Oficina de Estatistica Tutorial 2

Virgilio Mendes

15/07/2019

Tutorial 2

Antes de iniciarmos a próxima etapa, vamos limpar o ambiente de trabalho. A função `rm` serve para remover objetos do environment. Ela deve vir acompanhada do argumento `list`, que incluirá a lista de objetos a serem removido. No caso, removeremos todos os objetos do ambiente de trabalho, que são listados pela função `ls`.

```
rm(list = ls())
```

Pronto, agora nosso environment está sem nenhum objeto.

1. Estatísticas descritivas e tabelas de frequência

O cálculo de estatísticas descritivas no R é bastante simples. Existem funções para o cálculo das principais estatísticas descritivas, além de funções que apresentam um resumo destas.

Para calculá-las, vamos usar outras variáveis do questionário respondido pelos colegas. Contudo, dessa vez, para abrirmos o banco de dados, vamos usar a função `read.csv2`, ao invés da `read.table`. A `read.csv2` é uma função já programada para abrir arquivos em formato .csv separados por `;`, o que exige a inclusão de menos argumentos.

```
banco <- read.csv("/home/virgilioam/Área de trabalho/Modus 2019/Oficina de Estatística/Oficina de Estati
```

Começaremos com o cálculo de estatísticas descritivas de variáveis contínuas, destacando as estatísticas de ordenamento e as estatísticas de momento. Por fim, vamos descrever variáveis categóricas.

1.1 Estatísticas de ordenamento

Os nomes das funções para o cálculo de estatísticas descritivas são, em geral, o nome em inglês ou uma abreviatura do nome dessa estatística.

1.1.1 Mediana

Assim, para o cálculo da **mediana** de uma distribuição, usamos a função `median`. Vamos então calcular a mediana de minutos de deslocamento até a UFMG dos alunos dessa oficina e também a mediana do número de livros lidos por eles.

```
median(banco$minutos_deslocamento)
```

```
[1] 30
```

```r
median(banco$livros_lidos)
```

[1] 10

1.1.2 Mínimo e máximo

As estatísticas de valores **mínimo** e **máximo** são convenientes para a compreensão do ordenamento e alcance dos valores de uma variável contínua. Elas também podem ser úteis para identificação de erros ou valores discrepantes no banco de dados (por exemplo, se o valor máximo de uma variável "idade" é 135 anos, isso sugere que houve um erro de preenchimento).

```r
min(banco$minutos_deslocamento)
```

[1] 5

```r
max(banco$minutos_deslocamento)
```

[1] 90

```r
range(banco$minutos_deslocamento) # apresenta os valores minimo e maximo
```

[1] 5 90

```r
min(banco$livros_lidos)
```

[1] 0

```r
max(banco$livros_lidos)
```

[1] 25

```r
range(banco$livros_lidos)
```

[1] 0 25

1.1.3 Quantis

Como vimos, algumas das principais estatísticas para descrever uma distribuição de variável contínua são os quantis. O R nos dá grande flexibilidade para o seu cálculo. Para tanto, usamos a função `quantiles`. Geralmente, os quantis utilizados para descrever uma distribuição são os *quartis*.

```r
# Quartis
quantile(banco$minutos_deslocamento, probs = seq(0, 1, 0.25))
```

0% 25% 50% 75% 100% 5.00 16.25 30.00 40.00 90.00

```r
quantile(banco$minutos_deslocamento, probs = c(0, 0.25, 0.5, 0.75, 1))
```

0% 25% 50% 75% 100% 5.00 16.25 30.00 40.00 90.00

```r
# Quintis
quantile(banco$minutos_deslocamento, probs = seq(0, 1, 0.2))
```

0% 20% 40% 60% 80% 100% 5 15 24 32 46 90

```r
quantile(banco$minutos_deslocamento, probs = c(0, .2, .4, .6, .8, 1))
```

0% 20% 40% 60% 80% 100% 5 15 24 32 46 90

```r
# Decis
quantile(banco$minutos_deslocamento, probs = seq(0, 1, .1))
```

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% 5 10 15 20 24 30 32 40 46 62 90

```r
quantile(banco$minutos_deslocamento, probs = c(0, 0.1, 0.2, 0.3, 0.5,
                                0.6, 0.7, 0.8, 0.9, 1))
```

0% 10% 20% 30% 50% 60% 70% 80% 90% 100% 5 10 15 20 30 32 40 46 62 90

```r
# Percentis
quantile(banco$minutos_deslocamento, probs = seq(0, 1, 0.01))
```

