

# Estatística descritiva para *Data Science*

## 0.6 - Aula Prática: Visualização de dados através de gráficos

Profa. Dra. Amanda Buosi Gazon Milani

2024-07-20

### Conjunto de dados - Nascidos Vivos 2024 (DataSUS)

O conjunto de dados que será utilizado nesta disciplina foi obtido no site do OpenDataSUS, foi tratado e consiste da base de informações sobre nascidos vivos 2024 (parcial).

Inicialmente vamos importar (carregar) os dados no R, utilizando os códigos a seguir.

```
setwd <- "C:\\Users\\AmandaBGM\\Google Drive\\UFG\\Especialização_FEN_IME\\2024\\Scripts"

# 2 opções de importação (.csv com separador ponto-e-vírgula):

# dados <- read.csv2(file = "Dataframe_AulaAmanda.csv", header = TRUE)
# ou:
dados <- read.csv(file = "Dataframe_AulaAmanda.csv", sep=';', header = TRUE)

head(dados)
```

##	LOCNASC	IDADEMAE	ESTCIVMAE	QTDFILVIVO	QTDFILMORT	GESTACAO			
## 1	Hospital	24	Solteira	1	0	37 a 41 semanas			
## 2	Hospital	29	Casada	0	0	37 a 41 semanas			
## 3	Hospital	20	União consensual	0	0	37 a 41 semanas			
## 4	Hospital	40	Solteira	4	1	37 a 41 semanas			
## 5	Hospital	27	Casada	3	0	32 a 36 semanas			
## 6	Hospital	19	Solteira	0	0	37 a 41 semanas			
##	GRAVIDEZ	PARTO	CONSULTAS	DTNASC	SEXO	APGAR1	APGAR5	RACACOR	PESO
## 1	Única	Cesáreo	de 1 a 3	2024-02-14	Masculino	8	9	Parda	3120
## 2	Única	Cesáreo	7 e mais	2024-04-17	Masculino	8	9	Parda	3564
## 3	Única	Vaginal	7 e mais	2024-01-01	Masculino	8	9	Branca	3240
## 4	Única	Cesáreo	7 e mais	2024-01-01	Masculino	9	9	Parda	3960
## 5	Única	Vaginal	7 e mais	2024-01-01	Masculino	8	9	Parda	3610
## 6	Única	Cesáreo	7 e mais	2024-01-01	Masculino	9	9	Parda	3724
##	IDANOMAL	CODUFNATU	ESCMAC2010	RACACORMAE	QTDGESTANT				
## 1	Não	RO Fundamental II (5ª a 8ª série)		Parda	1				
## 2	Não	RO Superior completo		Parda	0				
## 3	Não	RO Médio (antigo 2º grau)		Branca	0				
## 4	Não	AC Superior completo		Parda	5				
## 5	Não	RO Médio (antigo 2º grau)		Parda	3				
## 6	Não	RO Médio (antigo 2º grau)		Parda	0				
##	QTDPARTNOR	QTDPARTCES	IDADEPAI	SEMAGESTAC	CONSPRENAT	STTRABPART	STCESPARTO		
## 1	0	1	NA	38	2	Não	Não		
## 2	0	0	41	39	8	Não	Sim		

```
## 3      0      0      NA      38      10      Não Não se aplica
## 4      1      3      NA      38      7      Não      Não
## 5      3      0      NA      36      10      Não Não se aplica
## 6      0      0      NA      41      8      Não      Não
##          TPNASCASSI MES.NASC SEMAGESTAC_cat CONSPRENAT_cat
## 1          Médico      Fev 37 a 41 semanas      1 a 3
## 2          Médico      Abr 37 a 41 semanas      7 ou mais
## 3 Enfermeira/obstetriz      Jan 37 a 41 semanas      7 ou mais
## 4          Médico      Jan 37 a 41 semanas      7 ou mais
## 5 Enfermeira/obstetriz      Jan 32 a 36 semanas      7 ou mais
## 6          Médico      Jan 37 a 41 semanas      7 ou mais
```

```
dim(dados)
```

```
## [1] 779927      31
```

## Construção de gráficos

A linguagem R é uma ferramenta muito útil e prática para construção de gráficos estatísticos. Vejamos a seguir como construir os gráficos vistos na aula teórica para as variáveis do conjunto de dados de nascidos vivos de 2024. Os gráficos são construídos a partir de uma tabela e, por esse motivo, salvo específicas exceções, a lógica das funções será construir uma tabela e aplicar a função que constrói o gráfico a esta tabela dos dados.

### Gráfico em barras

Como vimos, o gráfico em barras pode ser utilizado para variáveis qualitativas e ele pode ser construído no R utilizando a função `barplot(...)` juntamente com a opção `(horiz = TRUE)` para habilitar a opção de barras horizontais.

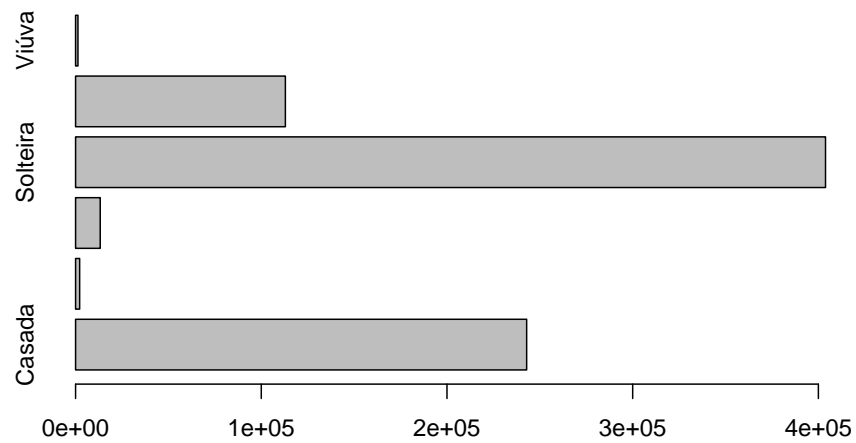
```
# Tabela de frequência absoluta da variável:
```

```
tab.estcivmae1 <- table(dados$ESTCIVMAE)
tab.estcivmae1
```

```
##
##          Casada          Ignorado Separada judicialmente
##          242908          2192          13273
##          Solteira      União consensual          Viúva
##          403848          112982          1286
```

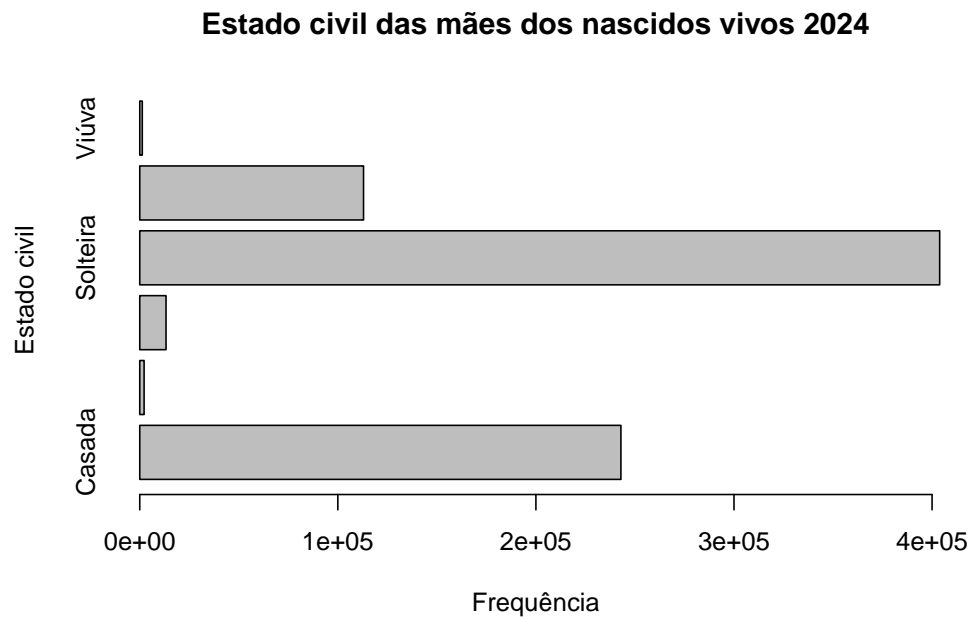
```
# Gráfico em barras:
```

```
barplot(tab.estcivmae1, horiz = TRUE)
```



O gráfico básico sai sem rótulos nos eixos, sem títulos, e nós devemos adicionar essas informações para que o gráfico fique completo.

```
# Gráfico em barras com frequência absoluta:
barplot(tab.estcivmae1,
        horiz = TRUE,
        main="Estado civil das mães dos nascidos vivos 2024", ## título do gráfico
        ylab="Estado civil", ## rótulo do eixo y
        xlab="Frequência") ## rótulo do eixo x
```

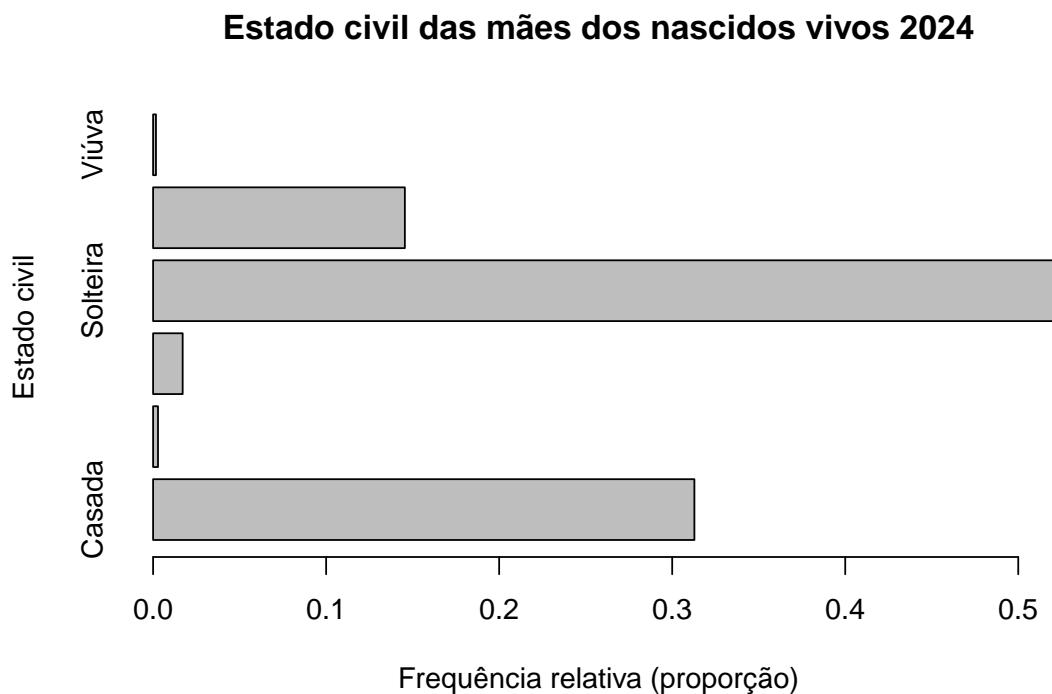


Note que utilizamos a frequência absoluta para construir o gráfico em barras, mas poderíamos adotar uma das frequências relativas (proporção ou porcentagem), bastando para isso, utilizar a tabela de frequência relativa no lugar da tabela de frequência absoluta, ao aplicar a função `barplot(...)`. Vejamos como ficaria o exemplo adotando a frequência relativa proporção:

```
# Tabela de frequência relativa da variável (proporção):
tab.estcivmae2 <- prop.table(table(dados$ESTCIVMAE))
tab.estcivmae2

##
##              Casada              Ignorado Separada judicialmente
##      0.312828643      0.002822963      0.017093610
##      Solteira      União consensual      Viúva
##      0.520094940      0.145503671      0.001656173

# Gráfico em barras com frequência relativa (proporção):
barplot(tab.estcivmae2,
        horiz = TRUE,
        main="Estado civil das mães dos nascidos vivos 2024",
        ylab="Estado civil",
        xlab="Frequência relativa (proporção)")
## para gráfico de barras
## título do gráfico
## rótulo do eixo y
## rótulo do eixo x
```

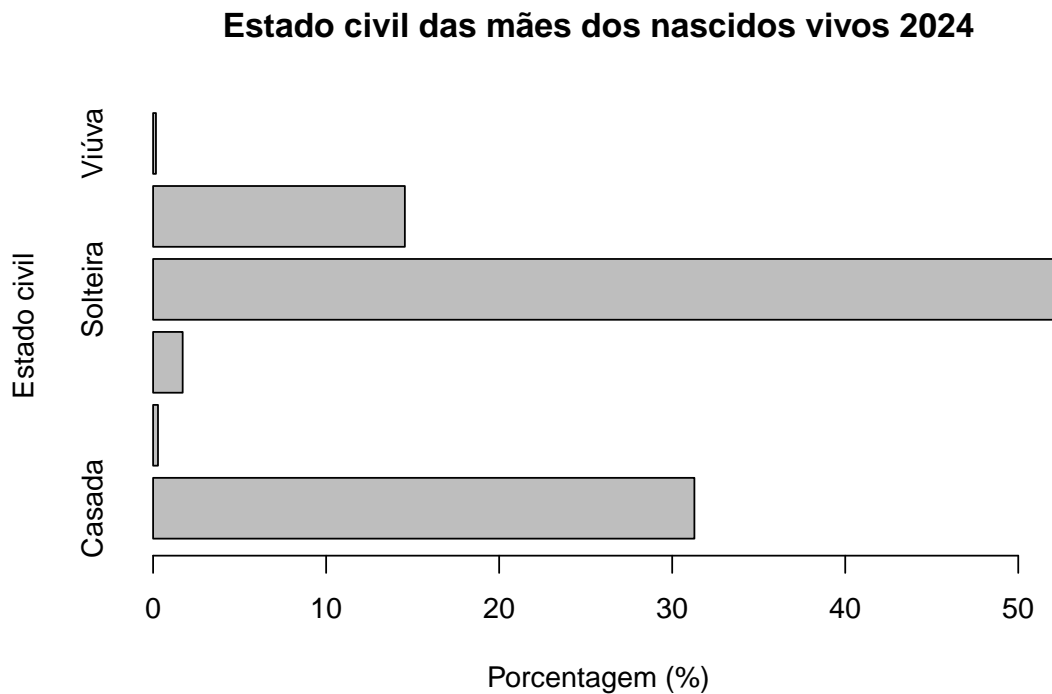


e adotando a porcentagem:

```
# Tabela de frequência relativa da variável (porcentagem):
tab.estcivmae3 <- 100*prop.table(table(dados$ESTCIVMAE))
tab.estcivmae3
```

