Análise de Sobrevivência

0.2 - Aula Prática

Prof. Dr. Eder Angelo Milani

04/04/2025

Manipulação dos dados da FOSP

As linhas de cógido a seguir executam as seguintes tarefas: - leitura dos dados completo - filtra os dados para CID C34 - cria uma variável do ano da última informação - calcula o tempo do diagnóstico até a última informação - variável tempo - constroi a variável censura - filtra os dados para algumas variáveis - salva o conjuto de dados após manipulações

```
# limpando o que tem na memoria
rm(list=ls())

# local onde esta o arquivo com os dados
setwd("G:\\Meu Drive\\UFG\\Especializacao\\Aulas Análise Sobrevivência\\Códigos")

### leitura
dados <- read.csv("dados_convertidos.csv")

# dimensao do conjunto de dados
dim(dados)
```

```
## [1] 1257217     104
```

```
# manipulacao do conjunto de dados
#install.packages(tidyverse)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
### pacotes para trabalhar com datas
#install.packages(devtools)
#install.packages(lubridate)
library(devtools)
```

```
## Carregando pacotes exigidos: usethis
```

```r
library(lubridate)

### Filtro por tipo de cancer

## CID - C34 (neoplasia maligna dos bronquios e dos pulmoes)

dados_c34 <- dados %>% filter(TOPOGRUP == "C34")
dim(dados_c34)
```

```
## [1] 58598   104
```

```r
# observe como diminuiu o conjunto de dados!!!

## pacientes diagnosticados entre 2014 e 2016, com segimento ate 2021
## Anos com registro de casos em andamento: 2022, 2023 e 2024

table(dados_c34$ANODIAG)
```

```
##
## 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
## 1694 1671 1744 1920 2058 2168 1966 1920 2203 2406 2378 2591 2619 2858 2996 2933
## 2016 2017 2018 2019 2020 2021 2022 2023 2024
## 3002 2999 2901 3000 2682 2519 2524 2279  567
```

```r
## vamos criar uma variavel que e o ano da ultima informacao

dados_c34 <- dados_c34 %>% mutate(`ANOULTINF`=year(DTULTINFO))
table(dados_c34$ANOULTINF)
```

```
##
## 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
##  702 1262 1438 1690 1788 1884 1831 1722 2016 2119 2229 2237 2369 2529 2622 2789
## 2016 2017 2018 2019 2020 2021 2022 2023 2024
## 2750 2803 2914 2925 2876 3073 2962 3255 3813
```

```r
dados_final <- dados_c34 %>%
  filter(ANODIAG >= 2014 & ANODIAG <= 2016)

dim(dados_final)
```

```
## [1] 8931  105
```

```r
#head(dados_final)

# vamos agora criar a variavel tempo e a variavel censura

### calculo do tempo

dados_final$TEMPO <- ifelse(dados_final$ANOULTINF <= 2021,
                            (ymd(dados_final$DTDIAG) %--%ymd(dados_final$DTULTINFO))/ddays(1),
                            (ymd(dados_final$DTDIAG) %--%ymd("2021-12-31"))/ddays(1))

dim(dados_final)
```

```
## [1] 8931  106
```

```r
summary(dados_final$TEMPO)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    80.0   253.0   543.7   690.0  2915.0
```

```r
## calculo da censura
### 1 - VIVO, COM CÂNCER / 2 - VIVO, SOE /
### 3 - OBITO POR CANCER / 4 - OBITO POR OUTRAS CAUSAS, SOE
table(dados_final$ULTINFO)
```

```
##
##    1    2    3    4
##  250  729 6950 1002
```

```r
dados_final$CENSURA <- ifelse(dados_final$ULTINFO==3 & dados_final$ANOULTINF <= 2021, 1, 0)

table(dados_final$CENSURA)
```

```
##
##    0    1
## 2022 6909
```

```r
#### formatacao dos dados para analise

dados <- dados_final %>%
        select(TOPOGRUP, TEMPO, CENSURA, ANODIAG, IDADE, SEXO, CIRURGIA, RADIO, QUIMIO, ECGRUP)

head(dados)
```

```
##   TOPOGRUP TEMPO CENSURA ANODIAG IDADE SEXO CIRURGIA RADIO QUIMIO ECGRUP
## 1      C34   292       1    2014    63    1        0     1      1    III
## 2      C34   132       1    2016    58    2        0     0      0      I
## 3      C34     3       0    2016    61    2        0     0      0     IV
## 4      C34    17       1    2016    67    1        0     0      0     IV
## 5      C34   182       1    2015    57    1        0     0      1    III
## 6      C34   287       1    2015    69    1        0     0      1     IV
```

```r
### Salvando os dados filtrados

f.out <- 'G:\\Meu Drive\\UFG\\Especializacao\\Aulas Análise Sobrevivência\\Códigos\\cancer_c34.csv'

write.csv(dados, f.out, row.names = F)
```

