Estatística descritiva para *Data Science*

0.4 - Aula Prática: Sumarização de dados em tabelas

Profa. Dra. Amanda Buosi Gazon Milani

2024-07-19

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-vígula):

# 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                     ESCMAE2010 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
```

Tabela de frequência absoluta

Para executarmos uma tabela de frequência absoluta no R, a função utilizada é a `table(...)`

Vamos construir a tabela de frequência absoluta para algumas variáveis do nosso conjunto de dados:

- **LOCNASC**

```
# Tabela de frequência absoluta:

table(dados$LOCNASC)
```

```
##
##   Aldeia Indígena         Domicílio          Hospital          Ignorado
##              1987              4113            768765                31
## Outro Estab Saúde            Outros
##              4829               202
```

- **ESTCIVMAE**

```
# Tabela de frequência absoluta:

table(dados$ESTCIVMAE)
```

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

Note que a função `table(...)` por padrão descarta os valores faltantes (NA's) da variável.

```r
length(dados$ESTCIVMAE)      ## para verificar o tamanho total
```

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

```r
sum(table(dados$ESTCIVMAE))  ## obtém a quantidade de valores (soma) da tabela
```

```
## [1] 776489
```

```r
length(dados$ESTCIVMAE) - sum(table(dados$ESTCIVMAE)) ## diferença (quantidade de NA's)
```

```
## [1] 3438
```

Caso seja uma informação importante, pode-se habilitar na função `table(...)` o parâmetro `useNA` para que os valores faltantes (NA's) sejam contados e apresentados na tabela de frequências.

```r
# Tabela de frequência absoluta com NA's:

table(dados$ESTCIVMAE, useNA = "always")
```

```
##
##                 Casada               Ignorado Separada judicialmente
##                 242908                   2192                  13273
##               Solteira       União consensual                  Viúva
##                 403848                 112982                   1286
##                   <NA>
##                   3438
```

- **GESTACAO**

```r
# Tabela de frequência absoluta:

table(dados$GESTACAO)
```

```
##
##     22 a 27 semanas     28 a 31 semanas     32 a 36 semanas     37 a 41 semanas
##                4540                8881               89511              658364
##   42 semanas e mais            Ignorado Menos de 22 semanas
##               11245                  52                 474
```

- **GRAVIDEZ**

```r
# Tabela de frequência absoluta:

table(dados$GRAVIDEZ)
```

```
##
##         Dupla      Ignorado Tripla e mais         Única
##         18191            16           302        760603
```

Tabela de frequência relativa

Para executarmos uma tabela de frequência relativa no R, a função utilizada é a `prop.table(...)` . Ela deve ser aplicada à uma tabela de frequência absoluta, portanto seu uso será conjunto com a função `table(...)` .

Vamos construir a tabela de frequência relativa para algumas variáveis do nosso conjunto de dados:

- **LOCNASC**

```
# Tabela de frequência relativa:

prop.table(table(dados$LOCNASC))
```

```
##
##   Aldeia Indígena         Domicílio          Hospital          Ignorado
##      2.547674e-03      5.273570e-03      9.856884e-01      3.974731e-05
## Outro Estab Saúde            Outros
##      6.191605e-03      2.589986e-04
```

Para visualizar melhor as frequências relativas deste exemplo, podemos usar a função `round(...)` para arredondar as proporções para um determinado número de casas decimais:

```
# Tabela de frequência relativa com arredondamento:

round( prop.table(table(dados$LOCNASC)) , 5)
```

```
##
##   Aldeia Indígena         Domicílio          Hospital          Ignorado
##           0.00255           0.00527           0.98569           0.00004
## Outro Estab Saúde            Outros
##           0.00619           0.00026
```

- **ESTCIVMAE**

```
# Tabela de frequência relativa com arredondamento:

round( prop.table(table(dados$ESTCIVMAE)) , 3 )
```

```
##
##                 Casada              Ignorado Separada judicialmente
##                  0.313                 0.003                 0.017
##               Solteira      União consensual                 Viúva
##                  0.520                 0.146                 0.002
```

Novamente, pode-se habilitar na função `table(...)` o parâmetro `useNA` para que os valores faltantes (NA's) sejam contados e apresentados na tabela de frequências e, ao executar a tabela de prequências relativas com `prop.table(...)`, os NA's serão considerados na tabela.