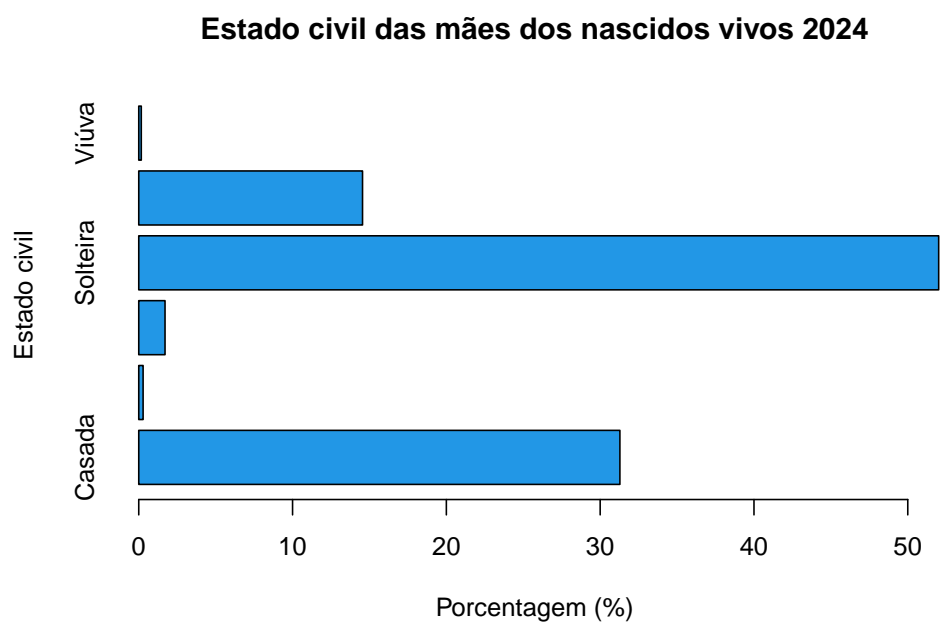
```
##
##          Casada          Ignorado Separada judicialmente
##      31.2828643      0.2822963      1.7093610
##      Solteira      União consensual      Viúva
##      52.0094940      14.5503671      0.1656173

# Gráfico em barras com frequência relativa (porcentagem):
barplot(tab.estcivmae3,
        horiz = TRUE,
        main="Estado civil das mães dos nascidos vivos 2024",
        ylab="Estado civil",
        xlab="Porcentagem (%)")
```



Vejamos a seguir que outros parâmetros podem ser configurados, como a cor das barras, os limites do eixo x, dentre outros.

```
# Ajustando a cor:
barplot(tab.estcivmae3,
        horiz = TRUE,
        main="Estado civil das mães dos nascidos vivos 2024",
        ylab="Estado civil",
        xlab="Porcentagem (%)",
        col = 4)
```



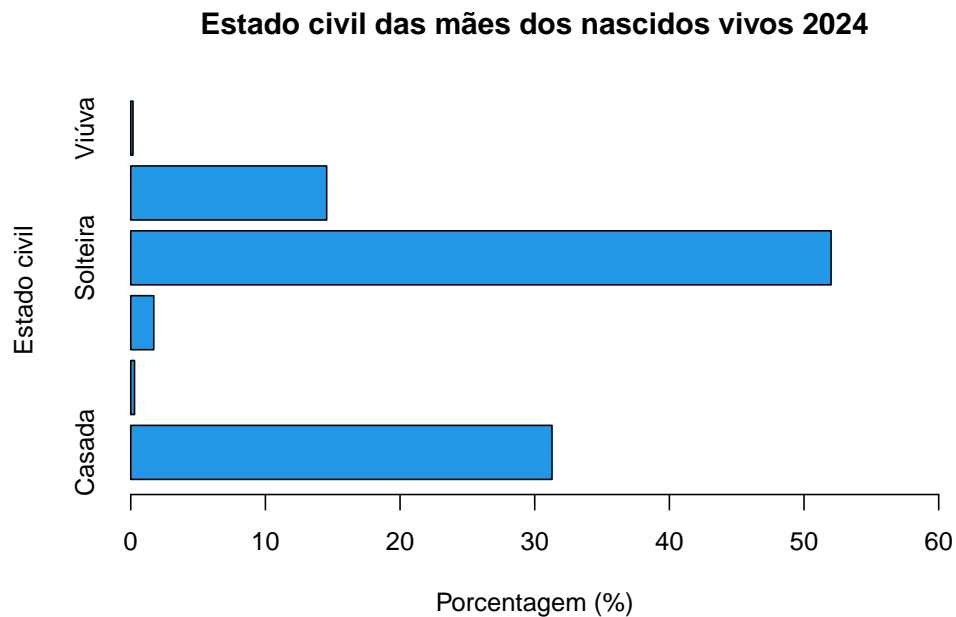
```
tab.estcivmae3 # vamos aumentar até 60% o eixo x para "abraçar" melhor todas as porcentagens
```

```
##
##          Casada          Ignorado Separada judicialmente
##      31.2828643      0.2822963      1.7093610
##      Solteira      União consensual      Viúva
##      52.0094940      14.5503671      0.1656173
```

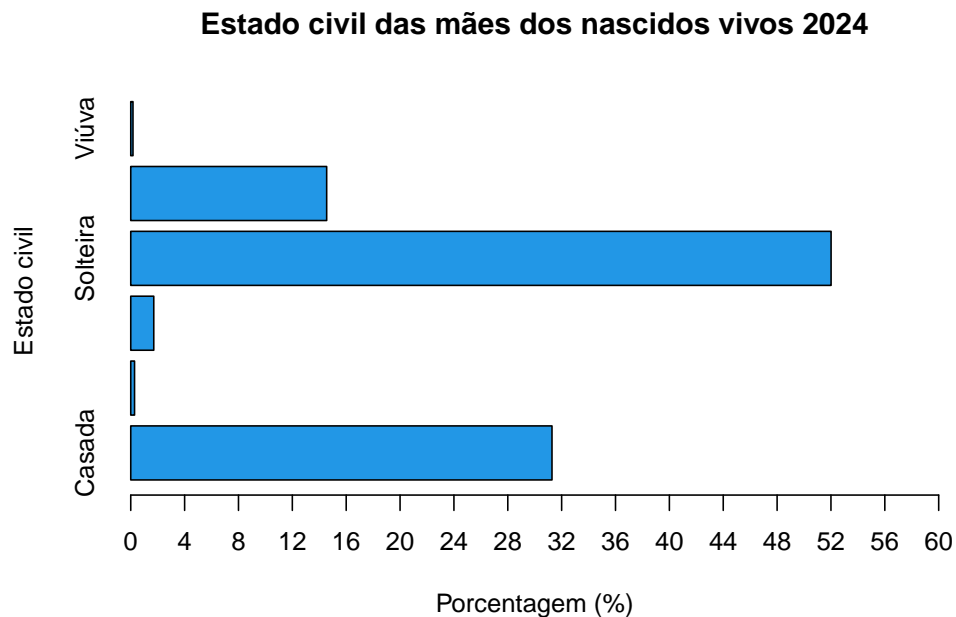
```
# Ajustando os limites do eixo x:
```

```
barplot(tab.estcivmae3,
        horiz = TRUE,
        main="Estado civil das mães dos nascidos vivos 2024",
        ylab="Estado civil",
        xlab="Porcentagem (%)",
        col = 4,
        xlim = c(0,60))
```

## para gráfico de barras  
## título do gráfico  
## rótulo do eixo y  
## rótulo do eixo x  
## cor das barras  
## limites do eixo x

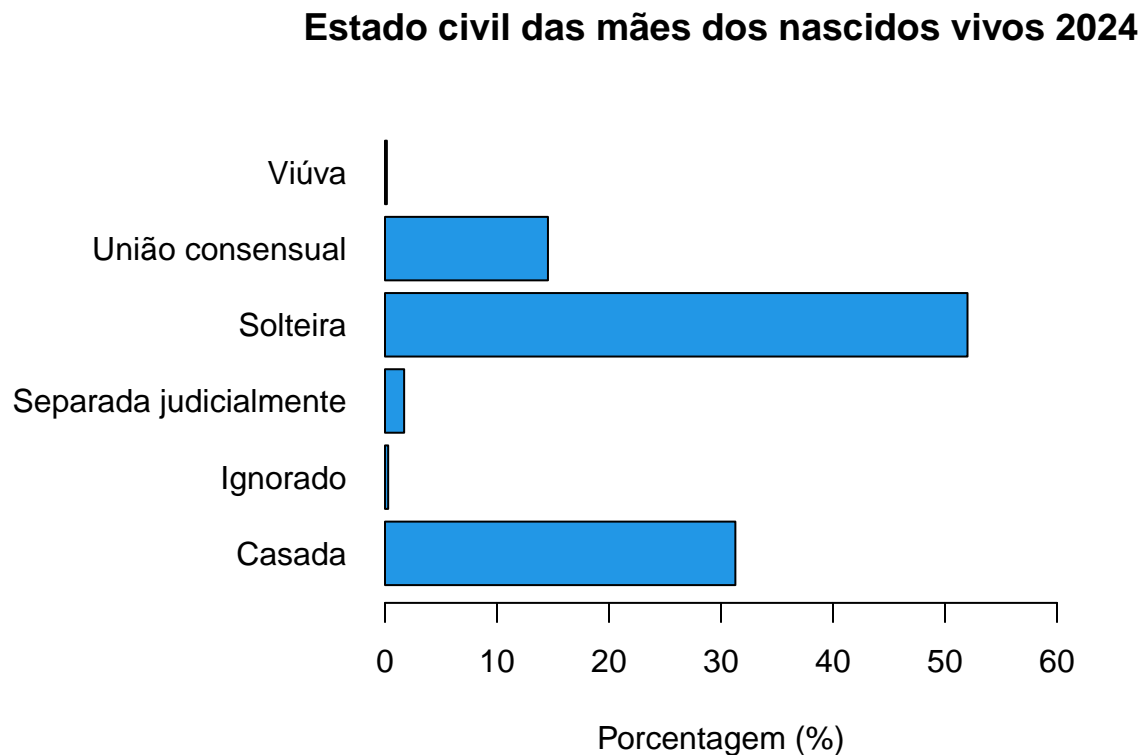


```
# Ajustando os limites do eixo x e mudando os valores impressos no eixo:
barplot(tab.estcivmae3,
        horiz = TRUE,                                ## para gráfico de barras
        main="Estado civil das mães dos nascidos vivos 2024",  ## título do gráfico
        ylab="Estado civil",                                ## rótulo do eixo y
        xlab="Porcentagem (%)",                            ## rótulo do eixo x
        col = 4,                                           ## cor das barras
        xlim = c(0,60),                                   ## limites do eixo x
        xaxp = c(0,60,15))                               ## controla eixo x : c(início, fim, qtde de intervalos)
```



Um dos principais motivos de se adotar as barras horizontais, além de fácil leitura do eixo e bom enquadramento no texto, é a presença de rótulos extensos. Note que nos gráficos anteriores, nem todos os rótulos estavam sendo impressos, e para resolver este problema, podemos rotacionar estes rótulos, o que inclusive deixará a leitura dos mesmos mais agradável, uma vez que ficará na posição horizontal, que é a orientação padrão dos textos.

```
# Rotacionando os rótulos:
par(mai=c(1,2,1,1)) # aumenta a margem esquerda : c(baixo, esquerda, cima, direita)
barplot(tab.estcivmae3,
        horiz = TRUE, ## para gráfico de barras
        las = 1,      ## para rótulos horizontais
        main="Estado civil das mães dos nascidos vivos 2024", ## título do gráfico
        ylab=" ",      ## rótulo do eixo y
        xlab="Porcentagem (%)", ## rótulo do eixo x
        col = 4,       ## cor das barras
        xlim = c(0,60)) ## limites do eixo x
```



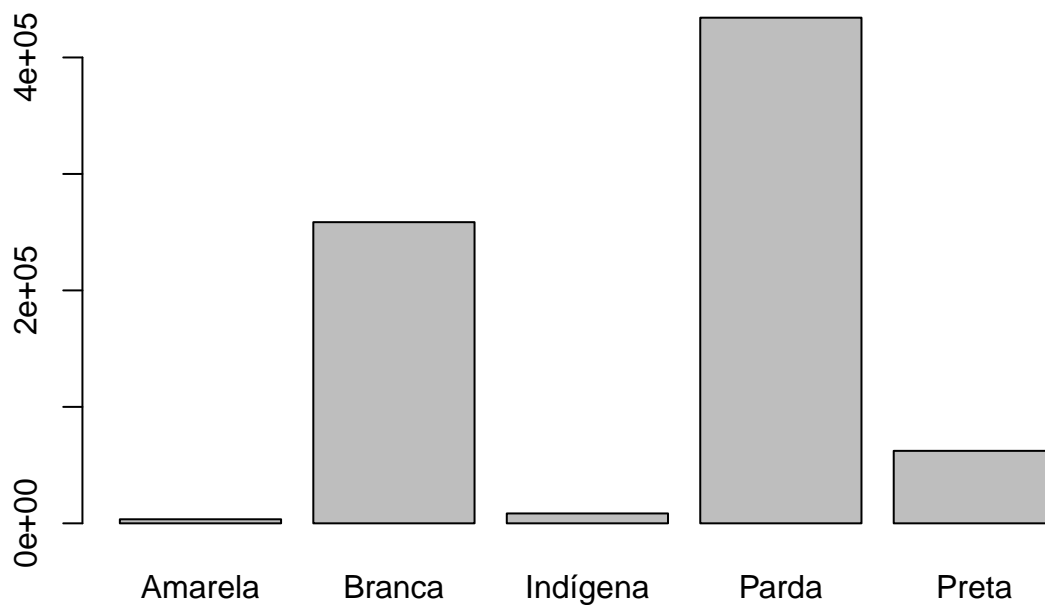


## Gráfico em colunas

Como vimos, o gráfico em barras pode ser utilizado para variáveis qualitativas e para variáveis quantitativas discretas (com os devidos cuidados). Ele pode ser construído no R utilizando a função `barplot(...)`.

