

Análise de Sobrevivência com o R: um breve tutorial dos pacotes `survival` e `survminer`

Material de apoio da disciplina EPI10 do PPGEPI - UFRGS

Rodrigo Citton P. dos Reis

13 de setembro de 2021

Introdução

Este tutorial tem como objetivo descrever as principais funções do pacote `survival` (THERNEAU, 2021), um dos pacotes estatísticos mais populares em análise de sobrevivência. As principais funcionalidades do pacote `survival` são:

- Organizar dados de sobrevivência na presença de censura.
- Estimar a função de sobrevivência.
- Ajustar modelos de regressão para dados de sobrevivência.
- Gerar saídas numéricas, em tabelas e gráficos.

Com relação a esta última funcionalidade, recentemente, o pacote `survminer` (KASSAMBARA; KOSINSKI; BIECEK, 2020) tem auxiliado na geração de gráficos elegantes, ricos em informações e com qualidade para publicações. Assim, sempre que for apropriado, as funções deste pacote também serão apresentadas aqui.

Antes de apresentar exemplos utilizando funções destes pacotes, a instalação do R e RStudio é brevemente descrita.

Baixando e instalando o R

Para instalação do R acesse o site <https://www.r-project.org/>:

1. Em **Download** clique em CRAN.
2. Escolha um repositório de sua preferência, por exemplo, Universidade Federal do Paraná (<https://cran-r.c3sl.ufpr.br/>).
3. Em **Download and Install R** clique no link adequado para o seu sistema operacional (no caso do Windows, clique no link **Download R for Windows**).
4. Clique no link **base** (no caso do sistema operacional ser o Windows).
5. Finalmente clique no link para baixar o arquivo executável (a versão mais atual **Download R 4.1.1 for Windows**).

Após baixar o arquivo executável, abra-o e siga as etapas de instalação conforme as [configurações padrão](#).

Baixando e instalando o RStudio

Para instalação do RStudio acesse o site <https://www.rstudio.com/products/rstudio/download/>. Em **Choose Your Version** clique no botão **Download** da versão **RStudio Desktop - Open Source License - Free**. Você será direcionado para uma seção do site que contém o botão **DOWNLOAD RSTUDIO FOR WINDOWS**. Clique neste botão para baixar a versão mais recente do instalador do RStudio para Windows (RStudio Desktop 1.4.1717).

Após baixar o arquivo executável, abra-o e siga as etapas de instalação conforme as configurações padrões. É importante salientar que o R deve ser instalado antes que o RStudio. Além disso, embora, o RStudio seja uma excelente interface gráfica para o R, com diversas funcionalidades de fácil acesso para o usuário, ele não substitui o R. Sem o R, o RStudio não consegue executar as funções de um programa R.

Instalando e carregando o pacote survival

A instalação do pacote `survival` (assim como boa parte dos mais de 18 mil pacotes do R) pode ser feita utilizando a função `install.packages`:

```
install.packages("survival")
```

Os pacotes, assim como R, são atualizados de tempos em tempos. As atualizações apresentam implementações de novos métodos, correções de antigos *bugs* e reorganizam o código do pacote para melhorar o seu desempenho computacional. Assim, é interessante que o usuário reinstale um pacote já instalado para ser contemplado com as atualizações do pacote. A periodicidade recomendada deste processo é seis meses.

Uma vez instalado o pacote, é preciso carregá-lo para que se tenha acesso às suas funções. Usualmente, a função `library` é utilizada para carregar um pacote:

```
library(survival)
```

A função `require` também pode ser utilizada para este fim.

```
require(survival)
```

É importante destacar que não é necessário reinstalar um determinado pacote sempre que uma nova sessão do R é iniciada. No entanto, é necessário carregar o pacote sempre que uma nova sessão é iniciada¹.

Uma boa prática do usuário do R é a sistemática e contínua consulta ao manual dos pacotes e funções. A função `help` acessa o manual do pacote ou função especificada:

```
help(package = "survival")
```

Organização dos dados de sobrevivência

Os dados de sobrevivência para o indivíduo i ($i = 1, \dots, n$) sob estudo, são representados, em geral, pelo par (t_i, δ_i) sendo t_i o **tempo de falha ou de censura** e δ_i a variável indicadora de falha ou censura, isto é,

$$\delta_i = \begin{cases} 1 & \text{se } t_i \text{ é um tempo de falha} \\ 0 & \text{se } t_i \text{ é um tempo de censura.} \end{cases}$$

Também é bastante comum que os conjuntos de dados contenham informações de covariáveis medidas. Assim, para o i -ésimo indivíduo, por exemplo, teríamos x_i representando o vetor de covariáveis ($sexo_i$, $idade_i$, $tratamento_i$). Dessa forma, os dados ficam representados por (t_i, δ_i, x_i) . No caso especial de dados de sobrevivência intervalar tem-se, ainda, a representação $(\ell_i, u_i, \delta_i, x_i)$ em que ℓ_i e u_i são, respectivamente, os limites inferior e superior do intervalo observado para o i -ésimo indivíduo.

Existem várias formas (funções) para carregar diversos formatos de arquivos de dados no R. Uma das mais conhecidas é função `read.table` para carregar dados do tipo texto. O arquivo `ipsec.csv` apresenta dados de provenientes de coortes hospitalares de pacientes portadores de HIV, é um conjunto de dados de exemplo apresentado em CARVALHO *et al.* (2011) e pode ser obtido em <http://sobrevida.fiocruz.br/aidsclassico.html>. É possível baixar o arquivo, e indicar o **caminho completo (diretório + arquivo com extensão)**² para o argumento `file`, ou indicar o próprio endereço *web* do arquivo.

¹A menos que o R seja configurado para que ao iniciar já carregue um certo conjunto de pacotes desejados pelo usuário. Este tutorial não apresenta esta possibilidade, mas o usuário interessado neste recurso encontrará facilmente referências para tal.

