

SME0821 - Análise de Sobrevida e Confiabilidade - Atividade III

Matheus Victal Cerqueira, nUSP: 10276661

15/07/2021

Introdução

O presente documento se trata de uma solução para os exercícios propostos na Atividade III da disciplina SME0821 - Análise de Sobrevida e Confiabilidade, oferecida pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo no primeiro semestre de 2021. As temáticas abordadas modelos de riscos proporcionais, modelos de longa duração e modelos de tempo de vida acelerados.

Bibliotecas utilizadas

```
# Limpeza do ambiente global
rm(list=ls(all=TRUE))
```

```
# Remoção de avisos de carregamento de pacotes
defaultW <- getOption("warn")
options(warn = -1)
```

```
# Exercícios 1 e 2
library(ggplot2)
library(ggpubr)
library(survminer)
library(survival)
library(ggfortify)
library(msm)
```

```
# Exercício 3
library(numDeriv)
library(xtable)
library(flexsurv)
```

```
##
## Attaching package: 'flexsurv'
```

```
## The following object is masked from 'package:msm':
##
##      qgeneric
```

Exercício 1

O exercício 1 fornece dados referentes à pacientes que, após serem submetidos à cirurgia de intestino, receberam tratamento quimioterápico com as drogas *Compath* ou *Zena*. Tais dados são referentes ao tempo até a observação de qualquer alteração de saúde nesses pacientes, os quais foram acompanhados por 250 dias. Abaixo apresentam-se os dados em duas estruturas diferentes, as quais foram utilizadas no decorrer do desenvolvimento do exercício.

[illegible]

Item a)

O primeiro item do exercício pede que verifiquemos se os dados apresentam a propriedade de riscos proporcionais. É denominada uma família de riscos proporcionais, uma classe de modelos na qual diferentes tipos de indivíduos apresentam funções de risco proporcionais (de razão contante no tempo). Ou seja, sendo $h_i(t)$ e $h_j(t)$ funções de risco de dois tipos diferentes de indivíduos, o risco relativo (dado por RR) para os dois indivíduos respeita a seguinte condição [1 - Capítulo 5]:

$$RR_{ij} = \frac{h_i(t)}{h_j(t)} = k,$$

onde k é contante em relação à variação do tempo t .

Existem algumas formas diferentes de se verificar a ocorrência dessa propriedade. Entre elas, a análise do gráfico $\log(t)$ vs $\log(-\log(S(t)))$ para as funções de sobrevivência $S(t)$ dos indivíduos de interesse. Assim sendo, para tal análise gráfica, estimou-se $S_C(t)$ e $S_Z(t)$ (funções de sobrevivência para os casos da droga *Compath* e

Zena, respectivamente) utilizando-se do estimador não paramétrico de Kaplan-Meier e foi realizada a análise gráfica sobre tais estimativas.

```
# Modelos Kaplan-Meier
mKM_C <- survfit(Surv(temposCompath, censuraCompath) ~ 1)
mKM_Z <- survfit(Surv(temposZena, censuraZena) ~ 1)

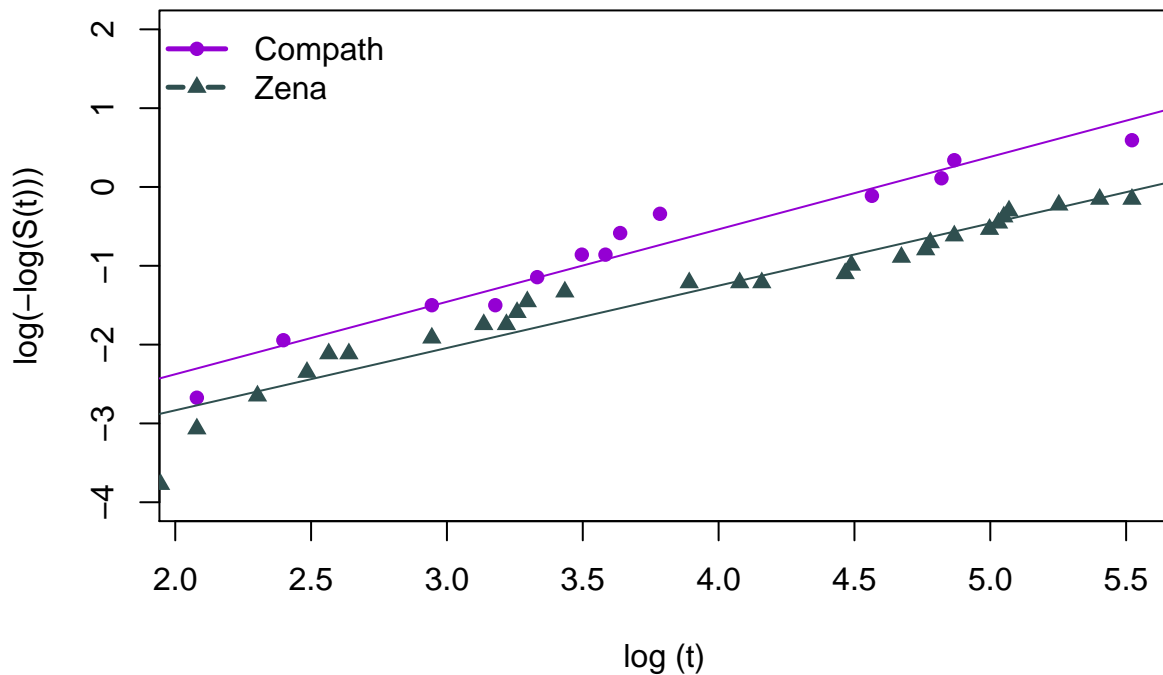
# Plotar gráfico simples log(t) vs log(-log(S(t))) para Compath
plot(log(mKM_C$time), log(-log(mKM_C$surv)) ,xlab="log (t)",
      ylab="log(-log(S(t)))",
      ylim=c(-4,2), lty=1, lwd=2.5, pch=16, col="darkviolet")

# Adiciona linha com regressão simples para melhor comparação
abline(lm(log(-log(mKM_C$surv))~log(mKM_C$time)), col = "darkviolet")

# Plotar gráfico simples log(t) vs log(-log(S(t))) para Zena
points(log(mKM_Z$time), log(-log(mKM_Z$surv)) ,xlab="log (t)",
       ylab="log(-log(S(t)))",
       lty=1, lwd=2.5, pch=17, col="darkslategrey")

# Adiciona linha com regressão simples para melhor comparação
abline(lm(log(-log(mKM_Z$surv))~log(mKM_Z$time)), col = "darkslategrey")

#Legenda
legend("topleft", c("Compath", "Zena"), bty="n", pch=c(16,17),
      col=c("darkviolet", "darkslategrey"), lty=1:3, lwd=2.5)
```



Observando-se as tendências apresentadas no gráfico acima para os dois grupos de dados, pode-se dizer que aparentemente a propriedade de riscos proporcionais é presente, já que as curvas parecem ser relativamente paralelas entre si para os dois grupos.

