Prática no R! Roteiro 2 - Análise Exploratória I

Elaborado por Luis Felipe Bortolatto da Cunha

21 de setembro de 2020

Contents

1. Introdução

Você pode baixar este roteiro em formato PDF neste endereço.

Este roteiro tem como objetivo apresentar algumas funções para a execução de uma **análise exploratória** no software R.

A apresentação será feita usando uma base de dados demográficos e de consumo de água de 2010, extraídos do Censo Demográfico (IBGE) e Sistema Nacional de Informações sobre Saneamento (SNIS), para uma amostra de 4.417 municípios, organizada por Carmo et al., 2013.

A base de dados está disponível para download no endereço abaixo: https://1drv.ms/u/s!AjettDH-3Gbni9kJkYXYWfg32AqrKg?e=WkJ00U

Baixe essa base de dados pelo endereço indicado para salvá-la em seu computador.

Se você ainda não criou uma pasta para salvar o conteúdo da disciplina, essa é a hora: crie uma pasta e coloque a base de dados nessa pasta.

2. Criando um novo script

Após a instalação do R e RStudio, **abra o RStudio**. Clique em `New file` e em seguida em `R Script` ou aperte os comando `Ctrl + Shift + N` para criar um novo script, onde você vai salvar a rotina de análise de dados.

É muito importante salvar a rotina de cada semana em um script, para lembrar as funções que foram usadas e reproduzir a análise de dados, se necessário.

3. Definindo o diretório de trabalho

O **diretório de trabalho** refere-se à **pasta** onde você está guardando a sua base de dados e outros arquivos.

Sempre que você abrir o RStudio, um diretório de trabalho será definido automaticamente. Para conferir qual o seu diretório de trabalho, você pode executar a função `getwd()`.

```
getwd()
```

```
## [1] "D:/OneDrive"
```

Neste exemplo, o diretório de trabalho retornado foi `"D:/OneDrive"`, mas os dados estão salvos em outro diretório. Para definir um novo diretório de trabalho pela interface, clique em `Session`, `Set Working Directory` e `Choose Directory...` ou aperte os comandos `Ctrl + Shift + H` e escolha a pasta onde você está salvando os seus arquivos.

A função `setwd()` será executada com um argumento para definir o diretório de trabalho, de forma semelhante ao exemplo abaixo.

```
setwd("~/MTI")
```

Como o objetivo de um script é tornar o código reprodutível, não se esqueça de copiar o código executado no `Console` para o script. Você pode fazer isso clicando em `To Source` na aba `History`.

Para conferir se deu tudo certo, você pode executar a função `getwd()` novamente para visualizar o seu diretório de trabalho.

```
getwd()
```

```
## [1] "C:/Users/luisf/Documents/MTI"
```

A definição do diretório de trabalho é uma etapa opcional, mas que pode economizar muito tempo no processo de análise de dados. Quando você for importar ou exportar uma base de dados, ao invés de escrever o caminho completo até os dados (por exemplo, `"C:/Users/luisf/Documents/MTI/dados/agua1.csv"`), você pode escrever apenas metade do caminho (seguindo o exemplo, seria `"dados/agua1.csv"`).

4. Importando a base de dados

A base de dados que você baixou está em formato textual, mais especificamente no formato `csv`, que é a forma mais comum para a disponibilização de dados na internet.

ATENÇÃO: o formato `csv` possui duas variações: `csv` - os valores são separados por vírgula e os decimais por ponto; `csv2` - os valores são separados por ponto e vírgula e os decimais por vírgula. Lembre-se de especificar o formato correto.

É possível importar essa base de dados pela interface, em `Environment`, ou código.

Para importar a base de dados pela interface, clique em `Import Dataset` e em seguida em `From Text (base)...`. Na janela que foi aberta, encontre o arquivo que deseja importar no diretório e clique em `Open`. Uma janela com opções adcionais será aberta. Você deve selecionar as seguintes opções para importar a base de dados baixada:

- Encoding: UTF-8 (Codificação: UTF-8)
- Heading: Yes (Cabeçalho: Sim)
- Separator: Semicolon (Separador: ponto e vírgula)
- Decimal: Comma (Decimal: vírgula)

Uma pré-visualização estará disponível em `Data Frame`, onde você pode conferir se a base de dados será importada corretamente. Quando estiver pronto, clique em `Import`.