²É importante utilizar “/” (ou dupla barra invertida “\\”) para separar os diretórios e subdiretórios do arquivo.

```
ipec <- read.table(file = "http://sobrevida.fiocruz.br/dados/ipec.csv",
                  sep = ";",
                  header = TRUE)
```

Ao carregar os dados no R é recomendado armazená-los em um **objeto**. Neste caso, o objeto criado foi o `ipec` (em referência ao nome do arquivo, mas outros nomes também seriam válidos) e o operador **atribuição** (`<-`) é responsável por esta ação. Uma vez que os dados foram carregados podemos visualizar o conteúdo do objeto, com a função `View` ou `head`³:

```
# View(ipec)
```

```
# Retorna as primeiras linhas do objeto
head(ipec)
```

```
##   id  ini  fim tempo status sexo  escola  idade  risco  acompan  obito  anotrat
## 1  1 1243 2095  852     1    M      3     34     0        1     S     1991
## 2  2 2800 2923  123     1    M      2     38     6        1     S      9
## 3  3 1250 2395 1145     1    M     NA     32     0        1     S    1992
## 4  4 1915 4670 2755     0    M     NA     43     6        0     N    1992
## 5  5 2653 4770 2117     0    M     NA     40     0        1     N    1992
## 6  6      3  332  329     0    M     NA     34     0        1     I      9
##   tratam  doenca  propcp
## 1      1      4      3
## 2      0      7      4
## 3      1      3      4
## 4      1     10      4
## 5      1      5      4
## 6      0      7      0
```

Algumas outras funções que nos permitem conhecer o objeto criado são: `class`, `summary`, `str`.

```
# Retorna a classe do objeto
class(ipec)
```

```
## [1] "data.frame"
```

```
# Retorna a estrutura do objeto
str(ipec)
```

```
## 'data.frame':  193 obs. of  15 variables:
## $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
## $ ini     : int 1243 2800 1250 1915 2653 3 36 1 544 71 ...
## $ fim     : int 2095 2923 2395 4670 4770 332 96 152 2107 1318 ...
## $ tempo   : int  852 123 1145 2755 2117 329 60 151 1563 1247 ...
## $ status  : int  1 1 1 0 0 0 1 1 1 1 ...
## $ sexo    : chr  "M" "M" "M" "M" ...
## $ escola  : int  3 2 NA NA NA NA NA 0 2 2 ...
## $ idade   : int  34 38 32 43 40 34 27 22 44 23 ...
## $ risco   : int  0 6 0 6 0 0 0 6 NA 0 ...
## $ acompan: int  1 1 1 0 1 1 2 2 0 2 ...
## $ obito   : chr  "S" "S" "S" "N" ...
## $ anotrat : int 1991 9 1992 1992 1992 9 9 9 9 9 ...
## $ tratam  : int  1 0 1 1 1 0 0 0 0 0 ...
## $ doenca  : int  4 7 3 10 5 7 7 3 10 3 ...
## $ propcp  : int  3 4 4 4 4 0 0 0 0 4 ...
```

³O R não mantém uma tabela de dados brutos em tela.

```
# Retorna um conjunto de estatística resumo
summary(ipec)
```

```
##          id          ini          fim          tempo          status
## Min.   : 1   Min.   : 1   Min.   : 96   Min.   : 16.0   Min.   :0.0000
## 1st Qu.: 49   1st Qu.:1406   1st Qu.:2095   1st Qu.: 290.0   1st Qu.:0.0000
## Median : 97   Median :2454   Median :3711   Median : 852.0   Median :0.0000
## Mean   : 97   Mean   :2397   Mean   :3335   Mean   : 938.2   Mean   :0.4663
## 3rd Qu.:145   3rd Qu.:3314   3rd Qu.:4790   3rd Qu.:1348.0   3rd Qu.:1.0000
## Max.   :193   Max.   :4593   Max.   :4822   Max.   :3228.0   Max.   :1.0000
##
##          sexo          escola          idade          risco
## Length:193   Min.   :0.000   Min.   :20.00   Min.   :0.000
## Class :character 1st Qu.:0.000   1st Qu.:30.00   1st Qu.:0.000
## Mode  :character Median :1.000   Median :35.00   Median :0.000
##              Mean   :1.242   Mean   :36.55   Mean   :1.506
##              3rd Qu.:2.000   3rd Qu.:43.00   3rd Qu.:3.000
##              Max.   :3.000   Max.   :68.00   Max.   :6.000
##              NA's   :11      NA's   :37
##
##      accompan      obito      anotrat      tratam
## Min.   :0.0000   Length:193   Min.   : 9   Min.   :0.000
## 1st Qu.:0.0000   Class :character 1st Qu.:1990   1st Qu.:1.000
## Median :1.0000   Mode  :character Median :1993   Median :1.000
## Mean   :0.8964           Mean   :1542   Mean   :1.098
## 3rd Qu.:1.0000           3rd Qu.:1996   3rd Qu.:2.000
## Max.   :2.0000           Max.   :2000   Max.   :3.000
##
##          doenca          propcp
## Min.   : 1.00   Min.   :0.000
## 1st Qu.: 3.00   1st Qu.:2.000
## Median : 8.00   Median :4.000
## Mean   :37.96   Mean   :2.948
## 3rd Qu.:99.00   3rd Qu.:4.000
## Max.   :99.00   Max.   :4.000
##
```

Note que o objeto `ipec` é um objeto do tipo `data.frame` que é uma **coleção de vetores colunas**, em que cada um destes pode ser um vetor de um tipo diferente: `numeric`, `integer`, `factor`, `character`, etc. Ou seja, um objeto do tipo `data.frame` pode ser utilizado para representar uma tabela de dados brutos, em que nas linhas armazenos a informação com respeito ao indivíduo (unidade de observação) e nas colunas armazenos a informação referente às variáveis (que podem ser contínuas, discretas, ordinais ou nominais). No caso do objeto `ipec`, temos 193 observações (indivíduos) e 15 variáveis, sendo que a variável `id` representa o identificador de cada unidade observacional.