Para agregar à conclusão da análise gráfica acima, façamos um teste sobre os resíduos padronizados de Schoenfeld. O teste aqui utilizado verifica a suposição de que tais resíduos e uma função do tempo $g(t)$, para cada covariável do modelo (no caso estudado temos apenas uma covariável binária) possuem correlação de Pearson estatisticamente próxima a 0, o que indicaria que os riscos podem ser considerados proporcionais. Assim, rejeitar a hipótese nula para tal teste significaria rejeitar a suposição de que os riscos são proporcionais para o modelo proposto.

```
# Primeiro, ajusta-se um modelo de Cox de riscos proporcionais para os dados de
# interesse. O método "Breslow" foi escolhido baseado na documentação da função
# coxph
fitCox <- coxph(Surv(tem, cen)~grupo, method = "breslow")

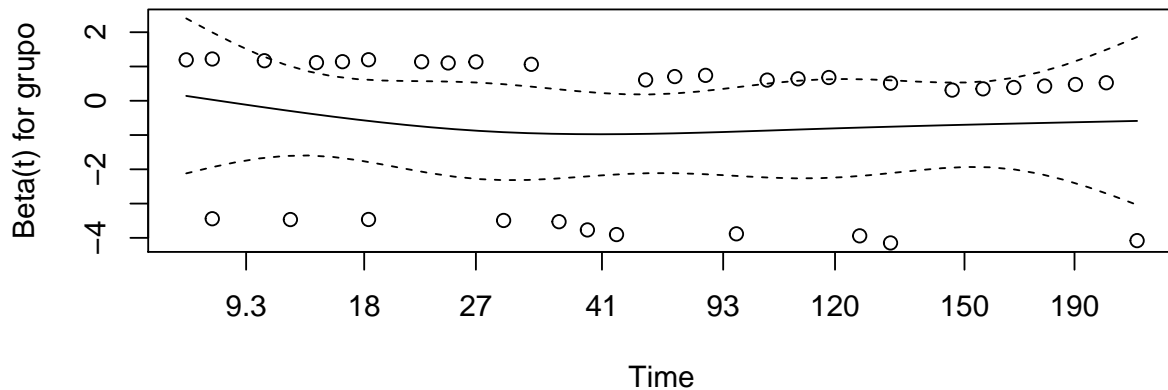
# Depois, é realizado o teste baseado no trabalho de [2]
test <- cox.zph(fitCox)
print(test)
```

```
##          chisq df    p
## grupo  0.152  1 0.7
## GLOBAL 0.152  1 0.7
```

O nível descritivo para o teste resultou em 0,7; o que indica que a probabilidade da estatística de teste assumir valores iguais ou mais extremos ao valor observado (0,152) dado que a suposição de riscos proporcionais é

verdadeira, é 0,7. Assim sendo, há fortes indícios na amostra que indiquem que os riscos são de fato proporcionais. Podemos também visualizar o gráfico de $\beta_q(t)$ vs t para cada grupo $q \in \{1, 2\}$ e é notório que pelas bandas do intervalo de confiança, podemos considerar que uma reta horizontal é formada para cada grupo, o que concorda com a análise até agora e corrobora para a suposição de riscos proporcionais ser válida.

```
plot(test)
```



Item b)

No presente item, é pedido que seja ajustado um modelo de riscos proporcionais de Cox e que suas estimativas sejam interpretadas, assumindo que os riscos podem ser considerados proporcionais para o caso estudado.

No item a), foi ajustado um modelo de Cox utilizando-se do método de *Breslow*, retomemos ao modelo ajustado para que possamos interpretar as saídas fornecidas pela função *coxph*.

```
summary(fitCox)
```

```
## Call:
## coxph(formula = Surv(tem, cen) ~ grupo, method = "breslow")
##
##    n= 59, number of events= 34
##
##           coef exp(coef) se(coef)      z Pr(>|z|)
## grupo -0.6671    0.5132   0.3703 -1.802  0.0716 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## grupo    0.5132      1.948    0.2484    1.06
##
## Concordance= 0.561 (se = 0.039 )
## Likelihood ratio test= 2.97 on 1 df,  p=0.08
```

```
## Wald test          = 3.25  on 1 df,   p=0.07
## Score (logrank) test = 3.36  on 1 df,   p=0.07
```

Sabe-se que o modelo de Cox para uma covariável é dado por:

$$h(t) = h_0(t) \exp\{\beta x\}$$

no qual $h_0(t)$ é o risco basal, β é o coeficiente da regressão e x é a covariável (no nosso caso, uma covariável binária de pertencimento ao grupo *Zena* (1) ou ao grupo *Compath* (0)) [1]. Pela saída do ajuste do modelo de Cox, temos que a estimativa do coeficiente β é dada por:

$$\hat{\beta} = -0,6671 \Rightarrow \exp\{\hat{\beta}x\} = 0,5132$$

Assim sendo, podemos estimar o risco relativo RR para o modelo proposto por:

$$\widehat{RR} = \frac{h_c(t)}{h_z(t)} = \frac{h_0(t)}{h_0(t) \exp\{\hat{\beta}\}} = \exp\{\hat{\beta}\} = 1,948$$

sendo $h_c(t)$ e $h_z(t)$ as funções de risco para o grupo *Compath* e *Zena* respectivamente, levando a conclusão de que o risco de falha é aproximadamente o dobro para os indivíduos tratados com *Compath* quando comparados com os indivíduos tratados com *Zena*, o seja, o tratamento com *Zena* parece ser consideravelmente mais eficaz, reduzindo o risco em quase o dobro ao longo de toda a reta real para t .

O resumo também apresenta os níveis descritivos para o teste *Score* (0,07), para o teste de Wald (0,07) e para o teste de razão de verossimilhança (0,08). Nenhum desses testes rejeita a hipótese nula ($H_0 : \beta = \hat{\beta}$) à um nível de significância de 5%.

Item c)

