Análise de Sobrevivência - Técnicas Não-Paramétricas

José Luiz Padilha

Março de 2024

Dados de Aleitamento Materno[1]

Os dados se referem a um estudo realizado no Centro de Saúde São Marcos, localizado em Belo Horizonte, com o objetivo principal de conhecer a prática de aleitamento materno de mães que utilizam este serviço, assim como os possíveis fatores de risco ou de proteção para o desmame precoce.

Um inquérito epidemiológico composto por questões demográficas e comportamentais foi aplicado a 150 mães de crianças menores de 2 anos de idade. A variável resposta de interesse foi estabelecida como o tempo máximo de aleitamento materno (em meses), ou seja, o tempo contado a partir do nascimento até o desmame completo da criança.

Foram registradas ainda 11 covariáveis. Algumas crianças não foram acompanhadas até o desmame, e, portanto, registra-se a presença de censuras. O quadro a seguir apresenta uma descrição das 11 covariáveis estudadas.

Código	Descrição	Categorias
V1	Experiência anterior de amamentação	0 se sim e 1 se não
V2	Número de filhos vivos	0 se dois ou menos e 1 se mais de dois
V3	Conceito materno sobre o tempo ideal de amamentação	0 se $>$ 6 meses e 1 se $\leq$ 6 meses
V4	Dificuldades para amamentar nos primeiros dias pós-parto	0 se não e 1 se sim
V5	Tipo de serviço em que realizou o pré-natal	0 se público e 1 se privado/convênios
V6	Recebeu exclusivamente leite materno na maternidade	0 se sim e 1 se não
V7	A criança teve contato com o pai	0 se sim e 1 se não
V8	Renda per capita (em SM/mês)	0 se $\geq$ 1 SM e 0 se $<$ 1 SM
V9	Peso ao nascimento	0 se $\geq$ 2,5kg e 1 se $<$ 2,5kg
V10	Tempo de separação mãe-filho pós-parto	0 se $\leq$ 6 horas e 1 se $>$ 6 horas
V11	Permanência no berçário	0 se não e 1 se sim

Todas as covariáveis são binárias, o que nos permite a aplicação direta dos testes não paramétricos. Procederemos com a construção e comparação das curvas de sobrevivência.

[1] Os dados são apresentados e analisados no livro Análise de Sobrevivência Aplicada (Colosimo e Giolo, 2006)

Leitura dos Dados

Utilizaremos extensivamente o pacote `survival`. Acesse a ajuda do pacote em `help(package="survival")`.

```
require(survival); require(survminer); require(tidyverse)
dados <- read.table("desmame.txt", header = TRUE, dec = ",")
head(dados)
```

```
##   id tempo cens V3 V2 V7 V11 V4 V1 V6 V10 V8 V9 V5
## 1  1   6.0    1  0  0  0   1  0  0  0   1  1  1  0
## 2  5   8.0    1  0  0  0   1  1  1  1   1  1  1  1
## 3  6   0.1    1  1  0  0   0  1  1  0   1  0  0  1
## 4  8   5.0    1  0  1  0   1  1  0  0   0  0  0  0
## 5  9   3.0    1  0  0  0   1  1  0  0   1  0  0  0
## 6 15   5.0    1  1  0  0   0  1  1  0   0  0  0  1
```

