Análise exploratória de dados de saneamento dos estados brasileiros

Mario Peres

October 20, 2021

Nesse projeto foi desenvolvida uma análise exploratória dos dados de condições de saneamento nos estados do Brasil. O principal objetivo era a criação de um gráfico de pontos, mostrando a relação entre a proporção da população atendida com abastecimento e a proporção da população atendida com esgotamento, considerando todos os estados (diferentes cores) e o tamanho da população (tamanho dos pontos). Porém, mais algumas análises, principalmente gráficas, foram executadas.

Finalmente, você pode acompanhar todo o desenvolvimento desse projeto, pois explico passo a passo todos os procedimentos que foram utilizados. Primeiramente, vamos carregar os pacotes necessários, os dados e vamos visualizar os dados em formato de tabela. Os dados foram extraídos da Confederação Nacional de Municípios (CNM). url: https://www.cnm.org.br/municipios/registros/100151/todos

```
# Carregando os pacotes
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(plotly)
```

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

```
## 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(ggpubr)
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
# Carregando os dados
df <- fread('data/planilha_resumo.csv', skip = 8)

# Visualizando os dados
View(df)
```

Como visualizado na figura, o cabeçalho da tabela de dados está bem desorganizada, e portanto, vamos fazer um trabalho de limpeza e organização dos dados.

```
# Fazendo o update do dataframe com as linhas 4, 1 e todas as outras partir da 5
df <- rbind(rbind(df[4,],df[1,]), df[5:length(df$V1),])

# Nomeando a variável que contem os estados, pois a linha 1 será utilizada como nome das variáveis (col
df[1,1] <- "State"

# Renomeando as colunas, utilizando a linha 1 como character
colnames(df) <- sapply(df[1,], as.character)

# Removendo a primeira linha dos dados, pois já não precisamos mais dela
df <- df[-1,]

# Visualizando as mudanças
View(df)
```

A tabela já parece bem mais limpa e organizada, mas ainda temos um pouco de trabalho para a preparação dos dados. Primeiro, vamos selecionar as variáveis de interesse, para então darmos prosseguimento com a limpeza.

```
# Selecionando as variáveis de interesse para esta análise exploratória
df <- df %>%
  select(State, GE05a, GE05b, FN002, FN003, AG001, AG007, AG010, ES001, ES005, ES006)
```

Nomeando as colunas que não tinham nome e renomeando algumas colunas para facilitar a interpretação das variáveis. Esses dados também serão usados para a criação de um dicionário de dados para consulta.

```
# Renomeando a primeira linha dos dados para facilitar o entendimento das variáveis bem como para a cri
df[1,4] <- "Receitas operacionais diretas com agua"
df[1,5] <- "Receitas operacionais diretas com esgoto"
df[1,7] <- "Volume de agua tratada em ETAs"
df[1,8] <- "Volume de agua consumido"
df[1,10] <- "Volume de esgoto coletado"
df[1,11] <- "Volume de esgoto tratado"
```

Agora vamos criar um novo dataframe com o dicionario dos dados, contendo o nome das variáveis e sua correspondente descrição. Depois, removemos a primeira linha da tabela de dados e a coluna Symbol, visto que a tabela possui como index esses mesmos valores. Finalmente visualizamos o dicionário de dados.

```
# Criando um dataframe com duas colunas (symbol e description)
df_dict <- data.frame(symbol = colnames(df),
                      description = t(df[1,])[,1])

# Removendo a primeira linha dos dados, pois não contém informação
df_dict <- df_dict[-1,]
```

```r
# Removendo a coluna Symbol, pois já temos os símbolos das variáveis como index
df_dict$symbol <- NULL

# Visualizando o dicionário de dados
df_dict
```

```
##                                                description
## GE05a Quantidade de municípios atendidos com abastecimen
## GE05b Quantidade de municípios atendidos com esgotamento
## FN002             Receitas operacionais diretas com agua
## FN003           Receitas operacionais diretas com esgoto
## AG001 População total atendida com abastecimento de água
## AG007                      Volume de agua tratada em ETAs
## AG010                           Volume de agua consumido
## ES001 População total atendida com esgotamento sanitário
## ES005                          Volume de esgoto coletado
## ES006                           Volume de esgoto tratado
```

```r
# Agora podemos remover a primeira linha do conjunto de dados df, pois já temos o dicionário de dados p
df <- df[-1,]
```

Vamos visualizar os tipos de dados das variáveis selecionadas.

```r
# Visualizando os tipos de dados
glimpse(df)
```

```
## Rows: 38
## Columns: 11
## $ State <chr> "N - Norte", "Acre (AC)", "Amapá (AP)", "Amazonas (AM)", "Pará (~
## $ GE05a <chr> "", "22", "16", "25", "86", "48", "15", "143", "355", "", "89", ~
## $ GE05b <chr> "", "1", "6", "3", "17", "10", "3", "18", "58", "", "22", "168",~
## $ FN002 <chr> "", "64.063.550,82", "56.785.323,26", "472.271.251,57", "377.845~
## $ FN003 <chr> "", "8.965.444,00", "12.178.855,22", "55.426.612,20", "36.918.40~
## $ AG001 <chr> "", "423.469", "290.944", "2.584.829", "2.967.413", "818.865", "~
## $ AG007 <chr> "", "54.977,40", "60.810,07", "196.716,84", "138.507,44", "74.02~
## $ AG010 <chr> "", "22.986,44", "17.394,68", "86.277,83", "159.523,15", "43.427~
## $ ES001 <chr> "", "88.199", "59.574", "441.358", "405.270", "103.461", "365.28~
## $ ES005 <chr> "", "4.472,62", "3.024,74", "19.754,57", "25.389,39", "3.989,32"~
## $ ES006 <chr> "", "4.472,62", "2.871,23", "19.544,57", "11.041,80", "3.073,45"~
```

Como pode ser observado, os dados está como character, porém são numéricos. Primeiro, vamos utilizar a função gsub() para trocar o é vírgula por ponto como separador decimal e remover os pontos como separador de milhares. Na sequência, faremos a transformação para numérico.

```r
# Substituindo vírgula por ponto com decimal e removendo pontos como separador de milhares para todas a
df$GE05a <- sapply(df$GE05a, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$GE05b <- sapply(df$GE05b, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$FN002 <- sapply(df$FN002, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$FN003 <- sapply(df$FN003, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$AG001 <- sapply(df$AG001, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$AG007 <- sapply(df$AG007, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$AG010 <- sapply(df$AG010, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$ES001 <- sapply(df$ES001, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$ES005 <- sapply(df$ES005, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
df$ES006 <- sapply(df$ES006, function(x) { ifelse(is.na(x), NA ,gsub(",", ".", gsub("\\.", "", x))) } )
```

```r
# Tranformando as variáveis numéricas para dados do tipo numérico
df$GE05a <- sapply(df$GE05a, function(x) { as.numeric(x) } )
df$GE05b <- sapply(df$GE05b, function(x) { as.numeric(x) } )
df$FN002 <- sapply(df$FN002, function(x) { as.numeric(x) } )
df$FN003 <- sapply(df$FN003, function(x) { as.numeric(x) } )
df$AG001 <- sapply(df$AG001, function(x) { as.numeric(x) } )
df$AG007 <- sapply(df$AG007, function(x) { as.numeric(x) } )
df$AG010 <- sapply(df$AG010, function(x) { as.numeric(x) } )
df$ES001 <- sapply(df$ES001, function(x) { as.numeric(x) } )
df$ES005 <- sapply(df$ES005, function(x) { as.numeric(x) } )
df$ES006 <- sapply(df$ES006, function(x) { as.numeric(x) } )
```

