa Interativo

# Aqui eu crio do data frame que forma um mapa do brasil, e ja acerto a variável com o nome do e
stado
# para que fique igual ao o do df para que eu possa fazer o join corretamente.
map <- read_state(code_state ="all", year=2019,)
```

```
## Using year 2019
```

```
## Loading data for the whole country
```

```
##
  |
  |                                                                      |   0%
  |
  |===                                                                   |   4%
  |
  |=====                                                                 |   7%
  |
  |========                                                              |  11%
  |
  |==========                                                            |  15%
  |
  |=============                                                         |  19%
  |
  |=================                                                     |  22%
  |
  |===================                                                   |  26%
  |
  |======================                                                |  30%
  |
  |========================                                              |  33%
  |
  |============================                                          |  37%
  |
  |==============================                                        |  41%
  |
  |================================                                      |  44%
  |
  |==================================                                    |  48%
  |
  |====================================                                  |  52%
  |
  |========================================                              |  56%
  |
  |==========================================                            |  59%
  |
  |============================================                          |  63%
  |
  |================================================                      |  67%
  |
  |==================================================                    |  70%
  |
  |====================================================                  |  74%
  |
  |======================================================                |  78%
  |
  |==========================================================            |  81%
  |
  |============================================================          |  85%
  |
  |==============================================================        |  89%
  |
  |================================================================      |  93%
```

```
  |
  |=================================================================    |  96%
  |
  |======================================================================| 100%
```

```
estados <- c("Rondonia","Acre","Amazonas" , "Roraima","Para","Amapa","Tocantins","Maranhao","Pia
ui",
             "Ceara","Rio_Grande_do_Norte","Paraiba", "Pernambuco", "Alagoas","Sergipe","Bahia"
             ,"Minas_Gerais","Espirito_Santo","Rio_de_Janeiro","Sao_Paulo","Parana","Santa_Catar
ina"
             ,"Rio_Grande_do_Sul","Mato_Grosso_do_Sul" ,"Mato_Grosso", "Goias","Distrito_Federa
l"  )
map$name_state <- estados
unique(df2$Estado)
```

```
##  [1] "Rondonia"            "Acre"                "Amazonas"
##  [4] "Roraima"             "Para"                "Amapa"
##  [7] "Tocantins"           "Maranhao"            "Piaui"
## [10] "Ceara"               "Rio_Grande_do_Norte" "Paraiba"
## [13] "Pernambuco"          "Alagoas"             "Sergipe"
## [16] "Bahia"               "Minas_Gerais"        "Espirito_Santo"
## [19] "Rio_de_Janeiro"      "Sao_Paulo"           "Parana"
## [22] "Santa_Catarina"      "Rio_Grande_do_Sul"   "Mato_Grosso_do_Sul"
## [25] "Mato_Grosso"         "Goias"               "Distrito_Federal"
```

```
unique(map$name_state)
```

```
##  [1] "Rondonia"            "Acre"                "Amazonas"
##  [4] "Roraima"             "Para"                "Amapa"
##  [7] "Tocantins"           "Maranhao"            "Piaui"
## [10] "Ceara"               "Rio_Grande_do_Norte" "Paraiba"
## [13] "Pernambuco"          "Alagoas"             "Sergipe"
## [16] "Bahia"               "Minas_Gerais"        "Espirito_Santo"
## [19] "Rio_de_Janeiro"      "Sao_Paulo"           "Parana"
## [22] "Santa_Catarina"      "Rio_Grande_do_Sul"   "Mato_Grosso_do_Sul"
## [25] "Mato_Grosso"         "Goias"               "Distrito_Federal"
```

```
# Pego apenas as duas colunas que me interessam no data frame do mapa, que é a do estado ja trad
ado e o
# e o geom que são as coordenadas de cada estado e coloco em um df chamado geom.
geom <- map %>%
  select (name_state, geom)
head(geom)
```

```
## Simple feature collection with 6 features and 1 field
## geometry type:  GEOMETRY
## dimension:      XY
## bbox:           xmin: -73.99045 ymin: -13.6937 xmax: -46.06151 ymax: 5.271841
## geographic CRS: SIRGAS 2000
##   name_state                           geom
## 1   Rondonia POLYGON ((-65.3815 -10.4290...
## 2       Acre POLYGON ((-71.07772 -9.8277...
## 3   Amazonas POLYGON ((-69.83766 -3.6865...
## 4    Roraima POLYGON ((-63.96008 2.47312...
## 5       Para MULTIPOLYGON (((-51.43248 -...
## 6      Amapa MULTIPOLYGON (((-50.45011 2...
```