Estimador de Kaplan-Meier

As linhas de cógido a seguir executam as seguintes tarefas: - leitura dos dados filtrados - calcula o estimador de Kaplan-Meier - faz o plot da função de sobrevivência estimada por Kaplan-Meier - calcula o estimador de Kaplan-Meier considerando a variável sexo - faz o plot da função de sobrevivência estimada por Kaplan-Meier considerando a variável sexo

```r
# limpando o que tem na memoria
rm(list=ls())

# local onde esta o arquivo com os dados
setwd("G:\\Meu Drive\\UFG\\Especializacao\\Aulas Análise Sobrevivência\\Códigos")

### leitura
```

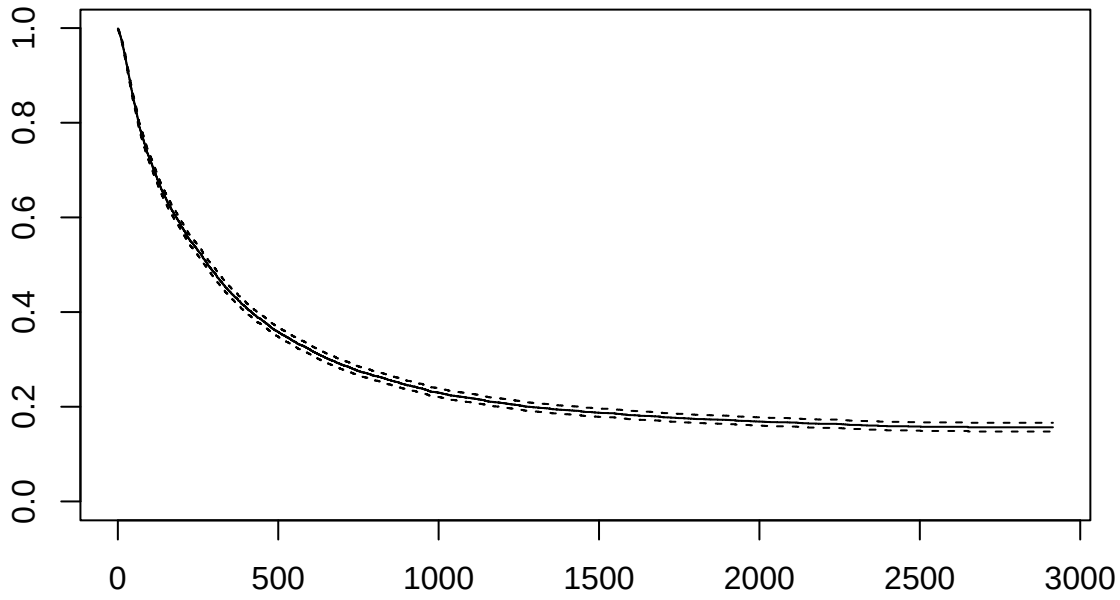
```r
dados <- read.csv("cancer_c34.csv")
head(dados)
```

```
##   TOPOGRUP TEMPO CENSURA ANODIAG IDADE SEXO CIRURGIA RADIO QUIMIO ECGRUP
## 1      C34   292       1    2014    63    1        0     1      1    III
## 2      C34   132       1    2016    58    2        0     0      0      I
## 3      C34     3       0    2016    61    2        0     0      0     IV
## 4      C34    17       1    2016    67    1        0     0      0     IV
## 5      C34   182       1    2015    57    1        0     0      1    III
## 6      C34   287       1    2015    69    1        0     0      1     IV
```