Estimador de Kaplan-Meier

O estimador de Kaplan-Meier pode ser obtido pela função `survfit`. Veja detalhes em `?Surv`, `?survfit` e `?survfit.formula` (em especial as opções para o intervalo de confiança `conf.type`).

```
ekm <- survfit(Surv(tempo,cens) ~ 1, data = dados)
summary(ekm)
```

```
## Call: survfit(formula = Surv(tempo, cens) ~ 1, data = dados)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   0.1    150       4    0.973  0.0132        0.948        0.999
##   0.5    143       3    0.953  0.0174        0.919        0.988
##   0.7    137       2    0.939  0.0197        0.901        0.978
##   0.9    135       1    0.932  0.0208        0.892        0.974
##   1.0    132       7    0.883  0.0268        0.832        0.937
##   1.5    115       1    0.875  0.0276        0.822        0.931
##   1.6    114       1    0.867  0.0284        0.813        0.925
##   1.8    112       1    0.860  0.0292        0.804        0.919
##   2.0    111       2    0.844  0.0307        0.786        0.906
##   2.5     94       6    0.790  0.0357        0.723        0.863
##   3.0     88       3    0.763  0.0377        0.693        0.841
##   3.5     80       2    0.744  0.0391        0.671        0.825
##   4.0     77       9    0.657  0.0440        0.576        0.749
##   5.0     60       7    0.580  0.0475        0.495        0.681
##   5.9     51       1    0.569  0.0479        0.483        0.671
##   6.0     50       3    0.535  0.0489        0.447        0.640
##   7.0     46       1    0.523  0.0492        0.435        0.629
##   8.0     44       4    0.476  0.0502        0.387        0.585
##   9.0     36       1    0.463  0.0505        0.373        0.573
##  10.0     31       2    0.433  0.0514        0.343        0.546
##  11.5     23       1    0.414  0.0525        0.323        0.531
##  12.0     22       1    0.395  0.0534        0.303        0.515
##  14.0     15       1    0.369  0.0560        0.274        0.496
##  18.0      5       1    0.295  0.0797        0.174        0.501
```

Podemos obter estimativas para tempos específicos usando o argumento `times`:

```
summary(ekm, times = c(12, 13.5, 14, 15, 16))
```

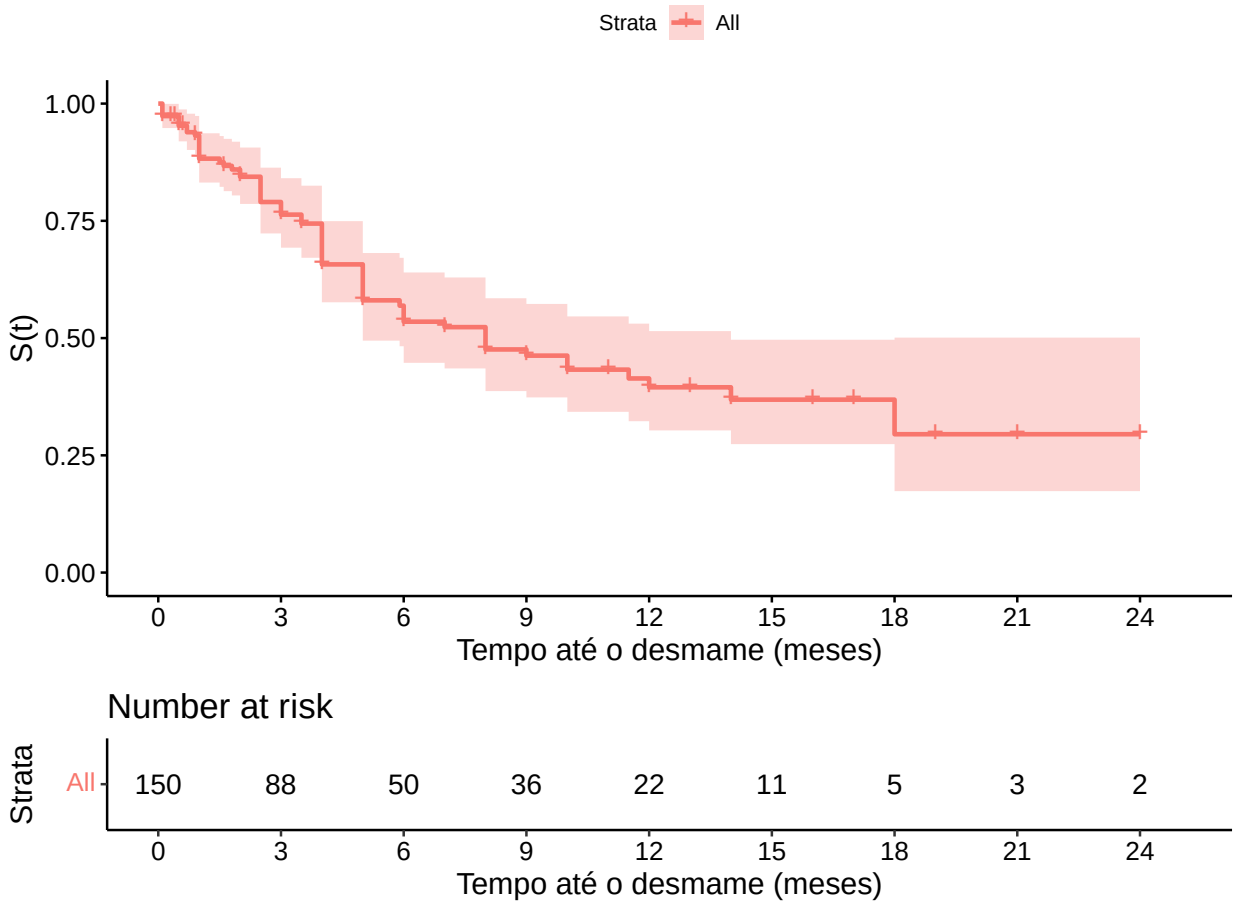
```
## Call: survfit(formula = Surv(tempo, cens) ~ 1, data = dados)
```

```
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##  12.0     22      63    0.395  0.0534        0.303        0.515
##  13.5     15       0    0.395  0.0534        0.303        0.515
##  14.0     15       1    0.369  0.0560        0.274        0.496
##  15.0     11       0    0.369  0.0560        0.274        0.496
##  16.0     11       0    0.369  0.0560        0.274        0.496
```