Agora, estudemos a adequabilidade para o modelo de Cox obtido. Alguns testes de hipóteses e análises gráficas já foram comentados nos itens a) e b), agora, serão analisados os resíduos de Cox-Snell do modelo.

```
# Obtenção dos resíduos de Martingale
residuos_martingale <- resid(fitCox, type = "martingale")

m <- residuos_martingale

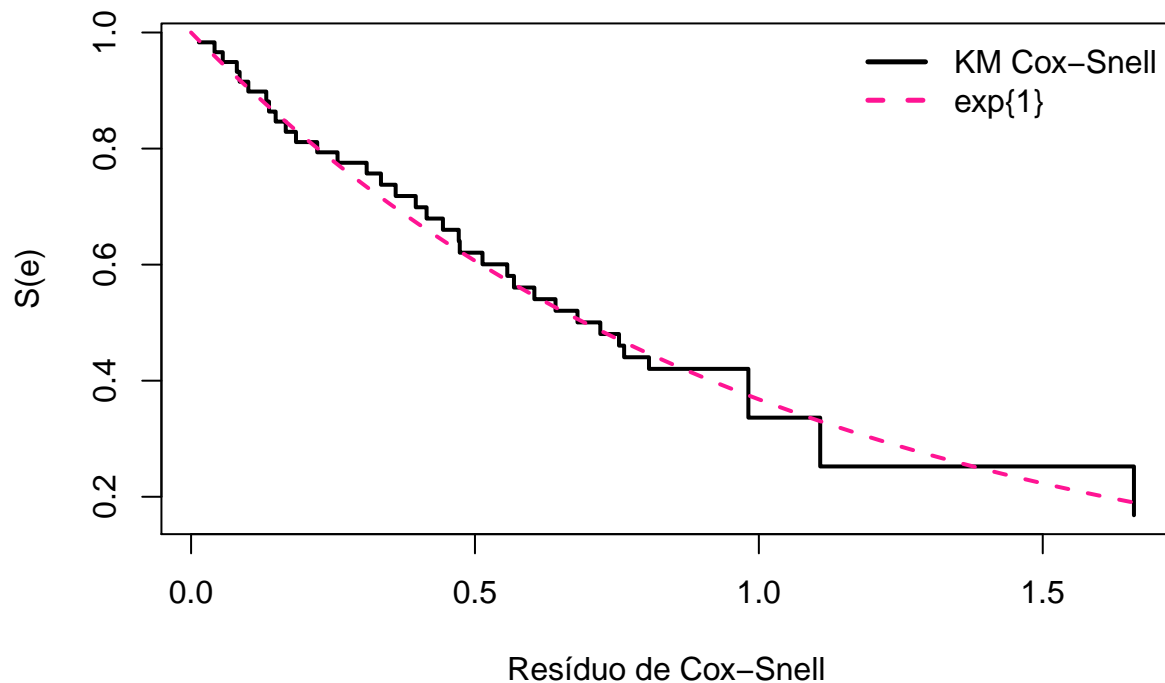
# Obtenção dos resíduos de Cox-Snell
e <- cen - m

# Sumário do modelo KM ajustado com os resíduos de Cox-Snell
cox_snell_sum <- summary(survfit(Surv(e,cen)~1))

plot(y = cox_snell_sum$surv, x = cox_snell_sum$time, ylab="S(e)",
      xlab="Resíduo de Cox-Snell", lty=1, lwd=2, main=" ", type = 's')

# Curva para uma distribuição exponencial com lambda = 1
curve(exp(-x), 0, max(cox_snell_sum$time), add=T, lwd=2, lty=2, col = "deeppink")

legend("topright", c("KM Cox-Snell", "exp{1}"), bty="n",
      col=c("black", "deeppink"), lty=1:3, lwd=2.5)
```



É notório que os resíduos de Cox-Snell parecem possuir uma distribuição com boa aderência à distribuição exponencial com $\lambda = 1$. Assim sendo, agregando-se tal resultado aos resultados obtidos anteriormente, pode-se concluir que o modelo aqui proposto é adequado, baseando-se nos testes de hipóteses performados e na análise gráfica.

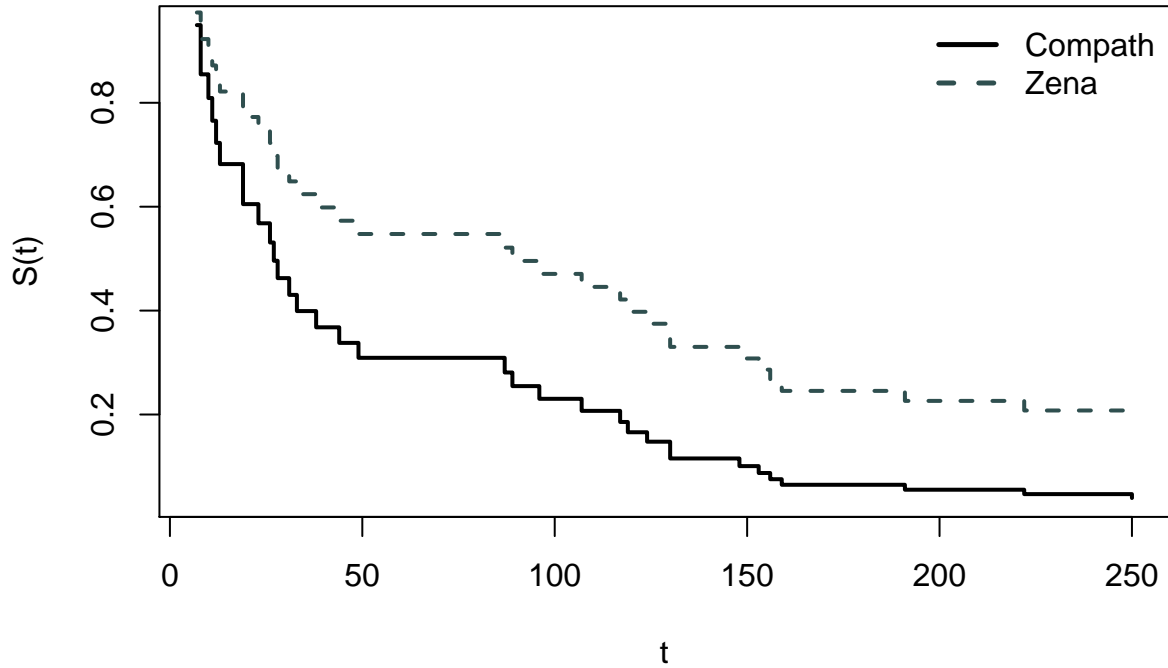
Item d)

O item d) sugere que as curvas de sobrevivência para cada tratamento sejam estimadas utilizando-se dos resultados obtidos em b). Isso pode ser feito utilizando-se da função *basehaz* para obter o risco acumulado basal do modelo de Cox ajustado. Abaixo é apresentada a solução e os gráficos para as estimativas (os valores numéricos estão armazenados nas variáveis *Sc* e *Sz* para as estimativas das funções de sobrevivência para os grupos *Compath* e *Zena* respectivamente).

```
#Função de risco acumulado (H(t)) para o modelo de Cox ajustado
Ht <- basehaz(fitCox, centered = F)
tC <- Ht$time
HC <- Ht$hazard

# Risco acumulado basal
S0 <- exp(-HC)
# Risco para o grupo Compath
Sc <- S0
#Risco para o grupo Zena
Sz <- S0^exp(fitCox$coefficients[1])
```

```
# Plotando-se as curvas
plot(y = Sc, x = tC, ylab="S(t)", xlab="t", lty=1, lwd=2, main=" ", type = 's')
lines(y = Sz, x = tC, lty=2, lwd=2, main=" ", type = 's', col = "darkslategrey")
legend("topright", c("Compath", "Zena"), bty="n",
      col=c("black", "darkslategrey"), lty=1:3, lwd=2.5)
```



Item e)

No item e), é pedido que seja ajustado um modelo de riscos proporcionais de Weibull aos dados. Tal modelo pode ser parametrizado da seguinte forma:

$$S(t|\mathbf{x}) = \exp \left\{ - \left(\frac{t}{\exp\{\mathbf{x}'\boldsymbol{\beta}\}} \right)^\alpha \right\}$$

Que para o caso aqui estudado, no qual temos apenas uma covariável envolvida, tem-se:

$$S(t|x) = \exp \left\{ - \left(\frac{t}{\exp\{\beta_0 + \beta_1 x\}} \right)^\alpha \right\}$$

Assim sendo, foi utilizada a função *survreg* para obter estimativas para os parâmetros do modelo, como segue.


```
fitWei <- survreg(Surv(tem,cen)~grupo, dist = "weibull")
summary(fitWei)
```

```
##
## Call:
## survreg(formula = Surv(tem, cen) ~ grupo, dist = "weibull")
##              Value Std. Error      z      p
## (Intercept)  3.924      0.763  5.15 2.7e-07
## grupo        0.864      0.442  1.95  0.051
## Log(scale)   0.173      0.146  1.18  0.236
##
## Scale= 1.19
##
## Weibull distribution
## Loglik(model)= -214.4   Loglik(intercept only)= -216.2
##  Chisq= 3.52 on 1 degrees of freedom, p= 0.061
## Number of Newton-Raphson Iterations: 5
## n= 59
```

Assim sendo, temos as seguintes estimativas para os parâmetros: $\hat{\beta}_0 = 3,924$ e $\hat{\beta}_1 = 0,864$. É notório que o valor para $\log(\alpha)$ é próximo de 0, o que indica que um modelo exponencial talvez seja adequado ($\alpha \approx 0$).