Agora vamos criar uma nova tabela de dados com os dados regionais para que possamos remover essas linhas da tabela de dados df. Para isso, primeiro, criamos uma nova variável de regiões, chamado Region, e inserimos no dataframe df. Depois, criamos a tabela de dados df_per_Region com os dados de cada região. Finalmente, renomeamos a das regiões como Region bem como criamos a variável Region nessa tabela de dados. Finalmente, visualizamos as transformações.

```r
# Criando a variável Region na tabela de dados df
df$Region <- c(rep('Norte',9), rep('Nordeste',11), rep('Sudeste',6), rep('Sul',5), rep('Centro_Oeste',6)

# Criando uma nova tabela de dados com os dados das regiões
df_per_Region <- df[c(9,20,26,31,37),]

# Renomeando a primeira coluna (regiões)
colnames(df_per_Region)[1] <- 'Region'

# Criando uma nova variável Region na tabela de dados df_per_Region
df_per_Region$Region <- c('Norte', 'Nordeste', 'Sudeste', 'Sul', 'Centro_Oeste')

# Visualizando a tabela de dados df_per_Region
df_per_Region
```

```
##          Region GE05a GE05b       FN002       FN003    AG001     AG007
## 1:        Norte   355    58  1590834832   266626564  8987680  617853.8
## 2:     Nordeste  1636   531  8418730605  2661381719 40523130 2526440.5
## 3:      Sudeste  1605  1408 21202953280 15516173738 80019344 7338231.1
## 4:          Sul  1159   415  8894765290  2835125424 26931471 2036564.9
## 5: Centro_Oeste   422   180  4092583204  2195027451 14342891  885503.8
##        AG010    ES001      ES005      ES006       Region
## 1:  423084.8  1927986   91613.93   75725.76        Norte
## 2: 1795163.1 15498076  702778.46  583822.67     Nordeste
## 3: 5286628.2 69879184 3921041.03 2811474.57      Sudeste
## 4: 1472843.1 13768012  675170.47  638806.03          Sul
## 5:  783633.3  9227084  436081.58  406284.52 Centro_Oeste
```

```r
# Visualizando a tabela de dados df
df
```

```
##                       State GE05a GE05b       FN002       FN003     AG001
##  1:               N - Norte    NA    NA          NA          NA        NA
##  2:               Acre (AC)    22     1    64063551     8965444    423469
##  3:              Amapá (AP)    16     6    56785323    12178855    290944
##  4:           Amazonas (AM)    25     3   472271252    55426612   2584829
##  5:               Pará (PA)    86    17   377845478    36918410   2967413
##  6:           Rondônia (RO)    48    10   170501397     8482893    818865
```

```
##  7:              Roraima (RR)    15     3    66628268    30019054    494697
##  8:            Tocantins (TO)   143    18   382739565   114635296   1407463
##  9:         Total por grupo:    355    58  1590834832   266626564   8987680
## 10:            NE - Nordeste     NA    NA          NA          NA        NA
## 11:             Alagoas (AL)    89    22   481866324   105737460   2357068
## 12:               Bahia (BA)   396   168  2642815114   948707100  11725988
## 13:               Ceará (CE)   177    98  1171507477   473834115   5238334
## 14:            Maranhão (MA)   167    18   541813830   190318367   3427119
## 15:             Paraíba (PB)   210    54   642062222   246324125   2910101
## 16:          Pernambuco (PE)   178    69  1341542878   390573480   7642886
## 17:               Piauí (PI)   181    27   455499311    63483723   2357765
## 18: Rio Grande do Norte (RN)   163    63   620417502   132243256   2887371
## 19:             Sergipe (SE)    75    12   521205947   110160093   1976498
## 20:         Total por grupo:  1636   531  8418730605  2661381719  40523130
## 21:             SE - Sudeste     NA    NA          NA          NA        NA
## 22:      Espírito Santo (ES)    77    67   839716890   368186803   3243410
## 23:        Minas Gerais (MG)   812   651  4283310535  2486150246  17165390
## 24:      Rio de Janeiro (RJ)    88    62  5252276374  2993824343  15603715
## 25:           São Paulo (SP)   628   628 10827649481  9668012347  44006829
## 26:         Total por grupo:  1605  1408 21202953280 15516173738  80019344
## 27:                 S - Sul     NA    NA          NA          NA        NA
## 28:              Paraná (PR)   396   225  3217344170  1862826385  10749898
## 29:   Rio Grande do Sul (RS)   473   116  3681143736   515199578   9799440
## 30:      Santa Catarina (SC)   291    74  1996277384   457099462   6382133
## 31:         Total por grupo:  1159   415  8894765290  2835125424  26931471
## 32:        CO - Centro-Oeste     NA    NA          NA          NA        NA
## 33:    Distrito Federal (DF)     1     1   938573359   800649729   2985115
## 34:              Goiás (GO)   242    87  1724865200   947746314   6195837
## 35:  Mato Grosso do Sul (MS)    77    59   847221870   289899821   2374606
## 36:         Mato Grosso (MT)   102    33   581922775   156731587   2787333
## 37:         Total por grupo:   422   180  4092583204  2195027451  14342891
## 38:     TOTALIZAÇÃO NACIONAL  5177  2592 44199867212 23474334897 170804516
##                        State GE05a GE05b       FN002       FN003     AG001
##           AG007      AG010     ES001      ES005      ES006       Region
##  1:          NA         NA        NA         NA         NA        Norte
##  2:    54977.40   22986.44     88199    4472.62    4472.62        Norte
##  3:    60810.07   17394.68     59574    3024.74    2871.23        Norte
##  4:   196716.84   86277.83    441358   19754.57   19544.57        Norte
##  5:   138507.44  159523.15    405270   25389.39   11041.80        Norte
##  6:    74025.25   43427.38    103461    3989.32    3073.45        Norte
##  7:    35921.51   22912.62    365286   17147.36   17117.36        Norte
##  8:    56895.33   70562.71    464838   17835.93   17604.73        Norte
##  9:   617853.84  423084.81   1927986   91613.93   75725.76        Norte
## 10:          NA         NA        NA         NA         NA     Nordeste
## 11:   124997.17  134980.95    679181   21921.20   19945.44     Nordeste
## 12:   729151.15  494395.79   5790656  269352.47  227716.03     Nordeste
## 13:   385118.73  263770.06   2290370   99789.28   90935.75     Nordeste
## 14:   174883.28  182036.19    813839   49070.93   21450.99     Nordeste
## 15:   204207.72  123869.09   1364157   61172.29   54690.97     Nordeste
## 16:   534867.98  280533.39   2669648  118196.52   89420.35     Nordeste
## 17:   135851.90  107459.10    512687   18060.38   16016.28     Nordeste
## 18:    80325.78  112214.79    895519   38602.64   37433.71     Nordeste
## 19:   157036.74   95903.71    482019   26612.75   26213.15     Nordeste
## 20:  2526440.45 1795163.07  15498076  702778.46  583822.67     Nordeste
```

```
## 21:          NA         NA        NA         NA         NA      Sudeste
## 22:   326814.73  203625.62   2229294  121479.10   87216.01      Sudeste
## 23:  1402843.22  995033.43  15280408  800064.27  435892.21      Sudeste
## 24:  1909548.97 1253953.09  11072314  801816.63  498972.97      Sudeste
## 25:  3699024.17 2834016.02  41297168 2197681.03 1789393.38      Sudeste
## 26:  7338231.09 5286628.16  69879184 3921041.03 2811474.57      Sudeste
## 27:          NA         NA        NA         NA         NA          Sul
## 28:   639479.31  550832.22   8324363  405988.29  405477.52          Sul
## 29:   863143.49  529154.50   3648316  168772.41  138696.54          Sul
## 30:   533942.11  392856.33   1795333  100409.77   94631.97          Sul
## 31:  2036564.91 1472843.05  13768012  675170.47  638806.03          Sul
## 32:          NA         NA        NA         NA         NA Centro_Oeste
## 33:   227996.00  158200.00   2698062  129923.00  129923.00 Centro_Oeste
## 34:   355307.76  316820.74   3975006  180014.40  159802.28 Centro_Oeste
## 35:    93355.86  136660.51   1437203   60951.14   60854.86 Centro_Oeste
## 36:   208844.15  171952.07   1116813   65193.04   55704.38 Centro_Oeste
## 37:   885503.77  783633.32   9227084  436081.58  406284.52 Centro_Oeste
## 38: 13404594.06 9761352.41 110300342 5826685.47 4516113.55       Brazil
##           AG007      AG010     ES001      ES005      ES006       Region
```

Agora podemos remover as linhas que não contém informação por estado na tabela de dados df. Depois disso, criamos duas novas variáveis, chamadas Num_Mun e Population, contendo dados extraídos do IBGE e adicionados manualmente devido à problemas de conexão com o banco de dados que se encontrava fora do ar. Esses dados se referem ao número de municípios e a população para cada estado.

```
# Removendo as linhas que não contém informação estadual
df <- df[-c(1,9,10,20,21,26,27,31,32,37,38),]

# Criando duas novas colunas contendo o número de municípios e a população
df <- df %>%
  mutate(
    Num_Mun = c(22,16,62,144,52,15,139,102,417,184,217,223,185,224,
                167,75,78,853,92,645,399,497,295,1,246,79,141),
    Population = c(894470,861773,4207714,8690745,1796460,631181,1590248,
                3351543,14930634,9187103,7114598,4039277,9616621,3281480,
                3534165,2318822,4064052,21292666,17366189,46289333,11516840,
                11422973,7252502,3055149,7113540,2809394,3526220)
  )
```