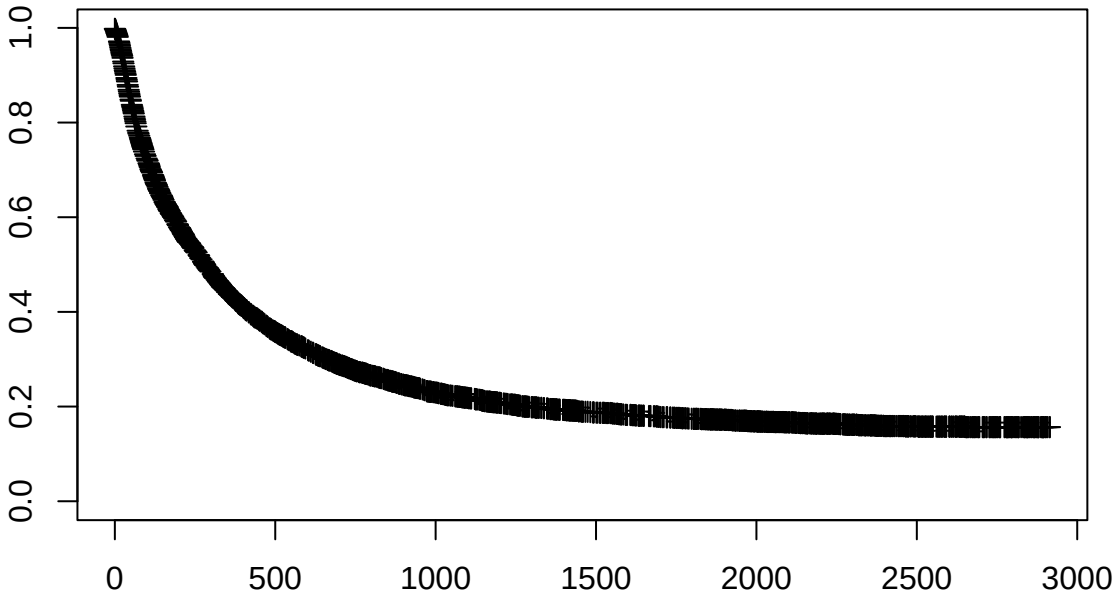
```r
# Estimador de Kaplan-Meier

#O objetivo é obter a estimativa de kaplan-Meier do conjunto de dados de cancer de pulmao (CID C34)

## Utilizando o plot comum

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