```r
# Tabela de frequência relativa com NA's:

round( prop.table(table(dados$ESTCIVMAE, useNA = "always")) , 3)
```

```
##
##                 Casada               Ignorado Separada judicialmente
##                  0.311                  0.003                  0.017
##               Solteira      União consensual                  Viúva
##                  0.518                  0.145                  0.002
##                   <NA>
##                  0.004
```

A tabela de frequência relativa adotando a porcentagens pode ser obtida multiplicando por 100 o resultado de uma tabela de frequência relativa com proporções!

- **LOCNASC**

```r
# Tabela de frequência relativa com arredondamento:

100*round( prop.table(table(dados$LOCNASC)) , 5)
```

```
##
##   Aldeia Indígena         Domicílio          Hospital          Ignorado
##             0.255             0.527            98.569             0.004
## Outro Estab Saúde            Outros
##             0.619             0.026
```

- **ESTCIVMAE**

```r
# Tabela de frequência relativa com arredondamento:

100*round( prop.table(table(dados$ESTCIVMAE)) , 2 )
```

```
##
##                 Casada               Ignorado Separada judicialmente
##                     31                      0                      2
##               Solteira      União consensual                  Viúva
##                     52                     15                      0
```

Em resumo, temos a seguinte estrutura de uso das funções para construção de tabelas de frequências absoluta e relativas:

```r
table(nome_variavel)                 ## frequência absoluta

prop.table(table(nome_variavel))     ## frequência relativa (proporção)

100*prop.table(table(nome_variavel)) ## frequência relativa (porcentagem)
```

Criando intervalos de classes

Quando a variável é quantitativa contínua, necessitamos criar intervalos de classes para construir a tabelas de frequências. O mesmo ocorre quando a variável é quantitativa discreta com muitos valores distintos de ocorrência, e também necessita da quebra em intervalos para que ocorra um efetivo resumo e organização dos dados.

• IDADEMAE

O primeiro passo é fazer um sumário da variável, para conhecer sua distribuição, seu mínimo e máximo e entender como proceder com a criação dos intervalos de classe.

```
summary(dados$IDADEMAE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    8.00   23.00   27.00   27.76   33.00   99.00       7
```

Geralmente 99 é um valor de controle, para valor ignorado, portanto vamos filtrar os dados somente para idades menores que 99 anos e retomar o `summary(...)` .

```
summary(dados$IDADEMAE[dados$IDADEMAE<99])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    8.00   23.00   27.00   27.75   33.00   65.00       7
```

Podemos ver que a idade mínima é de 8 anos e a máxima é de 65 anos, portanto, podemos adotar intervalos de 10 em 10 anos, por exemplo.

```
dados$IDADEMAE_cat <- cut(dados$IDADEMAE,                    ## variável que será categorizada
                          breaks = c(0,10,20,30,40,50,60,70),  ## limites dos intervalos
                          right = FALSE)                     ## para intervalos fechados à esquerda
table(dados$IDADEMAE_cat)
```

```
##
##  [0,10) [10,20) [20,30) [30,40) [40,50) [50,60) [60,70)
##       2   89337  384156  272475   33797     142       8
```

Note que nessa categorização da variável, conseguimos inclusive "excluir" os valores 99, pois eles foram incluídos aos NA's (uma vez que os intervalos de classe não contemplavam esse valor).

```
table(dados$IDADEMAE_cat, useNA = "always")
```

```
##
##  [0,10) [10,20) [20,30) [30,40) [40,50) [50,60) [60,70)    <NA>
##       2   89337  384156  272475   33797     142       8      10
```

```
table(dados$IDADEMAE>=99)  ## essa tabela verifica quantos "TRUE" satisfazem a condição
```

```
##
##  FALSE   TRUE
## 779917      3
```

- **PESO**

O primeiro passo é fazer um sumário da variável, para conhecer sua distribuição, seu mínimo e máximo e entender como proceder com a criação dos intervalos de classe.

```
summary(dados$PESO)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     100    2870    3185    3144    3490    7000      36
```

Podemos ver que o menor peso é de 100g e o maior peso é de 7000g, portanto, podemos adotar intervalos de 1000 em 1000 gramas, por exemplo.