Devido a uma falha no conjunto de dados, um dos dados (linha 7) foi ajustado, pois o número de municípios com atendimento de abastecimento era maior que o número de municípios presentes no estado. Desta forma, neste caso, simplesmente utilizamos o dado com o menor valor para o número de municípios (dado ofcial do IBGE). Finalmente, visualizamos as transformações nos dados.

```
# Ajustando o dado que estava estranho (row 7)
for (i in 1:length(df$GE05a)) {
  if (df$GE05a[i] > df$Num_Mun[i]) {
    df$GE05a[i] <- df$Num_Mun[i]
  }
}

# Visualizando as transformações na tabela df
df
```

```
##                       State GE05a GE05b       FN002      FN003    AG001
##  1:               Acre (AC)    22     1    64063551    8965444   423469
```

```
##  2:                Amapá (AP)    16     6    56785323   12178855   290944
##  3:              Amazonas (AM)    25     3   472271252   55426612  2584829
##  4:                  Pará (PA)    86    17   377845478   36918410  2967413
##  5:              Rondônia (RO)    48    10   170501397    8482893   818865
##  6:               Roraima (RR)    15     3    66628268   30019054   494697
##  7:             Tocantins (TO)   139    18   382739565  114635296  1407463
##  8:               Alagoas (AL)    89    22   481866324  105737460  2357068
##  9:                 Bahia (BA)   396   168  2642815114  948707100 11725988
## 10:                 Ceará (CE)   177    98  1171507477  473834115  5238334
## 11:              Maranhão (MA)   167    18   541813830  190318367  3427119
## 12:               Paraíba (PB)   210    54   642062222  246324125  2910101
## 13:            Pernambuco (PE)   178    69  1341542878  390573480  7642886
## 14:                Piauí (PI)   181    27   455499311   63483723  2357765
## 15: Rio Grande do Norte (RN)   163    63   620417502  132243256  2887371
## 16:               Sergipe (SE)    75    12   521205947  110160093  1976498
## 17:      Espírito Santo (ES)    77    67   839716890  368186803  3243410
## 18:        Minas Gerais (MG)   812   651  4283310535 2486150246 17165390
## 19:      Rio de Janeiro (RJ)    88    62  5252276374 2993824343 15603715
## 20:           São Paulo (SP)   628   628 10827649481 9668012347 44006829
## 21:              Paraná (PR)   396   225  3217344170 1862826385 10749898
## 22:   Rio Grande do Sul (RS)   473   116  3681143736  515199578  9799440
## 23:      Santa Catarina (SC)   291    74  1996277384  457099462  6382133
## 24:    Distrito Federal (DF)     1     1   938573359  800649729  2985115
## 25:                Goiás (GO)   242    87  1724865200  947746314  6195837
## 26:  Mato Grosso do Sul (MS)    77    59   847221870  289899821  2374606
## 27:         Mato Grosso (MT)   102    33   581922775  156731587  2787333
##                       State GE05a GE05b       FN002      FN003    AG001
##          AG007      AG010    ES001      ES005      ES006        Region Num_Mun
##  1:   54977.40   22986.44    88199    4472.62    4472.62        Norte      22
##  2:   60810.07   17394.68    59574    3024.74    2871.23        Norte      16
##  3:  196716.84   86277.83   441358   19754.57   19544.57        Norte      62
##  4:  138507.44  159523.15   405270   25389.39   11041.80        Norte     144
##  5:   74025.25   43427.38   103461    3989.32    3073.45        Norte      52
##  6:   35921.51   22912.62   365286   17147.36   17117.36        Norte      15
##  7:   56895.33   70562.71   464838   17835.93   17604.73        Norte     139
##  8:  124997.17  134980.95   679181   21921.20   19945.44     Nordeste     102
##  9:  729151.15  494395.79  5790656  269352.47  227716.03     Nordeste     417
## 10:  385118.73  263770.06  2290370   99789.28   90935.75     Nordeste     184
## 11:  174883.28  182036.19   813839   49070.93   21450.99     Nordeste     217
## 12:  204207.72  123869.09  1364157   61172.29   54690.97     Nordeste     223
## 13:  534867.98  280533.39  2669648  118196.52   89420.35     Nordeste     185
## 14:  135851.90  107459.10   512687   18060.38   16016.28     Nordeste     224
## 15:   80325.78  112214.79   895519   38602.64   37433.71     Nordeste     167
## 16:  157036.74   95903.71   482019   26612.75   26213.15     Nordeste      75
## 17:  326814.73  203625.62  2229294  121479.10   87216.01      Sudeste      78
## 18: 1402843.22  995033.43 15280408  800064.27  435892.21      Sudeste     853
## 19: 1909548.97 1253953.09 11072314  801816.63  498972.97      Sudeste      92
## 20: 3699024.17 2834016.02 41297168 2197681.03 1789393.38      Sudeste     645
## 21:  639479.31  550832.22  8324363  405988.29  405477.52          Sul     399
## 22:  863143.49  529154.50  3648316  168772.41  138696.54          Sul     497
## 23:  533942.11  392856.33  1795333  100409.77   94631.97          Sul     295
## 24:  227996.00  158200.00  2698062  129923.00  129923.00 Centro_Oeste       1
## 25:  355307.76  316820.74  3975006  180014.40  159802.28 Centro_Oeste     246
## 26:   93355.86  136660.51  1437203   60951.14   60854.86 Centro_Oeste      79
```

```
## 27:  208844.15  171952.07  1116813   65193.04   55704.38 Centro_Oeste     141
##          AG007      AG010    ES001      ES005      ES006       Region Num_Mun
##     Population
##  1:     894470
##  2:     861773
##  3:    4207714
##  4:    8690745
##  5:    1796460
##  6:     631181
##  7:    1590248
##  8:    3351543
##  9:   14930634
## 10:    9187103
## 11:    7114598
## 12:    4039277
## 13:    9616621
## 14:    3281480
## 15:    3534165
## 16:    2318822
## 17:    4064052
## 18:   21292666
## 19:   17366189
## 20:   46289333
## 21:   11516840
## 22:   11422973
## 23:    7252502
## 24:    3055149
## 25:    7113540
## 26:    2809394
## 27:    3526220
##     Population
```

Agora que os dados estão bem organizados, podemos começar nossa análise gráfica para ver se conseguímos alguns insights e depois vamos produzir o gráfico de pontos descrito no objetivo deste projeto. Primeiramente, vamos calcular algumas estatísticas.

```
# Apresentando estatística descritiva dos dados
summary(df)
```

```
##     State              GE05a           GE05b           FN002
##  Length:27          Min.   :  1.0   Min.   :  1.0   Min.   :5.679e+07
##  Class :character   1st Qu.: 76.0   1st Qu.: 14.5   1st Qu.:4.639e+08
##  Mode  :character   Median :139.0   Median : 54.0   Median :6.421e+08
##                     Mean   :191.6   Mean   : 96.0   Mean   :1.637e+09
##                     3rd Qu.:226.0   3rd Qu.: 80.5   3rd Qu.:1.861e+09
##                     Max.   :812.0   Max.   :651.0   Max.   :1.083e+10
##      FN003               AG001              AG007             AG010
##  Min.   :8.483e+06   Min.   :  290944   Min.   :  35922   Min.   :  17395
##  1st Qu.:8.461e+07   1st Qu.: 2357416   1st Qu.: 109177   1st Qu.: 101681
##  Median :2.463e+08   Median : 2967413   Median : 204208   Median : 159523
##  Mean   :8.694e+08   Mean   : 6326093   Mean   : 496466   Mean   : 361532
##  3rd Qu.:6.579e+08   3rd Qu.: 7012510   3rd Qu.: 534405   3rd Qu.: 354838
##  Max.   :9.668e+09   Max.   :44006829   Max.   :3699024   Max.   :2834016
##      ES001              ES005             ES006            Region
##  Min.   :   59574   Min.   :   3025   Min.   :   2871   Length:27
```

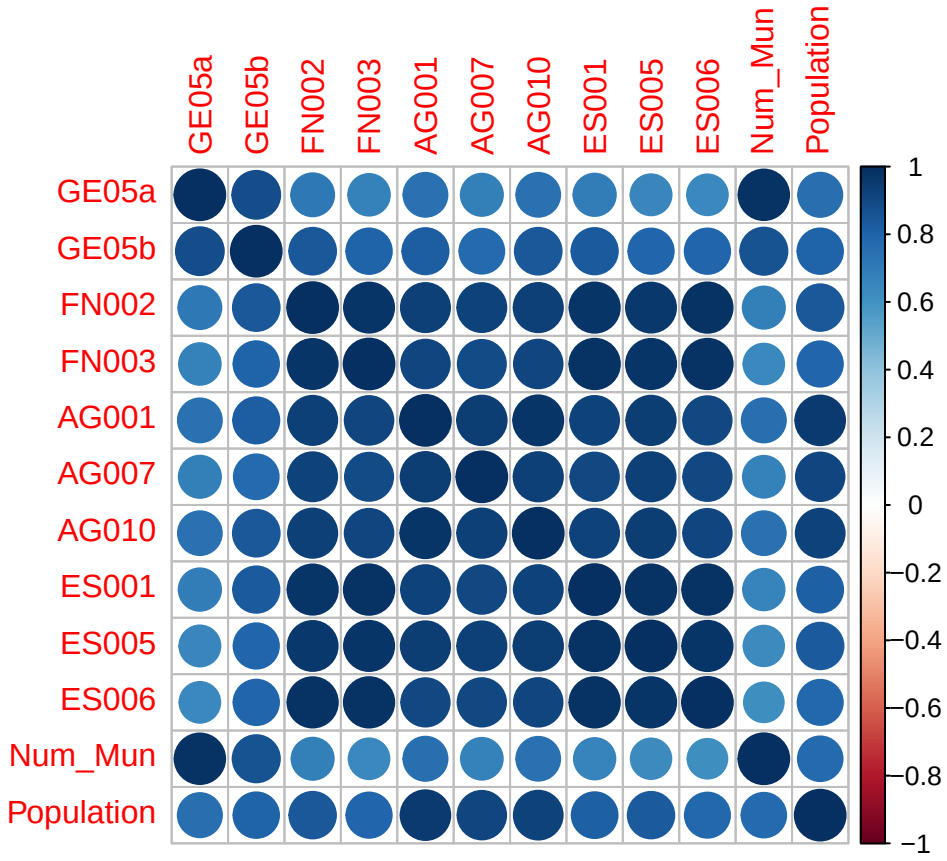
```
##  1st Qu.:  473428   1st Qu.:  20838   1st Qu.:  18575   Class :character
##  Median : 1364157   Median :  61172   Median :  55704   Mode  :character
##  Mean   : 4085198   Mean   : 215803   Mean   : 167264
##  3rd Qu.: 3173189   3rd Qu.: 149348   3rd Qu.: 134310
##  Max.   :41297168   Max.   :2197681   Max.   :1789393
##     Num_Mun        Population
##  Min.   :  1.0   Min.   :  631181
##  1st Qu.: 76.5   1st Qu.: 2932272
##  Median :144.0   Median : 4064052
##  Mean   :206.3   Mean   : 7842803
##  3rd Qu.:235.0   3rd Qu.: 9401862
##  Max.   :853.0   Max.   :46289333
```