Gráfico de Sobrevivência

É possível visualizar a curva de sobrevivência usando a função base `plot()`. Uma alternativa mais atraente é por meio da função `survminer::ggsurvplot()` que utiliza o pacote `ggplot2` para construção dos gráficos. Por exemplo, a seguir obtemos um gráfico da curva de sobrevivência juntamente com o número de indivíduos sob risco em diferentes tempos.

```
#?ggsurvplot
ggsurvplot(ekm, risk.table = TRUE, ylab = "S(t)", xlab = "Tempo até o desmame (meses)",
           break.time.by = 3)
```



Um *dataframe* com as estimativas de sobrevivência pode ser obtidas via `surv_summary`.

```
rbind(head(surv_summary(ekm)), tail(surv_summary(ekm)))
```

```
##    time n.risk n.event n.censor      surv    std.err     upper     lower
## 1   0.1    150       4        1 0.9733333 0.01351475 0.9994599 0.9478897
## 2   0.3    145       0        1 0.9733333 0.01351475 0.9994599 0.9478897
```

```
## 3   0.4    144       0        1 0.9733333 0.01351475 0.9994599 0.9478897
## 4   0.5    143       3        2 0.9529138 0.01823454 0.9875859 0.9194589
## 5   0.6    138       0        1 0.9529138 0.01823454 0.9875859 0.9194589
## 6   0.7    137       2        0 0.9390026 0.02099133 0.9784410 0.9011539
## 29 16.0     11       0        4 0.3687408 0.15176449 0.4964819 0.2738666
## 30 17.0      7       0        2 0.3687408 0.15176449 0.4964819 0.2738666
## 31 18.0      5       1        0 0.2949927 0.27024518 0.5010076 0.1736913
## 32 19.0      4       0        1 0.2949927 0.27024518 0.5010076 0.1736913
## 33 21.0      3       0        1 0.2949927 0.27024518 0.5010076 0.1736913
## 34 24.0      2       0        2 0.2949927 0.27024518 0.5010076 0.1736913
```

Note que são mostrados todos os tempos observados, independente de serem ou não evento.

```
dim(surv_summary(ekm))
```

```
## [1] 34  8
```

```
length(unique(dados$tempo))
```

```
## [1] 34
```

```
length(unique(dados$tempo[dados$cens==1]))
```

```
## [1] 24
```

Tempo Médio e Mediano

A estimativa do tempo mediano, obtida por meio de uma interpolação linear, é:

$$\frac{8-7}{0,476-0,523} = \frac{MED-7}{0,500-0,523} \Rightarrow MED = t_{0,50} = 7,49 \; meses.$$

Podemos obter estimativas de outros percentis de forma análoga. Já o tempo médio pode ser obtido fazendo:

```
aux <- summary(ekm)
t_m <- aux$time[1] + sum(diff(aux$time)*aux$surv[-length(aux$surv)])
t_m
```

```
## [1] 9.760153
```