Nesse momento o código foi executado no `Console`, um novo objeto está listado em `Environment` (agua1 - 5566 observações de 14 variáveis) e uma nova janela foi aberta em `Source`, que permite a visualização interativa da tabela, de forma semelhante ao Excel.

Embora a interface seja útil para importar dados, para tornar o código reprodutível, é necessário copiar o código que foi executado para o script. Faça isso copiando e colando do `Console` ou pela aba `History`.

O código copiado e colado deve ser semelhante ao código abaixo, com alterações apenas no diretório.

```
agua1 <- read.csv2("~/MTI/dados/agua1.csv", encoding="UTF-8")
```

Se você fechou a janela em `Source` que permite a visualização interativa da tabela e deseja abri-la de novo, execute a função `View()` com o nome da base de dados como argumento, sem aspas.

```
View(agua1)
```

5. Explorando os dados

Além da função `View()`, que permite a visualização interativa de uma tabela, existem diversas outras funções que permitem explorar uma base de dados.

A função `names()` exibe o nome de todas as variáveis da base de dados no `Console`.

```
names(agua1)
```

```
##  [1] "ID_IBGE"   "ID_SNIS"   "NOME_MUN"  "UF"        "REGIAO"    "PIB"
##  [7] "RENDAPITA" "GINI"      "IDH"       "IDH_CLASS" "GE012"     "AG001"
## [13] "AG020"     "AG022"
```

Os nomes das variáveis dessa base de dados estão codificados, o que é uma prática comum na disponibilização de bases de dados na internet. Para saber o que cada variável significa, as bases de dados geralmente vêm acompanhadas de um dicionário, que é um repositório centralizado com informações sobre os dados (metadados), tais como: código, descrição, origem, uso e formato. A tabela abaixo apresenta uma descrição de cada variável, de forma semelhante a um dicionário:

Código	Descrição
ID_IBGE	Código IBGE (7 dígitos)
ID_SNIS	Código IBGE (6 dígitos)
NOME_MUN	Nome do Município
UF	Unidade da Federação
REGIAO	Região do País
PIB	PIB 2010
RENDAPITA	Renda per Capita 2010
GINI	Índice GINI 2010

Código	Descrição
IDH	Índice de Desenvolvimento Humano 2010
IDH_CLASS	Classificação do Índice de Desenvolvimento Humano 2010: Muito Alto >= 0,9; Alto >= 0,8; Médio >= 0,5; Baixo < 0,5.
GE012	População Total Residente no Município
AG001	População Total Atendida com Abastecimento de Água
AG020	Volume Micromedido nas Economias Residenciais Ativas de Agua - 1.000 m3/ano
AG022	Quantidade de Economias Residenciais Ativas Micromedidas

A função `head()` exibe as primeiras observações da base de dados no `Console`, sem a necessidade de abrir uma nova aba em `Source`. Seu uso é altamente recomendado para exibir bases de dados muito grandes (mais de 100.000 observações) que podem travar o programa RStudio quando aberta a visualização interativa.

```
head(agua1)
```

```
##   ID_IBGE ID_SNIS              NOME_MUN UF REGIAO       PIB RENDAPITA   GINI
## 1 1100015  110001 Alta Floresta D'Oeste RO  Norte 167212.50    467.72 0.5893
## 2 1100023  110002             Ariquemes RO  Norte 613103.30    672.87 0.5496
## 3 1100031  110003                Cabixi RO  Norte  41686.84    446.58 0.5166
## 4 1100049  110004                Cacoal RO  Norte 582804.40    718.79 0.5890
## 5 1100056  110005            Cerejeiras RO  Norte  86885.25    553.47 0.5147
## 6 1100064  110006     Colorado do Oeste RO  Norte 109115.00    507.70 0.5154
##     IDH IDH_CLASS GE012 AG001   AG020 AG022
## 1 0.715     Médio 24392  9184   99.00   602
## 2 0.752     Médio 90353 35968 1145.40  8020
## 3 0.705     Médio  6313  1651   27.30   358
## 4 0.755     Médio 78574 61921 1828.99  9769
## 5 0.751     Médio 17029  8238  249.30  1923
## 6 0.739     Médio 18591 13435  515.80  3593
```