0% 1% 2% 3% 4% 5% 6% 7% 8% 9% 10% 11% 5.00 5.85 6.70 7.55 8.40 9.25 10.00 10.00 10.00 10.00 10.00 10.00 12% 13% 14% 15% 16% 17% 18% 19% 20% 21% 22% 23% 10.20 11.05 11.90 12.75 13.60 14.45 15.00 15.00 15.00 15.00 15.00 15.00 24% 25% 26% 27% 28% 29% 30% 31% 32% 33% 34% 35% 15.40 16.25 17.10 17.95 18.80 19.65 20.00 20.00 20.00 20.00 20.00 20.00 36% 37% 38% 39% 40% 41% 42% 43% 44% 45% 46% 47% 20.60 21.45 22.30 23.15 24.00 24.85 25.70 26.55 27.40 28.25 29.10 29.95 48% 49% 50% 51% 52% 53% 54% 55% 56% 57% 58% 59% 30.00 30.00 30.00 30.00 30.00 30.00 30.00 30.00 30.00 30.00 30.00 30.30 60% 61% 62% 63% 64% 65% 66% 67% 68% 69% 70% 71% 32.00 33.70 35.40 37.10 38.80 40.00 40.00 40.00 40.00 40.00 40.00 40.00 72% 73% 74% 75% 76% 77% 78% 79% 80% 81% 82% 83% 40.00 40.00 40.00 40.00 40.00 40.90 42.60 44.30 46.00 47.70 49.40 50.00 84% 85% 86% 87% 88% 89% 90% 91% 92% 93% 94% 95% 50.00 50.00 50.00 50.00 50.00 55.20 62.00 68.80 75.60 82.40 89.20 90.00 96% 97% 98% 99% 100% 90.00 90.00 90.00 90.00 90.00

A função `IQR` calcula o intervalo interquartil (IIQ), isto é, a diferença entre os valores do terceiro e do primeiro quartil da distribuição.

```r
IQR(banco$minutos_deslocamento)
```

[1] 23.75

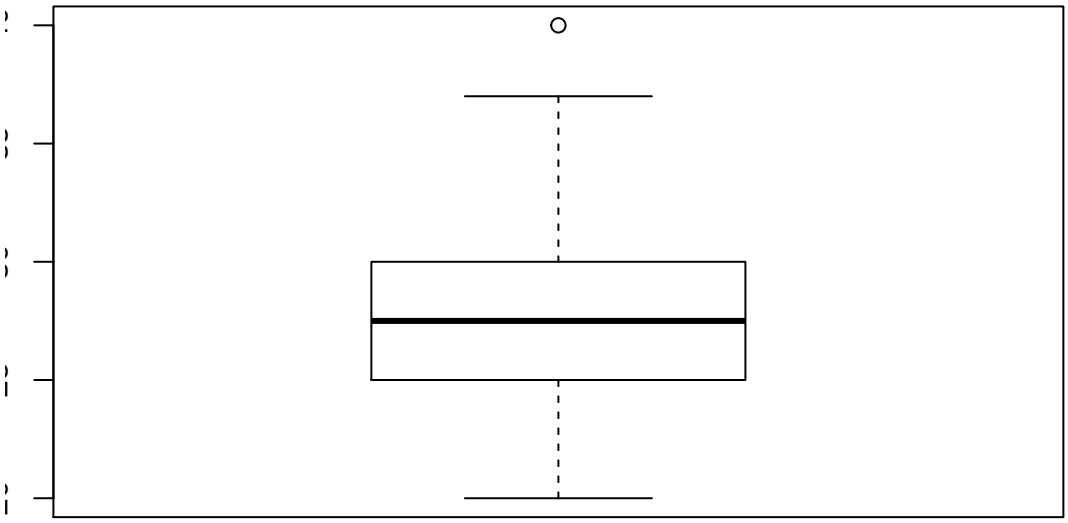
1.1.4 Representação gráfica das estatísticas de ordenamento

Como vimos na parte expositiva, as estatísticas de ordenamento são representadas graficamente por meio de um box-plot. Como o objetivo do nosso curso são as noções de estatística e não a programação em R, faremos todos os nossos gráficos a partir do R base (mas já fica a sugestão, para aqueles que não conhecem, de explorarem o pacote ggplot2).

As funções do R base para a produção de gráficos são simples, diretas e muito úteis para a exploração inicial dos dados (além disso, assim como em gráficos produzidos com o ggplot, ao especificar diversos dos argumentos das funções de gráficos do R base, podemos personalizar bastante essas imagens).