### Variável qualitativa

```
# Tabela de frequência absoluta da variável :  
tab.racacor1 <- table(dados$RACACOR)  
tab.racacor1  
  
##  
## Amarela Branca Indígena Parda Preta  
## 3475 258580 8501 434158 62226  
  
# Gráfico em colunas utilizando frequência absoluta:  
barplot(tab.racacor1)
```



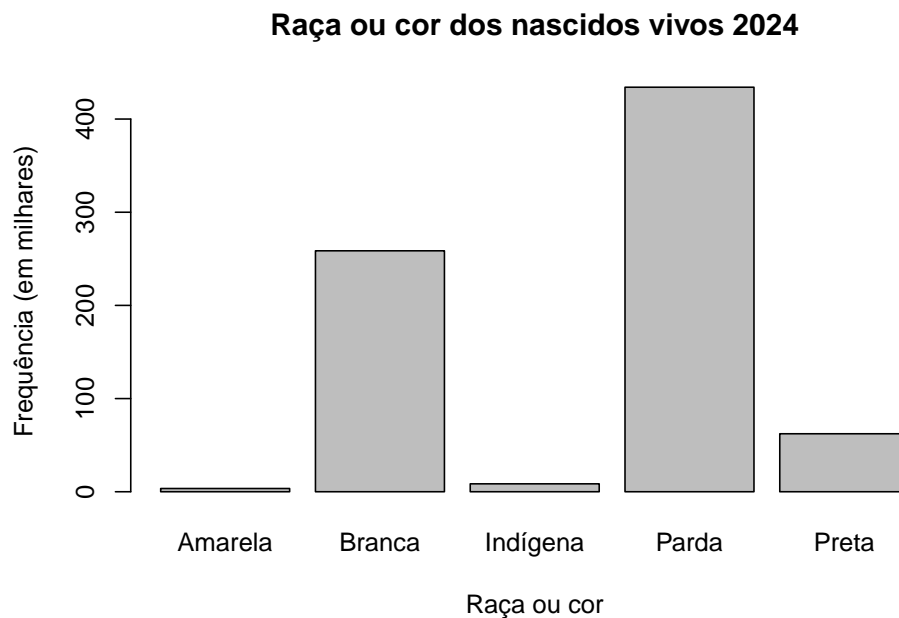
Novamente, o gráfico em barras básico vem sem rótulos nos eixos, sem título e precisamos adicioná-los. Além disso, como o conjunto de dados é relativamente grande, as frequências absolutas ficam nas casas dos milhares e o R utiliza notação científica para apresentar os valores das frequências absolutas no gráfico. Uma alternativa é utilizar outra unidade de medida para a frequência absoluta, por exemplo, a de milhares. Para

isso, fazemos a tabela nesta unidade de medida, dividindo as frequências absolutas por 1000 e em seguida executamos o gráfico em colunas.

```
# Tabela de frequência absoluta da variável (em milhares) :
tab.racacor1.2 <- table(dados$RACACOR)/1000
tab.racacor1.2

##
##  Amarela  Branca Indígena   Parda   Preta
##    3.475  258.580    8.501  434.158  62.226

# Gráfico em colunas utilizando frequência absoluta (em milhares):
barplot(tab.racacor1.2,
        main="Raça ou cor dos nascidos vivos 2024", ## título do gráfico
        xlab="Raça ou cor",                        ## rótulo do eixo x
        ylab="Frequência (em milhares)")            ## rótulo do eixo y
```

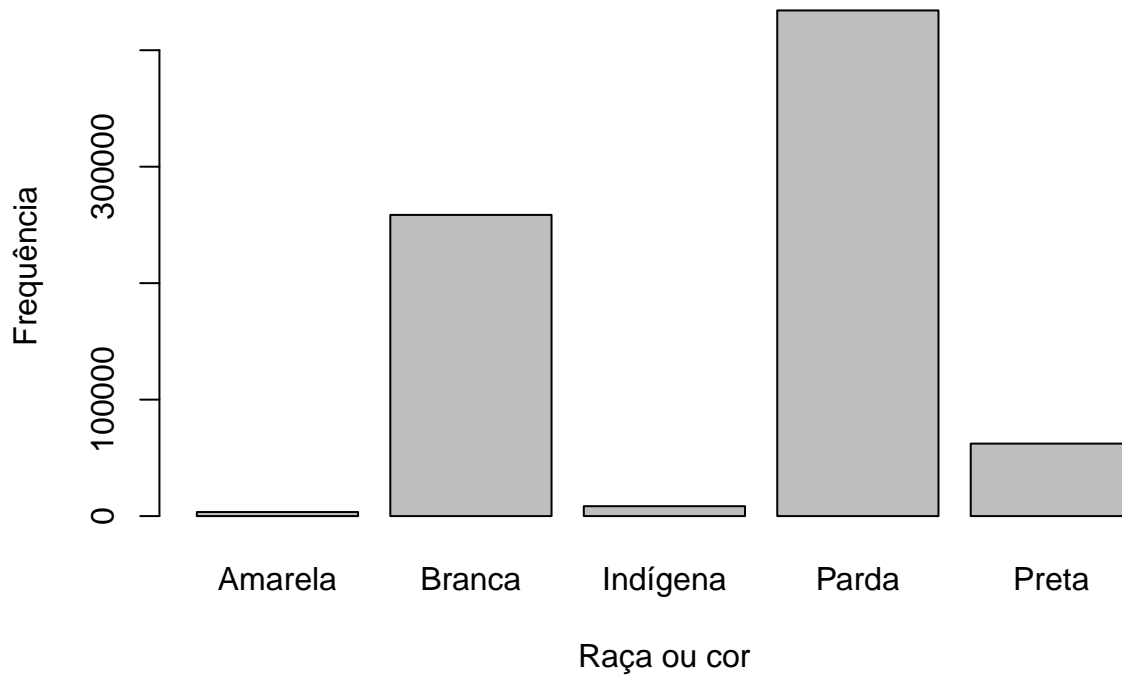


Outra maneira é ajustar as opções do R para que não apareça notação científica:

```
# ajustando para valor grande, para que os números não fiquem em notação científica:
options(scipen = 999)

# Gráfico em colunas utilizando frequência absoluta:
barplot(tab.racacor1,
        main="Raça ou cor dos nascidos vivos 2024", ## título do gráfico
        xlab="Raça ou cor",                        ## rótulo do eixo x
        ylab="Frequência")                          ## rótulo do eixo y
```

## Raça ou cor dos nascidos vivos 2024



```
# para voltar as opções ao normal:  
# options(scipen = 0)
```

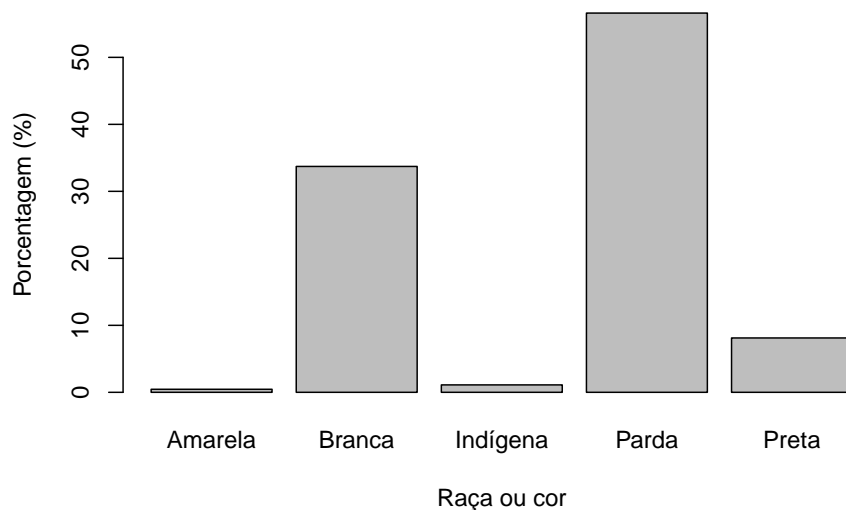
Ou podemos adotar a frequência relativa (proporção ou porcentagem) para executar o gráfico em colunas. Por exemplo, para a frequência relativa porcentagem:

```
# Tabela de frequência relativa da variável (porcentagem):  
tab.racacor3 <- 100*prop.table(table(dados$RACACOR))  
tab.racacor3
```

```
##  
##   Amarela   Branca  Indígena   Parda   Preta  
## 0.4530993 33.7158057 1.1084309 56.6091220 8.1135421
```

```
# Gráfico em colunas para a frequência relativa (porcentagem):  
barplot(tab.racacor3,  
        main="Raça ou cor dos nascidos vivos 2024", ## título do gráfico  
        xlab="Raça ou cor",                        ## rótulo do eixo x  
        ylab="Porcentagem (%)")                    ## rótulo do eixo y
```

**Raça ou cor dos nascidos vivos 2024**



```
tab.racacor3 # vamos aumentar até 60% o eixo x para "abraçar" melhor todas as porcentagens
```

```
##
```

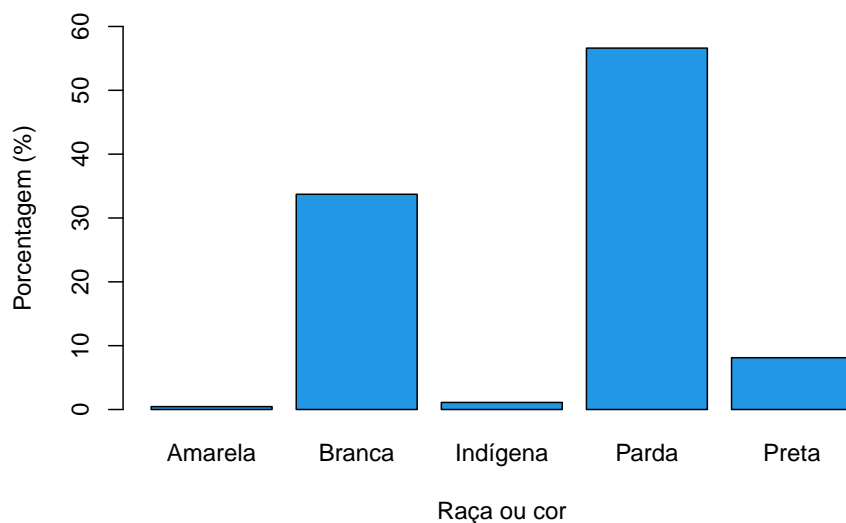
```
## Amarela Branca Indígena Parda Preta
```

```
## 0.4530993 33.7158057 1.1084309 56.6091220 8.1135421
```

```
# Gráfico em colunas alterando a cor das barras e os limites do eixo:
```

```
barplot(tab.racacor3,
  main="Raça ou cor dos nascidos vivos 2024", ## título do gráfico
  xlab="Raça ou cor", ## rótulo do eixo x
  ylab="Porcentagem (%)", ## rótulo do eixo y
  col = 4, ## cor das barras
  ylim = c(0,60)) ## limites do eixo y
```