A função `tail()`, de forma similar, exibe as últimas observações da base de dados no `Console`.

```
tail(agua1)
```

```
##      ID_IBGE ID_SNIS       NOME_MUN UF       REGIAO         PIB RENDAPITA
## 5561 5221908  522190         Varjão GO Centro-Oeste    19997.78    599.29
## 5562 5222005  522200     Vianópolis GO Centro-Oeste    86180.24    644.68
## 5563 5222054  522205 Vicentinópolis GO Centro-Oeste    80684.42    613.29
## 5564 5222203  522220       Vila Boa GO Centro-Oeste    17711.13    370.83
## 5565 5222302  522230  Vila Propício GO Centro-Oeste    48867.57    370.43
## 5566 5300108  530010       Brasília DF Centro-Oeste 43521630.00   1665.42
##        GINI   IDH IDH_CLASS   GE012   AG001     AG020  AG022
## 5561 0.4731 0.729     Médio    3659    2243     94.43    818
## 5562 0.4672 0.784     Médio   12548    9164    359.27   3058
## 5563 0.4824 0.773     Médio    7371    6321        NA     NA
## 5564 0.4935 0.674     Médio    4735    3497    121.67   1115
## 5565 0.5240 0.674     Médio    5145    1198     34.46    343
## 5566 0.6370 0.844      Alto 2570160 2556024 142390.00 857488
```

A função `str()` exibe a estrutura da base de dados, com a classe da base de dados, o número de observações e variáveis, o nome de cada variável, a classe de cada variável (se é numérica, lógica, caractére ou outra) e as primeiras observações.

```
str(agua1)
```

```
## 'data.frame':    5566 obs. of  14 variables:
```

```
##  $ ID_IBGE  : int  1100015 1100023 1100031 1100049 1100056 1100064 1100072 1100080 1100098 1100106 .
##  $ ID_SNIS  : int  110001 110002 110003 110004 110005 110006 110007 110008 110009 110010 ...
##  $ NOME_MUN : chr  "Alta Floresta D'Oeste" "Ariquemes" "Cabixi" "Cacoal" ...
##  $ UF       : chr  "RO" "RO" "RO" "RO" ...
##  $ REGIAO   : chr  "Norte" "Norte" "Norte" "Norte" ...
##  $ PIB      : num  167213 613103 41687 582804 86885 ...
##  $ RENDAPITA: num  468 673 447 719 553 ...
##  $ GINI     : num  0.589 0.55 0.517 0.589 0.515 ...
##  $ IDH      : num  0.715 0.752 0.705 0.755 0.751 0.739 0.668 0.693 0.738 0.743 ...
##  $ IDH_CLASS: chr  "Médio" "Médio" "Médio" "Médio" ...
##  $ GE012    : int  24392 90353 6313 78574 17029 18591 8783 13678 28729 41656 ...
##  $ AG001    : int  9184 35968 1651 61921 8238 13435 1268 2283 9177 18365 ...
##  $ AG020    : num  99 1145.4 27.3 1829 249.3 ...
##  $ AG022    : int  602 8020 358 9769 1923 3593 204 469 2160 3606 ...
```

IMPORTANTE: o operador $ é usado para referir-se a uma variável da base de dados.