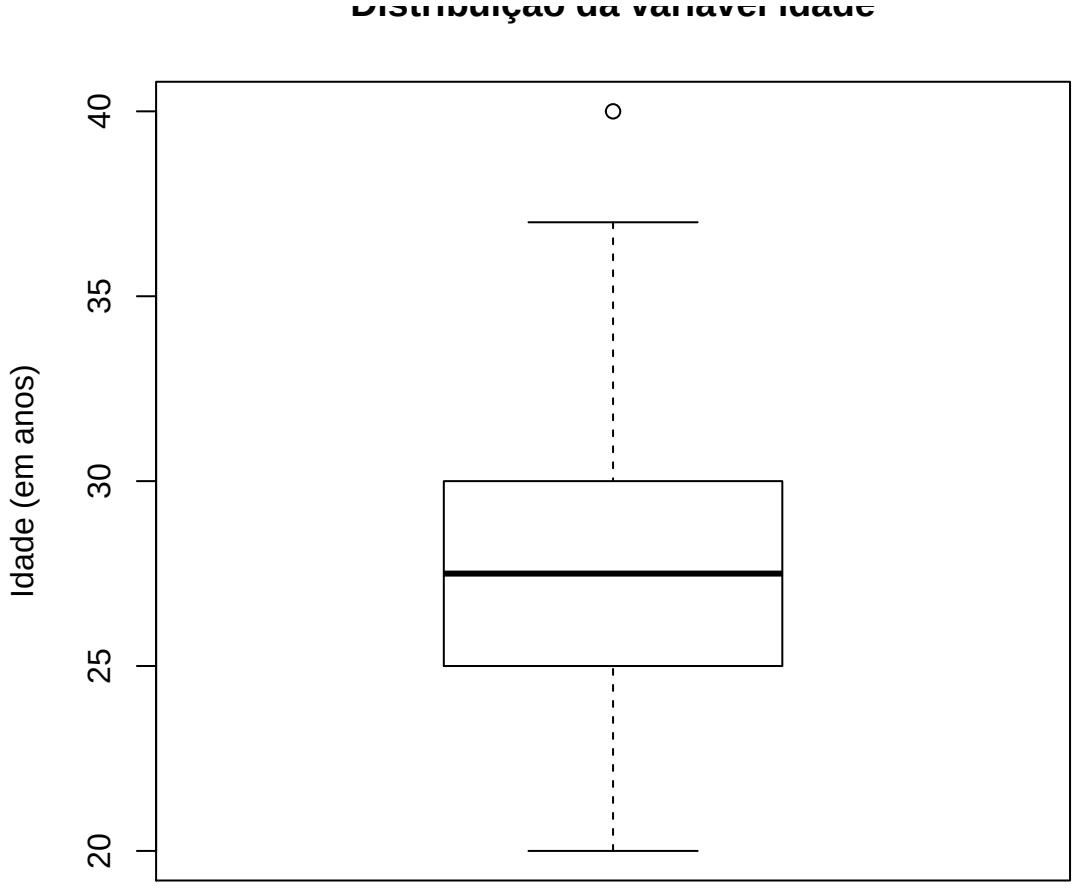
Para produzirmos um boxplot, utilizamos a função `boxplot`.

```
boxplot(banco$idade)
```



Para personalizar um pouco o gráfico (o que pode ser muito útil para relatórios, artigos etc), podemos especificar alguns dos argumentos:

```
boxplot(banco$idade, main = "Distribuição da variável idade",
        ylab = "Idade (em anos)")
```



1.2 Estatísticas de momento

1.2.1 Média

A principal estatística de momento é a **média**. Para calculá-la, usamos a função `mean`.

```
var(banco$minutos_deslocamento)
```

[1] 595.7516

```
sd(banco$minutos_deslocamento)
```

[1] 24.40802

1.2.2 Variância e Desvio-Padrão

O cálculo da **variância** (`var`) e do **desvio-padrão** (`sd`) seguem a mesma lógica das funções anteriores.

```
var(banco$minutos_deslocamento)
```