```
class(geom)
```

```
## [1] "sf"         "data.frame"
```

```
# Crio um dataset apenas com os dados da ultima data para plotar no grafico o total por estado,
 usando como referencia a ultima data reportada
df3 <- df2%>%
  select(Estado,Data_reportada,Qtd_Casos)%>%
  group_by(Estado)%>%
  filter(Data_reportada == '2016-06-11')%>%
  summarise(Total = (Qtd_Casos))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# Faço o join do df geom com o df2 e criou o df mapa_zica contendo os dados dos dois DFs.
# com isso eu teho um data frame com os dados das pesquisas e as coordenadas dos estados para pl
otar no mapa.
Mapa_zica <- left_join(geom, df3, by = c('name_state' = 'Estado'))
Mapa_zica = sf::st_cast(Mapa_zica, "MULTIPOLYGON")
class(Mapa_zica)
```

```
## [1] "sf"         "data.frame"
```

```
head(Mapa_zica)
```

```
## Simple feature collection with 6 features and 2 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -73.99045 ymin: -13.6937 xmax: -46.06151 ymax: 5.271841
## geographic CRS: SIRGAS 2000
##   name_state Total                           geom
## 1   Rondonia   898 MULTIPOLYGON (((-65.3815 -1...
## 2       Acre   846 MULTIPOLYGON (((-71.07772 -...
## 3   Amazonas  3713 MULTIPOLYGON (((-69.83766 -...
## 4    Roraima    83 MULTIPOLYGON (((-63.96008 2...
## 5       Para  2121 MULTIPOLYGON (((-51.43248 -...
## 6      Amapa   189 MULTIPOLYGON (((-50.45011 2...
```

```
tail(Mapa_zica)
```

```
## Simple feature collection with 6 features and 2 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -61.63335 ymin: -33.75118 xmax: -45.90716 ymax: -7.349028
## geographic CRS: SIRGAS 2000
##            name_state Total                           geom
## 22     Santa_Catarina    97 MULTIPOLYGON (((-48.60084 -...
## 23  Rio_Grande_do_Sul   360 MULTIPOLYGON (((-49.70392 -...
## 24 Mato_Grosso_do_Sul   762 MULTIPOLYGON (((-57.83371 -...
## 25        Mato_Grosso 19985 MULTIPOLYGON (((-52.61926 -...
## 26              Goias  4132 MULTIPOLYGON (((-52.36102 -...
## 27   Distrito_Federal   367 MULTIPOLYGON (((-47.81455 -...
```

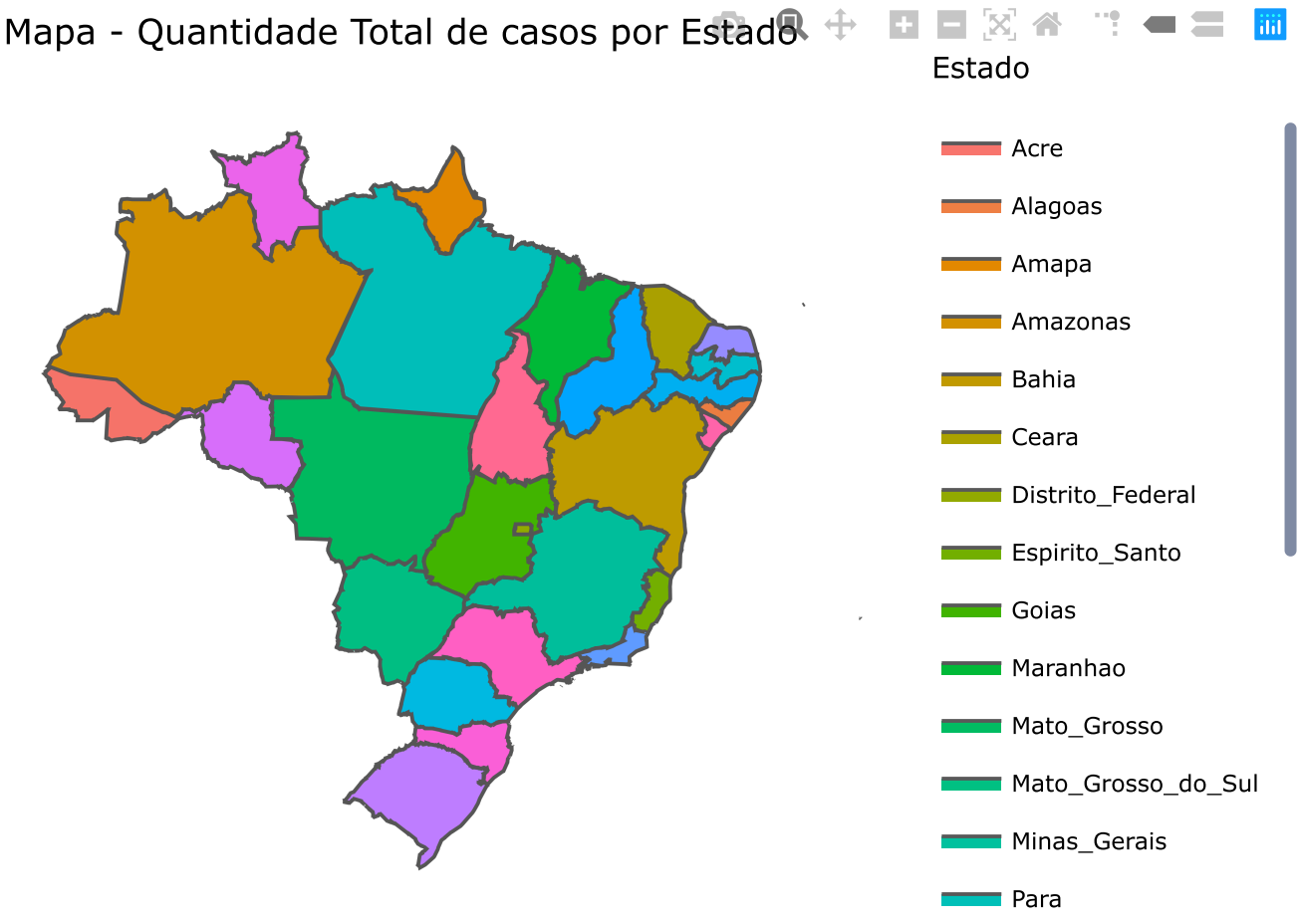
```
glimpse(Mapa_zica)
```