É possível alterar o tipo de uma variável no `data.frame` combinando o operador `<-`, `$`⁴ e a função correspondente a nova classe:

```
# Transforma a variável sexo em uma variável tipo factor
ipec$sexo <- factor(ipec$sexo)

summary(ipec$sexo)
```

```
##    F    M
## 49 144
```

⁴O `$` é como o possessivo (em inglês) “’s”; `ipec$idade` é a variável `idade` de `ipec`.

```
summary(ipec$idade)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    20.00  30.00   35.00   36.55  43.00   68.00
```

```
table(ipec$status)
```

```
##
##      0      1
## 103    90
```

Estes dados representam uma coorte de 49 mulheres e 144 homens, com idades entre 20 e 68 anos. Durante os quatro anos de estudo, foram registradas 103 censuras e 90 eventos (óbito). Todas as censuras são classificadas como censuras à direita, já que resultam de perda de acompanhamento, seja por abandono ou término do tempo de observação (CARVALHO *et al.*, 2011).

As variáveis `tempo` e `status` representam as variáveis de tempo até o evento e a indicadora de falha ou censura, respectivamente, formando o “desfecho-em-par” (t_i, δ_i) . O início do acompanhamento era definido pelo diagnóstico de Aids e era contado em dias. Este conjunto de dados registrou tempo de início e de fim do acompanhamento, o que permite também a representação do desfecho como (ℓ_i, u_i, δ_i) . As demais variáveis podem ser utilizadas nas análises para compor o vetor de covariáveis $x_i = (sexo_i, escola_i, idade_i, \dots)$.

A representação do desfecho de sobrevivência (t_i, δ_i) no R pode ser feita com a função `Surv` do pacote `survival`:

```
Surv(time = ipec$tempo, event = ipec$status)
```

```
##      [1] 852  123 1145 2755+ 2117+ 329+  60  151 1563 1247  84  214
##     [13] 25+ 1348  158  555  408 1116 998 1125 944+  54 151 855
##     [25] 116   80+ 1757+ 194  183  37 237+ 1506 168+ 134 803+  18
##     [37] 371  173 688  163 3178+  29  50+ 887  516 645 310 204
##     [49] 1344+ 1261 285  83  150 1307+ 1076+ 1226 865+ 811 2898  80
##     [61] 967  618 235 2236+ 152 892  81+ 1085 1073+ 1615+  35 290
##     [73] 1780+ 3228+  52  733 3213+ 1983 2304+ 572  21 1272+ 1646+ 304
##     [85] 418  854 2973+  40 850 1139 323 1507+ 2717+ 1735+ 388+ 145
##     [97] 905  927 1027+ 631 2495+ 1331+ 623 2568+ 2013+ 721 1952+ 397
##    [109] 254 1630+ 1523+ 146+ 108 1835+ 499 333 202+ 2437+ 1015 2138+
##    [121]  22 2090+ 179 2439+ 1063+ 85+ 343+ 2215+ 259 2258+ 1371  39
##    [133] 2371+ 975+ 952 2492+ 1478+ 295+ 992 1011+ 644 426 537+ 1454+
##    [145] 1869+ 714+ 1310+ 2084+ 1918+ 1649+ 290+ 1685+ 1348+ 652+ 1384+ 1471+
##    [157] 1512+ 378+ 1352+ 419 1426+ 1488+ 1315+ 643+ 1197+ 1343+ 1176+ 944
##    [169] 340 881+ 915+ 948+ 985+ 1242+ 955+ 987+ 899+ 1056+ 775 785+
##    [181] 731+ 16 680+ 21+ 444+ 524+ 217+ 440+ 470+ 390+ 344+ 578+
##    [193] 504+
```