O intercepto β_0 é interpretado como o fator de aceleração para o grupo $x = 0$, ou seja, *Compath*. Já o coeficiente β_1 é o fator de aceleração para o grupo $x = 1$, ou seja, *Zena*. Assim, com as estimativas obtidas, temos estimativas para os fatores de aceleração para os dois grupos.

Item f)

Aqui é proposto que, se possível, comparemos os modelos de b) e e) para determinarmos qual se ajusta melhor aos dados apresentados.

A comparação entre os modelos é possível e existem algumas formas de se fazer isso diretamente. No decorrer do desenvolvimento do exercício, levantou-se suspeitas de que o modelo de Cox fosse o mais adequado devido aos níveis descritivos dos testes discutidos e às análises gráficas. Porém, mais comparações podem ser realizadas.

Primeiramente, pode-se comparar o valor para a função log verossimilhança dos dois modelos.

```
logLik(fitCox)
```

```
## 'log Lik.' -120.4887 (df=1)
```

```
logLik(fitWei)
```

```
## 'log Lik.' -214.4206 (df=3)
```

É notório que a função log verossimilhança assume um valor maior para o modelo ajustado de Cox do que para o modelo de Weibull.

Também é possível comparar as medidas AIC (*Akaike's Information Criteria*) e BIC (*Bayesian Information Criteria*) [3] para tomarmos uma decisão entre qual modelo está melhor ajustado.

```
print(AIC(fitCox))
```

```
## [1] 242.9773
```

```
print(AIC(fitWei))
```

```
## [1] 434.8412
```

```
print(BIC(fitCox))
```

```
## [1] 244.5037
```

```
print(BIC(fitWei))
```

```
## [1] 441.0739
```

Tanto a medida de BIC quanto de AIC para o modelo Weibull ajustado são maiores do que as medidas correspondentes obtidas para o modelo de Cox ajustado. Assim sendo, pode-se concluir, baseando-se nas medidas aqui discutidas, que o modelo de Cox é mais adequado e se ajusta melhor aos dados de interesse do que o modelo Weibull.

Exercício 2

O segundo exercício, sugere a utilização de Testes de Falha Acelerada para o estudo do tempo de vida de uma bateria sob condições de estresse padrão (no caso do exercício, a bateria irá performar sob uma tensão de 20kV em situações normais). Assim sendo, foram realizados testes sobre um grupo de baterias o qual foi dividido em subgrupos, os quais foram colocados sob diferentes níveis de estresse (28kV, 30kV, 32kV). A partir destes testes, tem-se o objetivo de obter informações sobre a curva de sobrevivência deste tipo de bateria sob 20kV.

Assim sendo, abaixo estão apresentados os dados.

```
#### Testes Acelerados ####
```

```
# Para 28 kV:
```

```
## Tempos
```

```
t1 <- c(68.85, 70, 76.75, 108, 110.29, rep(120,5))
```

```
## Censuras
```

```
c1 <- c(rep(1,5),rep(0,5))
```

```
# Fator de estresse
```

```
x1 <- c(rep(28,10))
```

```
# Para 30 kV
```

```
## Tempos
```

```
t2 <- c(32.76, 35.66, 35.76, 39.85, 40, 40.25, 47.05, 54, 72, 81)
```

```
## Censuras
c2 <- rep(1,10)
## Fator de estresse
x2 <- c(rep(30,10))

# Para 32 kV

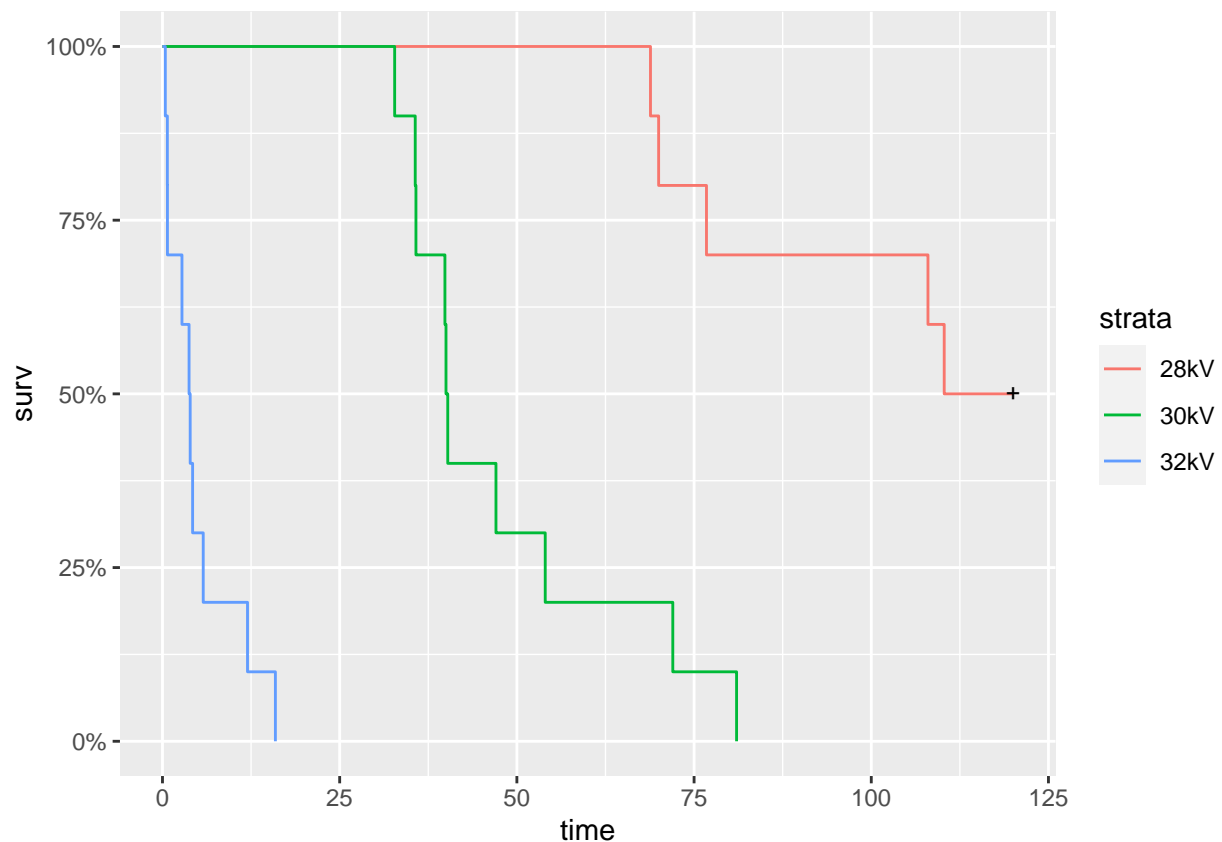
## Tempos
t3 <- c(0.4, 0.69, 0.7, 2.75, 3.75, 3.91, 4.25, 5.75, 12, 15.93)
## Censuras
c3 <- rep(1,10)
## Fator de estresse
x3 <- c(rep(32,10))
```

```
# Observação do comportamento das curvas de sobrevivência por ajustes KM

t_baterias <- c(t1,t2,t3)
c_baterias <- c(c1,c2,c3)
grupo_baterias <- c(rep("28kV",10), rep("30kV",10), rep("32kV",10))

fit <- survfit(Surv(t_baterias,c_baterias) ~ grupo_baterias)

autoplot(fit, conf.int = F)
```



Claramente, como esperado, o tempo de vida das baterias é extremamente prejudicado pelo aumento da

variável de estresse (tensão). Utilizando-se das propriedades dos modelos baseados em testes acelerados, pode-se estimar medidas da curva de sobrevivência da bateria para condições nas quais ela será operada normalmente.