Vamos verificar se há correlações entre as variáveis numéricas.

```
# Calculando a correlação para as variáveis numéricas
df_cor <- cor(df[,-c('State','Region')], method = "spearman")

# Resultados da análise de correlação
df_cor
```

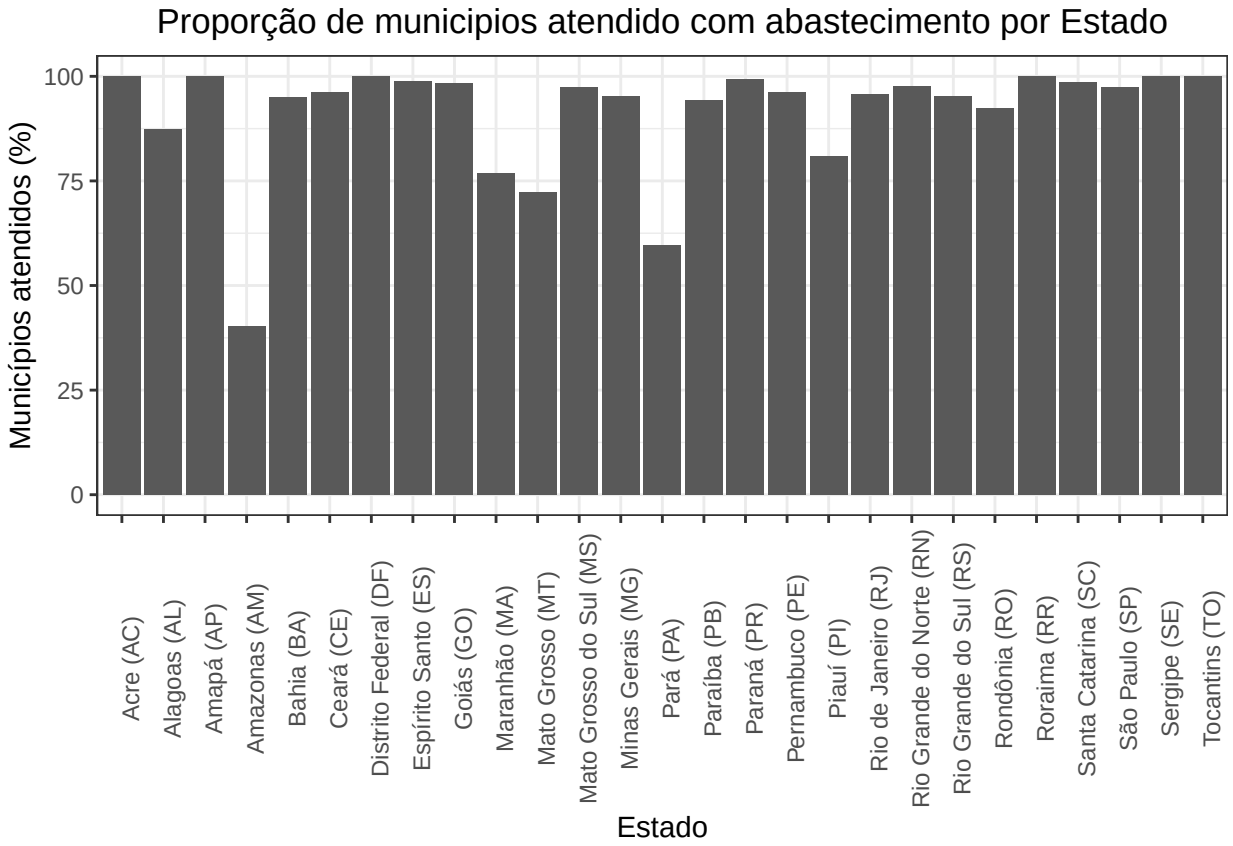
```
##                GE05a     GE05b     FN002     FN003     AG001     AG007
## GE05a      1.0000000 0.8883458 0.7157252 0.6711451 0.7487023 0.6885497
## GE05b      0.8883458 1.0000000 0.8468469 0.8013438 0.8251642 0.7790503
## FN002      0.7157252 0.8468469 1.0000000 0.9700855 0.9340659 0.9279609
## FN003      0.6711451 0.8013438 0.9700855 1.0000000 0.9157509 0.8968254
## AG001      0.7487023 0.8251642 0.9340659 0.9157509 1.0000000 0.9499389
## AG007      0.6885497 0.7790503 0.9279609 0.8968254 0.9499389 1.0000000
## AG010      0.7441222 0.8474577 0.9371184 0.9151404 0.9700855 0.9395604
## ES001      0.6946565 0.8318828 0.9774115 0.9859585 0.9230769 0.9090354
## ES005      0.6512977 0.7943198 0.9664225 0.9737485 0.9420024 0.9340659
## ES006      0.6442748 0.7989007 0.9810745 0.9841270 0.9053724 0.9059829
## Num_Mun    0.9899237 0.8621164 0.6880342 0.6422466 0.7527473 0.6776557
## Population 0.7554199 0.8025654 0.8406593 0.7942613 0.9566545 0.9120879
##                AG010     ES001     ES005     ES006   Num_Mun Population
## GE05a      0.7441222 0.6946565 0.6512977 0.6442748 0.9899237  0.7554199
## GE05b      0.8474577 0.8318828 0.7943198 0.7989007 0.8621164  0.8025654
## FN002      0.9371184 0.9774115 0.9664225 0.9810745 0.6880342  0.8406593
## FN003      0.9151404 0.9859585 0.9737485 0.9841270 0.6422466  0.7942613
## AG001      0.9700855 0.9230769 0.9420024 0.9053724 0.7527473  0.9566545
## AG007      0.9395604 0.9090354 0.9340659 0.9059829 0.6776557  0.9120879
## AG010      1.0000000 0.9261294 0.9462759 0.9120879 0.7442002  0.9242979
## ES001      0.9261294 1.0000000 0.9810745 0.9829060 0.6654457  0.8125763
## ES005      0.9462759 0.9810745 1.0000000 0.9798535 0.6361416  0.8388278
## ES006      0.9120879 0.9829060 0.9798535 1.0000000 0.6129426  0.7887668
## Num_Mun    0.7442002 0.6654457 0.6361416 0.6129426 1.0000000  0.7728938
## Population 0.9242979 0.8125763 0.8388278 0.7887668 0.7728938  1.0000000
```

```
# Resultados da análise de correlação de forma gráfica
corrplot(df_cor)
```



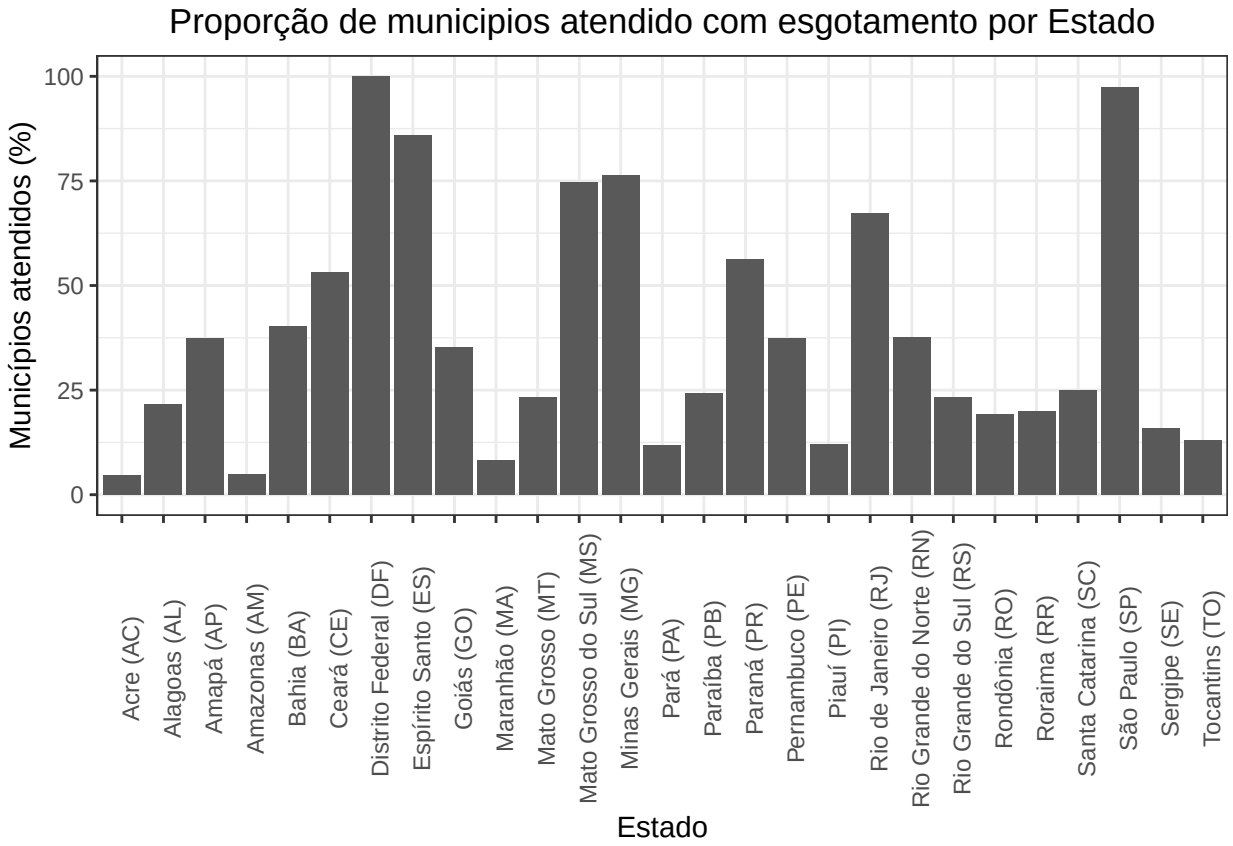
Como visto anteriormente, existe uma forte correlação entre algumas variáveis numéricas e moderada para outras. Vale a pena pegar o dicionário de dados e verificar essas correlações com calma. Esse primeiro gráfico mostra a porcentagem de municípios atendidos com abastecimento de água por Estado. Esse gráfico mostra que o Amazonas ainda deixa muito a desejar com relação ao fornecimento de água para os municípios, apresentando uma proporção bem baixa dos municípios com atendimento de abastecimento.Alguns outros estados como Pará, Mato Gross, Maranhão e Piauí também merecem destaque e precisam de aumentar o número de municípios com atendimento de abastecimento se quiserem se igualar à média, que se encontra acima dos 90%.

```
ggplot(df) +
  geom_bar(aes(x=State, y=(GE05a / Num_Mun)*100),
           stat = 'identity') +
  theme_bw() +
  ylab('Municípios atendidos (%)') +
  xlab('Estado') +
  ggtitle('Proporção de municipios atendido com abastecimento por Estado') +
  theme(
    axis.text.x = element_text(angle = 90),
    plot.title = element_text(hjust = 0.5)
  )
```