**Raça ou cor dos nascidos vivos 2024**



## Variável quantitativa discreta

Para variáveis quantitativas discretas, podemos executar o gráfico em colunas, desde que respeitando o caráter numérico da variável, isto é, se a variável tiver “buracos”, no sentido de valores que não tiverem ocorrências, o ideal é que o gráfico em colunas também tenha essa representação. Veremos que o gráfico isso não ocorre automaticamente, porque em geral as tabelas não costumam vir com a frequência zerada para os valores em que não houve ocorrência. Sendo assim, as tabelas precisam ser adaptadas para que então o gráfico fique correto. Vamos mostrar primeiro como ficaria um gráfico feito sem o devido cuidado e, posteriormente, como adaptar a tabela e fazer o gráfico de forma correta.

```
# Tabela de frequência absoluta:
```

```
table(dados$QTDPARTCES)
```

```
##
```

```
##      0      1      2      3      4      5      6      7      9     10     11
## 529146 168930 49862 10778 1873  322   61   13    4   48   14
##    12    14    20    21    22    23    25    30    31    32    33
##     1     2    16     6     9     2     1     8     1     2     2
##    40    41    63    99
##     2     1     1     7
```

```
# Tabela de frequência relativa (porcentagem) - arredondado para 4 casas decimais:
```

```
tab.qtdpartces3 <- round(100*prop.table(table(dados$QTDPARTCES)), 4)
```

```
tab.qtdpartces3
```

```
##
```

```
##      0      1      2      3      4      5      6      7      9     10
## 69.5228 22.1952  6.5512  1.4161  0.2461  0.0423  0.0080  0.0017  0.0005  0.0063
##    11    12    14    20    21    22    23    25    30    31
##  0.0018  0.0001  0.0003  0.0021  0.0008  0.0012  0.0003  0.0001  0.0011  0.0001
##    32    33    40    41    63    99
##  0.0003  0.0003  0.0003  0.0001  0.0001  0.0009
```

```
barplot(tab.qtdpartces3,
```

```
  main="Quantidade de partos cesárea das mães dos nascidos vivos 2024", ## título do gráfico
```

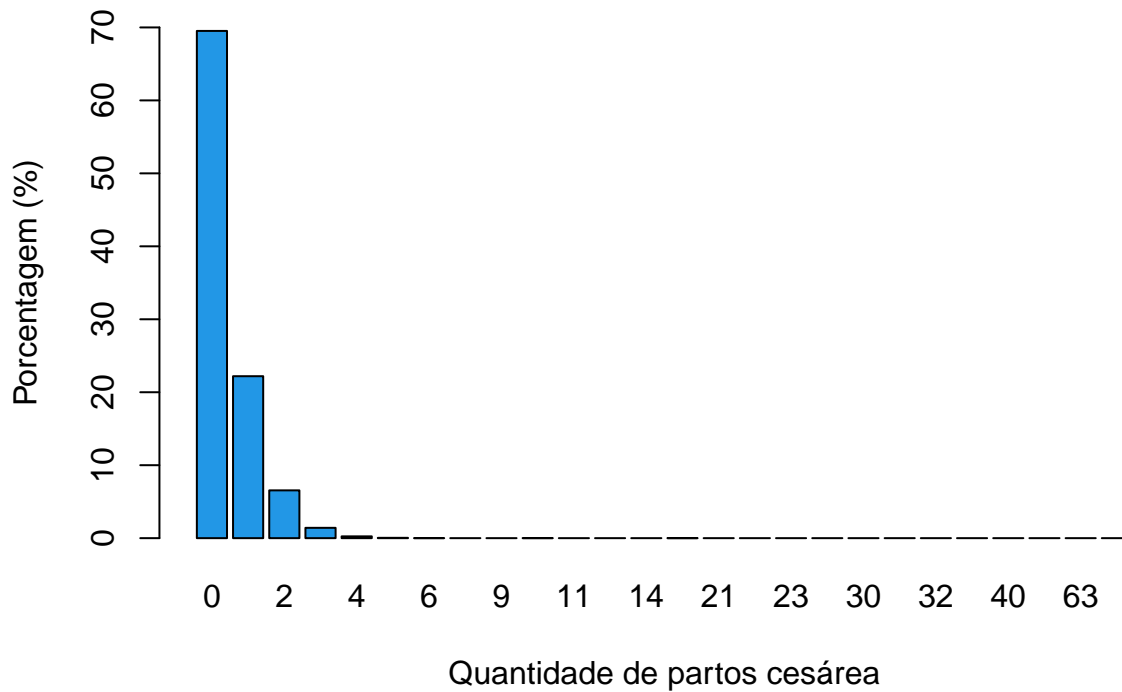
```
  xlab="Quantidade de partos cesárea", ## rótulo do eixo x
```

```
  ylab="Porcentagem (%)", ## rótulo do eixo y
```

```
  col = 4, ## cor das colunas
```

```
  ylim = c(0,70)) ## limites do eixo y
```

## Quantidade de partos cesárea das mães dos nascidos vivos 2024



Como já vimos na aula teórica, pelo tipo de variável (quantitativa), o ideal é que o eixo x seja respeitado, mesmo para valores que não apresentaram ocorrências no conjunto de dados. Nestas situações, até podemos usar o gráfico de colunas, mas precisamos adaptar (completar) a tabela de frequências, para que a função `barplot` execute o gráfico corretamente.

*#Para melhor visualização do processo, vamos considerar apenas os valores menores ou iguais a 10, # pois na base temos poucos valores maiores que 10 (mas deve ser feito para todos os valores):*

```
table(dados$QTDPARTCES)
```

```
##
##      0      1      2      3      4      5      6      7      9     10     11
## 529146 168930 49862 10778 1873  322   61   13    4   48   14
##      12     14     20     21     22     23     25     30     31     32     33
##      1      2     16      6      9      2      1      8      1      2      2
##     40     41     63     99
##      2      1      1      7
```

```
table(dados$QTDPARTCES>10)
```

```
##
## FALSE  TRUE
## 761037    75
```

```

# Criando a tabela de frequências
tab.qtdpartces <- table(dados$QTDPARTCES[dados$QTDPARTCES<=10])
tab.qtdpartces

##
##      0      1      2      3      4      5      6      7      9     10
## 529146 168930 49862 10778 1873  322   61   13   4    48

# Convertendo a tabela para um vetor para adicionar os valores sem ocorrências
frequencias <- as.vector(tab.qtdpartces)      #vetor com as frequências
frequencias

## [1] 529146 168930 49862 10778 1873  322   61   13   4    48
nomes <- as.numeric(names(tab.qtdpartces))    #vetor com os valores que a variável assume (capturados p
nomes

## [1] 0 1 2 3 4 5 6 7 9 10

# Incluindo os valores sem ocorrência com frequência zero
todos_nomes <- 0:10
todas_frequencias <- rep(0, length(todos_nomes))      #começa com todas as frequências nulas
todas_frequencias[todos_nomes %in% nomes] <- frequencias      #substitui as frequências existentes

# Criando uma tabela de frequências completa
tabela_completa <- setNames(todas_frequencias, todos_nomes)

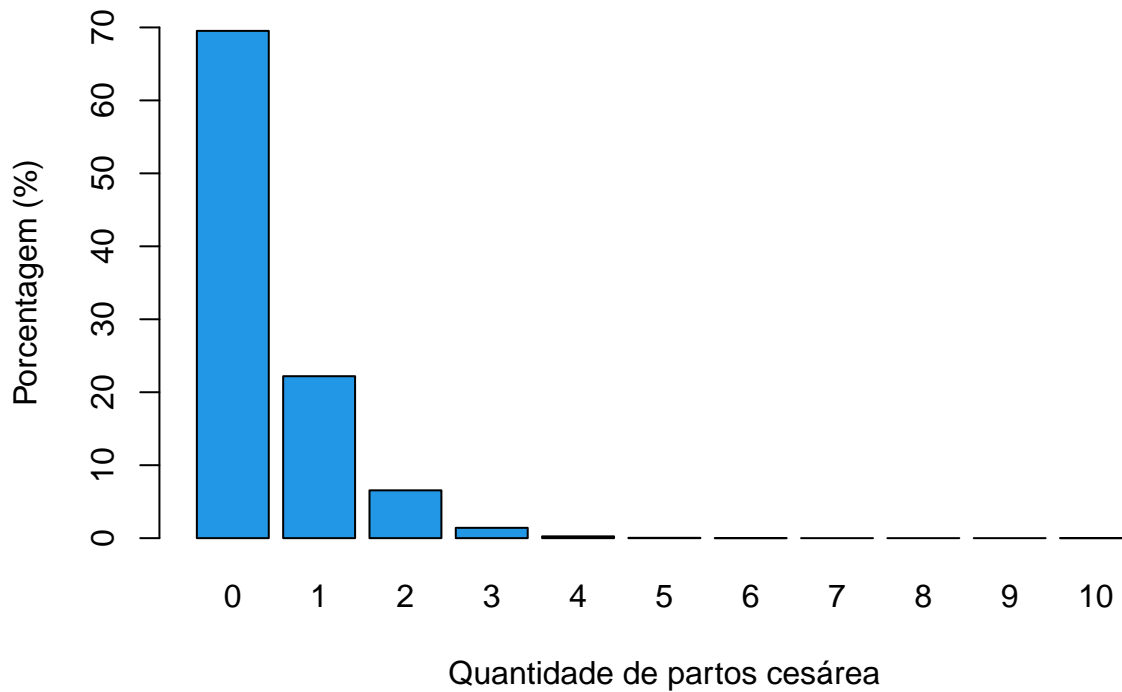
# Exibindo a tabela de frequências completa
print(tabela_completa)

##      0      1      2      3      4      5      6      7      8      9     10
## 529146 168930 49862 10778 1873  322   61   13   0    4    48

# Criando o gráfico de colunas
barplot(100*prop.table(tabela_completa),
      main="Quantidade de partos cesárea das mães dos nascidos vivos 2024",    ## título do gráfico
      xlab="Quantidade de partos cesárea",      ## rótulo do eixo x
      ylab="Porcentagem (%)",      ## rótulo do eixo y
      col = 4,      ## cor das colunas
      ylim = c(0,70))      ## limites do eixo y

```

## Quantidade de partos cesárea das mães dos nascidos vivos 2024



### Gráfico em setores

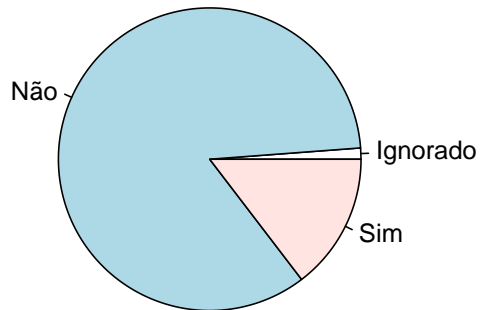
O gráfico de setores, também conhecido como gráfico de pizza, é um gráfico utilizado para variáveis qualitativas e pode ser obtido no R utilizando a função `pie(...)`. Tanto a frequência absoluta quanto a relativa (proporção ou porcentagem) podem ser adotadas para construir o gráfico.

```
# Tabela de frequência absoluta:  
tab.tpinduz <- table(dados$STTRABPART)  
tab.tpinduz
```

```
##  
## Ignorado      Não      Sim  
##      9211    650262    112850
```

```
#Gráfico de pizza:  
pie(tab.tpinduz)
```

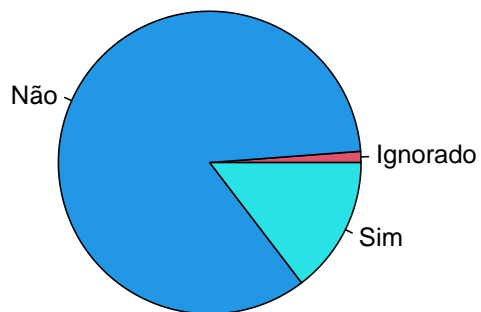




Podemos adicionar título e mudar as cores dos setores do gráfico:

```
# Adicionando título e cores:
pie(tab.tpinduz,
    main = "Trabalho de parto induzido?", # Título do gráfico
    col = c(2,4,5))                      # cores dos setores
```

### Trabalho de parto induzido?



Além disso, para o gráfico ficar mais completo, podemos adicionar como rótulos, as frequências (absolutas ou relativas) dos setores. Podemos fazer isso de duas formas: 1) deixar somente as porcentagens como rótulos e construir uma legenda para os nomes das categorias ou 2) concatenar nomes e frequências das categorias e colocar tudo como rótulo. Vejamos como fazer as duas opções a seguir.

```
# Opção 1: Adicionando as frequências como rótulos e nomes das categorias como legenda
```

```
# Salvando as porcentagens em uma variável:
```

```
porc.tpinduz <- round(100*prop.table(table(dados$STTRABPART)),1)
porc.tpinduz
```

```
##
## Ignorado      Não      Sim
##      1.2      84.2     14.6
```

```
# Criando os rótulos: valor porcentagem + símbolo % :
```

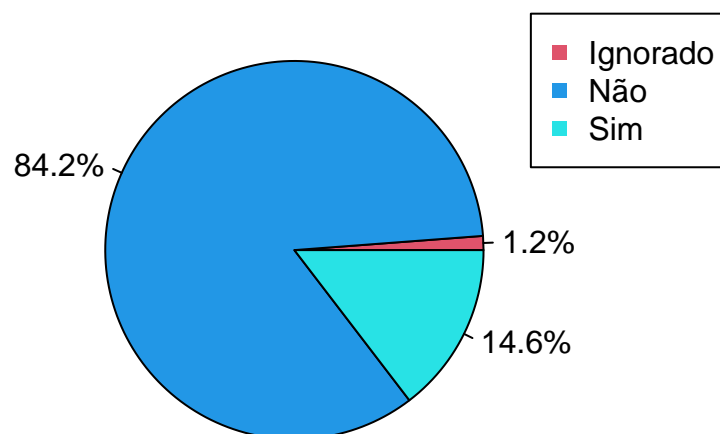
```
rotulos1.tpinduz<-paste(porc.tpinduz,"%",sep="")
rotulos1.tpinduz
```

```
## [1] "1.2%" "84.2%" "14.6%"
```

```
# Adicionando rótulos aos setores e legenda ao gráfico:
```

```
pie(porc.tpinduz,
    main = "Trabalho de parto induzido?", # Título do gráfico
    col = c(2,4,5),                       # cores dos setores
    labels = rotulos1.tpinduz)             # rótulos dos setores (porcentagens)
legend(1,1, names(porc.tpinduz), col = c(2,4,5), pch=15) #note que para a legenda usamos as mesmas cor
```