Como o maior tempo é censura, a curva de Kaplan-Meier não atinge o valor zero e tempo médio de vida encontrado fica subestimado. Podemos fazer uso da função `survfit()` que permite a obtenção do tempo médio, tempo mediano e seu intervalo de confiança de 95%:

```
#?print.survfit
print(ekm, print.rmean = TRUE)
```

```
## Call: survfit(formula = Surv(tempo, cens) ~ 1, data = dados)
##
##        n events rmean* se(rmean) median 0.95LCL 0.95UCL
## [1,] 150     65   11.5      1.02      8       5      14
##     * restricted mean with upper limit =  24
```

A função `quantile.survfit()` retorna quantis e intervalo de confiança a partir de um objeto do tipo `survfit`. O k-ésimo quantil para $S(t)$ é obtido como o ponto no qual uma linha horizontal na altura $p = 1-k$ intercepta o gráfico de $S(t)$. Por consistência com outras funções de quantis, o argumento `prob` se aplica à função de distribuição acumulada $F(t) = 1 - S(t)$. Como exemplo, considere

```
quantile(ekm, probs = c(0.25, 0.5, 0.75))
```

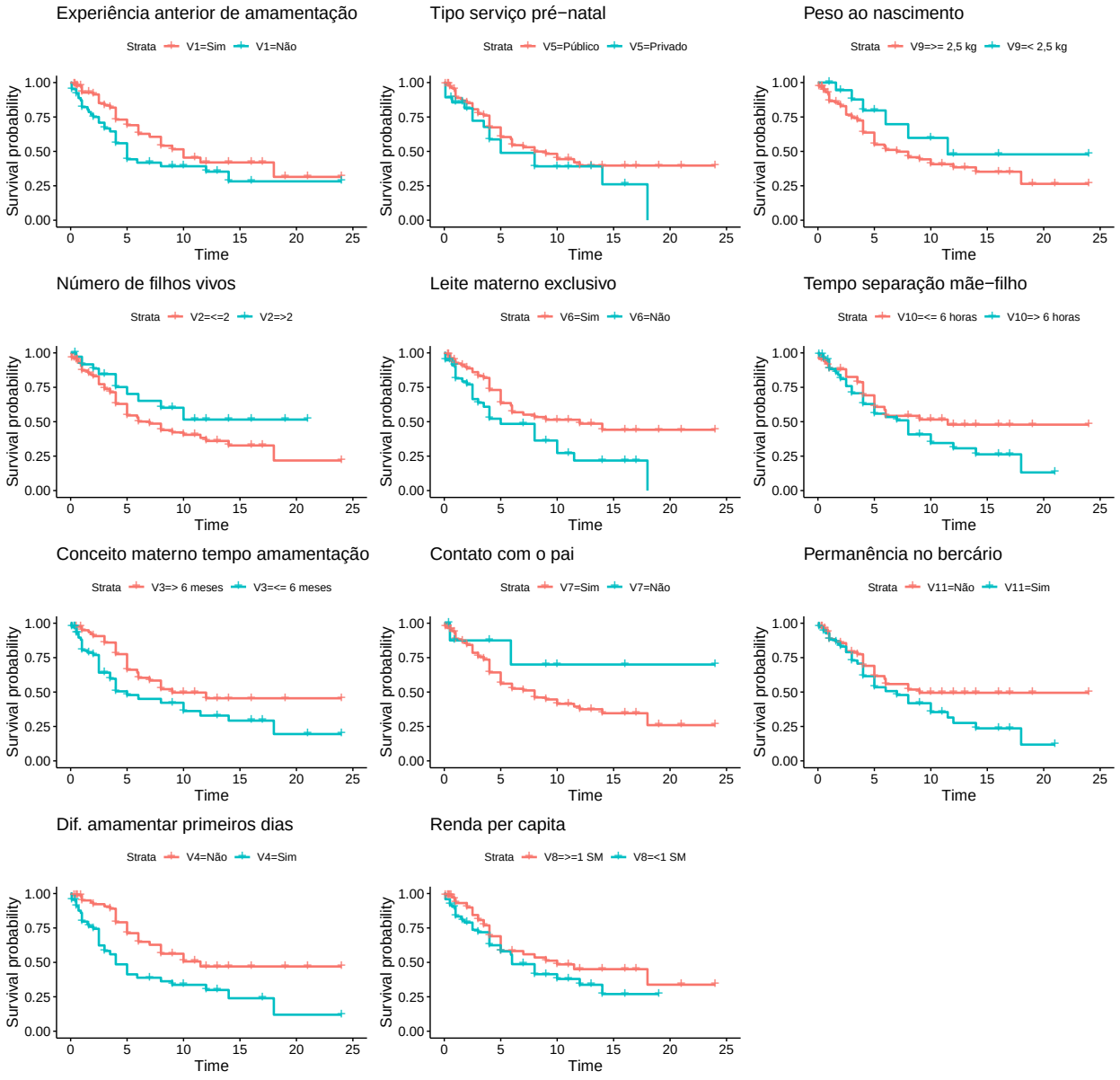
```
## $quantile
##  25  50  75
## 3.5 8.0  NA
##
## $lower
##   25   50   75
##  2.5  5.0 18.0
##
## $upper
## 25 50 75
##  4 14 NA
```