A função `summary()` apresenta um sumário de estatísticas descritivas (mínimo, 1º quartil, mediana, média, 3º quartil, máximo e valores faltantes - NA's) para todas as variáveis numéricas.

```
summary(agua1)
```

```
##     ID_IBGE           ID_SNIS         NOME_MUN             UF
##  Min.   :1100015   Min.   :110001   Length:5566        Length:5566
##  1st Qu.:2512126   1st Qu.:251213   Class :character   Class :character
##  Median :3146230   Median :314623   Mode  :character   Mode  :character
##  Mean   :3253242   Mean   :325324
##  3rd Qu.:4119078   3rd Qu.:411908
##  Max.   :5300108   Max.   :530010
##
##     REGIAO               PIB              RENDAPITA           GINI
##  Length:5566        Min.   :        0   Min.   :   0.0   Min.   :0.0000
##  Class :character   1st Qu.:    25236   1st Qu.: 275.0   1st Qu.:0.4586
##  Mode  :character   Median :    53505   Median : 456.3   Median :0.5025
##                     Mean   :   349732   Mean   : 483.3   Mean   :0.5030
##                     3rd Qu.:   138168   3rd Qu.: 636.9   3rd Qu.:0.5459
##                     Max.   :160637500   Max.   :2009.0   Max.   :0.8082
##                     NA's   :627
##       IDH          IDH_CLASS             GE012              AG001
##  Min.   :0.0000   Length:5566        Min.   :     805   Min.   :       0
##  1st Qu.:0.6330   Class :character   1st Qu.:    5593   1st Qu.:    2990
##  Median :0.7160   Mode  :character   Median :   11525   Median :    6385
##  Mean   :0.6979                      Mean   :   37194   Mean   :   30046
##  3rd Qu.:0.7700                      3rd Qu.:   24682   3rd Qu.:   15826
##  Max.   :0.9190                      Max.   :11253503   Max.   :11253503
##  NA's   :627                         NA's   :627        NA's   :627
##      AG020            AG022
##  Min.   :     0   Min.   :      1
##  1st Qu.:    90   1st Qu.:    848
##  Median :   210   Median :   1873
##  Mean   :  1368   Mean   :   9247
##  3rd Qu.:   592   3rd Qu.:   4697
##  Max.   :627684   Max.   :3900531
##  NA's   :1149     NA's   :986
```

6. Criando um subconjunto (filtrar observações e selecionar variáveis)

No decorrer de uma análise de dados, você pode precisar selecionar variáveis e filtrar observações, ou seja criar um subconjunto. Na linguagem R, subconjuntos podem ser criados com o uso de colchetes, seguindo o formato: `base_da_dados[observações,variáveis]`.

Por exemplo, é possível selecionar as seis primeiras observações, de forma similar à função `head()`, com o comando:

```
agua1[1:6,]
```

```
##   ID_IBGE ID_SNIS              NOME_MUN UF REGIAO       PIB RENDAPITA   GINI
## 1 1100015  110001 Alta Floresta D'Oeste RO  Norte 167212.50    467.72 0.5893
## 2 1100023  110002             Ariquemes RO  Norte 613103.30    672.87 0.5496
## 3 1100031  110003                Cabixi RO  Norte  41686.84    446.58 0.5166
## 4 1100049  110004                Cacoal RO  Norte 582804.40    718.79 0.5890
## 5 1100056  110005            Cerejeiras RO  Norte  86885.25    553.47 0.5147
## 6 1100064  110006     Colorado do Oeste RO  Norte 109115.00    507.70 0.5154
##     IDH IDH_CLASS GE012 AG001   AG020 AG022
## 1 0.715     Médio 24392  9184   99.00   602
## 2 0.752     Médio 90353 35968 1145.40  8020
## 3 0.705     Médio  6313  1651   27.30   358
## 4 0.755     Médio 78574 61921 1828.99  9769
## 5 0.751     Médio 17029  8238  249.30  1923
## 6 0.739     Médio 18591 13435  515.80  3593
```

Para visualizar apenas o IDH desses seis municípios, podemos adicionar um segundo argumento com o número ou nome da variável:

```
agua1[1:6,9]
```

```
## [1] 0.715 0.752 0.705 0.755 0.751 0.739
```

```
agua1[1:6,"IDH"]
```

```
## [1] 0.715 0.752 0.705 0.755 0.751 0.739
```

Outra forma de acessar uma coluna pelo seu nome é usando o operador `$` (cifrão) após os colchetes.

```
agua1[1:6,]$IDH
```

```
## [1] 0.715 0.752 0.705 0.755 0.751 0.739
```

Para ver o nome do município e IDH, podemos adicionar um vetor no segundo argumento com as variáveis de interesse:

```
agua1[1:6,c("NOME_MUN", "IDH")]
```

```
##                NOME_MUN   IDH
## 1 Alta Floresta D'Oeste 0.715
## 2             Ariquemes 0.752
## 3                Cabixi 0.705
## 4                Cacoal 0.755
## 5            Cerejeiras 0.751
## 6     Colorado do Oeste 0.739
```

IMPORTANTE: é possível criar vetores com a função `c()` (concatenar).