```
## Carregando pacotes exigidos: survival
```

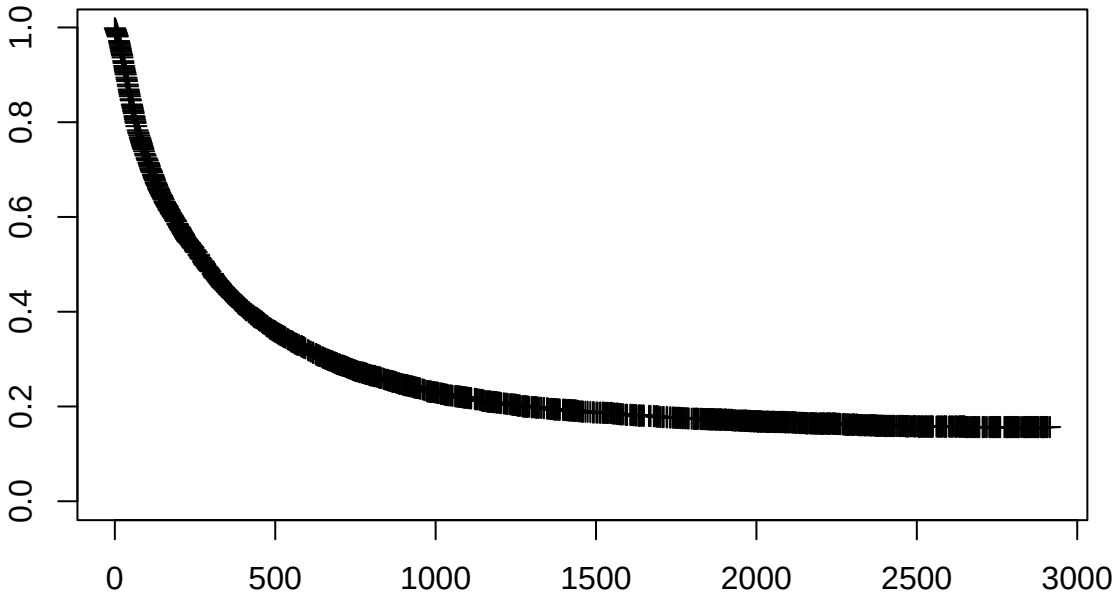
```r
ekm <- survfit(Surv(TEMPO, CENSURA) ~ 1, data=dados)
#summary(ekm)
plot(ekm)
```



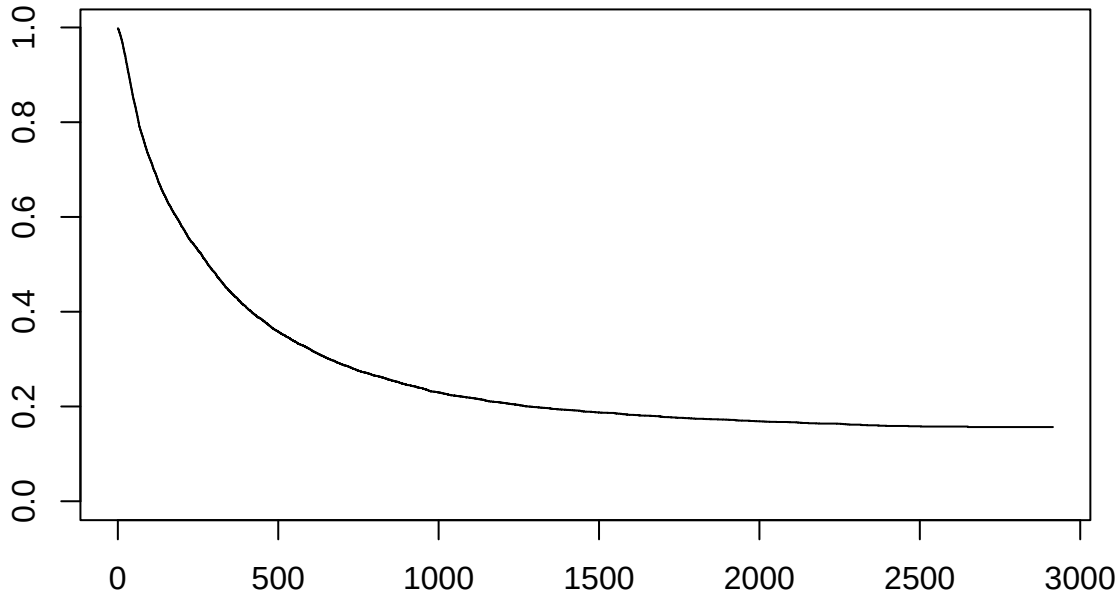