Aqui será considerado um modelo utilizando-se da relação da potência inversa, o qual segue a seguinte forma:

$$T = \frac{A}{V^{\beta_1}} = \exp\{\log A - \beta_1 \log V\}$$

O qual pode ser linearizado aplicando-se a função logarítmica para:

$$\log T = \log A - \beta_1 \log V = \log A + \beta_1(-\log V) = \beta_0 + \beta_1 x$$

Onde T é o tempo até a falha, $\beta_0 = \log A$ e β_1 são constantes características do equipamento e V é o valor assumido pela variável de estresse. O fator de aceleração entre o tempo de falha T no nível de estresse V e o tempo de falha T' no nível de estresse V' é dado por:

$$\gamma = \left[\frac{V'}{V} \right]^{\beta_1}$$

Primeiramente, comparemos alguns modelos paramétricos para que seja escolhido um modelo com um ajuste razoável para que as estimações de interesse sejam feitas.

```
# Divisão por grupos com valores numéricos para regressão
grupo_num <- c(rep(28,10), rep(30,10), rep(32,10))
# Aplicação do logarítimo
log_g_baterias <- -log(grupo_num)

# Ajuste modelo paramétrico Weibull
TFA_wei <- survreg(Surv(t_baterias,c_baterias) ~ log_g_baterias,
                  dist="weibull")
# Ajuste modelo paramétrico exponencial
TFA_exp <- survreg(Surv(t_baterias,c_baterias) ~ log_g_baterias,
                  dist="exponential")
# Ajuste modelo paramétrico Lognormal
TFA_logn <- survreg(Surv(t_baterias,c_baterias) ~ log_g_baterias,
                  dist="lognormal")
# Ajuste modelo paramétrico Log-logístico
TFA_logl <- survreg(Surv(t_baterias,c_baterias) ~ log_g_baterias,
                  dist="loglogistic")
# Ajuste modelo paramétrico Rayleigh
TFA_ray <- survreg(Surv(t_baterias,c_baterias) ~ log_g_baterias,
                  dist="rayleigh")
# Ajuste modelo paramétrico Gaussiano
TFA_gauss <- survreg(Surv(t_baterias,c_baterias) ~ log_g_baterias,
                  dist="gaussian")
```

Abaixo, apresenta-se um estudo de adequabilidade gráfico baseado nas medidas BIC, AIC e log verossimilhança:

```
df <- data.frame("Modelo"= c("Weibull","Exponencial","Lognormal",
                             "Log-logístico","Rayleigh","Gaussiano"),
```

```

"AIC" = c(AIC(TFA_wei), AIC(TFA_exp), AIC(TFA_logn),
          AIC(TFA_logl), AIC(TFA_ray), AIC(TFA_gauss) ),

"BIC" = c(BIC(TFA_wei), BIC(TFA_exp), BIC(TFA_logn),
          BIC(TFA_logl), BIC(TFA_ray), BIC(TFA_gauss) ),

"Log Likelihood" =
  c(logLik(TFA_wei), logLik(TFA_exp), logLik(TFA_logn),
    logLik(TFA_logl), logLik(TFA_ray), logLik(TFA_gauss) )
)
df

```

##	Modelo	AIC	BIC	Log.Likelihood
## 1	Weibull	215.3762	219.5798	-104.6881
## 2	Exponencial	218.1341	220.9365	-107.0671
## 3	Lognormal	219.5853	223.7889	-106.7927
## 4	Log-logístico	219.6127	223.8162	-106.8063
## 5	Rayleigh	217.6758	220.4782	-106.8379
## 6	Gaussiano	231.7370	235.9406	-112.8685

É notório que as medidas BIC e AIC possuem valores menores para o modelo de Weibull ajustado, além disso, a função log cerossimilhança para esse modelo possui o maior valor dentre os estudados. Assim sendo, a análise será continuada utilizando-se desse modelo.

Item a)

Aqui iremos considerar a mesma parametrização utilizada no item e) do exercício 1, ou seja:

$$S(t|\mathbf{x}) = \exp \left\{ - \left(\frac{t}{\exp\{\mathbf{x}'\boldsymbol{\beta}\}} \right)^\alpha \right\}$$

Que para o caso aqui estudado, no qual temos apenas uma covariável envolvida, tem-se:

$$S(t|x) = \exp \left\{ - \left(\frac{t}{\exp\{\beta_0 + \beta_1 x\}} \right)^\alpha \right\}$$

O primeiro item pede que seja obtida uma estimativa para o tempo médio de falha das baterias operando em 20kV e seu respectivo erro-padrão. Sabe-se que um estimador para o valor médio é dado por:

$$\widehat{E(T|x)} = \exp\{\hat{\beta}_0 + \hat{\beta}_1 x\} \Gamma \left(1 + \frac{1}{\hat{\alpha}} \right)$$

Assim sendo, foi utilizada da relação acima para a obtenção de uma estimativa para o tempo médio de falha e do método *delta* para a obtenção de seu respectivo erro-padrão.

```

# Valores estimado para beta1 e beta2
Beta <- TFA_wei$coefficients
# Parametrização do R: sigma = 1/alpha
sigma <- TFA_wei$scale
alpha <- 1/sigma

```

```

# Tempo médio de falha estimado  $\hat{E}(T/x)$ 
# (Objetiva-se estimar em  $x = -\log(20)$ , ou seja, sobre estresse de 20kV)
tm_est <- exp(Beta[1] + Beta[2]*(-log(20)))*gamma(1/alpha + 1)

# Aplicação do método Delta
cov <- TFA_wei$var
emv <- c(Beta,TFA_wei$icoef[2])
ep_20 <- deltamethod(list(~exp(x1+x2*(-log(20)))*gamma(exp(x3)+1)), emv, cov)

# Convertendo-se os valores de minutos para anos, tem-se:
cat("\nValor estimado para o tempo médio de falha para 20kV:",
    tm_est/60/24/365, "anos")

```

```

##
## Valor estimado para o tempo médio de falha para 20kV: 2.905289 anos

```

```

cat("\nRespectivo erro-padrão para 20kV: ", ep_20/60/24/365, "anos")

```

```

##
## Respectivo erro-padrão para 20kV: 5.810997 anos

```

Item b)