Subconjuntos também funcionam com operadores lógicos, sendo possível filtrar as observações que satisfazem

uma determinada condição usando a função `which()`. No exemplo abaixo, visualizamos o nome e IDH dos municípios que satisfazem as condições: Unidade da Federação é igual a São Paulo E IDH maior que 0,85.

```
agua1[which(agua1$UF == "SP" & agua1$IDH > 0.85),c("NOME_MUN", "IDH")]
```

```
##                 NOME_MUN   IDH
## 3273  Águas de São Pedro 0.908
## 3375            Campinas 0.852
## 3560             Jundiaí 0.857
## 3754      Ribeirão Preto 0.855
## 3773            Saltinho 0.851
## 3795 Santana de Parnaíba 0.853
## 3809              Santos 0.871
## 3812  São Caetano do Sul 0.919
## 3903             Vinhedo 0.857
```

7. Criando tabelas de contingência

Se você estiver trabalhando com variáveis categóricas, pode ser necessário criar tabelas de contingência para apresentar a contagem de ocorrências daquela variável. Para exibir a distribuição de municípios de acordo com a classificação do IDH, usaremos a função `table()`.

```
table(agua1$IDH_CLASS)
```

```
##
##       Alto      Baixo      Médio Muito alto
##        533         48       4356          2
```

A função `table()` também pode ser usada com dois argumentos, conforme mostra o exemplo abaixo com a classificação do IDH por região.

```
table(agua1$REGIAO, agua1$IDH_CLASS)
```

```
##
##                Alto Baixo Médio Muito alto
##   Centro-Oeste   26    15   376          0
##   Nordeste        2    15  1569          0
##   Norte           2     1   340          0
##   Sudeste       207     1  1295          2
##   Sul           296    16   776          0
```

8. Definindo uma nova variável

É possível criar uma nova variável usando o operador `<-` ou `=`. Neste exemplo, vamos criar duas novas variáveis, chamadas `CONSUMO1` e `CONSUMO2`:

- `CONSUMO1`: Consumo de Água per capita - População Total - m3/ano (AG020/GE012)
- `CONSUMO2`: Consumo de Água per capita - População Atendida - m3/ano (AG020/AG001)

Como a unidade do consumo de água (AG020) é **1.000 m³/ano**, primeiro é preciso multiplicar AG020 por 1.000 para obter a unidade **m³/ano** e depois dividir pela população para obter o consumo de água per capita em m3/ano.

```
agua1$CONSUMO1 <- agua1$AG020 * 1000 / agua1$GE012
```

```
agua1$CONSUMO2 <- agua1$AG020 * 1000 / agua1$AG001
```

Exibindo as primeiras observações, podemos ver que as duas novas variáveis foram definidas com sucesso.

```r
head(agua1)
```

```
##   ID_IBGE ID_SNIS              NOME_MUN UF REGIAO       PIB RENDAPITA   GINI
## 1 1100015  110001 Alta Floresta D'Oeste RO  Norte 167212.50    467.72 0.5893
## 2 1100023  110002             Ariquemes RO  Norte 613103.30    672.87 0.5496
## 3 1100031  110003                Cabixi RO  Norte  41686.84    446.58 0.5166
## 4 1100049  110004                Cacoal RO  Norte 582804.40    718.79 0.5890
## 5 1100056  110005            Cerejeiras RO  Norte  86885.25    553.47 0.5147
## 6 1100064  110006     Colorado do Oeste RO  Norte 109115.00    507.70 0.5154
##     IDH IDH_CLASS GE012 AG001   AG020 AG022  CONSUMO1 CONSUMO2
## 1 0.715     Médio 24392  9184   99.00   602  4.058708 10.77962
## 2 0.752     Médio 90353 35968 1145.40  8020 12.676945 31.84497
## 3 0.705     Médio  6313  1651   27.30   358  4.324410 16.53543
## 4 0.755     Médio 78574 61921 1828.99  9769 23.277293 29.53748
## 5 0.751     Médio 17029  8238  249.30  1923 14.639732 30.26220
## 6 0.739     Médio 18591 13435  515.80  3593 27.744608 38.39226
```