```r
plot(ekm, mark.time = T)
```



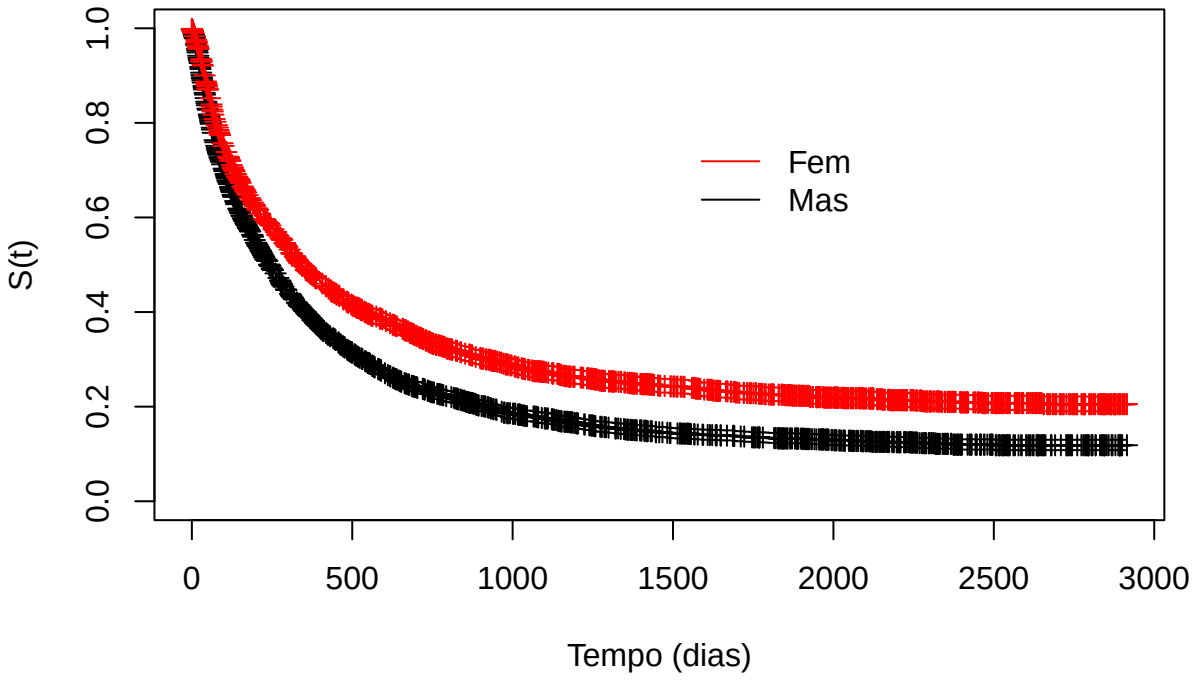

```
plot(ekm, mark.time = T, conf.int = F)
```



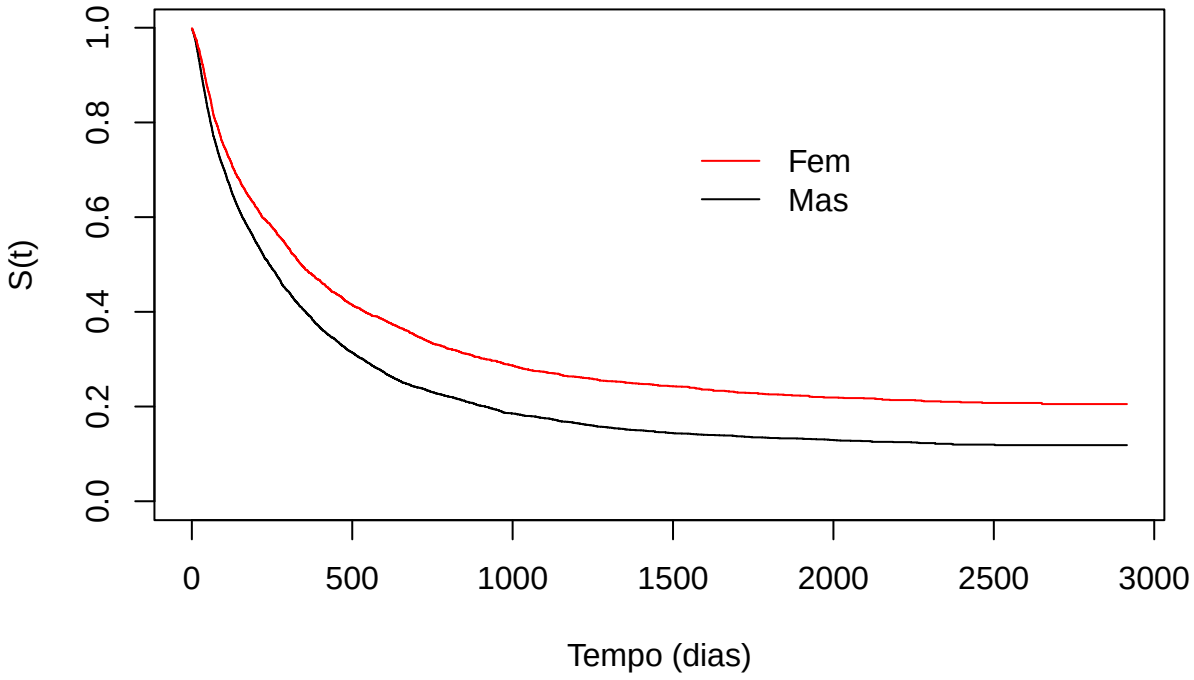