[1] 595.7516

```r
sd(banco$minutos_deslocamento)
```

[1] 24.40802

```r
var(banco$livros_lidos)
```

[1] 72.18301

```r
sd(banco$livros_lidos)
```

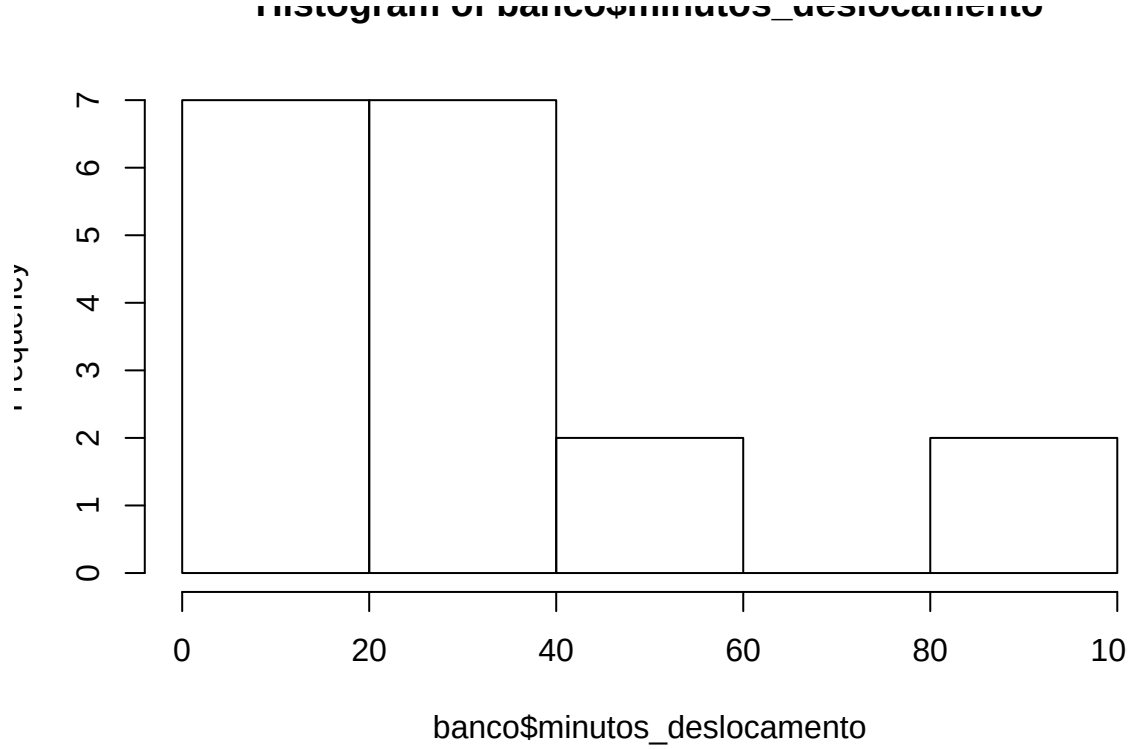
[1] 8.496058

1.2.3 Representações gráficas

Tradicionalmente, as representações gráficas de variáveis contínuas são feitas por (1) histogramas; ou (2) gráficos de densidade de *kernel*.

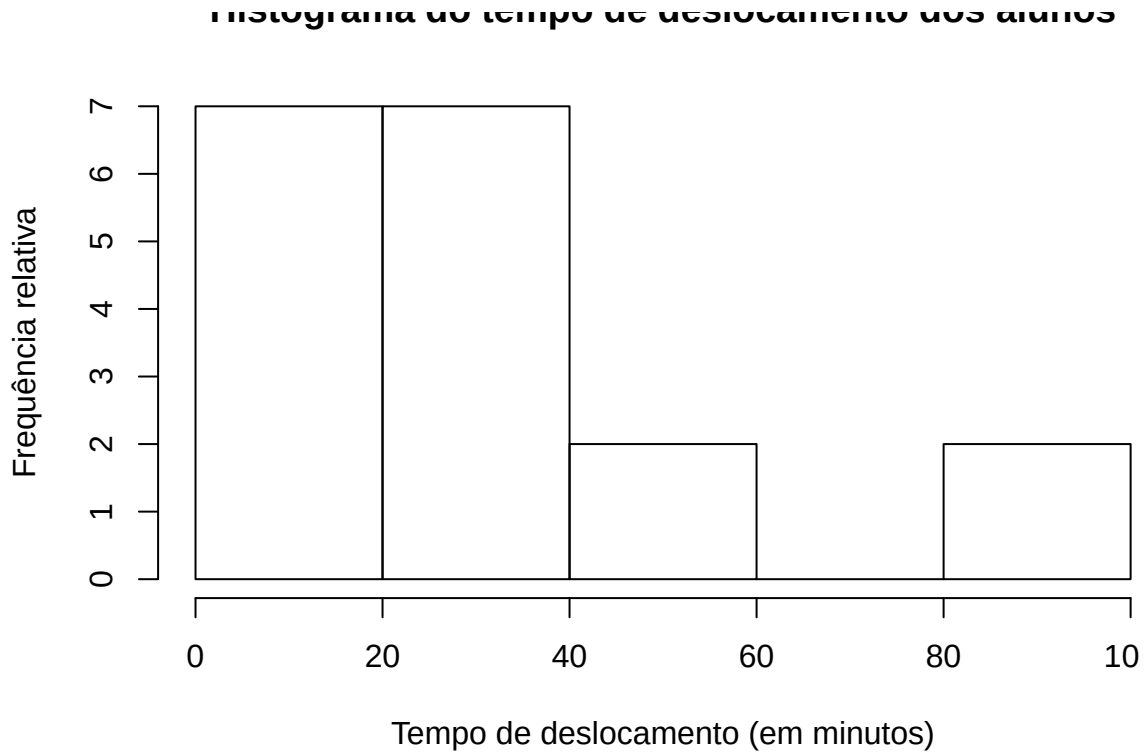
Histogramas apresentam a distribuição de uma variável segundo intervalos (pré-definidos) desta e a frequência relativa de ocorrências em cada intervalo. A função para plotar um histograma é a `hist`.

```r
hist(banco$minutos_deslocamento)
```