9. Calculando estatísticas básicas (média, mediana, variância, desvio padrão)

Com o uso do operador `$` podemos aplicar funções em uma variável para obter estatísticas básicas, como média, mediana, desvio padrão e variância. Vamos executar essas funções na nova variável `CONSUMO1`. Em todas as funções usaremos o argumento opcional `na.rm = TRUE` para remover os valores faltantes (NA). Se a variável não possuir valores faltantes, esse argumento não é necessário.

```r
mean(agua1$CONSUMO1, na.rm = TRUE)
```

```
## [1] 24.76889
```

```r
median(agua1$CONSUMO1, na.rm = TRUE)
```

```
## [1] 22.35445
```

```r
var(agua1$CONSUMO1, na.rm = TRUE)
```

```
## [1] 270.1714
```

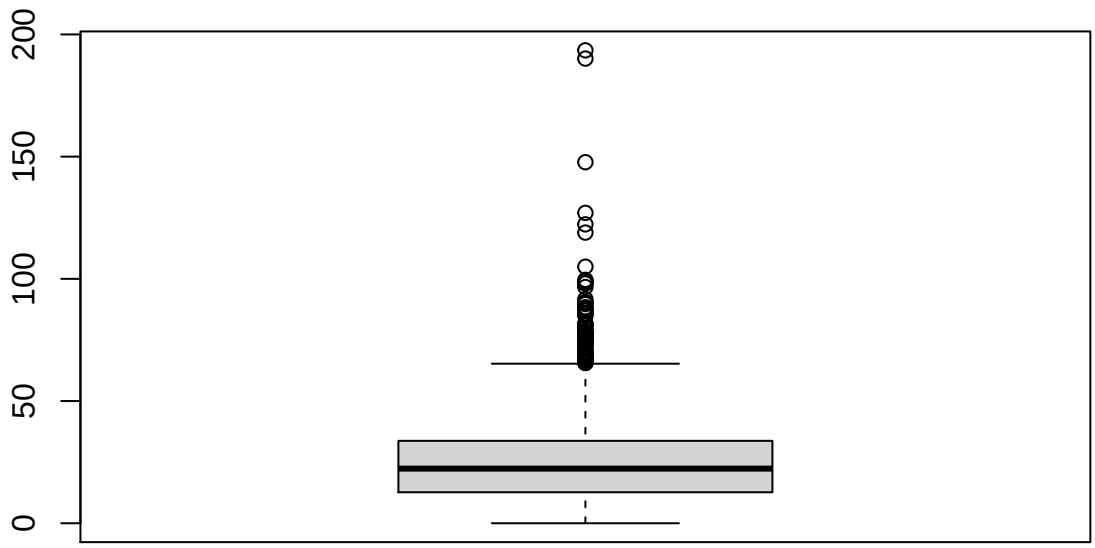
```r
sd(agua1$CONSUMO1, na.rm = TRUE)
```

```
## [1] 16.43689
```

10. Desenhando gráficos - Box-plot

O **box-plot** é um gráfico utilizado para **avaliar a distribuição empírica de uma variável**, exibindo a mediana, quartis e limites superior e inferior. Para desenhar um box-plot, use a função `boxplot()`.

```r
boxplot(agua1$CONSUMO1)
```