Curvas de Sobrevivência por Covariável

Vamos construir curvas de sobrevivência, uma para cada covariável.

```
# Transforma em fatores
dados <- dados %>% mutate(across(starts_with("V"), .fns = factor))
# Define os níveis
levels(dados$V1) <- c("Sim", "Não")
levels(dados$V2) <- c("<=2", ">2")
levels(dados$V3) <- c("> 6 meses", "<= 6 meses")
levels(dados$V4) <- c("Não", "Sim")
levels(dados$V5) <- c("Público", "Privado")
levels(dados$V6) <- c("Sim", "Não")
levels(dados$V7) <- c("Sim", "Não")
levels(dados$V8) <- c(">=1 SM", "<1 SM")
levels(dados$V9) <- c(">= 2,5 kg", "< 2,5 kg")
levels(dados$V10) <- c("<= 6 horas", "> 6 horas")
levels(dados$V11) <- c("Não", "Sim")
# Estimador de Kaplan-Meier
ekm_V1 <- survfit(Surv(tempo, cens) ~ V1, data = dados)
ekm_V2 <- survfit(Surv(tempo, cens) ~ V2, data = dados)
ekm_V3 <- survfit(Surv(tempo, cens) ~ V3, data = dados)
ekm_V4 <- survfit(Surv(tempo, cens) ~ V4, data = dados)
ekm_V5 <- survfit(Surv(tempo, cens) ~ V5, data = dados)
ekm_V6 <- survfit(Surv(tempo, cens) ~ V6, data = dados)
ekm_V7 <- survfit(Surv(tempo, cens) ~ V7, data = dados)
ekm_V8 <- survfit(Surv(tempo, cens) ~ V8, data = dados)
ekm_V9 <- survfit(Surv(tempo, cens) ~ V9, data = dados)
ekm_V10 <- survfit(Surv(tempo, cens) ~ V10, data = dados)
ekm_V11 <- survfit(Surv(tempo, cens) ~ V11, data = dados)
# Lista de gráficos
splots <- list()
splots[[1]] <- ggsurvplot(ekm_V1, title = "Experiência anterior de amamentação")
splots[[2]] <- ggsurvplot(ekm_V2, title = "Número de filhos vivos")
splots[[3]] <- ggsurvplot(ekm_V3, title = "Conceito materno tempo amamentação")
splots[[4]] <- ggsurvplot(ekm_V4, title = "Dif. amamentar primeiros dias")
splots[[5]] <- ggsurvplot(ekm_V5, title = "Tipo serviço pré-natal")
splots[[6]] <- ggsurvplot(ekm_V6, title = "Leite materno exclusivo")
splots[[7]] <- ggsurvplot(ekm_V7, title = "Contato com o pai")
splots[[8]] <- ggsurvplot(ekm_V8, title = "Renda per capita")
splots[[9]] <- ggsurvplot(ekm_V9, title = "Peso ao nascimento")
splots[[10]] <- ggsurvplot(ekm_V10, title = "Tempo separação mãe-filho")
splots[[11]] <- ggsurvplot(ekm_V11, title = "Permanência no bercário")
```

```r
# Junta os ggsurvplots
arrange_ggsurvplots(splots, print = TRUE, ncol = 3, nrow = 4)
```



As curvas de sobrevida estimadas apontam eventuais diferenças entre as categorias comparadas. Por exemplo, para `V5` (tipo de serviço que realizou o pré-natal) não há indícios de diferenças nos perfis de sobrevida enquanto para `V4` (dificuldade para amamentar nos primeiros dias) notamos uma marcada diferença.