Não é necessário armazenar esta “variável” no objeto `data.frame`. Esta função será utilizada em combinação com diversas outras funções do pacote `survival` para representar o desfecho. Outra forma (neste caso) de representar o desfecho é pelo trio “tempo de início - tempo final - indicadora de falha e censura” (ℓ_i, u_i, δ_i) :

```
Surv(time = ipec$ini, time2 = ipec$fim, event = ipec$status)
```

```
##      [1] (1243,2095] (2800,2923] (1250,2395] (1915,4670+] (2653,4770+]
##      [6] (  3, 332+] ( 36, 96] (  1, 152] ( 544,2107] ( 71,1318]
##     [11] ( 946,1030] ( 802,1016] ( 266, 291+] (1544,2892] ( 57, 215]
##     [16] (1270,1825] (2753,3161] ( 940,2056] ( 393,1391] (1000,2125]
##     [21] ( 238,1182+] ( 423, 477] ( 206, 357] ( 480,1335] ( 226, 342]
##     [26] ( 249, 329+] (3052,4809+] (1802,1996] (1395,1578] ( 354, 391]
##     [31] ( 493, 730+] (1113,2619] ( 638, 806+] ( 655, 789] (1189,1992+]
```

```
## [36] ( 943, 961] (1715,2086] ( 792, 965] (1037,1725] ( 820, 983]
## [41] ( 884,4062+] (2262,2291] (1121,1171+] (1131,2018] ( 878,1394]
## [46] (1316,1961] (1107,1417] (1190,1394] ( 393,1737+] (1274,2535]
## [51] (1172,1457] (2360,2443] (2074,2224] (1019,2326+] ( 605,1681+]
## [56] (1915,3141] (3948,4813+] (1314,2125] (1502,4400] (1347,1427]
## [61] (1379,2346] (2352,2970] (2625,2860] (2586,4822+] (1406,1558]
## [66] (1466,2358] (3314,3395+] (3413,4498] (3712,4785+] (3207,4822+]
## [71] (1592,1627] (1537,1827] (3018,4798+] (1555,4783+] (1541,1593]
## [76] (1589,2322] (1609,4822+] (1682,3665] (2465,4769+] (1243,1815]
## [81] (1667,1688] (1605,2877+] (3157,4803+] (2066,2370] (1929,2347]
## [86] (2216,3070] (1809,4782+] (1670,1710] (1983,2833] (2883,4022]
## [91] (1766,2089] (3313,4820+] (1977,4694+] (3087,4822+] (2286,2674+]
## [96] (1877,2022] (1852,2757] (1549,2476] (3795,4822+] (2475,3106]
## [101] (2310,4805+] (2870,4201+] (1935,2558] (2199,4767+] (2800,4813+]
## [106] (2990,3711] (2857,4809+] (3586,3983] (2143,2397] (3124,4754+]
## [111] (3276,4799+] (2208,2354+] (2209,2317] (2976,4811+] (2626,3125]
## [116] (3838,4171] (2314,2516+] (2311,4748+] (2280,3295] (2684,4822+]
## [121] (2454,2476] (2713,4803+] (2311,2490] (2370,4809+] (3756,4819+]
## [126] (2565,2650+] (2599,2942+] (2553,4768+] (2601,2860] (2553,4811+]
## [131] (2726,4097] (2739,2778] (2447,4818+] (3830,4805+] (2429,3381]
## [136] (2311,4803+] (3299,4777+] (4510,4805+] (2384,3376] (3749,4760+]
## [141] (2676,3320] (2985,3411] (4192,4729+] (3159,4613+] (2921,4790+]
## [146] (4078,4792+] (2934,4244+] (2645,4729+] (2857,4775+] (3173,4822+]
## [151] (4509,4799+] (3082,4767+] (3465,4813+] (3188,3840+] (3271,4655+]
## [156] (3276,4747+] (3287,4799+] (4439,4817+] (3446,4798+] (3305,3724]
## [161] (3391,4817+] (3307,4795+] (3425,4740+] (4117,4760+] (3612,4809+]
## [166] (3479,4822+] (3572,4748+] (3796,4740] (3527,3867] (3921,4802+]
## [171] (3798,4713+] (3808,4756+] (3772,4757+] (3557,4799+] (3867,4822+]
## [176] (3594,4581+] (3923,4822+] (3733,4789+] (4019,4794] (4033,4818+]
## [181] (4040,4771+] (4053,4069] (4137,4817+] (4208,4229+] (4362,4806+]
## [186] (4279,4803+] (4593,4810+] (4320,4760+] (4343,4813+] (4419,4809+]
## [191] (4406,4750+] (4199,4777+] (4301,4805+]
```

Duas observações merecem ser feitas:

1. Por padrão (*default*), a função `Surv` considera o `Status = 0` como uma **censura**, e `Status = 1` como uma **falha** (ocorrência do evento). Caso, o conjunto de dados apresentasse uma configuração distinta, poderíamos informar o código referente ao evento na própria função⁵. Por exemplo, se o óbito fosse codificado com o número 0, então a função deveria ser especificada da seguinte maneira: `Surv(time = ipec$tempo, event = ipec$status == 0)`; se o óbito fosse codificado pela letra S, então a função deveria ser especificada como: `Surv(time = ipec$tempo, event = ipec$status == 'S')`.
2. Por padrão, a função `Surv` especifica a censura como sendo uma censura à direita. No caso em que são analisados dados com censura à esquerda ou intervalar, o argumento `type` deve ser configurado apropriadamente (`Surv(tempo, cens, type = "left")` no caso de censura à esquerda; consulte o `help` da função para mais opções).

Estimação da curva de sobrevivência

A estimação da função de sobrevivência $S(t) = \Pr(T > t)$ pode ser obtida pela função `survfit` do pacote `survival`. A função `survfit` pode ser utilizada para obtenção da estimativa de diversas funções que caracterizam o tempo até o evento. No caso de um evento simples (um único evento por observação) e censura à direita, as estimativas $\hat{S}(t)$ para $S(t)$ são geradas pelo estimador de Kaplan-Meier. Para a estimativa da sobrevivência de toda a coorte *IPEC*, podemos especificar a função `survival` da seguinte forma:

⁵ Assim não é necessária a formatação prévia do conjunto de dados para esta finalidade.

```
ekm <- survfit(Surv(time = tempo, event = status) ~ 1,
               data = ipec,
               conf.type = "log-log")
```

O objeto criado `ekm` armazena a saída (*output*) da função `survival`. Podemos acessar a saída (ou partes desta) com as funções `print`, `summary`⁶ e `plot`

```
print(ekm)
```

```
## Call: survfit(formula = Surv(time = tempo, event = status) ~ 1, data = ipec,
##      conf.type = "log-log")
##
##      n  events  median 0.95LCL 0.95UCL
##    193     90   1247    967     NA
```

```
summary(ekm)
```

```
## Call: survfit(formula = Surv(time = tempo, event = status) ~ 1, data = ipec,
##      conf.type = "log-log")
##
##   time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    16    193      1   0.995 0.00517    0.964    0.999
##    18    192      1   0.990 0.00729    0.959    0.997
##    21    191      1   0.984 0.00890    0.953    0.995
##    22    189      1   0.979 0.01027    0.946    0.992
##    29    187      1   0.974 0.01147    0.939    0.989
##    35    186      1   0.969 0.01255    0.932    0.986
##    37    185      1   0.964 0.01353    0.925    0.982
##    39    184      1   0.958 0.01443    0.918    0.979
##    40    183      1   0.953 0.01527    0.912    0.975
##    52    181      1   0.948 0.01607    0.905    0.972
##    54    180      1   0.943 0.01682    0.899    0.968
##    60    179      1   0.937 0.01753    0.892    0.964
##    80    178      1   0.932 0.01821    0.886    0.960
##    83    175      1   0.927 0.01887    0.879    0.956
##    84    174      1   0.921 0.01950    0.873    0.952
##   108    172      1   0.916 0.02011    0.867    0.948
##   116    171      1   0.911 0.02069    0.860    0.943
##   123    170      1   0.905 0.02125    0.854    0.939
##   134    169      1   0.900 0.02179    0.848    0.935
##   145    168      1   0.895 0.02231    0.841    0.931
##   150    166      1   0.889 0.02282    0.835    0.926
##   151    165      2   0.878 0.02378    0.823    0.917
##   152    163      1   0.873 0.02423    0.817    0.913
##   158    162      1   0.868 0.02468    0.810    0.909
##   163    161      1   0.862 0.02511    0.804    0.904
##   173    159      1   0.857 0.02553    0.798    0.900
##   179    158      1   0.851 0.02593    0.792    0.895
##   183    157      1   0.846 0.02633    0.786    0.890
##   194    156      1   0.841 0.02671    0.780    0.886
##   204    154      1   0.835 0.02709    0.774    0.881
```