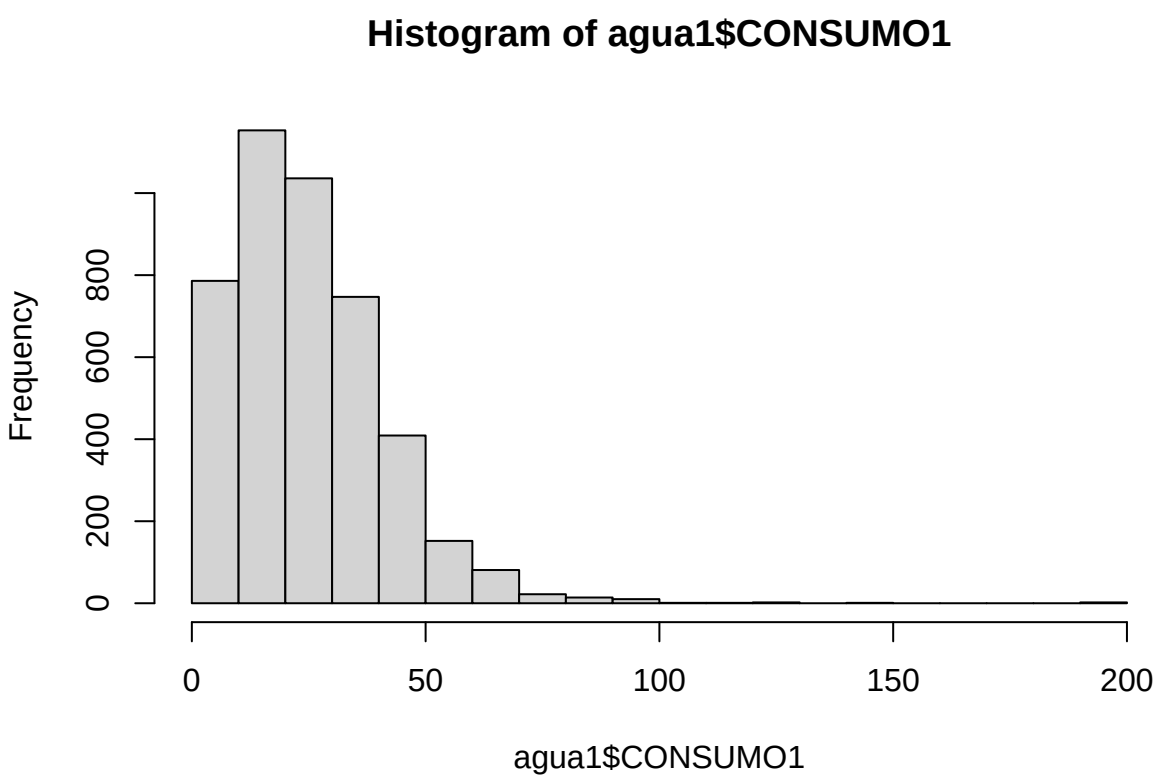


O box-plot de `CONSUMO1` indica que existem diversos valores atípicos (*outliers*), ou seja, fora dos limites.

11. Desenhando gráficos - Histograma

O **histograma** é um gráfico que exibe a **frequência, ou distribuição, dos valores de uma variável**. É possível desenhar um histograma no R básico com a função `hist()`.

```
hist(agua1$CONSUMO1)
```