Na sequência, comparamos formalmente as curvas de sobrevivência para cada covariável por meio de testes não-paramétricos.

Testes Não-paramétricos

Testes não-paramétricos são realizados por meio da função `survdiff()`. Tal função testa se há diferença entre duas ou mais curvas de sobrevivência usando uma família de testes (pesos de Harrington-Fleming). O argumento `rho` determina o peso adotado. Veja os detalhes em `?survdiff`.

Para fins ilustrativos compararemos os resultados dos testes para os valores extremos de ρ. É importante

ressaltar que o tipo de teste a ser adotado deve ser definido *a priori* de acordo com os objetivos de pesquisa e não após o conhecimento dos dados e das diferenças observadas.

- $\rho = 0$

Usando o argumento `rho = 0` obtemos o teste de logrank. Por exemplo, para `V1` (indicadora de experiência anterior de amamentação), temos:

```
logrank1 <- survdiff(Surv(tempo, cens) ~ V1, rho = 0, data = dados)
logrank1
```

```
## Call:
## survdiff(formula = Surv(tempo, cens) ~ V1, data = dados, rho = 0)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## V1=Sim 87       32     39.6      1.46      3.95
## V1=Não 63       33     25.4      2.28      3.95
##
##  Chisq= 3.9  on 1 degrees of freedom, p= 0.05
```

O valor-p não fica armazenado no objeto da classe `survdiff` mas podemos acessá-lo facilmente usando a função `broom::glance()`:

```
library(broom)
glance(logrank1)$p.value
```

```
## [1] 0.04690154
```

Ao nível de 5% de significância o teste aponta diferença estatisticamente significativa entre as curvas de sobrevivência.

- $\rho = 1$

Já para `rho = 1` o peso é o Kaplan-Meier no tempo de falha anterior e tem-se um teste similar ao de Wilcoxon.

```
wilcox1 <- survdiff(Surv(tempo, cens) ~ V1, rho = 1, data =  dados)
wilcox1
```

```
## Call:
## survdiff(formula = Surv(tempo, cens) ~ V1, data = dados, rho = 1)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## V1=Sim 87     22.5     29.6      1.69      5.67
## V1=Não 63     26.3     19.2      2.61      5.67
##
##  Chisq= 5.7  on 1 degrees of freedom, p= 0.02
```

```
glance(wilcox1)$p.value
```

```
## [1] 0.01729705
```

O resultado aponta diferença estatisticamente significativa entre os grupos. Vamos comparar os valores-p para todas as covariáveis.