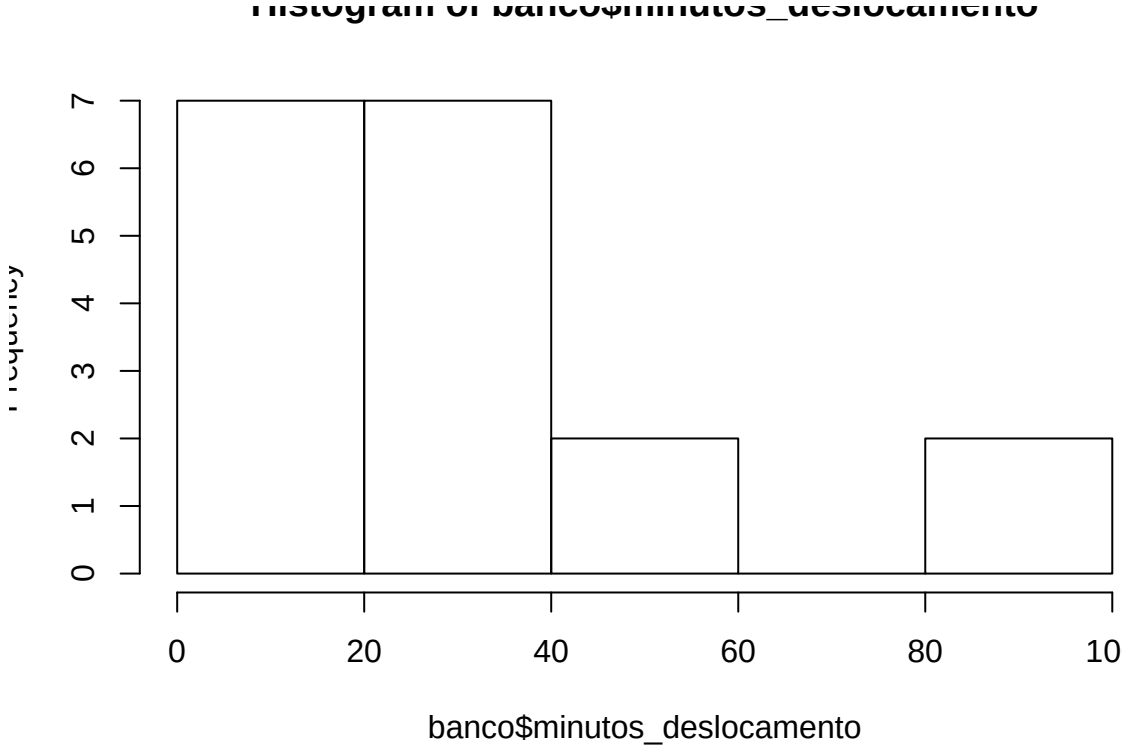
O padrão desta função utiliza uma fórmula (de Sturges) para definir o número de intervalos. Porém, podemos definir os nossos próprios intervalos, de acordo com o que julgarmos que mais ajudaria a visualização desta distribuição (com o argumento `breaks`).

```r
hist(banco$minutos_deslocamento,
     breaks = 6,
     main = "Histograma do tempo de deslocamento dos alunos",
     xlab = "Tempo de deslocamento (em minutos)",
     ylab = "Frequência relativa")
```



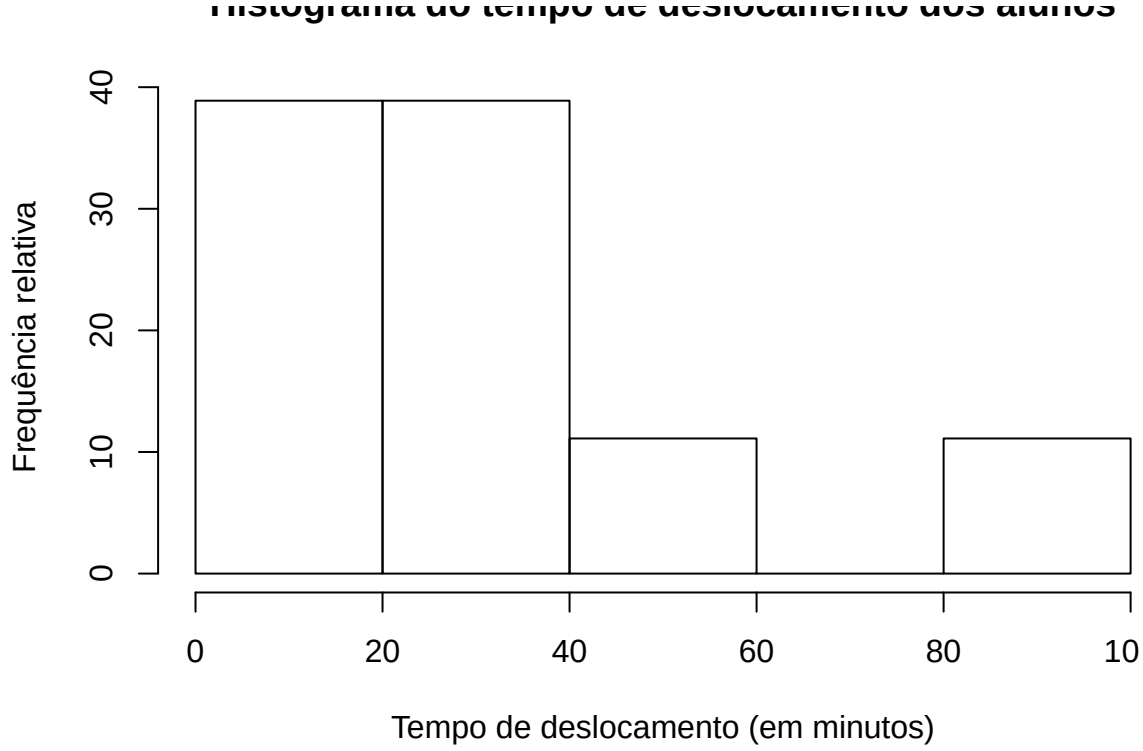
Além disso, percebam também que o padrão da função retorna a frequência *absoluta* e não a relativa. Para conseguir produzir um histograma com a frequência relativa, precisamos fazer alguns ajustes (como o do bloco de código abaixo, onde transformamos a densidade em percentuais).

```r
histograma_minutos <- hist(banco$minutos_deslocamento,breaks = 6)
```