Aqui, objetiva-se obter a mediana estimada para a curva de sobrevivência em $V = 20kV$. Pode-se estimar tal quantidade por:

$$\widehat{t_{0.5}} = \exp\{\beta_0 + \beta_1 x\}(-\log 0,5)^{1/\hat{\alpha}} = \exp\{\beta_0 + \beta_1 x\}(\log 2)^{1/\hat{\alpha}}$$

```

#Tempo mediano estimado
t05 <- exp(Beta[1] + Beta[2]*(-log(20)))*log(2)^(1/alpha)
#Método delta para a obtenção do erro-padrão do estimador
ep_t05 <- deltamethod(list(~log(2)^exp(x3)*exp(x1 + x2*(-log(20))))), emv, cov)

# Convertendo-se os valores de minutos para anos, tem-se:
cat("\nValor estimado para o tempo mediano de falha para 20kV:",
    t05/60/24/365, "anos")

```

```

##
## Valor estimado para o tempo mediano de falha para 20kV: 2.499003 anos

```

```

cat("\nRespectivo erro-padrão para 20kV: ", ep_t05/60/24/365, "anos")

```

```

##
## Respectivo erro-padrão para 20kV: 2.726491 anos

```

Item c)

Como o fabricante pretende arcar com o custo de 5% das baterias, tem-se interesse no valor $t_{0.05}$, o qual pode ser estimado, a partir do desenvolvimento de $S(T|x)$ por:

$$t_{0.05} = \exp\{\beta_0 + \beta_1 x\}(-\log 0,95)^{1/\hat{\alpha}}$$

```
#Estimativa de t0.05
t005 <- exp(Beta[1] + Beta[2]*(-log(20)))*(-log(0.95))^(1/alpha)
#Método delta para a obtenção do erro-padrão
ep_t005 <- deltamethod(list(~exp(x1+x2*(-log(20)))*(-log(1-0.05))^(exp(x3))),
                        emv, cov)

# Convertendo-se os valores de minutos para anos, tem-se:
cat("\nValor estimado para o tempo mediano de falha para 20kV:",
    t005/60/24/365, "anos")
```

```
##
## Valor estimado para o tempo mediano de falha para 20kV: 0.4230193 anos
```

```
cat("\nRespectivo erro-padrão para 20kV: ", ep_t005/60/24/365, "anos")
```

```
##
## Respectivo erro-padrão para 20kV: 0.07429516 anos
```

Item d)

É possível obter a proporção de baterias que falharam em 2 anos a partir da função de sobrevivência estimada, a qual deriva de:

$$S(t|x) = \exp\left\{-\left(\frac{t}{\exp\{\beta_0 + \beta_1 x\}}\right)^\alpha\right\}$$

ao aplicar as estimativas obtidos para os parâmetros α , β_0 e β_1 .

Assim, basta apenas converter os 2 anos (considerando-se 12 horas de uso por dia) para minutos e aplicar a equação acima.

```
# Conversão de 2 anos para minutos
t <- 2*365*12*60
#Sobrevivencia estimada para 2 anos
S <- exp(-(t/exp(Beta[1] + Beta[2]*(-log(20))))^alpha)
cat(1-S)
```

```
## 0.1655929
```

Assim sendo, a proporção de baterias que falham em 2 anos nas condições citadas é de aproximadamente 0,165.

Item e)

Sabe-se que o fator de aceleração γ é dado por:

$$\gamma = \left[\frac{V'}{V} \right]^{\beta_1}$$

Assim sendo, pode-se estimar o fator de aceleração entre 20kV e 25 kV por:

$$\gamma = \left[\frac{25}{20} \right]^{\hat{\beta}_1}$$

```
cat((25/20)^Beta[2])
```

```
## 357.8453
```

Como o fator de aceleração estimado de 20kV para 25kV foi de aproximadamente 357,84; pode-se interpretar que o tempo de vida para baterias operando a 25kV será acelerado nesse valor quando comparada com o tempo de vida de baterias operando a 20kV.

Exercício 3

No presente exercício, o objetivo é ajustar diferentes modelos de longa duração com o intuito de modelar o comportamento da curva de sobrevivência em um estudo sobre melanoma. Segue a descrição do conjunto de dados sendo utilizado:

"Os dados deste exemplo foram coletados em um estudo sobre melanoma com o objetivo de avaliar o desempenho da aplicação de uma dosagem alta de interferon alfa-2b como forma de prevenir recorreência do câncer. Os pacientes foram incluídos no estudo entre 1991 e 1995, tendo sido acompanhados até 1998. Uma descrição mais detalhada dos dados pode ser vista em Ibrahim et al (2001). A variável resposta (T) representa o tempo até a morte do paciente ou o tempo de censura. Da amostra original foram removidos 10 pacientes para os quais há valores faltantes em uma das covariáveis (x_3), restando $n = 417$ pacientes, com 56% de observações censuradas. As variáveis incluem:

- t : tempo (em anos; média = 3,18 e desvio padrão = 1,69);
- x_1 : tipo de tratamento (0: sem tratamento, $n = 204$; 1: interferon, $n = 213$);
- x_2 idade (1: > 48 ; $n = 220$; ≤ 48 , $n = 197$);
- x_3 categoria do nódulo (1, $n = 82$; 2, $n = 87$; 3, $n = 137$; 4, $n = 111$);
- x_4 sexo (0: masculino, $n = 263$; 1: feminino, $n = 154$);
- x_5 capacidade funcional (0: ativo, $n = 363$; 1: outras, $n = 54$) ;
- x_6 espessura do tumor (em mm, média = 3,94 e desvio padrão = 3,20)."

Assim sendo, foram ajustados, assumindo que $S(t)$ é associado aos modelos log-logístico e log-normal, três modelos de longa duração (mistura padrão, tempo de promoção e tempo de promoção com dispersão).


```

# Carregando os dados de interesse para a variável "dados"
dados <- read.table("melanoma.txt", header=TRUE, sep="\t")

# Tempos observados
y = dados$t

# Indicador de censura
status = dados$d1

# Tipo de tratamento
x1 = dados$x1

# Idade (com encoding binário)
x2 = ifelse(dados$x2>48,1,0)

# Categoria de nódulo
x3 = as.factor(dados$x3)

# Sexo (com encoding binário)
x4 = as.factor(dados$x4)

# Capacidade funcional
x5 = as.factor(dados$x5)

# Espessura do tumor
x6 = dados$x6

```

Para o ajuste dos modelos propostos, algumas funções foram criadas para a obtenção da função de distribuição e de distribuição acumulada dos modelos probabilísticos log-logístico e log-normal. Basicamente, tais funções utilizam das funções prontas do pacote *flexsurv* e do R para obter tais quantidades mas limitam os valores que podem ser inseridos para os parâmetros ao espaço paramétrico para tais modelos, alertando quando valores não aceitos são atribuídos. As funções também alertam para valores que não pertence ao suporte da função.

```

# Modelo Log-logístico

## Função de distribuição
d_log_logistica <- function(y, mu = 1, sigma = 0, log = FALSE){

  # Verificação de parâmetros e entradas
  if(any(mu <= 0)){
    stop(paste("mu precisa ser positivo", "\n", ""))
  }

  if(any(y < 0)){
    stop(paste("y precisa ser > 0 ", "\n", ""))
  }

  alpha = exp(mu) # Parâmetro de escala
  beta = 1/sigma # Parâmetro de forma

  # Obtenção dos valores da função de distribuição log-logística para y
  fy = dllogis(y, shape = beta, scale = alpha, log = log)
  return(fy)
}