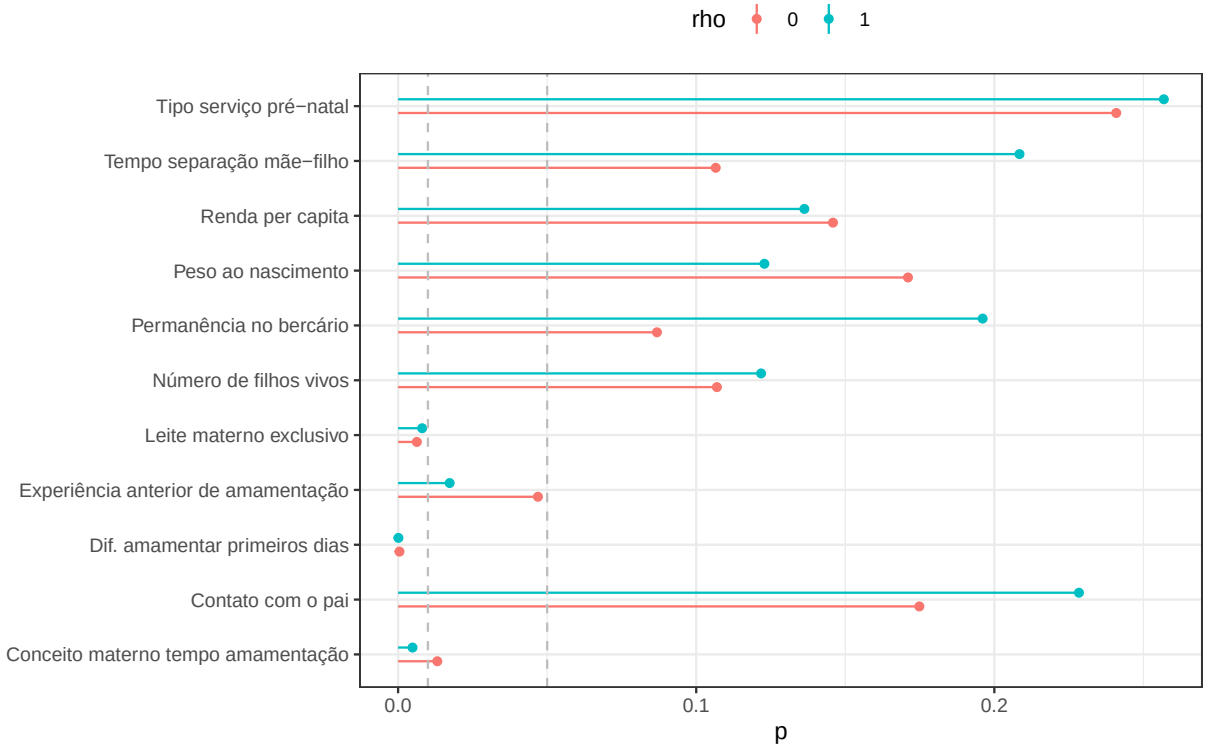
```
logrank2 <- survdiff(Surv(tempo, cens) ~ V2, rho = 0, data = dados)
logrank3 <- survdiff(Surv(tempo, cens) ~ V3, rho = 0, data = dados)
logrank4 <- survdiff(Surv(tempo, cens) ~ V4, rho = 0, data = dados)
logrank5 <- survdiff(Surv(tempo, cens) ~ V5, rho = 0, data = dados)
logrank6 <- survdiff(Surv(tempo, cens) ~ V6, rho = 0, data = dados)
logrank7 <- survdiff(Surv(tempo, cens) ~ V7, rho = 0, data = dados)
logrank8 <- survdiff(Surv(tempo, cens) ~ V8, rho = 0, data = dados)
```

```r
logrank9 <- survdiff(Surv(tempo, cens) ~ V9, rho = 0, data = dados)
logrank10 <- survdiff(Surv(tempo, cens) ~ V10, rho = 0, data = dados)
logrank11 <- survdiff(Surv(tempo, cens) ~ V11, rho = 0, data = dados)
#
wilcox2 <- survdiff(Surv(tempo, cens) ~ V2, rho = 1, data = dados)
wilcox3 <- survdiff(Surv(tempo, cens) ~ V3, rho = 1, data = dados)
wilcox4 <- survdiff(Surv(tempo, cens) ~ V4, rho = 1, data = dados)
wilcox5 <- survdiff(Surv(tempo, cens) ~ V5, rho = 1, data = dados)
wilcox6 <- survdiff(Surv(tempo, cens) ~ V6, rho = 1, data = dados)
wilcox7 <- survdiff(Surv(tempo, cens) ~ V7, rho = 1, data = dados)
wilcox8 <- survdiff(Surv(tempo, cens) ~ V8, rho = 1, data = dados)
wilcox9 <- survdiff(Surv(tempo, cens) ~ V9, rho = 1, data = dados)
wilcox10 <- survdiff(Surv(tempo, cens) ~ V10, rho = 1, data = dados)
wilcox11 <- survdiff(Surv(tempo, cens) ~ V11, rho = 1, data = dados)
#
res <- data.frame(cov = paste("V", 1:11, sep=""),
                  desc = c("Experiência anterior de amamentação",
                           "Número de filhos vivos",
                           "Conceito materno tempo amamentação",
                           "Dif. amamentar primeiros dias",
                           "Tipo serviço pré-natal",
                           "Leite materno exclusivo",
                           "Contato com o pai",
                           "Renda per capita",
                           "Peso ao nascimento",
                           "Tempo separação mãe-filho",
                           "Permanência no bercário"),
                  p.rho.0 = c(glance(logrank1)$p.value, glance(logrank2)$p.value,
                              glance(logrank3)$p.value, glance(logrank4)$p.value,
                              glance(logrank5)$p.value, glance(logrank6)$p.value,
                              glance(logrank7)$p.value, glance(logrank8)$p.value,
                              glance(logrank9)$p.value, glance(logrank10)$p.value,
                              glance(logrank11)$p.value),
                  p.rho.1 = c(glance(wilcox1)$p.value, glance(wilcox2)$p.value,
                              glance(wilcox3)$p.value, glance(wilcox4)$p.value,
                              glance(wilcox5)$p.value, glance(wilcox6)$p.value,
                              glance(wilcox7)$p.value, glance(wilcox8)$p.value,
                              glance(wilcox9)$p.value, glance(wilcox10)$p.value,
                              glance(wilcox11)$p.value))
res %>% mutate(dplyr::across(where(is.numeric), ~ round(., digits = 3)))
```

```
##    cov                                desc p.rho.0 p.rho.1
## 1   V1 Experiência anterior de amamentação   0.047   0.017
## 2   V2              Número de filhos vivos   0.107   0.122
## 3   V3  Conceito materno tempo amamentação   0.013   0.005
## 4   V4       Dif. amamentar primeiros dias   0.000   0.000
## 5   V5              Tipo serviço pré-natal   0.241   0.257
## 6   V6             Leite materno exclusivo   0.006   0.008
## 7   V7                   Contato com o pai   0.175   0.228
## 8   V8                    Renda per capita   0.146   0.136
## 9   V9                  Peso ao nascimento   0.171   0.123
## 10 V10           Tempo separação mãe-filho   0.107   0.208
## 11 V11             Permanência no bercário   0.087   0.196
```