```
dados$PESO_cat <- cut(dados$PESO,                      ## variável que será categorizada
                      breaks = seq(0,7000,1000),       ## limites dos intervalos
                      right = FALSE,                   ## para intervalos fechados à esquerda
                      include.lowest = TRUE)           ## fechar o último intervalo à direita também

table(dados$PESO_cat)
```

```
##
##     [0,1e+03) [1e+03,2e+03) [2e+03,3e+03) [3e+03,4e+03) [4e+03,5e+03)
##          5524         20652        239027        483422         30928
## [5e+03,6e+03) [6e+03,7e+03]
##           318            20
```

Podemos mudar os rótulos (nomes) das classes, como a seguir. Neste caso, usando a unidade de medida quilograma (kg) em vez de grama (g), conseguimos rótulos mais claros e legíveis.

```
nomes_classes <- c('[0,1)','[1,2)','[2,3)','[3,4)','[4,5)','[5,6)','[6,7]')  # rótulos para as classes

dados$PESO_cat <- cut(dados$PESO,                      ## variável que será categorizada
                      breaks = seq(0,7000,1000),       ## limites dos intervalos
                      right = FALSE,                   ## para intervalos fechados à esquerda
                      include.lowest = TRUE,           ## fechar o último intervalo à direita também
                      labels = nomes_classes)          ## adiciona os rótulos das classes

table(dados$PESO_cat)
```

```
##
##  [0,1)  [1,2)  [2,3)  [3,4)  [4,5)  [5,6)  [6,7]
##   5524  20652 239027 483422  30928    318     20
```

E para variáveis categorizadas com intervalos de classes podemos fazer também tabelas de frequências relativas:

```
round( prop.table(table(dados$PESO_cat)) , 5 )     ## frequência relativa (proporção)
```

```
##
##   [0,1)   [1,2)   [2,3)   [3,4)   [4,5)   [5,6)   [6,7]
## 0.00708 0.02648 0.30649 0.61986 0.03966 0.00041 0.00003
```

```r
100*round( prop.table(table(dados$PESO_cat)) , 5 ) ## frequência relativa (porcentagem)
```

```
##
##  [0,1)  [1,2)  [2,3)  [3,4)  [4,5)  [5,6)  [6,7]
##  0.708  2.648 30.649 61.986  3.966  0.041  0.003
```

Tabela de frequência acumulada

Para as variáveis quantitativas, podemos construir a tabela de frequência acumulada. Para isso, utilizaremos o pacote `fdth` e você precisará instalá-lo antes de usá-lo pela primeira vez.

```r
#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
```

```r
# Tabela de frequências (absoluta, relativa e acumulada) para variável quantitativa:
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
```

Note que a tabela ficou com muitos intervalos de classes e estes com valores quebrados. Podemos manipular os valores dos limites das classes:

```r
# Alterando os valores dos limites das classes na função fdt(...)
tabela.pesos <- fdt(dados$PESO, na.rm = TRUE, start = 0, end = 7000, h = 1000)
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
```

Vamos aplicar ao exemplo da variável IDADEMAE, replicando os mesmos intervalos de classes que utilizamos anteriormente:

```r
# Tabela de frequências (absoluta, relativa e acumulada) para variável quantitativa:
tabela.idademae <- fdt(dados$IDADEMAE, na.rm = TRUE, start = 0, end = 70, h = 10)
tabela.idademae
```

```
##  Class limits      f   rf rf(%)     cf  cf(%)
##        [0,10)      2 0.00  0.00      2   0.00
##       [10,20)  89337 0.11 11.45  89339  11.45
##       [20,30) 384156 0.49 49.26 473495  60.71
##       [30,40) 272475 0.35 34.94 745970  95.65
##       [40,50)  33797 0.04  4.33 779767  99.98
##       [50,60)    142 0.00  0.02 779909 100.00
##       [60,70)      8 0.00  0.00 779917 100.00
```

Tabela completa

Retomando os resultados obtidos com as funções `table(...)` e `prop.table(...)`, podemos agrupar os resultados das tabelas de frequência absoluta e relativa em um única tabela, e obter tabelas completas, como visto nos exemplos da aula teórica.