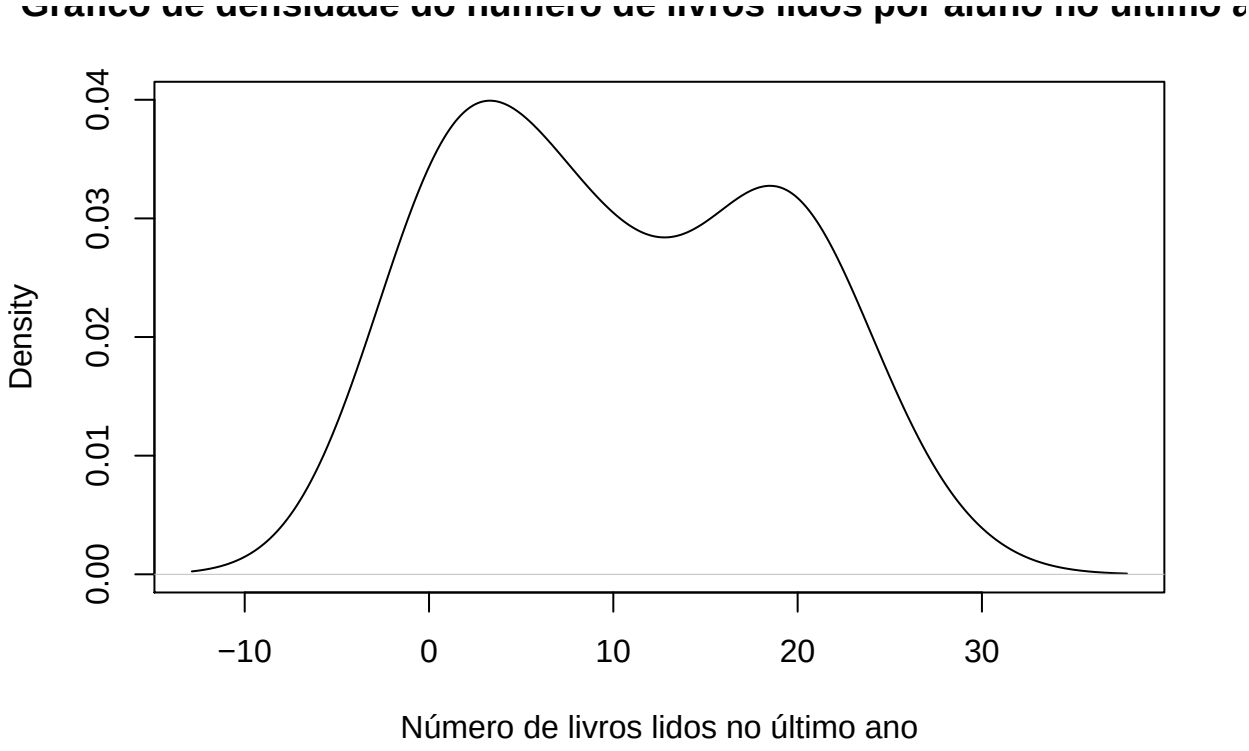

```
histograma_minutos$density <- histograma_minutos$counts/sum(histograma_minutos$counts)*100
```

```
plot(histograma_minutos, freq = F,
     main = "Histograma do tempo de deslocamento dos alunos",
     xlab = "Tempo de deslocamento (em minutos)",
     ylab = "Frequência relativa")
```



Por fim, temos o gráfico de densidade de *kernel*, amplamente utilizado na academia (ainda que com alguma dificuldade de compreensão do público em geral). Para produzi-lo, calculamos a densidade da variável desejada, plotando-a num gráfico.

```
densidade_livros <- density(banco$livros_lidos)
plot(densidade_livros,
     main = "Gráfico de densidade do número de livros lidos por aluno no último ano",
     xlab = "Número de livros lidos no último ano")
```



1.3 Resumo de estatísticas descritivas

Existem também funções que trazem um **resumo das nossas estatísticas descritivas**, gerando um output com várias das estatísticas referidas acima.