Ou, em forma de gráfico:

```
res_long <- gather(res, "p.rho.0", "p.rho.1", key = "rho", value = "p")
res_long$rho <- as.factor(res_long$rho)
levels(res_long$rho) <- c("0", "1")
ggplot(res_long) + geom_linerange(aes(x = desc, ymin = 0, ymax = p, colour = rho),
                                  position = position_dodge(width = .5)) +
  geom_point(aes(x = desc, y = p, colour = rho), position = position_dodge(width = 0.5)) +
  coord_flip() + geom_hline(yintercept = c(0.01, 0.05), linetype = "dashed",
                            color = "gray") +
  theme_bw() + theme(legend.position = "top") + labs(x = "")
```



Vários outros testes para comparação de curvas de sobrevivência estão implementados no pacote `coin`. Veja os detalhes em `?logrank_test`. Por exemplo, para o teste Tarone-Ware, obtemos:

```
library(coin)
logrank_test(Surv(tempo, cens) ~ factor(V1), type = "Tarone-Ware", data=dados)
```

```
##
##  Asymptotic Two-Sample Tarone-Ware Test
##
## data:  Surv(tempo, cens) by factor(V1) (Sim, Não)
## Z = 2.364, p-value = 0.01808
## alternative hypothesis: true theta is not equal to 1
```

Comentários gerais:

As técnicas ilustradas foram importantes para descrever o tempo de sobrevida geral e por variável. Testes não-paramétricos nos permitiram verificar a igualdade de curvas de sobrevivência sem assumir qualquer distribuição de probabilidade para os tempos de falha.

Apesar da simplicidade de aplicação, temos algumas limitações:

- Até agora nos limitamos a covariáveis categóricas. No caso de covariáveis contínuas, a estimação da curva de sobrevivência e aplicação de testes deve proceder a algum procedimento de dicotomização, o que quase sempre resulta em perda de informação, principalmente na inexistência de um ponto de corte bem estabelecido.
- Ainda, tais técnicas não nos permitem a inclusão conjunta das covariáveis na análise. Isso será feito por meio de modelos de regressão apropriados para dados censurados. Neste caso, podemos assumir ou não uma abordagem completamente paramétrica. Nos próximos módulos da disciplina discutiremos o modelo de regressão semi-paramétrico de Cox e modelos de regressão paramétricos.