```r
## tabela de frequência absoluta
absoluta <- table(dados$PESO_cat)

# só os valores da tabela de frequência relativa (proporção)
proporcao <- as.numeric( round( prop.table(table(dados$PESO_cat)) , 5 ) )

# só os valores da tabela de frequência relativa (porcentagem)
porcentagem <- as.numeric( 100*round( prop.table(table(dados$PESO_cat)) , 5 ) )

#Criando dataframe com as 3 tabelas:
pesos.frame<-data.frame(absoluta,proporcao,porcentagem)
pesos.frame
```

```
##    Var1   Freq proporcao porcentagem
## 1 [0,1)   5524   0.00708       0.708
## 2 [1,2)  20652   0.02648       2.648
## 3 [2,3) 239027   0.30649      30.649
## 4 [3,4) 483422   0.61986      61.986
## 5 [4,5)  30928   0.03966       3.966
## 6 [5,6)    318   0.00041       0.041
## 7 [6,7]     20   0.00003       0.003
```

```
#Editando os nomes das colunas:
names(pesos.frame) <- c('Pesos (kg)', 'Frequência', 'Proporção', 'Porcentagem')
pesos.frame
```

```
##   Pesos (kg) Frequência Proporção Porcentagem
## 1      [0,1)       5524   0.00708       0.708
## 2      [1,2)      20652   0.02648       2.648
## 3      [2,3)     239027   0.30649      30.649
## 4      [3,4)     483422   0.61986      61.986
## 5      [4,5)      30928   0.03966       3.966
## 6      [5,6)        318   0.00041       0.041
## 7      [6,7]         20   0.00003       0.003
```

```
#Adicionando linha de 'Total' :
pesos.frame[8,] <- c('Total', sum(pesos.frame[,2]),
                     round(sum(pesos.frame[,3])),
                     round(sum(pesos.frame[,4])))
```

```
## Warning in `[<-.factor`(`*tmp*`, iseq, value = "Total"): nível de fator
## inválido, NA gerado
```

```
pesos.frame
```

```
##   Pesos (kg) Frequência Proporção Porcentagem
## 1      [0,1)       5524   0.00708       0.708
## 2      [1,2)      20652   0.02648       2.648
## 3      [2,3)     239027   0.30649      30.649
## 4      [3,4)     483422   0.61986      61.986
## 5      [4,5)      30928   0.03966       3.966
## 6      [5,6)        318   0.00041       0.041
## 7      [6,7]         20     3e-05       0.003
## 8       <NA>     779891         1         100
```

Exportando a tabela

Exportar o resultado de uma tabela obtido no R pode ser útil para elaboração de um relatório, por exemplo. Após transformar a tabela em um dataframe, podemos exportá-la, por exemplo, para um arquivo .csv :

```
#Exportando dataframe (tabela de frequências) para arquivo csv:

write.table(pesos.frame, "tabela_pesos.csv", fileEncoding="latin1", sep=";")

getwd()
```

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

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.

Além disso, podemos exportar a tabela para o `latex`, o que pode ser muito útil na escrita de relatórios utilizando tal compilador. Para isso, utilizaremos o pacote `xtable` e você precisará instalá-lo antes de habilitá-lo, caso ainda não o tenha instalado em seu computador.

```
#Exportando dataframe (tabela de frequências) para o latex:

#install.packages("xtable")  # remover o # para instalar, caso necessário
library(xtable)

xtable(pesos.frame)
```

```
## % latex table generated in R 4.2.1 by xtable 1.8-4 package
## % Fri Jul 19 01:32:21 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rllll}
##   \hline
##  & Pesos (kg) & Frequência & Proporção & Porcentagem \\
##   \hline
## 1 & [0,1) & 5524 & 0.00708 & 0.708 \\
##   2 & [1,2) & 20652 & 0.02648 & 2.648 \\
##   3 & [2,3) & 239027 & 0.30649 & 30.649 \\
##   4 & [3,4) & 483422 & 0.61986 & 61.986 \\
##   5 & [4,5) & 30928 & 0.03966 & 3.966 \\
##   6 & [5,6) & 318 & 0.00041 & 0.041 \\
##   7 & [6,7] & 20 & 3e-05 & 0.003 \\
##   8 &  & 779891 & 1 & 100 \\
##    \hline
## \end{tabular}
## \end{table}
```