A função `summary`, do R base, nos retorna o mínimo, 1o quartil, mediana, média, 3o quartil e máximo da distribuição de uma variável.

```
summary(banco$minutos_deslocamento)
```

Min. 1st Qu. Median Mean 3rd Qu. Max. 5.00 16.25 30.00 33.89 40.00 90.00

Ela também apresenta um resumo de todas as variáveis do banco:

```
summary(banco)
```

```
                    area_atuacao     idade
```

Ciência Política :13 Min. :20.00
Direito / Ciências do Estado : 1 1st Qu.:25.00
Outros : 2 Median :27.50
Sociologia ou Ciências Sociais: 2 Mean :28.33
3rd Qu.:30.00
Max. :40.00

```
                          time   minutos_deslocamento
```

Atlético :2 Min. : 5.00
Cruzeiro :5 1st Qu.:16.25
Não gosto / sou indiferente a futebol:3 Median :30.00
Outro :8 Mean :33.89
3rd Qu.:40.00
Max. :90.00

meio_transporte regional livros_lidos
A pé :2 Barreiro :1 Min. : 0.00
Carro :6 Centro-Sul:1 1st Qu.: 3.25
Ônibus:8 Contagem :1 Median :10.00
Outro :2 Leste :3 Mean :10.22
Nordeste :2 3rd Qu.:19.00
Noroeste :1 Max. :25.00
Pampulha :9

Uma função semelhante, porém mais completa, é a `skim` do pacote `skimr`.

```
#install.packages("skimr")
library(skimr)
#skim(banco$minutos_deslocamento)
```

Outra opção é a função `describe` do pacote `describer`. Ela faz um resumo das estatísticas descritivas de um banco de dados. Vamos testá-la com nosso `banco` completo.

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

describe(banco)
```

```
.column_name .column_class .column_type .count_elements
```

1 area_atuacao factor integer 18 2 idade integer integer 18 3 time factor integer 18 4 minutos_deslocamento integer integer 18 5 meio_transporte factor integer 18 6 regional factor integer 18 7 livros_lidos integer integer 18 .mean_value .sd_value .q0_value .q25_value .q50_value .q75_value 1 NA NA Ciência Política NA NA NA 2 28.33333 4.862824 20 25.00 27.5 30 3 NA NA Atlético NA NA NA 4 33.88889 24.408024 5 16.25 30.0 40 5 NA NA A pé NA NA NA 6 NA NA Barreiro NA NA NA 7 10.22222 8.496058 0 3.25 10.0 19 .q100_value 1 Sociologia ou Ciências Sociais 2 40 3 Outro 4 90 5 Outro 6 Pampulha 7 25

Ele ficou com muita informação. Por que não selecionamos somente as variáveis que se referem aos minutos de deslocamento?

```
describe(banco$minutos_deslocamento)
```

.count_elements .mean_value .sd_value .q0_value .q25_value .q50_value 1 18 33.88889 24.40802 5 16.25 30 .q75_value .q100_value 1 40 90

Se quisermos, podemos renomear as colunas desse output, assim como fizemos com a `skim`. Tente fazer isso!

1.4 Tabelas de Frequência

Tabelas de frequência com contagem simples são facilmente realizadas com funções do R base:

```r
table(banco$time)                    # frequencia absoluta
```

```
                        Atlético
                               2
                        Cruzeiro
                               5
```

Não gosto / sou indiferente a futebol 3 Outro 8

```r
prop.table(table(banco$time)) * 100 # frequencia relativa
```

```
                        Atlético
                        11.11111
                        Cruzeiro
                        27.77778
```

Não gosto / sou indiferente a futebol 16.66667 Outro 44.44444

```r
table(banco$area_atuacao)
```

```
          Ciência Política   Direito / Ciências do Estado
                        13                              1
                    Outros Sociologia ou Ciências Sociais
                         2                              2
```

```r
prop.table(table(banco$area_atuacao)) * 100
```

```
          Ciência Política   Direito / Ciências do Estado
                 72.222222                       5.555556
                    Outros Sociologia ou Ciências Sociais
                11.111111                      11.111111
```

Podemos unir essas duas tabelas com a frequência relativa em uma única tabela:

```r
time_absoluta <- table(banco$time)
time_relativa <- prop.table(table(banco$time)) * 100

tabela_time <- rbind(time_absoluta, time_relativa)
row.names(tabela_time) <- c("Frequência Absoluta", "Frequência Relativa")
```

E para incluir a linha/coluna com os totais, usamos a função `addmargins`.