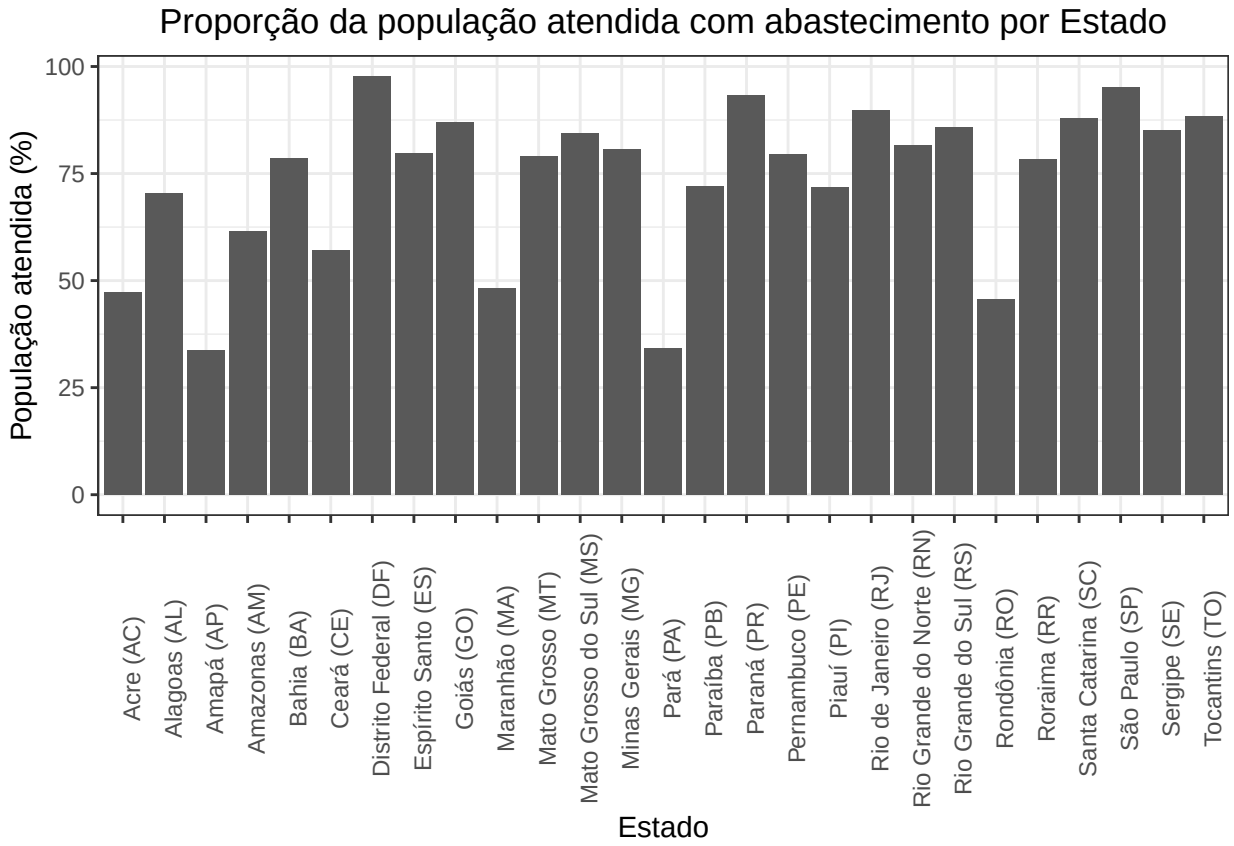
O segundo gráfico ilustra a porcentagem de municípios que são atendidos com esgotamento por estado. Ao contrário do que observamos para o abastecimento, onde somente alguns estados estão bem abaixo da média, nesse caso, apenas alguns estados apresentam uma alta proporção de municípios atendidos com esgotamento, tais como DF e SP. Outros estados que também apresentam proporções bem acima da média são ES, MG, MS, RJ, PR e CE. Todos os outros estados não possuem nem 50% dos seus municípios com atendimento de esgotamento. Triste realidade! :/

```
ggplot(df) +
  geom_bar(aes(x=State, y=(GE05b / Num_Mun)*100),
           stat = 'identity') +
  theme_bw() +
  ylab('Municípios atendidos (%)') +
  xlab('Estado') +
  ggtitle('Proporção de municipios atendido com esgotamento por Estado') +
  theme(
    axis.text.x = element_text(angle = 90),
    plot.title = element_text(hjust = 0.5)
  )
```



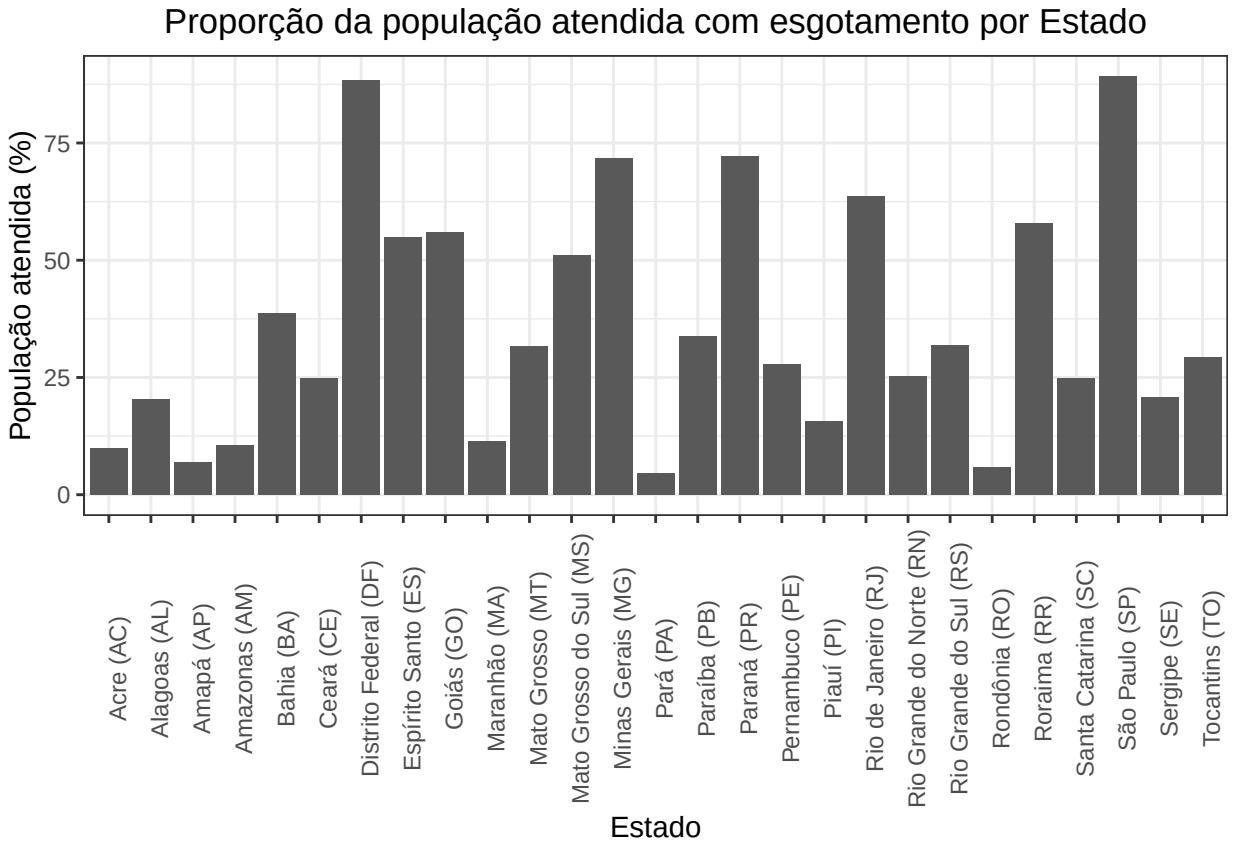
O próximo gráfico ilustra a proporção da população atendida com abastecimento de água por Estado.

```
ggplot(df) +
  geom_bar(aes(x=State, y=AG001 / Population * 100),
           stat = 'identity') +
  theme_bw() +
  ylab('População atendida (%)') +
  xlab('Estado') +
  ggtitle('Proporção da população atendida com abastecimento por Estado') +
  theme(
    axis.text.x = element_text(angle = 90),
    plot.title = element_text(hjust = 0.5)
  )
```