```
plot(ekm, mark.time = F, conf.int = F)
```



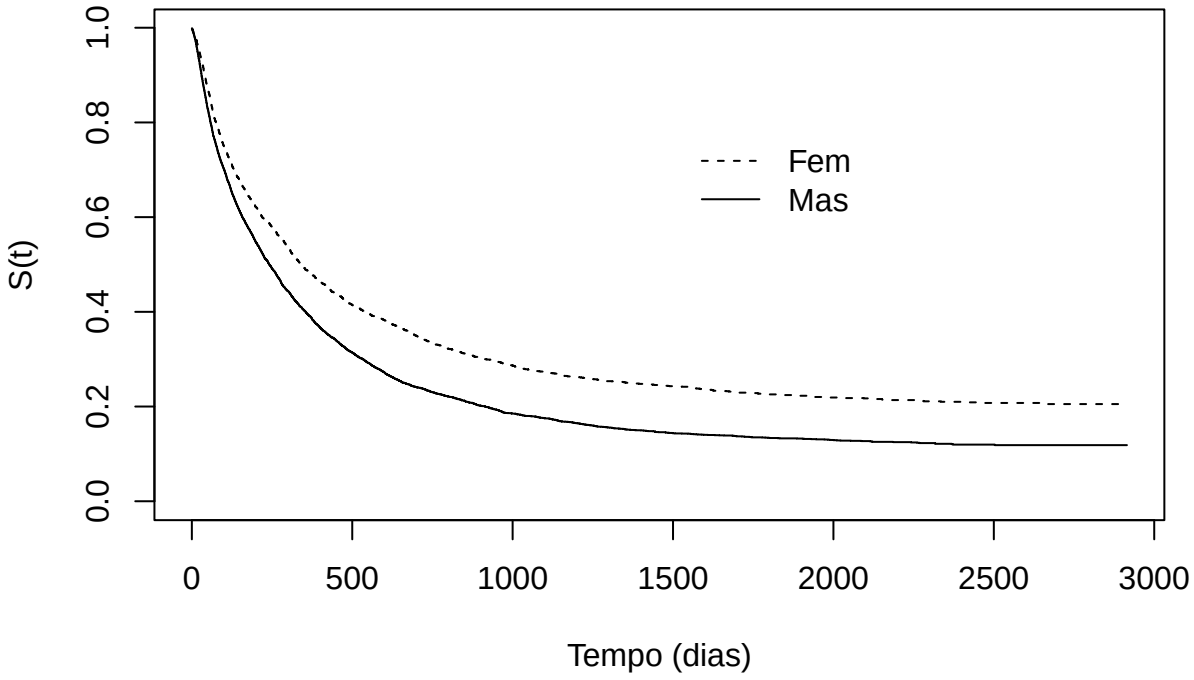

```
# sexo = 1 = masculino
# sexo = 2 = feminino
ekm2 <- survfit(Surv(TEMPO, CENSURA) ~ SEXO, data=dados)

#summary(ekm2)
plot(ekm2, lty=c(1,1), xlab= "Tempo (dias)", ylab="S(t)", mark.time = T,  conf.int = T, col=c("black",
legend(1500, 0.8, lty=c(1,1), c("Fem", "Mas"), col=c("red", "black"), bty="n")
```




```
plot(ekm2, lty=c(1,1), xlab= "Tempo (dias)", ylab="S(t)", mark.time = F,  conf.int = F, col=c("black",
legend(1500, 0.8, lty=c(1,1), c("Fem", "Mas"), col=c("red", "black"), bty="n")
```




```
plot(ekm2, lty=c(1,2), xlab= "Tempo (dias)", ylab="S(t)", mark.time = F,  conf.int = F)
legend(1500, 0.8, lty=c(2,1), c("Fem", "Mas"), bty="n")
```