## Trabalho de parto induzido?



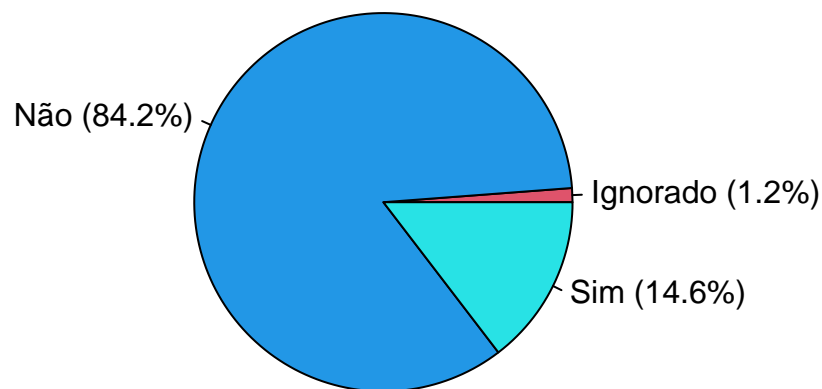
```
# Opção 2: Adicionando as tanto nomes das categorias quanto frequências como rótulos:

# Criando os rótulos: nomes das categorias + parênteses + valor porcentagem + símbolo % :
rotulos2.tpinduz<-paste(names(porc.tpinduz)," (",porc.tpinduz,"%)",sep="")
rotulos2.tpinduz

## [1] "Ignorado (1.2%)" "Não (84.2%)"      "Sim (14.6%)"

# Adicionando rótulos aos setores:
pie(porc.tpinduz,
    main = "Trabalho de parto induzido?", # Título do gráfico
    col = c(2,4,5),                      # cores dos setores
    labels = rotulos2.tpinduz)            # rótulos dos setores (nomes e porcentagens)
```

## Trabalho de parto induzido?



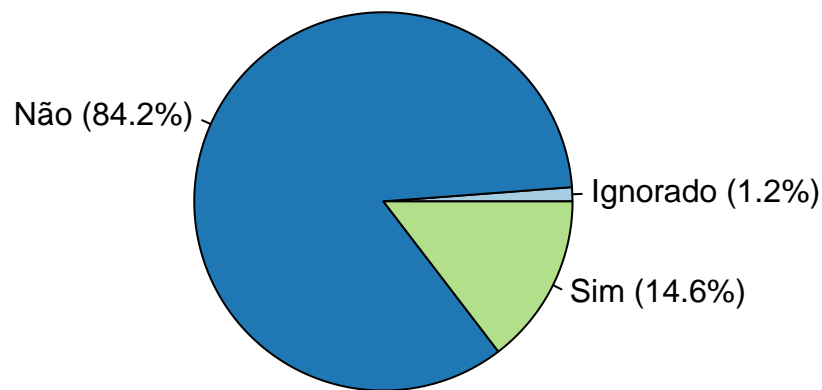
Além de escolher manualmente as cores, podemos usar paletas de cores prontas, de pacotes específicos do R. Por exemplo, o pacote `RColorBrewer` trás várias opções de paletas de cores e basta você selecionar uma paleta que os setores dos gráficos serão coloridos automaticamente com suas cores. As opções de paletas podem ser encontradas em <https://r-graph-gallery.com/38-rcolorbrewers-palettes.html>.

```
#install.packages("RColorBrewer") # remover o # para instalar, caso necessário
library(RColorBrewer)             # habilita o pacote

# Escolhendo uma paleta e salvando em cores.esp (no caso 3 cores da paleta "Paired"):
cores.especiais <- brewer.pal(3, "Paired")
```

```
pie(porc.tpinduz,
    main = "Trabalho de parto induzido?", # Título do gráfico
    col = cores.especiais,                # cores dos setores pacote RColorBrewer
    labels = rotulos2.tpinduz)            # rótulos dos setores (labels e porcentagens)
```

## Trabalho de parto induzido?



## Gráfico de dispersão unidimensional

O gráfico de dispersão unidimensional é utilizado para variáveis quantitativas discretas e construído no software R através da função `plot(...)`. Temos duas opções de gráfico de dispersão unidimensional: ou construímos ele com segmentos de retas para cada valor observado com altura até correspondente à frequência, ou com pontos para cada valor observado, na respectiva altura. Vejamos como construir as duas opções.

```
table(dados$APGAR1)
```

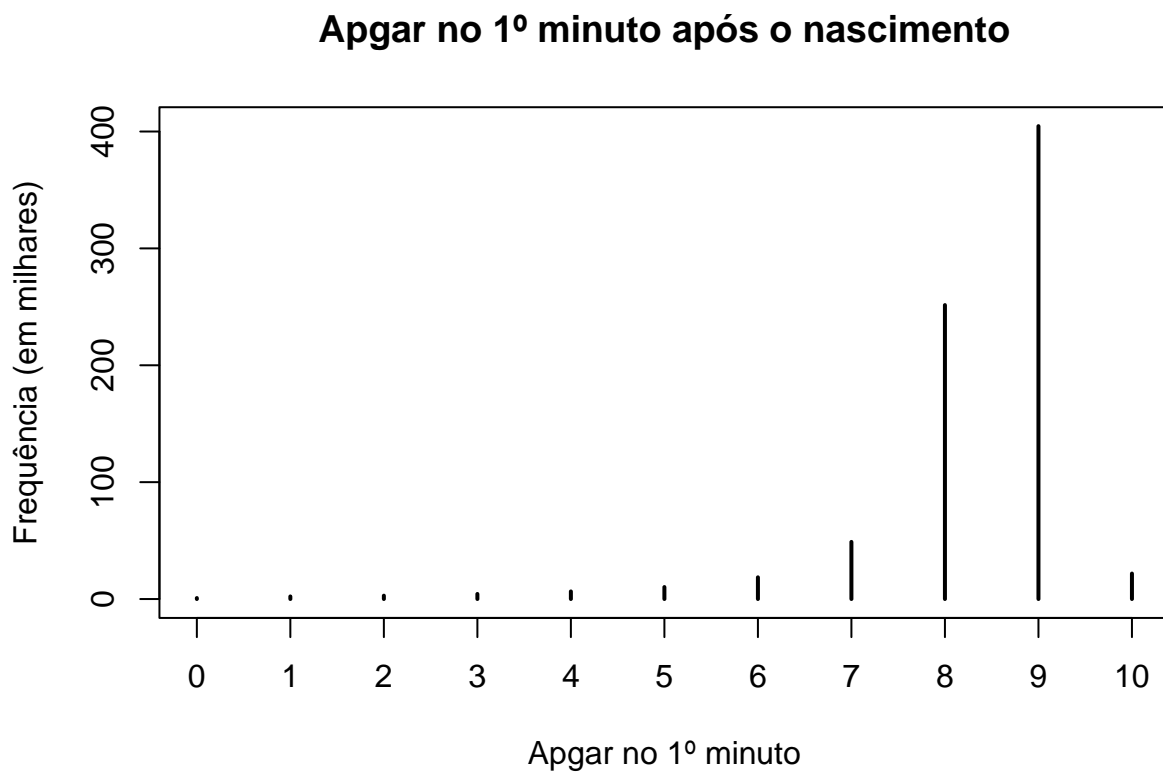
```
##
##      0      1      2      3      4      5      6      7      8      9     10
##  863   2231   2895   4342   6509   10266   18591   48881   251514  404597  21806
##    99
##   637
```

```
# Tabela de frequência absoluta (em milhares)
```

```
tab.apgar1 = table(dados$APGAR1[dados$APGAR1<=10])/1000 ## excluindo o 99 (código de valor ignorado)
tab.apgar1
```

```
##
##      0      1      2      3      4      5      6      7      8      9
## 0.863 2.231 2.895 4.342 6.509 10.266 18.591 48.881 251.514 404.597
##    10
## 21.806
```

```
# Opção 1: Segmentos até a altura da frequência
plot(tab.apgar1,
     main="Apgar no 1º minuto após o nascimento",
     xlab="Apgar no 1º minuto",
     ylab = "Frequência (em milhares)",
     pch=19)
```

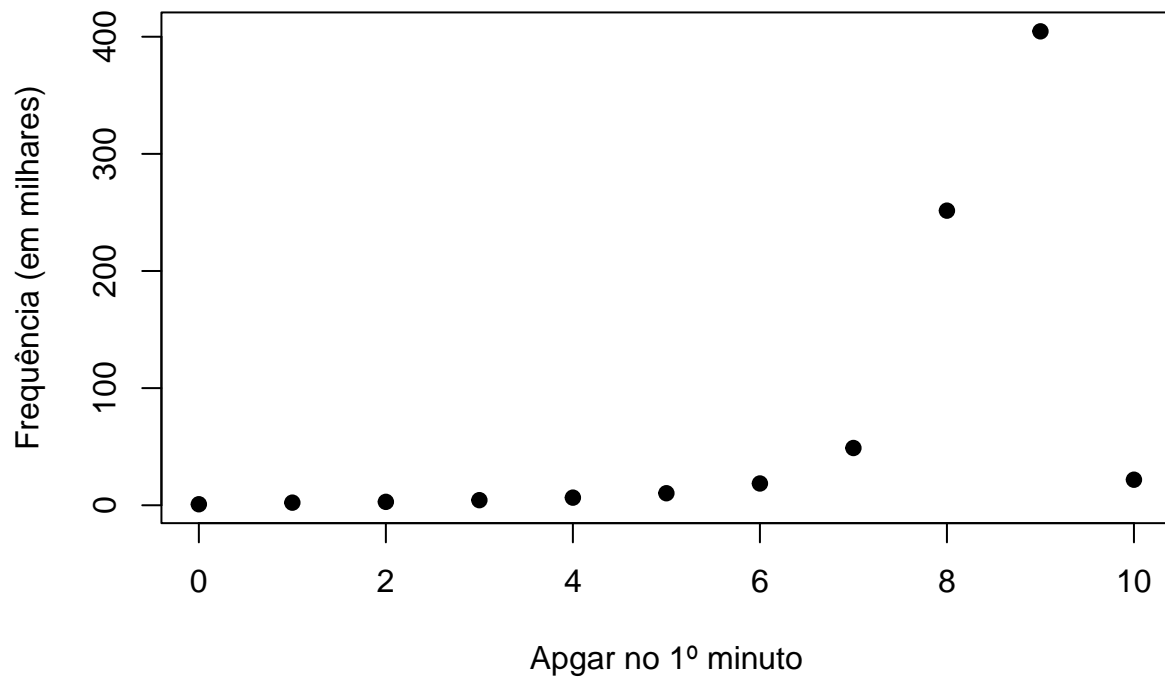


```
# Opção 2: Só os pontos na altura da frequência

valores_apgar1 = as.vector(labels(tab.apgar1)[[1]])
freq_apgar1 = as.vector(tab.apgar1)

plot(valores_apgar1, freq_apgar1,
     main="Apgar no 1º minuto após o nascimento",
     xlab="Apgar no 1º minuto",
     ylab = "Frequência (em milhares)",
     pch=19)
```

## Apgar no 1º minuto após o nascimento



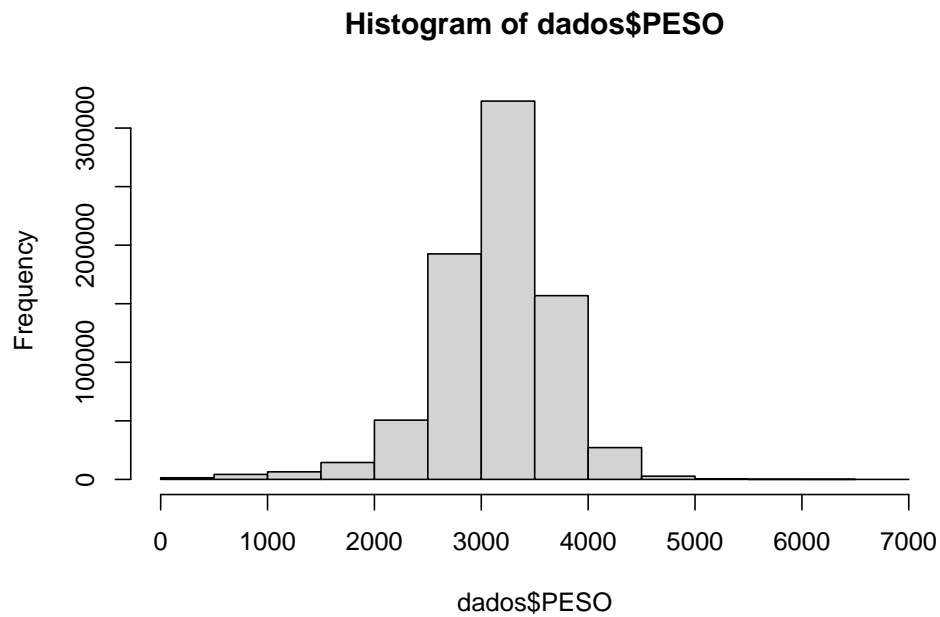
## Histograma

O histograma é um gráfico muito importante para visualização das variáveis quantitativas. Ele é feito para variáveis com intervalos de classe (inclusive pode ser feito adotando intervalos de classe de tamanho 1 e, portanto, pode ser feito para variáveis discretas). Sua construção, diferente das vistas até agora, não consiste em aplicar a função a uma tabela, mas sim direto à variável em questão e a função utilizada é a `hist(...)`.