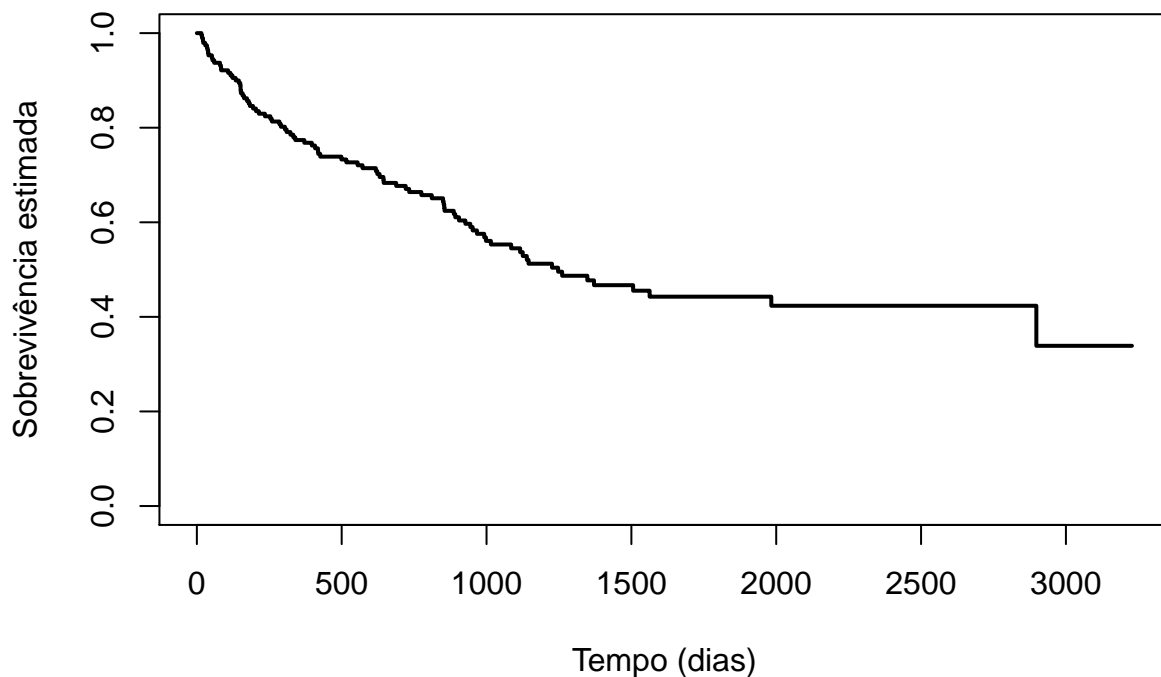
⁶Veja que função `summary` já foi utilizada para retornar as estatísticas resumo de variáveis de um `data.frame` ou de um vetor, e agora é utilizada para retornar a saída de um objeto tipo `survfit`. Esta função exemplifica que uma função no R retorna uma saída de acordo com tipo de objeto de entrada (ou seja, no R, os modos e classes de dados definem os atributos fundamentais e o comportamento de um objeto de dados).