```
## Utilizando o ggplot
require(survminer)
```

```
## Carregando pacotes exigidos: survminer
```

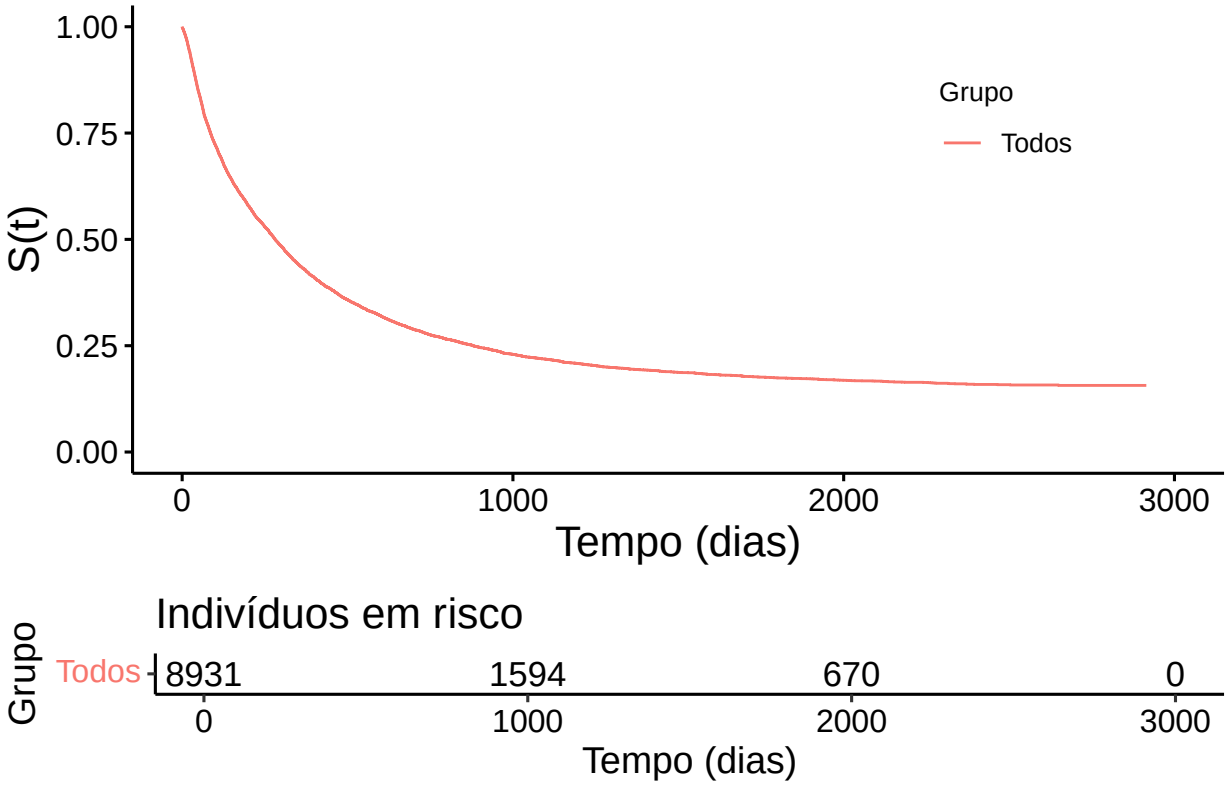
```
## Carregando pacotes exigidos: ggpubr
```