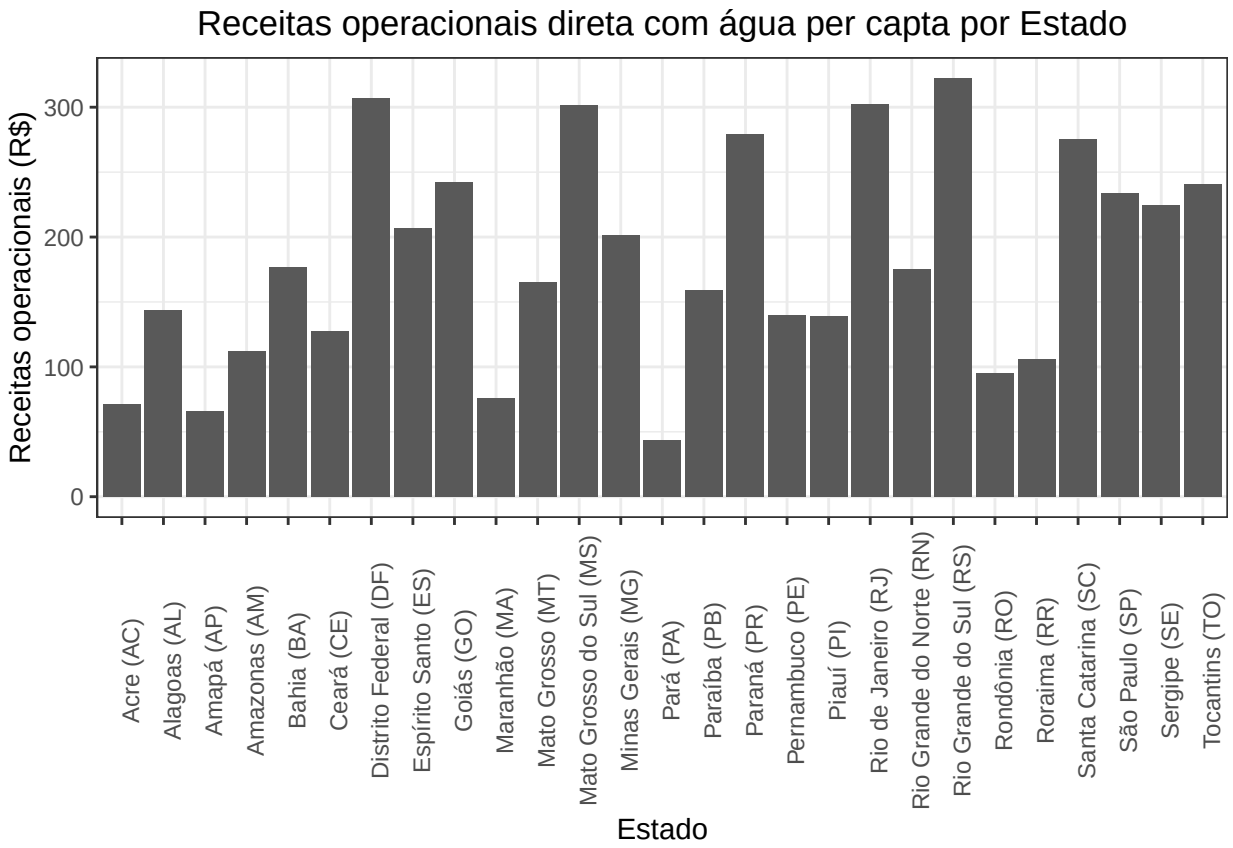
O próximo gráfico apresenta a proporção da população atendida com esgotamento por Estado.

```
ggplot(df) +
  geom_bar(aes(x=State, y=ES001 / Population * 100),
           stat = 'identity') +
  theme_bw() +
  ylab('População atendida (%)') +
  xlab('Estado') +
  ggtitle('Proporção da população atendida com esgotamento por Estado') +
  theme(
    axis.text.x = element_text(angle = 90),
    plot.title = element_text(hjust = 0.5)
  )
```



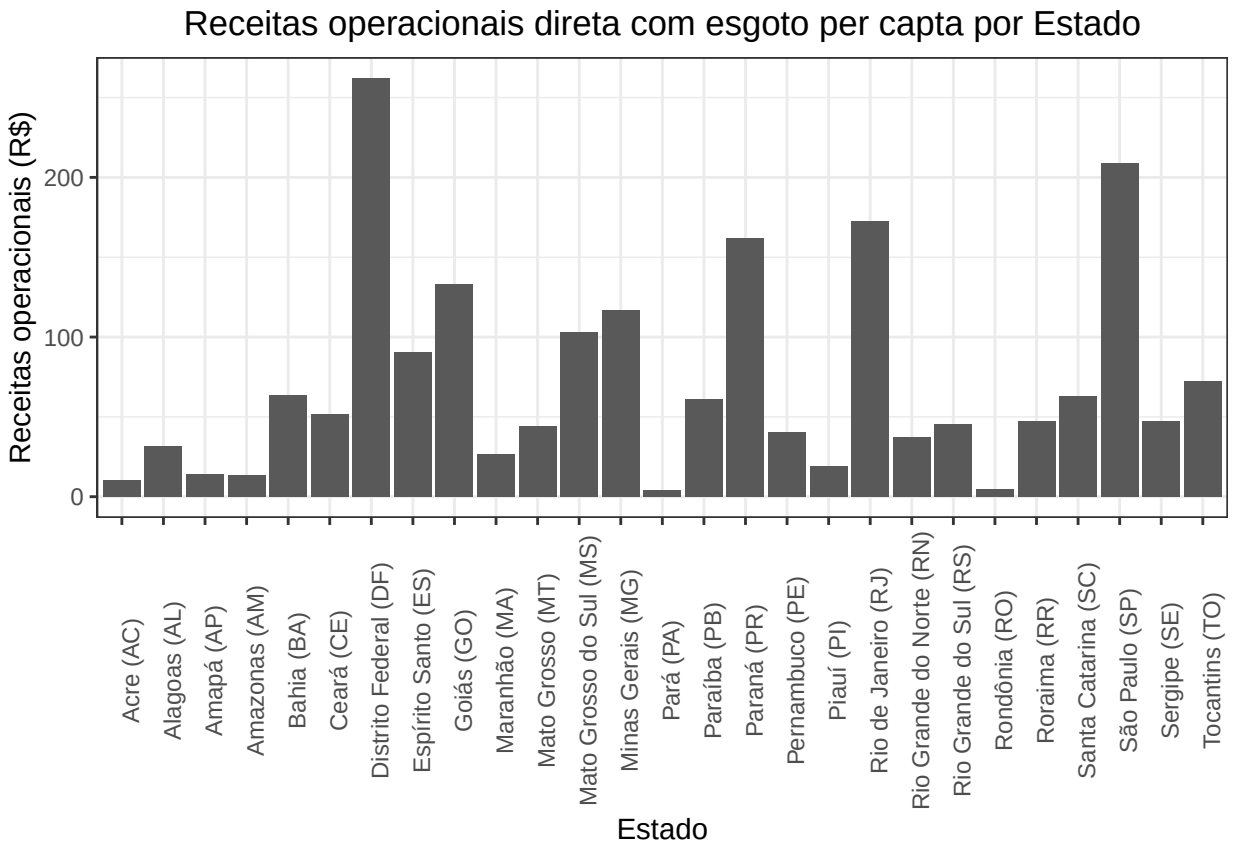
No próximo gráfico foi plotado as receitas operacionais diretas com abastecimendo de água per capta por Estado. A maior receita operacional per capta encontra-se no RS ($>$ 320 reais / por pessoa) enquanto que a menor receita operacional foi observada no estado do Pará ($<$ 45 reais). Esses valores estão bem distante da média de aproximadamente 183 reais, o que causa essa disparidade nos gastos, e consequentemente, mostra o porque temos grandes diferenças na proporção de abastecimento nos estados Brasileiros.

```
ggplot(df) +
  geom_bar(aes(x=State, y=FN002 / Population),
           stat = 'identity') +
  theme_bw() +
  ylab('Receitas operacionais (R$)') +
  xlab('Estado') +
  ggtitle('Receitas operacionais direta com água per capta por Estado') +
  theme(
    axis.text.x = element_text(angle = 90),
    plot.title = element_text(hjust = 0.5)
  )
```