```

```

}

## Função de distribuição acumulada
p_log_logistica <- function(q,mu = 1,sigma = 0,lower.tail = TRUE,log.p = FALSE){

  # Verificação de parâmetros e entradas
  if(any(mu <= 0)){
    stop(paste("mu precisa ser positivo", "\n", ""))
  }

  if(any(q < 0)){
    stop(paste("y precisa ser > 0 ", "\n", ""))
  }

  alpha = exp(mu) # Parâmetro de escala
  beta = 1/sigma # Parâmetro de forma

  # Obtenção dos valores da função de distribuição acumulada
  # log-logística para q
  cdf = pllogis(q, shape = beta, scale = alpha, lower.tail = lower.tail,
                log.p = log.p)
  return(cdf)
}

# Modelo log-normal

## Função de distribuição
d_log_normal <- function(y, mu = 1, sigma = 0, log = FALSE){
  # Verificação de parâmetros e entradas
  if(any(mu <= 0)){
    stop(paste("mu precisa ser positivo", "\n", ""))
  }

  if(any(y < 0)){
    stop(paste("y precisa ser > 0 ", "\n", ""))
  }

  # Obtenção dos valores da função de distribuição log-normal para y
  fy = dlnorm(y, meanlog = mu, sdlog = sigma, log = log)
  return(fy)
}

## Função de distribuição acumulada
p_log_normal <- function(q,mu = 1,sigma = 0,lower.tail = TRUE,log.p = FALSE){
  # Verificação de parâmetros e entradas
  if(any(mu <= 0)){
    stop(paste("mu precisa ser positivo", "\n", ""))
  }

  if(any(q < 0)){
    stop(paste("y precisa ser > 0 ", "\n", ""))
  }
}

```

```

# Obtenção dos valores da função de distribuição acumulada
# log-normal para q
cdf = plnorm(q, meanlog = mu, sdlog = sigma, lower.tail = lower.tail,
             log.p = log.p)
return(cdf)
}

```

Os modelos serão comparados utilizando-se das medidas AIC e do valor da função log verossimilhança para os parâmetros estimados. Assim sendo, declaremos vetores para armazenar tais medidas para cada modelo.

```

AIC <- c()
LogL <- c()

```

Antes de finalmente ajustar os modelos, é necessária a criação de funções de obtenção do valor da função log verossimilhança para os seis casos estudados (cada um dos modelos de longa duração com as duas distribuições de interesse).

```

# Função de obtenção da log verossimilhança para o modelo de mistura padrão,
# tendo-se como distribuição da função de sobrevivência a lognormal

```

```

Log_vero_mistura_ln <- function(vpar) {
  ph0 = vpar[1]
  ph1 = vpar[2]
  beta = vpar[-c(1:2)]
  linear = c(X%*%beta)
  theta = plogis(linear) # ligação logística
  vf = d_log_normal(y,mu=ph0,sigma=ph1)
  vF = p_log_normal(y,mu=ph0,sigma=ph1)
  Spop <- theta+(1-theta)*(1-vF)
  fpop <- (1-theta)*vf
  loglik <- sum(status*log(fpop)+(1-status)*log(Spop))
  loglik
}

```

```

# Função de obtenção da log verossimilhança para o modelo de mistura padrão,
# tendo-se como distribuição da função de sobrevivência a log-logística

```

```

Log_vero_mistura_ll <- function(vpar) {
  ph0 = vpar[1]
  ph1 = vpar[2]
  beta = vpar[-c(1:2)]
  linear = c(X%*%beta)
  theta = plogis(linear) # ligação logística
  vf = d_log_logistica(y,mu=ph0,sigma=ph1)
  vF = p_log_logistica(y,mu=ph0,sigma=ph1)
  Spop <- theta+(1-theta)*(1-vF)
  fpop <- (1-theta)*vf
  loglik <- sum(status*log(fpop)+(1-status)*log(Spop))
  loglik
}

```

*# Função de obtenção da log verossimilhança para o modelo de Tempo de Promoção,
tendo-se como distribuição da função de sobrevivência a lognormal*

```
Log_vero_p_ln <- function(vpar) {
  ph0 = (vpar[1])
  ph1 = vpar[2]
  beta = vpar[-c(1:2)]
  linear = c(X%*%beta)
  theta = exp(linear)
  vf = d_log_normal(y,mu=ph0,sigma=ph1)
  vF = p_log_normal(y,mu=ph0,sigma=ph1)
  Spop <- exp(-theta*(vF))
  fpop <- theta*Spop*vf
  loglik <- sum(status*log(fpop)+(1-status)*log(Spop))
  loglik
}
```

*# Função de obtenção da log verossimilhança para o modelo de Tempo de Promoção,
tendo-se como distribuição da função de sobrevivência a log-logística*

```
Log_vero_p_ll <- function(vpar) {
  ph0 = (vpar[1])
  ph1 = vpar[2]
  beta = vpar[-c(1:2)]
  linear = c(X%*%beta)
  theta = exp(linear)
  vf = d_log_logistica(y,mu=ph0,sigma=ph1)
  vF = p_log_logistica(y,mu=ph0,sigma=ph1)
  Spop <- exp(-theta*(vF))
  fpop <- theta*Spop*vf
  loglik <- sum(status*log(fpop)+(1-status)*log(Spop))
  loglik
}
```

*# Função de obtenção da log verossimilhança para o modelo de Modelo de Tempo de
Promoção com Dispersão, tendo-se como distribuição da função de sobrevivência
a lognormal*

```
Log_vero_b_ln <- function(vpar) {
  gamma=vpar[1]
  ph0 = vpar[2]
  ph1 = vpar[3]
  beta = vpar[-c(1:3)]
  linear=(X%*%beta)
  theta=exp(linear) # ligação log
  vf = d_log_normal(y,mu=ph0,sigma=ph1)
  vF = p_log_normal(y,mu=ph0,sigma=ph1)
  Spop=(1+theta*gamma*vF)^(-1/gamma)
  fpop=theta*vf*(1+theta*gamma*vF)^(-1/gamma-1)
  loglik <- sum(status*log(fpop)+(1-status)*log(Spop))
  loglik
}
```

```
# Função de obtenção da log verossimilhança para o modelo de Modelo de Tempo de
# Promoção com Dispersão, tendo-se como distribuição da função de sobrevivência
# a log-logística
```

```
Log_vero_b_ll <- function(vpar) {
  gamma=vpar[1]
  ph0 = vpar[2]
  ph1 = vpar[3]
  beta = vpar[-c(1:3)]
  linear=(X%*%beta)
  theta=exp(linear) # ligação log
  vf = d_log_logistica(y,mu=ph0,sigma=ph1)
  vF = p_log_logistica(y,mu=ph0,sigma=ph1)
  Spop=(1+theta*gamma*vF)^(-1/gamma)
  fpop=theta*vf*(1+theta*gamma*vF)^(-1/gamma-1)
  loglik <- sum(status*log(fpop)+(1-status)*log(Spop))
  loglik
}
```

Modelo de Mistura Padrão

Abaixo, ajustam-se 2 modelos de Mistura Padrão; um com a função lognormal e outro com a função log_logística