```
##
## Anexando pacote: 'survminer'
```

```
## O seguinte objeto é mascarado por 'package:survival':
##
##     myeloma
```

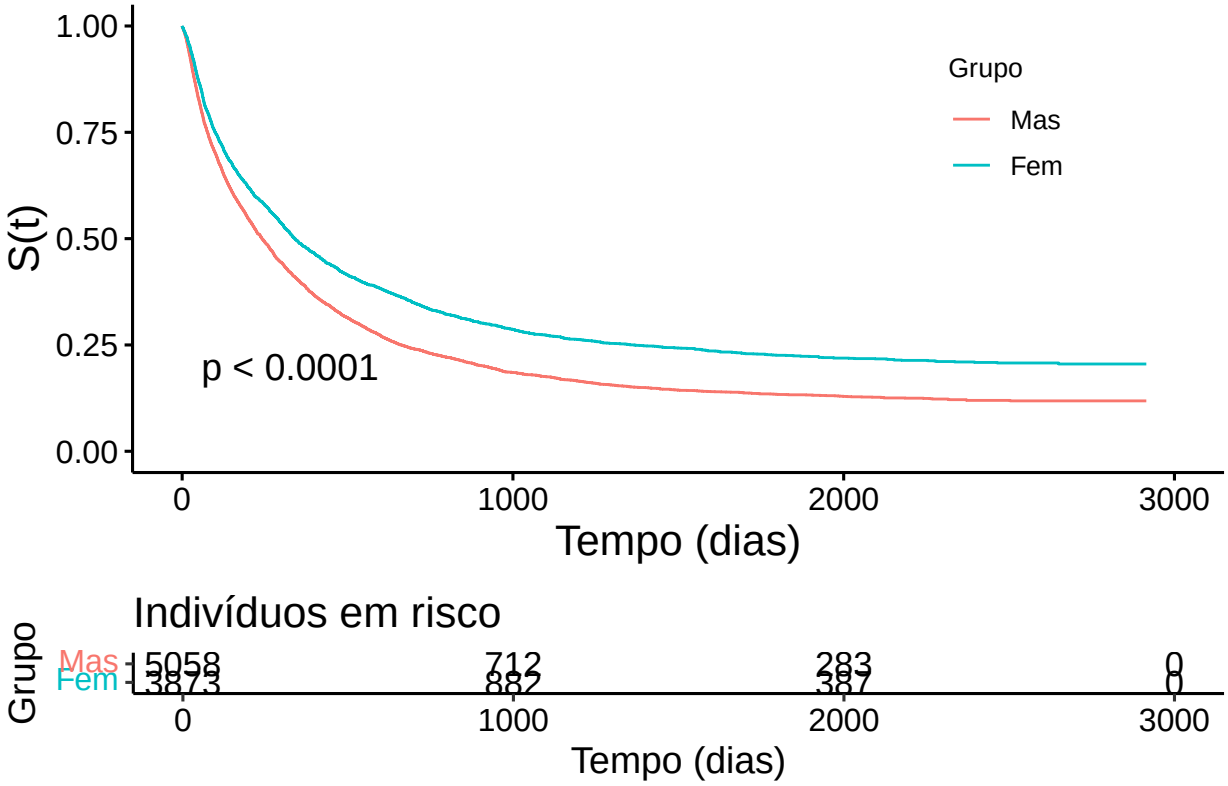
```
EKM1 <- survfit(Surv(TEMPO, CENSURA) ~ 1, data = dados)

ggsurvplot(EKM1,
           data = dados,
           ylab="S(t)",
           xlab="Tempo (dias)",
           break.x.by = 1000,
           size=0.5,
           censor.shape=" ",
           title="",
           conf.int = FALSE,
           font.x = c(16, "plain", "black"),
           font.y = c(16, "plain", "black"),
           legend.labs = "Todos",
```

```
            legend.title ="Grupo",
            legend=c(0.8,0.75),
            risk.table = T,
            risk.table.title="Indivíduos em risco",
            #pval=T

)
```




```
EKM2 <- survfit(Surv(TEMPO, CENSURA) ~ SEXO, data = dados)

ggsurvplot(EKM2,
            data = dados,
            ylab="S(t)",
            xlab="Tempo (dias)",
            break.x.by = 1000,
            size=0.5,
            censor.shape=" ",
            title="",
            font.x = c(16, "plain", "black"),
            font.y = c(16, "plain", "black"),
            legend.labs = c("Mas", "Fem"),
            legend.title ="Grupo",
            legend=c(0.8,0.75),
            risk.table = T,
            risk.table.title="Indivíduos em risco",
            pval=T
```

```
)
```



Teste logrank

O código a seguir calcula o teste de logrank.

```
require(survival)

survdiff(Surv(TEMPO, CENSURA) ~ SEXO, data = dados)
```

```
## Call:
## survdiff(formula = Surv(TEMPO, CENSURA) ~ SEXO, data = dados)
##
##           N Observed Expected (O-E)^2/E (O-E)^2/V
## SEXO=1 5058     4086     3606      63.8       134
## SEXO=2 3873     2823     3303      69.6       134
##
##  Chisq= 134  on 1 degrees of freedom, p= <2e-16
```

Quando a função *ggsurvplot* é utilizada para apresentar a estimativa da função de sobrevivência, o p-valor do teste logrank pode ser incluído no gráfico simplesmente adicionando o opção *pval=T*.