### Exemplo 1: Variável “peso”

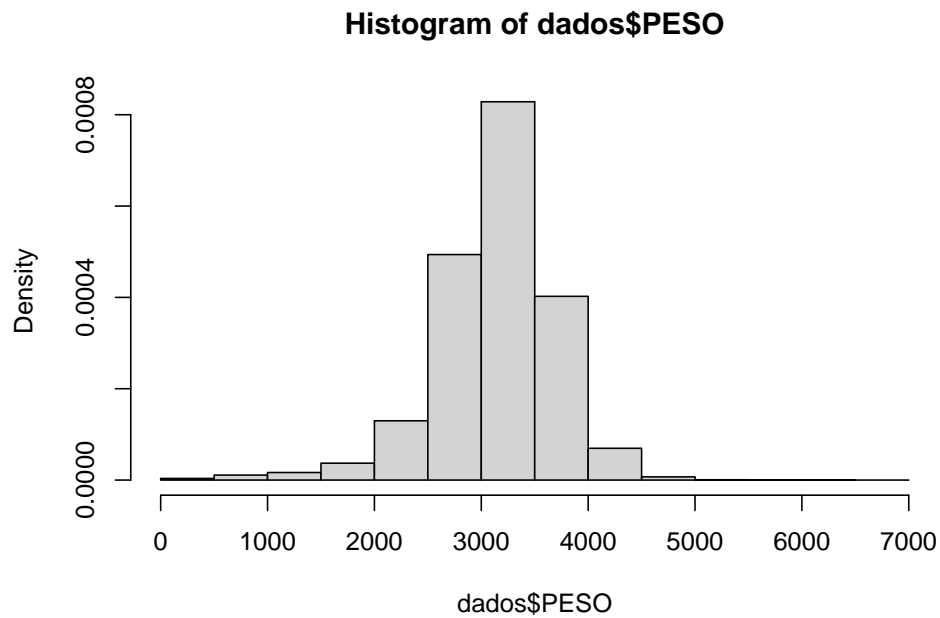
O histograma, por definição, é um gráfico com área igual a 1 e que contém no eixo y a densidade de frequência. Entretanto, é comum encontrarmos histogramas utilizando a frequência (absoluta ou relativa). O próprio R contrói, como default, o histograma utilizando a frequência absoluta.

```
# Histograma com frequência absoluta  
hist(dados$PESO)
```



Para mudar para densidade de frequência, precisamos alterar no `hist(...)` o parâmetro `freq = FALSE`.

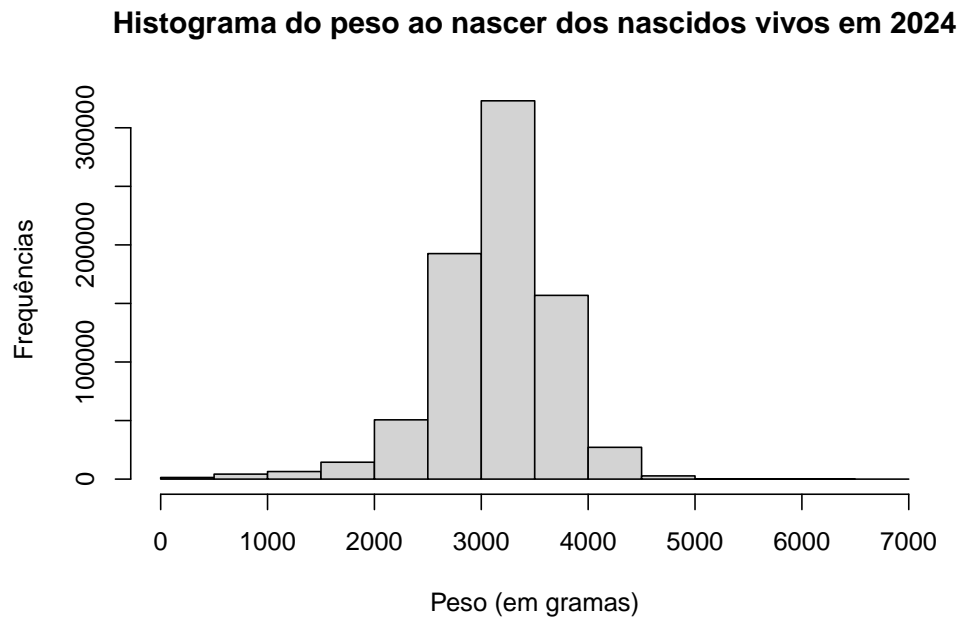
```
# Histograma com densidade de frequência  
hist(dados$PESO,  
      freq = FALSE) ## para densidade de frequências
```



O histograma pode ser formatado, mudando o título, rótulos dos eixos (até porque eles vêm em inglês),

alterando os eixos, adicionando cores às barras, etc.

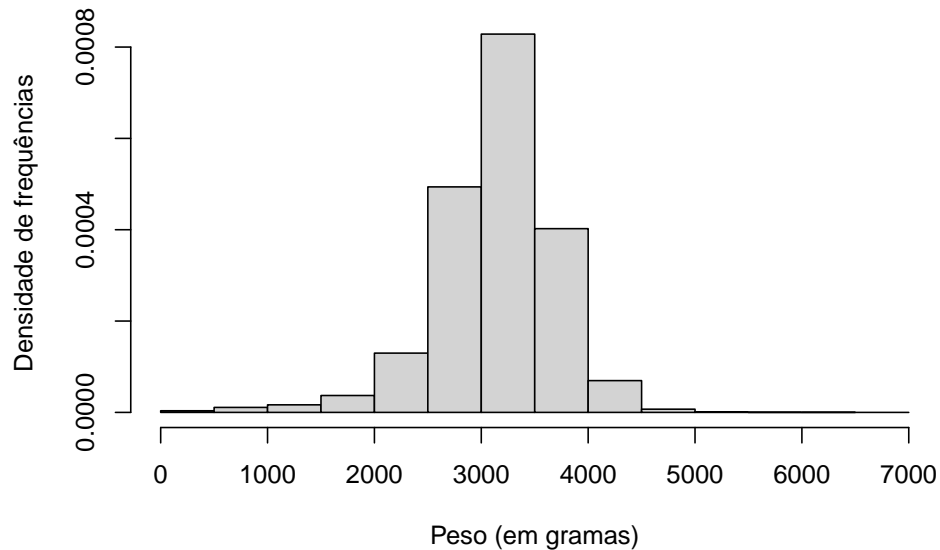
```
# Adicionando título e rótulos dos eixos (histograma com frequências):  
hist(dados$PESO,  
      main="Histograma do peso ao nascer dos nascidos vivos em 2024", ## título  
      xlab="Peso (em gramas)", ## rótulo do eixo x  
      ylab = "Frequências") ## rótulo do eixo y
```



```
# Adicionando título e rótulos dos eixos (histograma com densidade de frequências):  
hist(dados$PESO,  
      freq = FALSE, ## para densidade de frequências  
      main="Histograma do peso ao nascer dos nascidos vivos em 2024", ## título  
      xlab="Peso (em gramas)", ## rótulo do eixo x  
      ylab = "Densidade de frequências") ## rótulo do eixo y
```

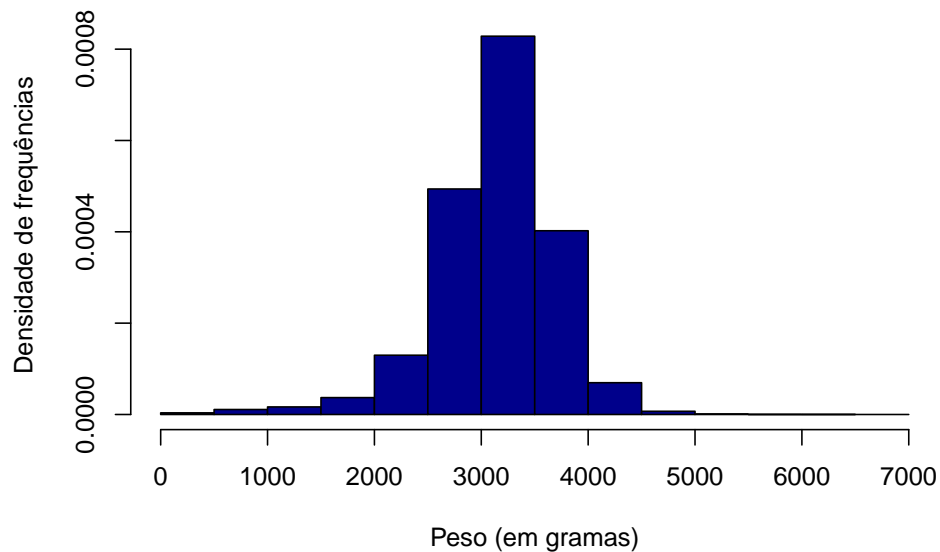


Histograma do peso ao nascer dos nascidos vivos em 2024



```
# Escolhendo a cor das barras:  
hist(dados$PESO,  
      freq = FALSE,                                ## para densidade de frequências  
      main="Histograma do peso ao nascer dos nascidos vivos em 2024", ## título  
      xlab="Peso (em gramas)",                      ## rótulo do eixo x  
      ylab = "Densidade de frequências",            ## rótulo do eixo y  
      col='darkblue')                               ## usa a cor roxa nas barras
```

Histograma do peso ao nascer dos nascidos vivos em 2024



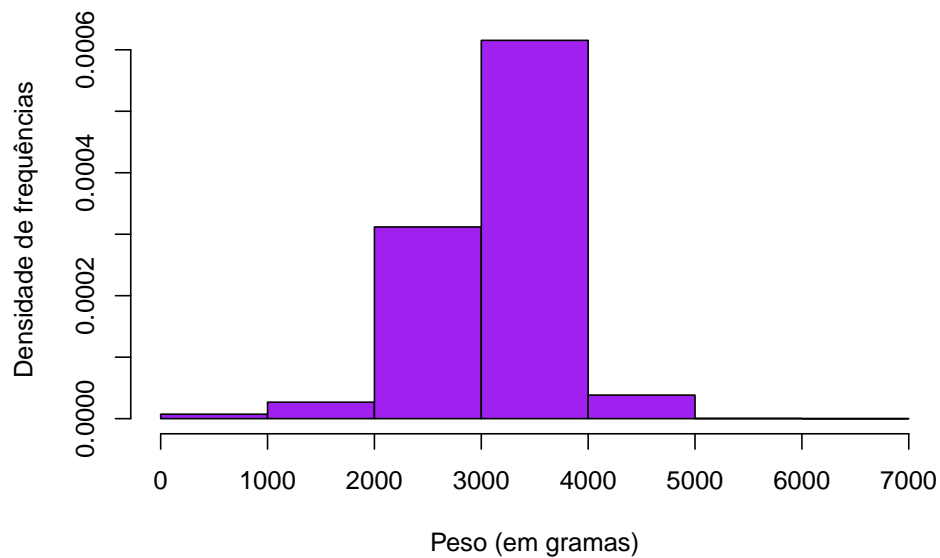
```
# Escolhendo o número de classes do histograma:  
hist(dados$PESO,
```

```

freq = FALSE,                                ## para densidade de frequências
nclass = 6,                                  ## N° de classes
main="Histograma do peso ao nascer dos nascidos vivos em 2024", ## título
xlab="Peso (em gramas)",                     ## rótulo do eixo x
ylab = "Densidade de frequências",           ## rótulo do eixo y
col='purple')                               ## usa a cor roxa nas barras

```

**Histograma do peso ao nascer dos nascidos vivos em 2024**

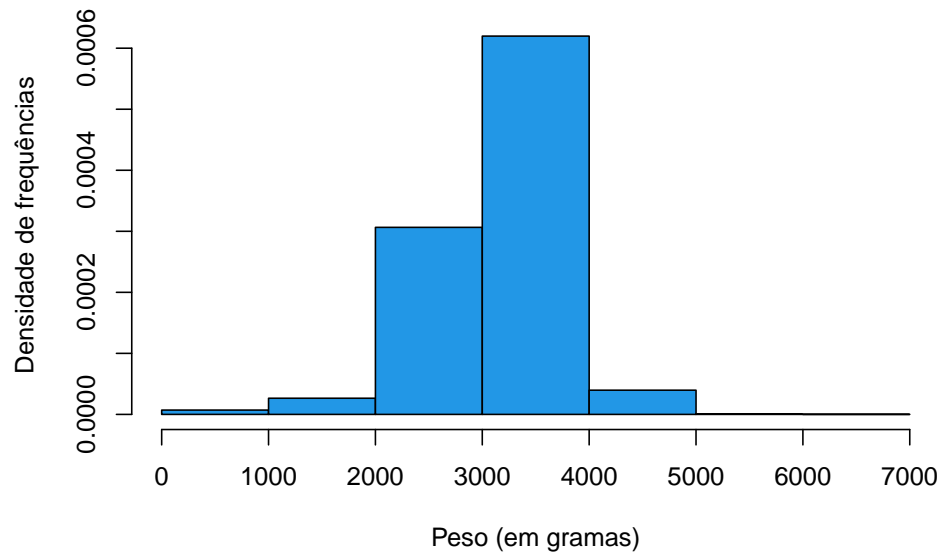


```

# Escolhendo as quebras dos intervalos de classes do histograma:
hist(dados$PESO,
     freq = FALSE,                                ## para densidade de frequências
     breaks = seq(0,7000,1000),                  ## Quebras das classes
     right = F,                                   ## p/ interv. fechados à esquerda
     main="Histograma do peso ao nascer dos nascidos vivos em 2024", ## título
     xlab="Peso (em gramas)",                     ## rótulo do eixo x
     ylab = "Densidade de frequências",           ## rótulo do eixo y
     # xlim=c(0,8000),                            ## limites do eixo x
     # ylim=c(0,0.001),                            ## limites do eixo y
     col=4)                                       ## cor das barras