```
## Rows: 27
## Columns: 3
## $ name_state <chr> "Rondonia", "Acre", "Amazonas", "Roraima", "Para", "Amap...
## $ Total      <dbl> 898, 846, 3713, 83, 2121, 189, 2795, 2840, 241, 2358, 23...
## $ geom       <MULTIPOLYGON [°]> MULTIPOLYGON (((-65.3815 -1..., MULTIPOLYGO...
```

```r
# Mapa interativo onde eu ao passar o mouse pelo Estado no mapa, ele mostra o nome do Estado e o
# valor total de casos.

MapaP_zica <- Mapa_zica %>%
    ggplot(aes(fill = name_state,
               text =paste0(
                 "Estado: ",name_state,"\n",
                 "Casos: ",Total)))+
  geom_sf()+
  theme(
    legend.position = "bottom",
    panel.background = element_blank(),
    panel.grid.major = element_line(color = "transparent"),
    axis.text = element_blank(),
    axis.ticks = element_blank()
  ) +
  labs(title = "Mapa - Quantidade Total de casos por Estado",
       fill = 'Estado')

ggplotly(MapaP_zica,tooltip = "text")
```



Conclusão

Com a análise exploratórias, conseguimos mostrar de várias formas diferentes informações que nos ajudam a tirar conclusões sobre a analise realizada.

1 – Em um primeiro momento percebemos um aumento continuo dos casos desde a primeira data até a última reportada, que naturalmente é o padrão uma vez que com a pouca informação inicial menos medidas preventivas são tomadas ocasionando esse aumento gradativo.

2 – Constata-se que os maiores casos se concentram nas regiões Nordeste e Sudeste seguindo um pouco mais de trás da região centro-Oeste

3 – Porem com uma análise mais detalhada, percebe-se que essa discrepância nessas 3 Regiões não ocorre de forma equilibrada entre os Estados, mas sim uma em Estado especifico para cada Região, criando assim o que é chamado de Outliers, Estados esses que fogem da distribuição normal dos casos por Região.

4 – Finalizando a Analise, é constatado que os Estados mais atingidos pelo zica vírus foram Bahia, Rio de janeiro, Mato grosso e Minas Gerais, já os demais com uma proporção bem menor de casos e equilibrada entre eles.