##	214	153	1	0.830	0.02746	0.768	0.876
##	235	151	1	0.824	0.02782	0.762	0.872
##	254	149	1	0.819	0.02818	0.755	0.867
##	259	148	1	0.813	0.02853	0.749	0.862
##	285	147	1	0.808	0.02886	0.743	0.857
##	290	146	1	0.802	0.02919	0.737	0.852
##	304	143	1	0.796	0.02952	0.731	0.848
##	310	142	1	0.791	0.02984	0.725	0.843
##	323	141	1	0.785	0.03015	0.719	0.838
##	333	139	1	0.780	0.03046	0.713	0.833
##	340	138	1	0.774	0.03076	0.707	0.828
##	371	135	1	0.768	0.03106	0.700	0.823
##	397	131	1	0.762	0.03137	0.694	0.817
##	408	130	1	0.756	0.03167	0.688	0.812
##	418	129	1	0.751	0.03197	0.681	0.807
##	419	128	1	0.745	0.03225	0.675	0.802
##	426	127	1	0.739	0.03252	0.669	0.796
##	499	123	1	0.733	0.03281	0.662	0.791
##	516	121	1	0.727	0.03309	0.656	0.786
##	555	118	1	0.721	0.03338	0.649	0.780
##	572	117	1	0.714	0.03366	0.642	0.775
##	618	115	1	0.708	0.03393	0.636	0.769
##	623	114	1	0.702	0.03420	0.629	0.763
##	631	113	1	0.696	0.03446	0.623	0.758
##	644	111	1	0.690	0.03471	0.616	0.752
##	645	110	1	0.683	0.03496	0.609	0.746
##	688	107	1	0.677	0.03521	0.602	0.740
##	721	105	1	0.670	0.03546	0.596	0.735
##	733	103	1	0.664	0.03571	0.589	0.729
##	775	102	1	0.657	0.03595	0.582	0.723
##	811	99	1	0.651	0.03619	0.575	0.717
##	850	98	1	0.644	0.03643	0.568	0.710
##	852	97	1	0.638	0.03665	0.561	0.704
##	854	96	1	0.631	0.03687	0.554	0.698
##	855	95	1	0.624	0.03707	0.547	0.692
##	887	92	1	0.617	0.03728	0.540	0.686
##	892	91	1	0.611	0.03749	0.533	0.679
##	905	89	1	0.604	0.03769	0.526	0.673
##	927	87	1	0.597	0.03789	0.519	0.667
##	944	86	1	0.590	0.03808	0.511	0.660
##	952	83	1	0.583	0.03828	0.504	0.654
##	967	81	1	0.576	0.03848	0.496	0.647
##	992	77	1	0.568	0.03869	0.489	0.640
##	998	76	1	0.561	0.03890	0.481	0.633
##	1015	74	1	0.553	0.03911	0.473	0.626
##	1085	68	1	0.545	0.03937	0.465	0.618
##	1116	67	1	0.537	0.03961	0.456	0.611
##	1125	66	1	0.529	0.03984	0.448	0.603
##	1139	65	1	0.521	0.04005	0.439	0.596
##	1145	64	1	0.512	0.04024	0.431	0.588
##	1226	61	1	0.504	0.04045	0.422	0.580
##	1247	59	1	0.495	0.04065	0.414	0.572
##	1261	58	1	0.487	0.04084	0.405	0.564
##	1348	50	1	0.477	0.04117	0.395	0.555

##	1371	47	1	0.467	0.04152	0.384	0.546
##	1506	40	1	0.455	0.04210	0.371	0.535
##	1563	36	1	0.443	0.04278	0.358	0.524
##	1983	23	1	0.423	0.04505	0.334	0.510
##	2898	5	1	0.339	0.08389	0.184	0.501

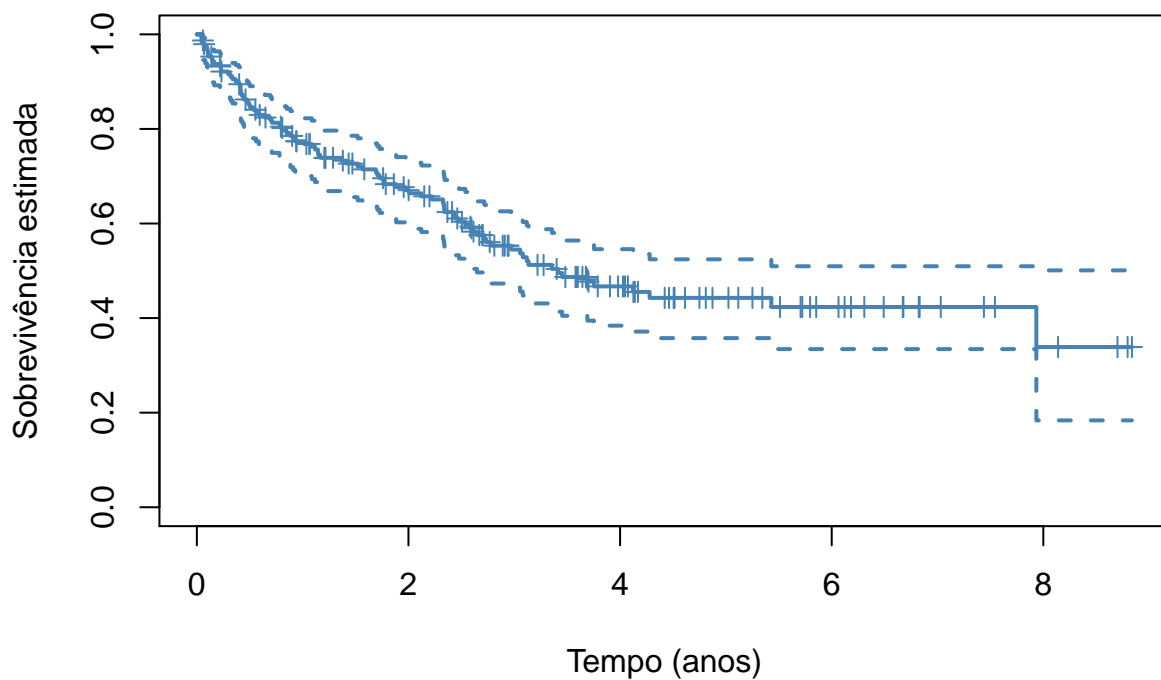
Como pode ser visto, a função `summary` apresenta uma tabela com os tempos distintos de falha (`time`), o número de indivíduos em risco em cada respectivo tempo (`n.risk`), o número de eventos no tempo de falha (`n.event`), a sobrevivência estimada pelo Kaplan-Meier naquele tempo (`survival`), o respectivo erro padrão estimado (`std.err`), e os limites inferior (`lower 95% CI`) e superior (`upper 95% CI`) do intervalo de confiança de 95%. Para a obtenção de intervalos com outros níveis de confiança é necessário especificar o argumento `conf.int` na função `survfit` (`conf.int = 0.90` para intervalos de confiança de 90%). A função `plot` desenha o gráfico da sobrevivência estimada pela função `survfit`:

```
plot(ekm, conf.int = FALSE,
     lwd = 2, xlab = "Tempo (dias)",
     ylab = "Sobrevivência estimada")
```