```

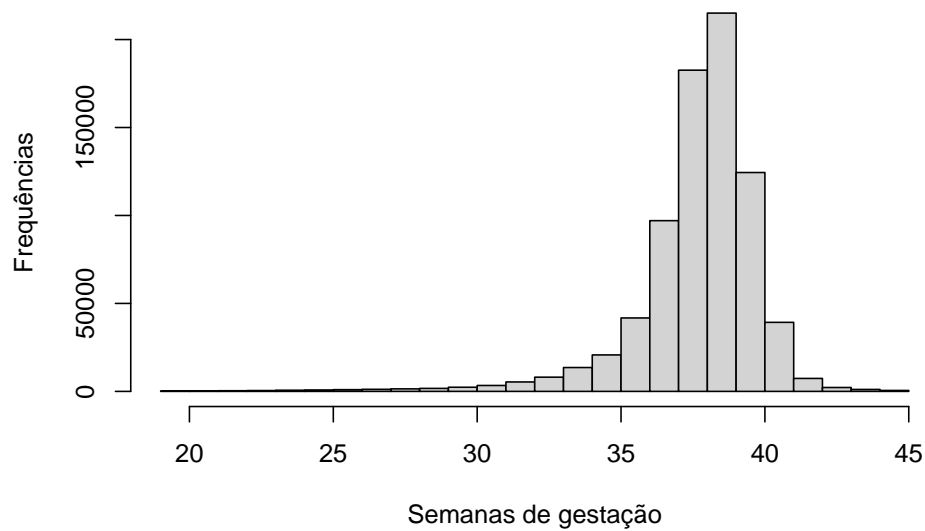
### Histograma do peso ao nascer dos nascidos vivos em 2024



### Exemplo 2: Variável “Semanas de gestação”

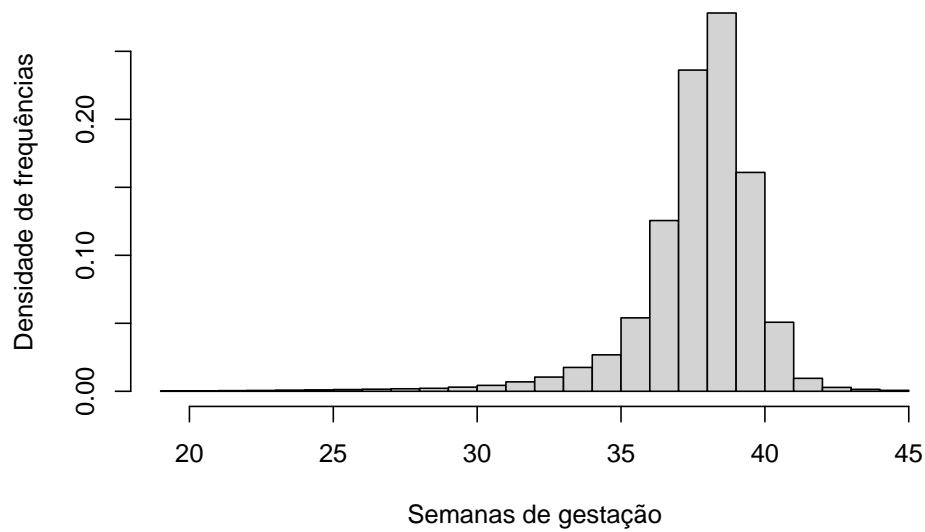
```
# Histograma com frequência absoluta
hist(dados$SEMAGESTAC,
     main="Histograma de semanas de gestação",      ## título
     xlab="Semanas de gestação",                  ## rótulo do eixo x
     ylab = "Frequências")                        ## rótulo do eixo y
```

**Histograma de semanas de gestação**



```
# Histograma com densidade de frequência
hist(dados$SEMAGESTAC,
     freq = FALSE,                                     ## para densidade de frequências
     main="Histograma de semanas de gestação",        ## título
     xlab="Semanas de gestação",                      ## rótulo do eixo x
     ylab = "Densidade de frequências")               ## rótulo do eixo y
```

**Histograma de semanas de gestação**



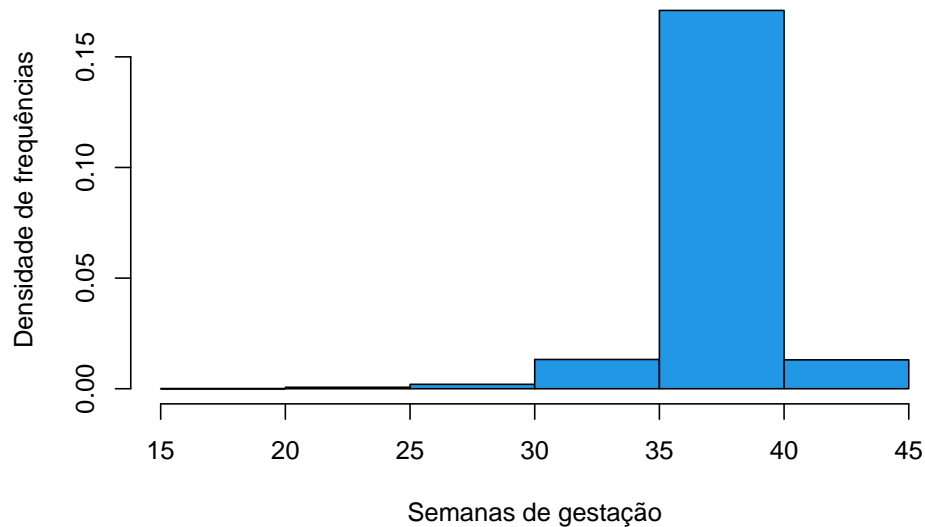
```
# Escolhendo o número de classes do histograma:
hist(dados$SEMAGESTAC,
     freq = FALSE,                                     ## para densidade de frequências
```

```

nclass = 6,                                ## Nº de classes
main="Histograma de semanas de gestação",   ## título
xlab="Semanas de gestação",                ## rótulo do eixo x
ylab = "Densidade de frequências",         ## rótulo do eixo y
col=4)                                     ## usa a cor azul nas barras

```

**Histograma de semanas de gestação**



```
summary(dados$SEMAGESTAC)
```

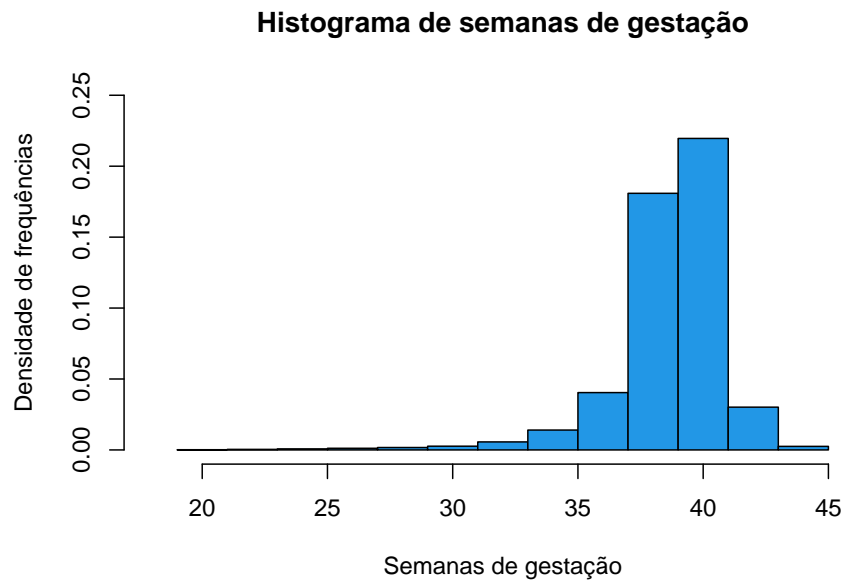
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##      19.00  37.00   39.00   38.16  39.00   45.00    6912
```

```
# Escolhendo as quebras dos intervalos de classes do histograma:
```

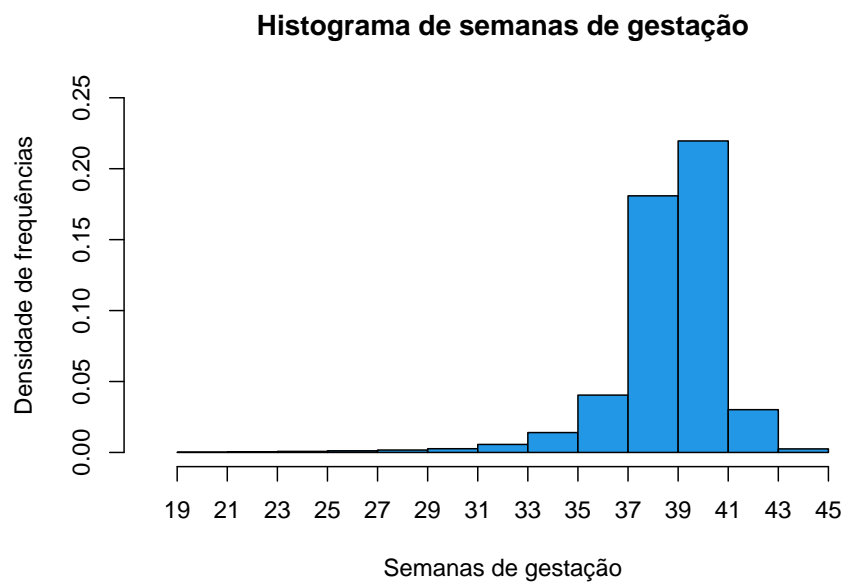
```

hist(dados$SEMAGESTAC,
     freq = FALSE,                                ## para densidade de frequências
     breaks = seq(19,45,2),                      ## Quebras das classes
     right = F,                                  ## para intervalos fechados à esquerda
     main="Histograma de semanas de gestação",   ## título
     xlab="Semanas de gestação",                ## rótulo do eixo x
     ylab="Densidade de frequências",           ## rótulo do eixo y
     col=4,                                       ## cor das barras
     xlim=c(18,46),                             ## limites do eixo x
     ylim=c(0,0.25))                           ## limites do eixo y

```

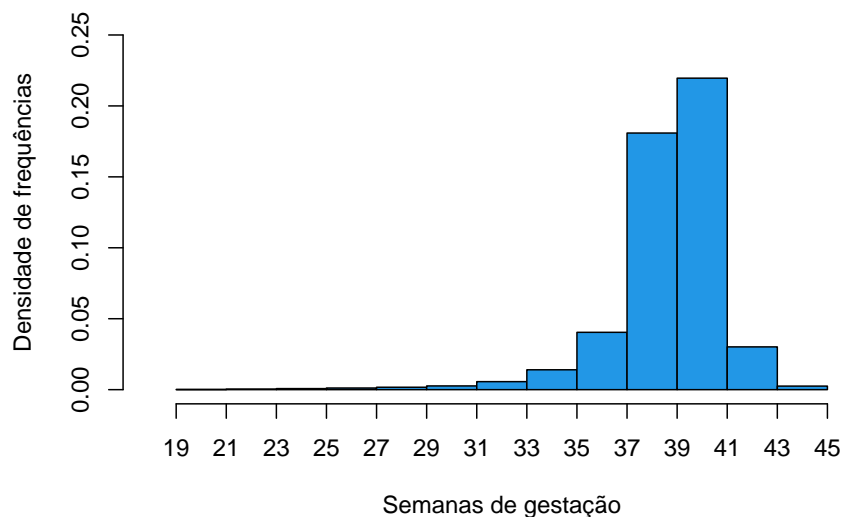


```
# Escolhendo as quebras dos intervalos de classes do histograma e controla o eixo x (opção 1):
hist(dados$SEMAGESTAC,
     freq = FALSE,
     breaks = seq(19,45,2),
     right = F,
     main="Histograma de semanas de gestação",
     xlab="Semanas de gestação",
     ylab="Densidade de frequências",
     col=4,
     xlim=c(18,46),
     ylim=c(0,0.25),
     xaxp = c(19,45,13))
## para densidade de frequências
## Quebras das classes
## para intervalos fechados à esquerda
## título
## rótulo do eixo x
## rótulo do eixo y
## cor das barras
## limites do eixo x
## limites do eixo y
## controla eixo x : c(início, fim, qtde de i
```



```
# Escolhendo as quebras dos intervalos de classes do histograma e controla o eixo x (opção 2):
hist(dados$SEMAGESTAC,
     freq = FALSE,
     breaks = seq(19,45,2),
     right = F,
     main="Histograma de semanas de gestação",
     xlab="Semanas de gestação",
     ylab="Densidade de frequências",
     col=4,
     xlim=c(18,46),
     ylim=c(0,0.25),
#     yaxt="n",
#     xaxt="n")
#     axis(2, at=y)
#     axis(2, at=y, las=2)
#     axis(1, at=seq(19,45,2))
     ## para densidade de frequências
     ## Quebras das classes
     ## para intervalos fechados à esquerda
     ## título
     ## rótulo do eixo x
     ## rótulo do eixo y
     ## cor das barras
     ## limites do eixo x
     ## limites do eixo y
     # para não imprimir valores do eixo y
     # para não imprimir valores do eixo x
     # para adicionar valores desejados no eixo y
     # para adicionar valores eixo y e rotacionar labels do
     # para adicionar valores desejados no eixo x
```

**Histograma de semanas de gestação**



## Gráfico de Ogiva (Polígono de frequências acumuladas)

O gráfico de ogiva pode ser utilizado para variáveis quantitativas e para construí-lo vamos utilizar o pacote `fdth`. Este pacote, através de sua função `fdt(...)` será muito útil para a construção de intervalos de classes e da tabela de frequência acumulada e, portanto, da construção do gráfico. Após a construção da tabela, basta aplicar a função `plot(...)`, utilizando a opção `type = 'cfp'`, de polígono de frequência acumulada (em inglês, *cumulative frequency polygon*).