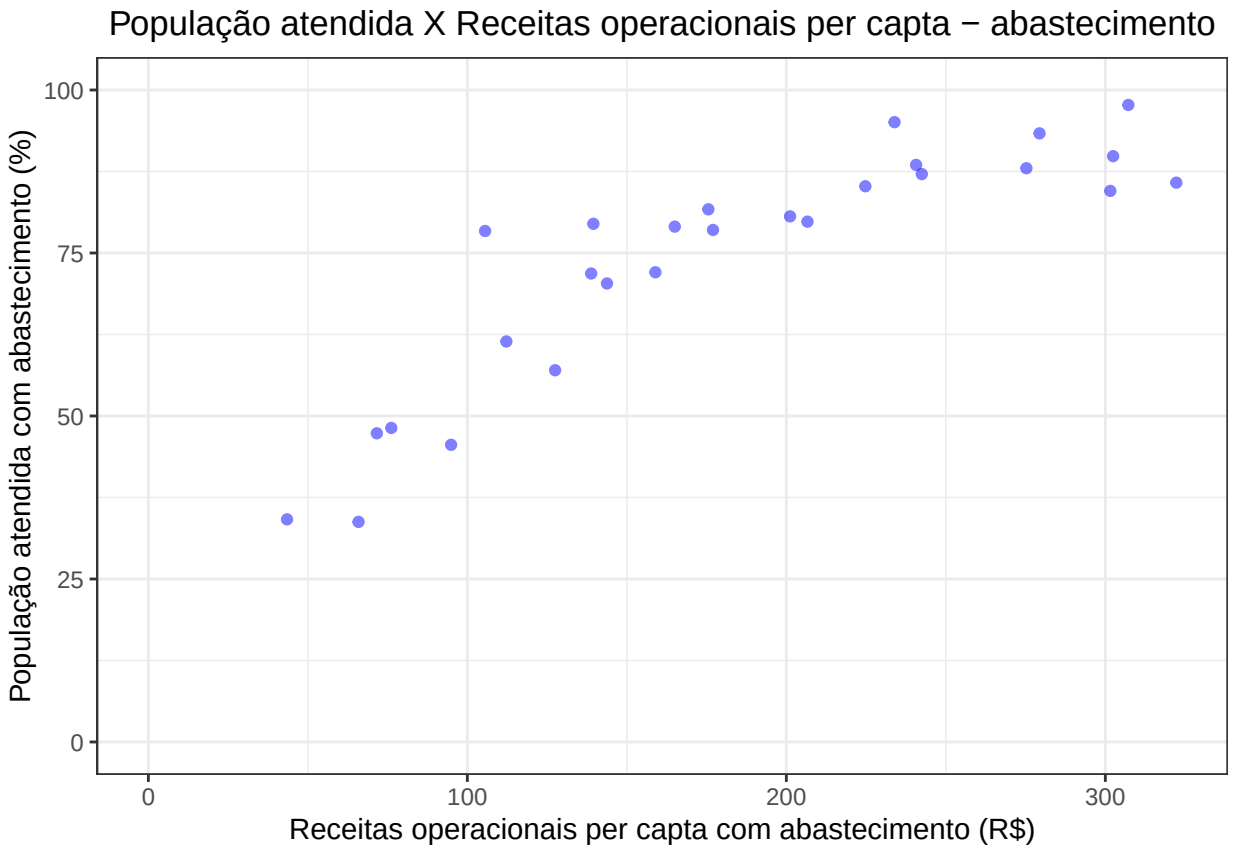
Para o atendimento com esgotamento, pode ser observado que os estados com maior receita operacional percapta são DF, SP, RJ and PR (> 150 reais), following this order. Por outro lado, RO e PA possuem gastos operacionais abaixo de 5 reais para o esgotamento. Como pode ser observado na figura abaixo, a discrepância de receitas operacionais para o esgotamento entre os estados é ainda maior que a observada para o abastecimento.

```
ggplot(df) +
  geom_bar(aes(x=State, y=FN003 / Population),
           stat = 'identity') +
  theme_bw() +
  ylab('Receitas operacionais (R$)') +
  xlab('Estado') +
  ggtitle('Receitas operacionais direta com esgoto per capta por Estado') +
  theme(
    axis.text.x = element_text(angle = 90),
    plot.title = element_text(hjust = 0.5)
  )
```

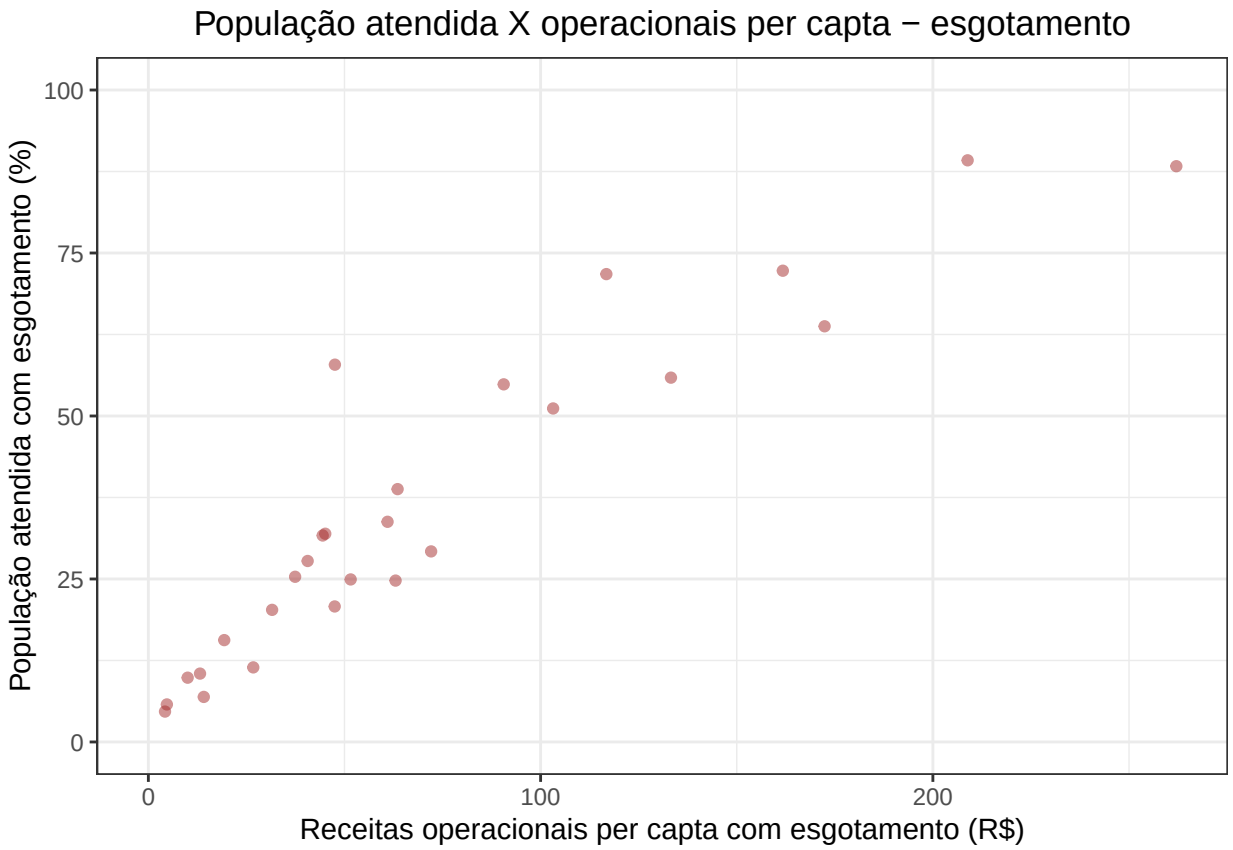


Agora vamos plotar dois gráficos de pontos mostrando a relação entre a proporção da população atendida e as receitas operacionais per capta, tanto para o abastecimento de água (blue), quanto para esgotamento (brown). Podemos observar que há uma relação entre as receitas e a proporção de atendimento, o que era esperado. Mas é interessante observar que alguns estados conseguem atender uma proporção maior da população utilizando receitas operacionais menores.

```
ggplot(df) +
  geom_point(aes(x=FN002 / Population, y=AG001 / Population * 100),
             color='blue', alpha=0.5) +
  theme_bw() +
  ylab('População atendida com abastecimento (%)') +
  xlab('Receitas operacionais per capta com abastecimento (R$)') +
  ggtitle('População atendida X Receitas operacionais per capta - abastecimento') +
  ylim(0,100) +
  xlim(0,NA) +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```



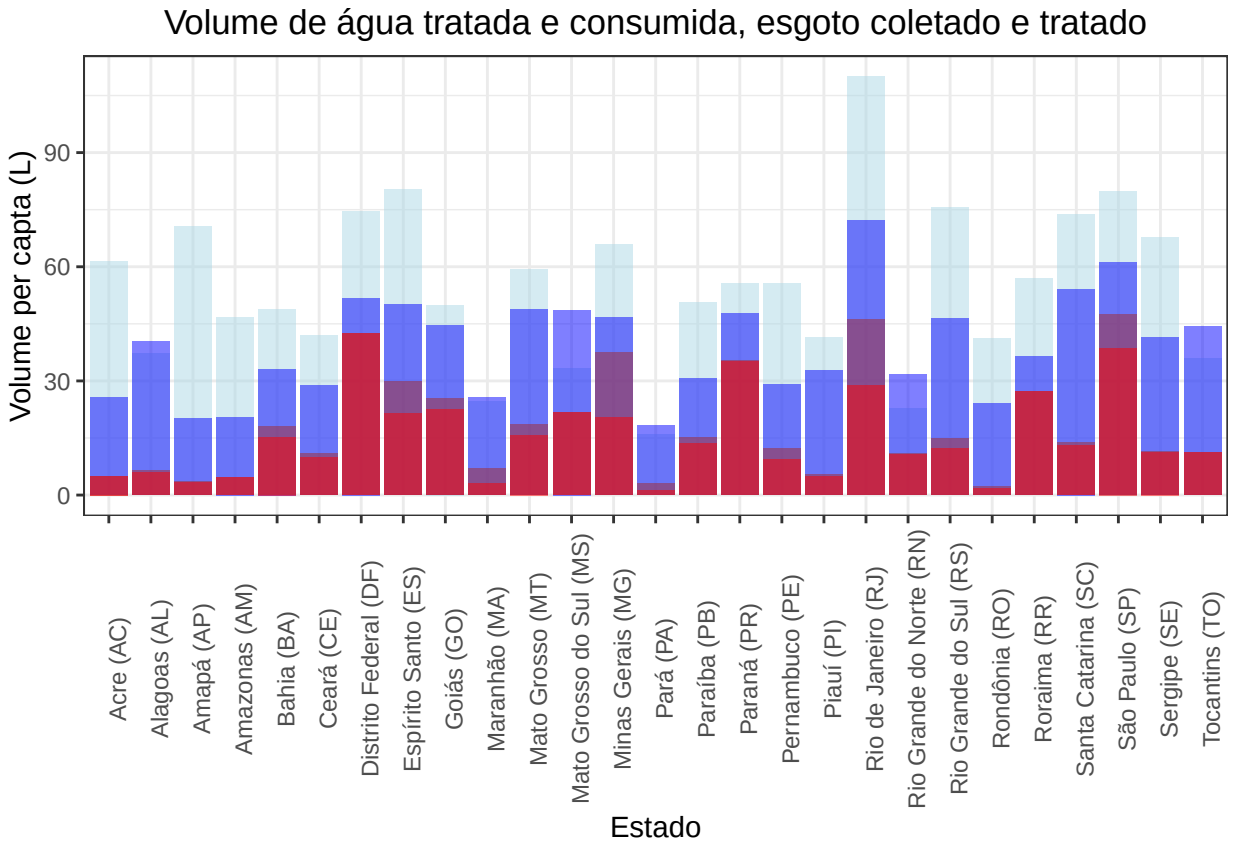