A mesma função pode nos apresentar diferentes versões do gráfico, quando seus argumentos são especificados para tal:

```
plot(ekm, conf.int = TRUE,
     mark.time = TRUE,
     lwd = 2, col = "steelblue",
     xscale = 365.25,
     xlab = "Tempo (anos)",
     ylab = "Sobrevivência estimada")
```



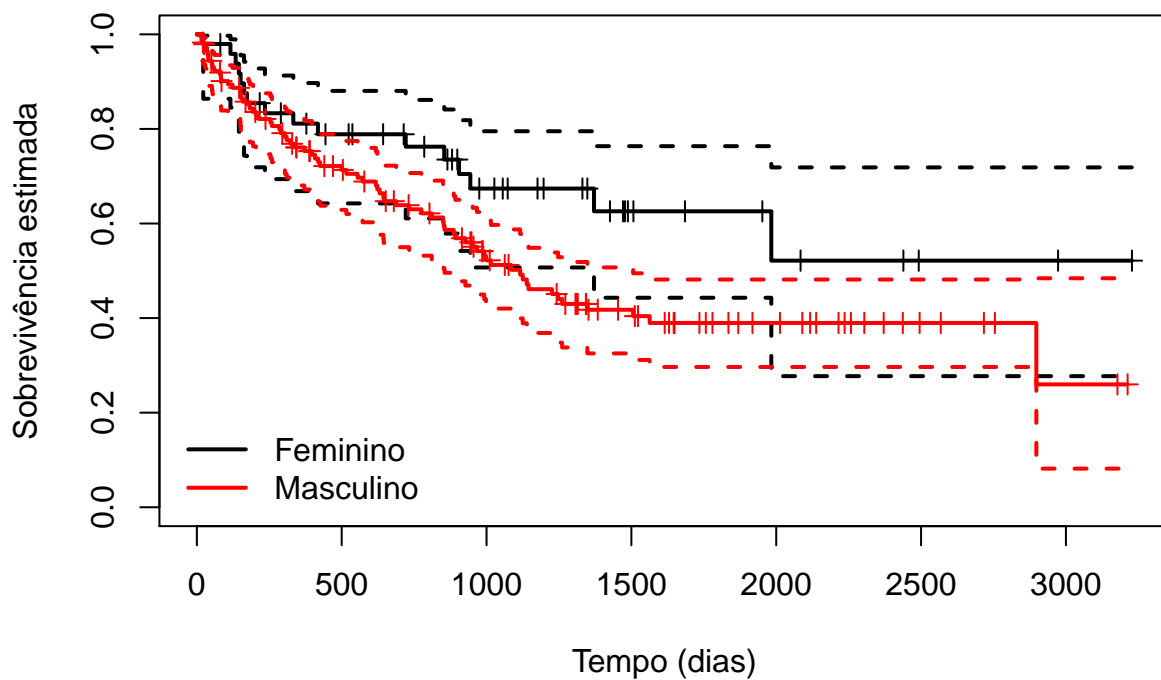
No gráfico acima, as linhas tracejadas representam a banda formada pelos intervalos de confiança (`conf.int = TRUE`), e as pequenas marcas na curva estimada representam os tempos de censura do estudo (`mark.time = TRUE`).

A função `survfit` também pode ser utilizada para estimar a sobrevivência de subgrupos da população de estudo:

```
ekm <- survfit(Surv(time = tempo, event = status) ~ sexo,
               data = ipec,
               conf.type = "log-log")

plot(ekm, conf.int = TRUE,
     mark.time = TRUE,
     col = c("black", "red"),
     lwd = 2, xlab = "Tempo (dias)",
     ylab = "Sobrevivência estimada")

legend("bottomleft",
      c("Feminino", "Masculino"),
      col = c("black", "red"),
      lwd = 2, bty = "n")
```



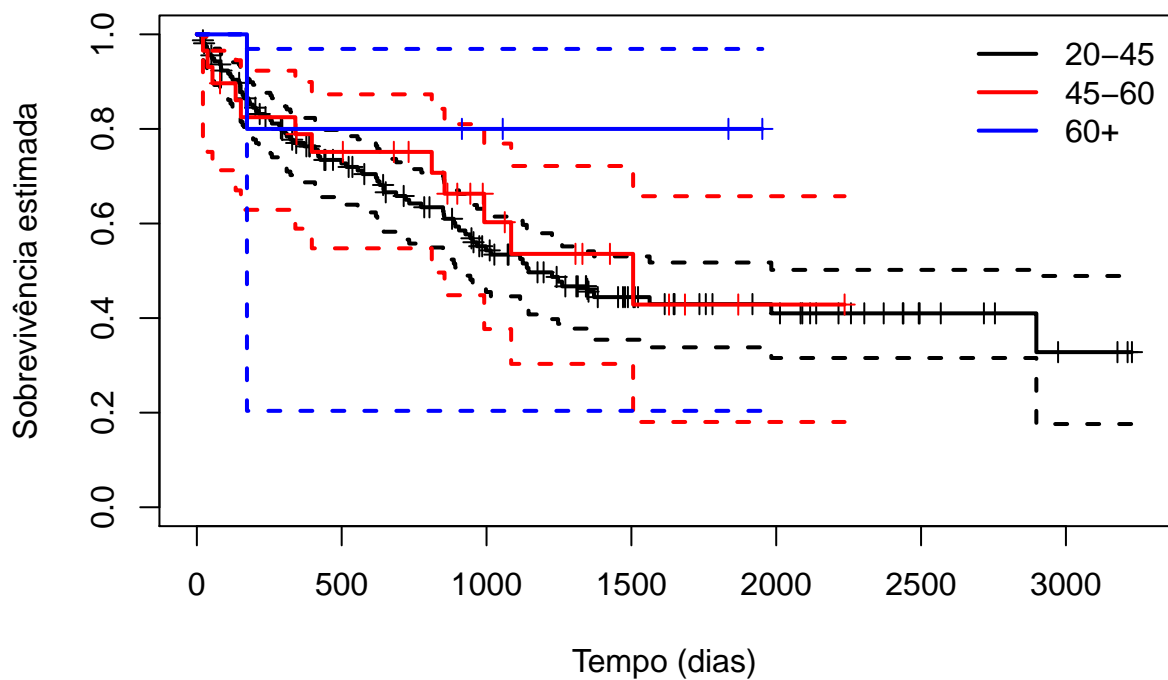
A variável de estrato pode possuir mais que duas categorias:

```
ipec$idade.cat <- cut(x = ipec$idade,
                      breaks = c(20, 45, 60, Inf),
                      include.lowest = T,
                      labels = c("20-45", "45-60", "60+"))

ekm <- survfit(Surv(time = tempo, event = status) ~ idade.cat,
              data = ipec,
              conf.type = "log-log")

plot(ekm, conf.int = TRUE,
     mark.time = TRUE,
     col = c("black", "red", "blue"),
     lwd = 2, xlab = "Tempo (dias)",
     ylab = "Sobrevivência estimada")

legend("topright",
     levels(ipec$idade.cat),
     col = c("black", "red", "blue"),
     lwd = 2, bty = "n")
```