```
#install.packages("fdth") # remover o # para rodar e instalar, caso necessário
library(fdth)             # habilitando o pacote
```

```
##
## Attaching package: 'fdth'
```

```
## The following objects are masked from 'package:stats':
##
## sd, var
```

```
# Construindo a tabela com intervalos de classes arbitrários:
tabela.pesos <- fdt(dados$PESO, na.rm = TRUE)
tabela.pesos
```

```
## Class limits      f   rf rf(%)    cf cf(%)
##      [99,431)    1041 0.00  0.13   1041  0.13
##     [431,762.9)   2316 0.00  0.30   3357  0.43
##    [762.9,1095)   3229 0.00  0.41   6586  0.84
##   [1095,1427)    4319 0.01  0.55  10905  1.40
##   [1427,1759)    6845 0.01  0.88  17750  2.28
##   [1759,2091)   13484 0.02  1.73  31234  4.00
##   [2091,2423)   32363 0.04  4.15  63597  8.15
##   [2423,2755)   82151 0.11 10.53 145748 18.69
##   [2755,3087)  176992 0.23 22.69 322740 41.38
##   [3087,3419)  220868 0.28 28.32 543608 69.70
##   [3419,3750)  153293 0.20 19.65 696901 89.35
##   [3750,4082)   61455 0.08  7.88 758356 97.23
##   [4082,4414)   16843 0.02  2.16 775199 99.39
##   [4414,4746)    3700 0.00  0.47 778899 99.87
##   [4746,5078)    760 0.00  0.10 779659 99.97
##   [5078,5410)    158 0.00  0.02 779817 99.99
##   [5410,5742)    41 0.00  0.01 779858 99.99
##   [5742,6074)    17 0.00  0.00 779875 99.99
##   [6074,6406)     7 0.00  0.00 779882 99.99
##   [6406,6738)     7 0.00  0.00 779889 100.00
##   [6738,7070)     2 0.00  0.00 779891 100.00
```

```
# Gráfico de ogiva: usa o pacote fdth e função fdt() para tabela juntamente com
# o plot() e type='cfp'
```

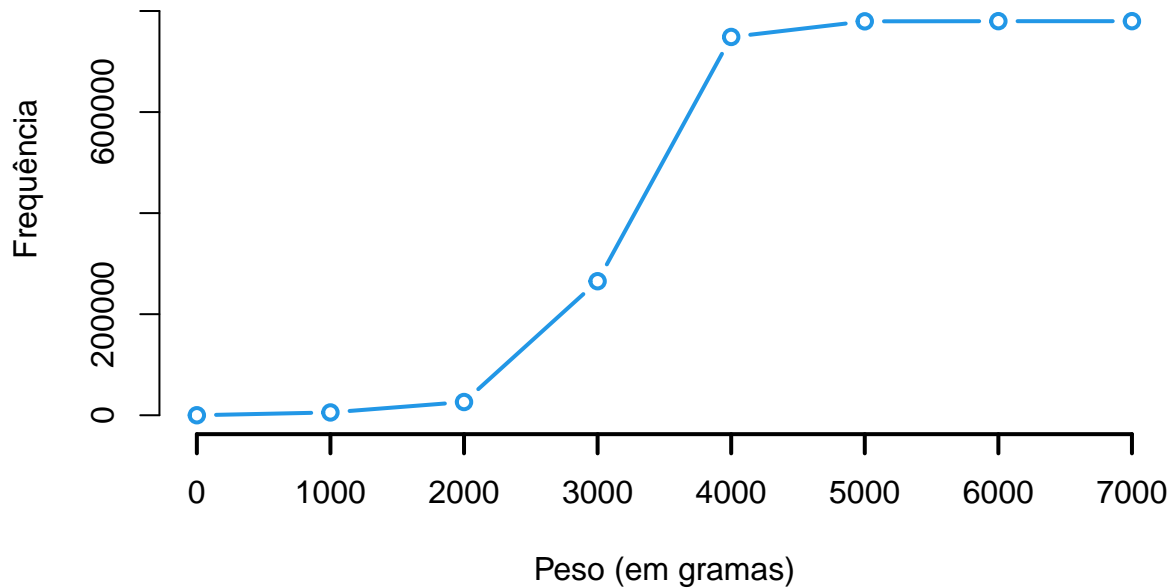
```
# Construindo a tabela com intervalos de classes a partir de início, fim e amplitude:
tabela.pesos = fdt(dados$PESO, na.rm = TRUE, start=0, end=7000, h=1000) # h = amplitude das classes
tabela.pesos
```

```
## Class limits      f   rf rf(%)    cf cf(%)
##      [0,1000)    5524 0.01  0.71   5524  0.71
##     [1000,2000)  20652 0.03  2.65  26176  3.36
##     [2000,3000)  239027 0.31 30.65 265203 34.00
##     [3000,4000)  483422 0.62 61.98 748625 95.99
##     [4000,5000)  30928 0.04  3.97 779553 99.95
##     [5000,6000)    318 0.00  0.04 779871 99.99
##     [6000,7000)    19 0.00  0.00 779890 100.00
```

```
# Gráfico de ogiva (polígono de frequência acumulada): usa o plot() com type='cfp'
plot(tabela.pesos,                                     ## tabela contruída com a função fdt()
      type='cfp',                                       ## cfp: cumulative frequency polygon
      xlab="Peso (em gramas)",                          ## rótulo do eixo x
      ylab = "Frequência",                             ## rótulo do eixo y
      main = "Ogiva - Peso ao nascer dos nascidos vivos em 2024", ## título
      col=4,                                           ## cor da linha
      lwd=2)                                           ## espessura da linha
```



## Ogiva – Peso ao nascer dos nascidos vivos em 2024



### Polígono de frequências

O polígono de frequências é um gráfico de linhas que pode ser utilizado para variáveis quantitativas e para construí-lo vamos utilizar a tabela fornecida pela função `fdt(...)` do pacote `fdth`. Após a construção da tabela, basta aplicar a função `plot(...)`, utilizando a opção `type = 'fp'`, de polígono de frequências (em inglês, *frequency polygon*).

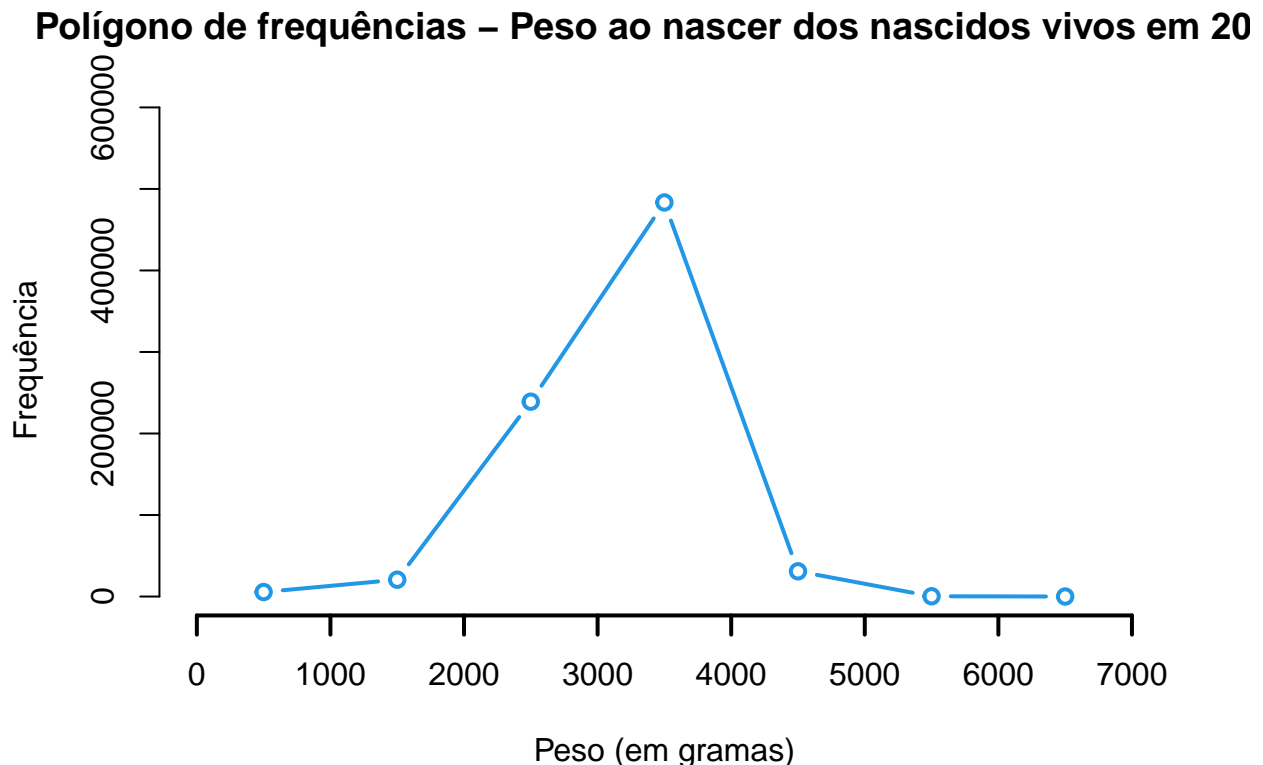
```
# install.packages("fdth") # remover o # para rodar e instalar, caso necessário
# library(fdth)           # habilitando o pacote

# Construindo a tabela com intervalos de classes a partir de início, fim e amplitude:
tabela.pesos = fdt(dados$PESO, na.rm = TRUE, start=0, end=7000, h=1000) # h = amplitude das classes
tabela.pesos
```

##	Class limits	f	rf	rf(%)	cf	cf(%)
##	[0,1000)	5524	0.01	0.71	5524	0.71
##	[1000,2000)	20652	0.03	2.65	26176	3.36
##	[2000,3000)	239027	0.31	30.65	265203	34.00
##	[3000,4000)	483422	0.62	61.98	748625	95.99
##	[4000,5000)	30928	0.04	3.97	779553	99.95
##	[5000,6000)	318	0.00	0.04	779871	99.99
##	[6000,7000)	19	0.00	0.00	779890	100.00

```
# Polígono de frequência: usa o pacote fdth e função fdt() para tabela juntamente com
# o plot() e type='fp'

plot(tabela.pesos,                                     ## tabela contruída com a função fdt()
     type='fp',                                       ## fp: frequency polygon
     xlab="Peso (em gramas)",                         ## rótulo do eixo x
     ylab = "Frequência",                             ## rótulo do eixo y
     main = "Polígono de frequências - Peso ao nascer dos nascidos vivos em 2024", ## título
     col=4,                                           ## cor da linha
     lwd=2)                                           ## espessura da linha
```



## Exportando gráficos

Quando fazemos gráficos no software R e temos o objetivo de utilizá-los em um trabalho, relatório, artigo científico, precisamos salvá-lo, exportá-lo. Podemos realizar essa tarefa utilizando a opção Export -> Save As da própria janela do RStudio, por exemplo, ou executar um código para realizar a exportação, como a seguir.

Como exemplo, vamos exportar o histograma da variável semanas de gestação.

```
## Inicie com o comando de salvar o gráfico em um arquivo com o formato desejado:

png("meu_grafico.png")      # formato png
pdf("meu_grafico.pdf")      # formato pdf
jpeg("meu_grafico.jpeg")    # formato jpeg
```

```

#tiff("meu_grafico.tiff") # formato tiff
#svg("meu_grafico.svg")  # formato svg

# Execute o gráfico

hist(dados$SEMAGESTAC,
     freq = FALSE,                ## para densidade de frequências
     breaks = seq(19,45,2),      ## Quebras das classes
     right = F,                  ## para intervalos fechados à esquerda
     main="Histograma de semanas de gestação", ## título
     xlab="Semanas de gestação",  ## rótulo do eixo x
     ylab="Densidade de frequências", ## rótulo do eixo y
     col=4,                      ## cor das barras
     xlim=c(18,46),              ## limites do eixo x
     ylim=c(0,0.25),             ## limites do eixo y
     xaxp = c(19,45,13))        ## controla eixo x : c(início, fim, qtde de intervalos)

# Dê o comando de encerrar o processo:
dev.off()

## pdf
## 2

```

**Observação:** O arquivo exportado será salvo na pasta definida em `setwd`. Se você não definiu uma pasta na sessão ou quer conferir qual pasta está definida, basta usar o comando `getwd()` que o caminho da pasta onde o arquivo será salvo será exibido.

```

getwd()

## [1] "C:/Users/AmandaBGM/Google Drive/UFG/Especialização_FEN_IME/2024/Scripts"

```