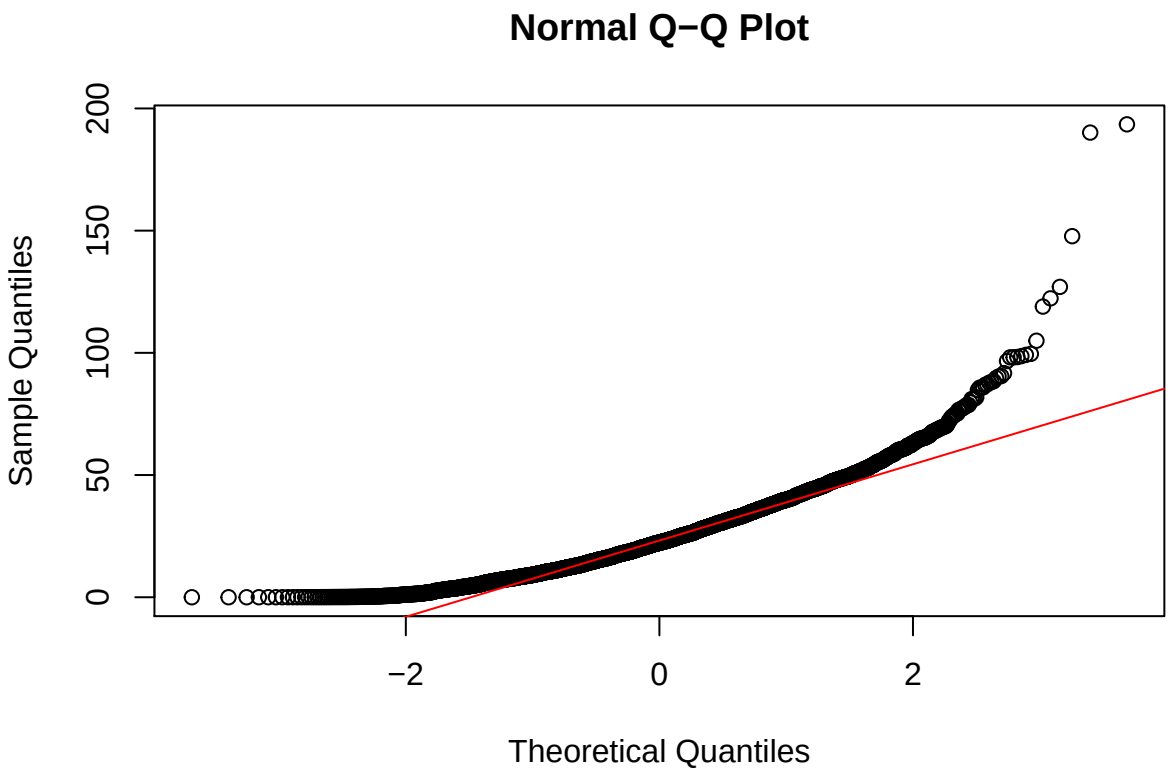


O histograma de `CONSUMO1` sugere uma distribuição positivamente assimétrica dessa variável. A barra maior indica a `moda`, que neste caso está entre 10-20 m3/ano, sendo menor do que a `média` (24,77 m3/ano) e `mediana` (22,35 m3/ano). Como a média, mediana e moda não coincidem, a distribuição de `CONSUMO1` não representa uma curva normal.

12. Desenhando gráficos - Qqplot

A distribuição de `CONSUMO1` também pode ser visualizada com um **qqplot**, que compara duas distribuições de dados traçando quantis. Para desenhar um qqplot comparando a distribuição de `CONSUMO1` com uma curva normal, usaremos as funções `qqnorm` e `qqline`, conforme o exemplo abaixo.

```
qqnorm(agua1$CONSUMO1)
qqline(agua1$CONSUMO1, col = "red")
```



O qqplot de `CONSUMO1` sugere uma certa linearidade pela proximidade com a curva normal (em vermelho) nos quantis centrais, desviando da curva em valores muito altos e muito baixos (*outliers*).

13. Exportando um gráfico

É possível exportar os gráficos pela interface ou código. Para exportar pela interface, clique em `Export` e `Save as Image...` na aba `Plots`. Escolha o diretório (`Directory`), nome do arquivo (`File name`), largura (`Width`) e altura (`Height`). Uma pré-visualização estará disponível ao clicar em `Update Preview`. Quando estiver satisfeito com o seu gráfico, clique em `Save`.

Para exportar um gráfico pela linha de código, primeiro execute a função `png()` para criar um arquivo (com o diretório, nome do arquivo e formato entre aspas como argumento), depois execute a função do seu gráfico e em seguida a função `dev.off()` para salvar o arquivo. Argumentos adicionais da função `png()` podem ser acessados na sua documentação e incluem a largura e altura, por exemplo.

```
png("graficos/boxplot_CONSUMO1.png")
boxplot(agua1$CONSUMO1)
dev.off()
```

```
png("graficos/histograma_CONSUMO1.png")
hist(agua1$CONSUMO1)
dev.off()
```

```
png("graficos/qqplot_CONSUMO1.png")
qqnorm(agua1$CONSUMO1)
qqline(agua1$CONSUMO1, col = "red")
dev.off()
```

14. Exportando a base de dados

Após executar a rotina, você fez alterações na base de dados (novas variáveis - CONSUMO1 e CONSUMO2), por isso é importante salvá-la para acesso futuro. O R não vai exportar a sua base de dados a menos que você especifique esse comando.

É possível exportar a base de dados no formato `csv2` com a função `write.csv2()`. Ela exige dois argumentos: primeiro o nome do objeto que deseja exportar e depois o diretório, nome do arquivo e formato entre aspas. Para não perder a base de dados original, recomendamos salvar essa base de dados com um novo nome - `agua2`.

```
write.csv2(agua1, "dados/agua2.csv")
```

15. Salvando o script

Antes de encerrar a sessão no RStudio, é muito importante salvar o script para tornar a análise exploratória proposta reprodutível.

Para salvar o script, clique no disquete azul em `Source` ou aperte o comando `Ctrl + S`.

IMPORTANTE: Todo o conteúdo após o símbolo # (jogo da velha/*hashtag*), recebe destaque e é considerado um comentário. Isso significa que todo o conteúdo após o # não será executado pelo R. É comum adicionar comentários que explicam o código com o uso de #.