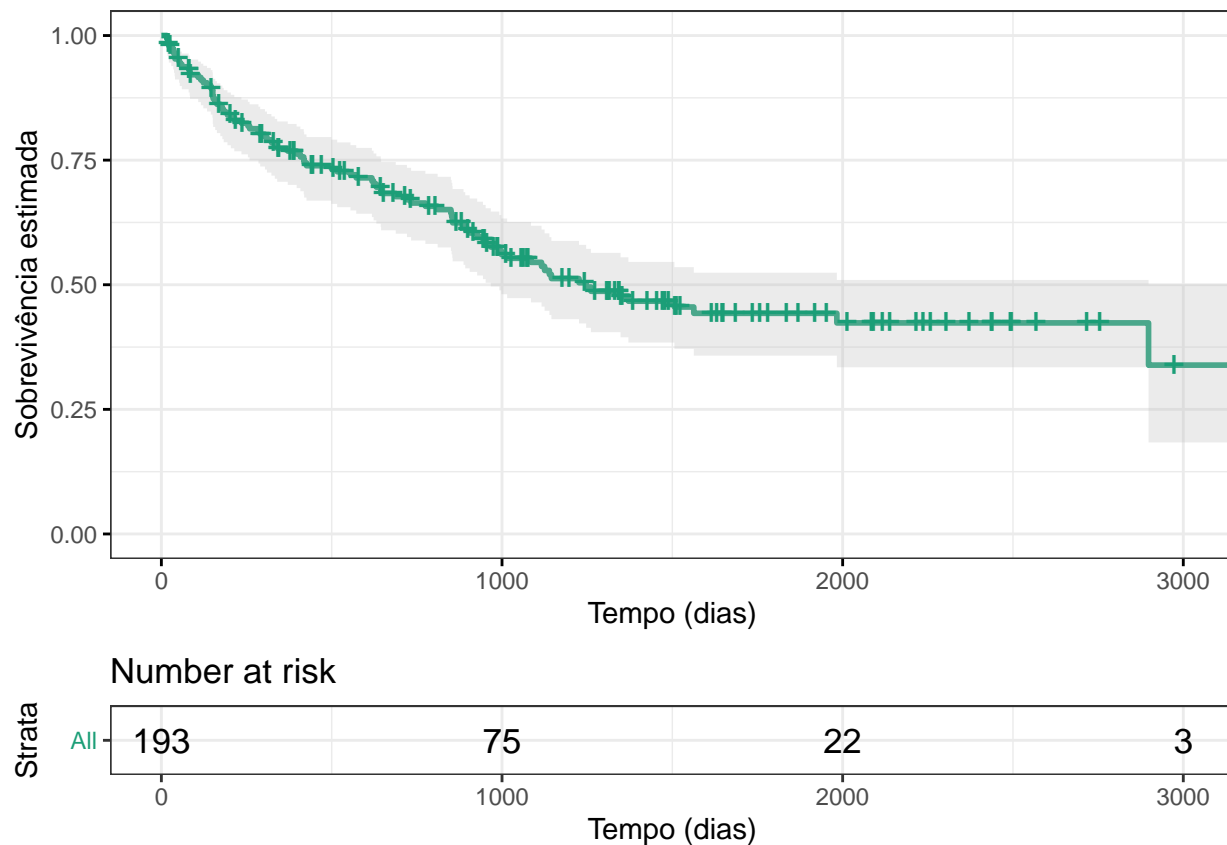


O pacote `survminer` apresenta uma forma gráfica alternativa (no formato dos gráficos `ggplot`), e ainda possibilita a criação de tabela de indivíduos em risco (muito utilizada nas publicações de estudos de sobrevivência):

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

ekm <- survfit(Surv(time = tempo, event = status) ~ 1,
               data = ipec,
               conf.type = "log-log")

ggsurvplot(ekm,
            data = ipec,
            risk.table = TRUE,
            conf.int = TRUE,
            xlab = "Tempo (dias)",
            ylab = "Sobrevivência estimada",
            palette = "Dark2",
            legend = "none",
            ggtheme = theme_bw())
```



Referências

CARVALHO, M. S. *et al.* **Análise de Sobrevivência: teoria e aplicações em saúde**. 2. ed. Rio de Janeiro: Editora Fiocruz, 2011.

KASSAMBARA, A.; KOSINSKI, M.; BIECEK, P. **survminer: Drawing Survival Curves using 'ggplot2'**. [S. l.: s. n.]. Disponível em: <https://CRAN.R-project.org/package=survminer>

THERNEAU, T. M. **A Package for Survival Analysis in R**. [S. l.: s. n.]. Disponível em: <https://CRAN.R-project.org/package=survival>