```
ggplot(df) +
  geom_point(aes(x=FN003 / Population, y=ES001 / Population * 100),
             color='brown', alpha=0.5) +
  theme_bw() +
  ylab('População atendida com esgotamento (%)') +
  xlab('Receitas operacionais per capta com esgotamento (R$)') +
  ggtitle('População atendida X operacionais per capta - esgotamento') +
  ylim(0,100) +
  xlim(0,NA) +
  theme(
    plot.title = element_text(hjust = 0.5)
  )
```



Agora vamos apresentar um gráfico mostrando o volume (L) de água tratada (azul claro) e consumida (azul), bem como o volume (L) de esgoto coletado (marrom) e tratado (vermelho) por estado. Nesse gráfico, a área em azul claro representa o volume de água que foi tratado e não foi consumido (perdas). Para vários estados, podemos observar que esse volume é bem significativo. É relevante salientar que os estados AL, MA, MS, PA, RN e TO possuem maiores volumes para o consumo de água do que para o tratamento, e isto não é nenhum erro. Isso ocorre porque alguns estados importam água de outros estados e operadoras. Na cor violeta podemos observar o volume de água consumida que não retornou como esgoto. Normalmente, para o cálculo de dimensionamento de estações de tratamento de esgoto, considera-se um retorno de aproximadamente 80% da água consumida, na forma de esgoto. Porém nesses dados, podemos observar que alguns estados possuem uma proporção bem menor, o que reflete que a maior parte dos esgotos nessas localidades ainda não são coletados. Os esgotos coletados que não foram tratados estão representados com cor de vinho, enquanto que em vermelho, temos a proporção de esgoto tratado. Podemos observar que o esgotamento é um problema muito grande na maior parte dos estados, correspondendo à um volume de esgoto coletado e tratado muito pequeno comparado com o volume consumido.

```
ggplot(df) +
  geom_bar(aes(x=State, y=AG007 / Population * 1000),
           stat = 'identity', fill='lightblue', alpha=0.5) +
  geom_bar(aes(x=State, y=AG010 / Population * 1000),
           stat = 'identity', fill='blue', alpha=0.5) +
  geom_bar(aes(x=State, y=ES005 / Population * 1000),
           stat = 'identity', fill='brown', alpha=0.5) +
  geom_bar(aes(x=State, y=ES006 / Population * 1000),
           stat = 'identity', fill='red', alpha=0.5) +
  theme_bw() +
  ylab('Volume per capta (L)') +
  xlab('Estado') +
```

```
  ggtitle('Volume de água tratada e consumida, esgoto coletado e tratado') +
  theme(
    axis.text.x = element_text(angle = 90),
    plot.title = element_text(hjust = 0.5)
  )
```



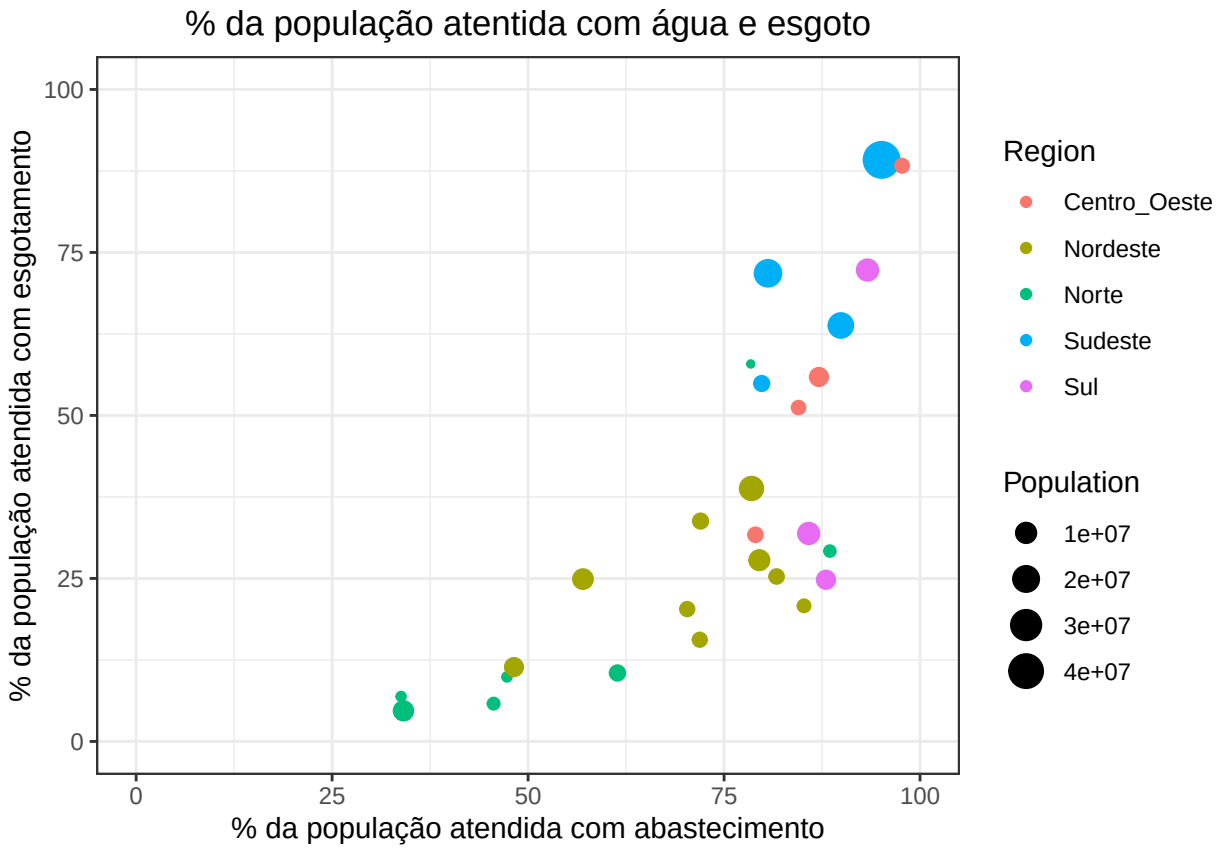
Finalmente, vamos criar duas novas colunas na tabela de dados df para poder criar o plot que relaciona a proporção da população com abastecimento de água e de esgotamento com o tamanho da população e estados.

```
# Criando a variável pop_water_prop que representa a proporção da população com abastecimento
df$pop_water_prop <- round(df$AG001/df$Population*100,1)

# Criando a variável pop_sewage_prop que representa a proporção da população com esgotamento
df$pop_sewage_prop <- round(df$ES001/df$Population*100,1)
```

Vamos gerar um objeto ggplot com a proporção da população atendida com abastecimento no eixo x, a população atendida com esgotamento no eixo y, o tamanho da população representado pelo tamanho dos pontos e com cores diferentes para cada região.

```
gg <- ggplot(df) +
  geom_point(aes(x=pop_water_prop,
                 y=pop_sewage_prop,
                 color=Region,
                 size=Population,
                 group=State)) +
  theme_bw() +
  xlab("% da população atendida com abastecimento") +
  ylab("% da população atendida com esgotamento") +
```

```
  ggtitle("% da população atentida com água e esgoto") +
  ylim(0,100) +
  xlim(0,100) +
  guides(color= guide_legend(), size=guide_legend()) +
  theme(
    plot.title = element_text(hjust = 0.5)
  )

#Visualizing a static version of the gg plot
gg
```



FIM