```r
atuacao_absoluta <- addmargins(table(banco$area_atuacao))
atuacao_relativa <- prop.table(table(banco$area_atuacao)) * 100

tabela_atuacao <- rbind(atuacao_absoluta, atuacao_relativa)
row.names(tabela_atuacao) <- c("Frequência Absoluta", "Frequência Relativa")

tabela_atuacao
```

```
                Ciência Política Direito / Ciências do Estado   Outros
```

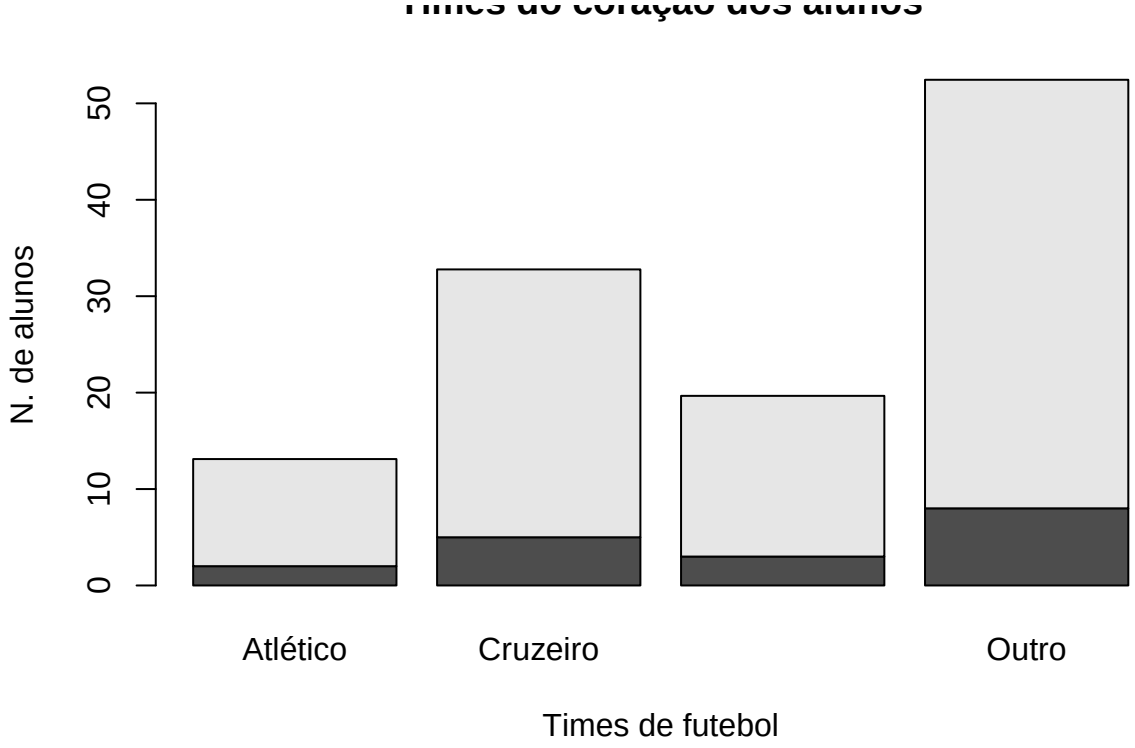
Frequência Absoluta 13.00000 1.000000 2.00000 Frequência Relativa 72.22222 5.555556 11.11111 Sociologia ou Ciências Sociais Sum Frequência Absoluta 2.00000 18.00000 Frequência Relativa 11.11111 72.22222

1.4.1 Representação gráfica (gráficos de barras)

Uma maneira simples (talvez óbvia) de representar variáveis categóricas é por meio de gráficos de barra, produzidos com a função `barplot()`. Para fazê-los, primeiro precisamos criar uma tabela de frequência da variável desejada e, em seguida, aplicar a função `barplot` a essa tabela.

```
tabela_times <- table(banco$times)


barplot(tabela_time,main = "Times do coração dos alunos",
        xlab = "Times de futebol", ylab = "N. de alunos")
```



Resumo do conteúdo trabalhado:

- Neste tutorial, aprendemos a calcular estatísticas descritivas das nossas variáveis contínuas e categóricas.
- Para realizar esses cálculos de variáveis contínuas, usamos as funções:
  - `median` (mediana);
  - `min` (mínimo);
  - `max` (máximo);
  - `range` (mínimo e máximo);
  - `quantile` (quantis);
  - `IQR` (intervalo interquartil);
  - `mean` (média);
  - `var` (variância);
  - `sd` (desvio-padrão).

- Vimos ainda três funções diferentes que retornam um conjunto de estatísticas descritivas: a `summary`, a `skim` (do pacote `skimr`) e a `describe` (do pacote `describer`).
- Para descrever nossas variáveis categóricas, conhecemos as funções `table` e `prop.table`, usadas para produzir tabelas de frequencia.