Lognormal

```
X=model.matrix(~1+x1+x2+x3+x4+x5+x6)
nc=ncol(X)
```

```
vpar1 = c(1,1,rep(0.1,nc))
Log_vero_mistura_ln(vpar1)
```

```
## [1] -586.8452
```

```
fitm=optim(vpar1,Log_vero_mistura_ln,control=list(fnscale=-1),method="L-BFGS-B",
          hessian = T,
          lower=c(0.0001,-Inf,rep(-Inf,(nc))),
          upper=c(rep(Inf,length(vpar1))))
EMVm=fitm$par
covm=solve(-fitm$hessian)
Epm=sqrt(diag(covm))

Estm=EMVm/Epm
p_valorm=2*(1-pnorm(abs(Estm)))
Saidam=cbind(EMVm, Epm,Estm,p_valorm)
rownames(Saidam)=c("phi_1","phi_2",paste("beta_", 0:(nc-1), sep = ""))
#print(Saidam, digits = 3)
AIC_m=-2*fitm$value+2*length(EMVm)
```

```

LogL <- c(LogL,Log_vero_mistura_ln(vpar1))
AIC <- c(AIC, AIC_m)
library(xtable)
xtable(Saidam,digits=3)

## % latex table generated in R 4.0.3 by xtable 1.8-4 package
## % Sun Jul 18 15:26:46 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
## \hline
## & EMVm & EPm & Estm & p\_valorm \\
## \hline
## phi\_1 & 0.653 & 0.125 & 5.243 & 0.000 \\
## phi\_2 & 0.867 & 0.084 & 10.322 & 0.000 \\
## beta\_0 & 1.255 & 0.514 & 2.444 & 0.015 \\
## beta\_1 & -0.135 & 0.263 & -0.512 & 0.609 \\
## beta\_2 & -0.333 & 0.268 & -1.244 & 0.214 \\
## beta\_3 & -0.776 & 0.418 & -1.856 & 0.063 \\
## beta\_4 & -1.176 & 0.500 & -2.353 & 0.019 \\
## beta\_5 & -2.203 & 0.650 & -3.392 & 0.001 \\
## beta\_6 & 0.228 & 0.269 & 0.848 & 0.397 \\
## beta\_7 & -0.151 & 0.406 & -0.372 & 0.710 \\
## beta\_8 & -0.104 & 0.069 & -1.507 & 0.132 \\
## \hline
## \end{tabular}
## \end{table}

```

Log-logística

```

X=model.matrix(~1+x1+x2+x3+x4+x5+x6)
nc=ncol(X)

vpar1 = c(1,1,rep(0.1,nc))
Log_vero_mistura_ll(vpar1)

## [1] -624.0537

fitm=optim(vpar1,Log_vero_mistura_ll,control=list(fnscale=-1),method="L-BFGS-B",
          hessian = T,
          lower=c(0.0001,-Inf,rep(-Inf,(nc))),
          upper=c(rep(Inf,length(vpar1))))
EMVm=fitm$par
covm=solve(-fitm$hessian)
EPm=sqrt(diag(covm))

Estm=EMVm/EPm
p_valorm=2*(1-pnorm(abs(Estm)))
Saidam=cbind(EMVm, EPm,Estm,p_valorm)
rownames(Saidam)=c("phi_1","phi_2",paste("beta_", 0:(nc-1), sep = ""))

```

```

# print(Saidam, digits = 3)
AIC_m = -2*fitm$value + 2*length(EMVm)

LogL <- c(LogL, Log_vero_mistura_ll(vpar1))
AIC <- c(AIC, AIC_m)
library(xtable)
xtable(Saidam, digits=3)

## % latex table generated in R 4.0.3 by xtable 1.8-4 package
## % Sun Jul 18 15:26:48 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
## \hline
## & EMVm & EPm & Estm & p\_valorm \\
## \hline
## phi\_1 & 0.602 & 0.084 & 7.180 & 0.000 \\
## phi\_2 & 0.475 & 0.042 & 11.270 & 0.000 \\
## beta\_0 & 1.205 & 0.463 & 2.601 & 0.009 \\
## beta\_1 & -0.137 & 0.250 & -0.547 & 0.584 \\
## beta\_2 & -0.305 & 0.251 & -1.215 & 0.224 \\
## beta\_3 & -0.705 & 0.377 & -1.873 & 0.061 \\
## beta\_4 & -1.070 & 0.425 & -2.517 & 0.012 \\
## beta\_5 & -2.025 & 0.508 & -3.982 & 0.000 \\
## beta\_6 & 0.225 & 0.257 & 0.878 & 0.380 \\
## beta\_7 & -0.160 & 0.383 & -0.418 & 0.676 \\
## beta\_8 & -0.088 & 0.057 & -1.557 & 0.119 \\
## \hline
## \end{tabular}
## \end{table}

```

Modelo de Tempo de Promoção

Abaixo, ajustam-se um modelo de de Tempo de Promoção com a função lognormal e um com a log logística.

Lognormal

```

vpar1 = c(1, 1, rep(0.1, nc))
Log_vero_p_ln(vpar1)

## [1] -635.1584

fitp=optim(vpar1, Log_vero_p_ln, control=list(fnscale=-1), method="L-BFGS-B",
          hessian = T,
          lower=c(0.00001, -Inf, rep(-Inf, (nc))),
          upper=c(rep(Inf, length(vpar1))))
EMVp=fitp$par
covp=solve(-fitp$hessian)
EPp=sqrt(diag(covp))

```

```

Estp=EMVp/EPp
p_valorp=2*(1-pnorm(abs(Estp)))
Saidap=cbind(EMVp, EPp,p_valorp)
#print(Saidap, digits = 3)
AIC_p=-2*fitp$value+2*length(EMVp)
library(xtable)
LogL <- c(LogL,Log_vero_p_ln(vpar1))
AIC <- c(AIC, AIC_p)
xtable(Saidap,digits=3)

```

```

## % latex table generated in R 4.0.3 by xtable 1.8-4 package
## % Sun Jul 18 15:26:49 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrr}
## \hline
## & EMVp & EPp & p\_valorp \\
## \hline
## 1 & 0.918 & 0.160 & 0.000 \\
## 2 & 0.898 & 0.086 & 0.000 \\
## 3 & -0.838 & 0.279 & 0.003 \\
## 4 & 0.067 & 0.150 & 0.655 \\
## 5 & 0.108 & 0.152 & 0.475 \\
## 6 & 0.411 & 0.236 & 0.082 \\
## 7 & 0.639 & 0.237 & 0.007 \\
## 8 & 1.164 & 0.225 & 0.000 \\
## 9 & -0.186 & 0.156 & 0.233 \\
## 10 & 0.142 & 0.211 & 0.501 \\
## 11 & 0.023 & 0.023 & 0.312 \\
## \hline
## \end{tabular}
## \end{table}

```

Log-logística

```

vpar1 = c(1,1,rep(0.1,nc))
Log_vero_p_ll(vpar1)

```

```
## [1] -674.0988
```

```

fitp=optim(vpar1,Log_vero_p_ll,control=list(fnscale=-1),method="L-BFGS-B",
          hessian = T,
          lower=c(0.00001,-Inf,rep(-Inf,(nc))),
          upper=c(rep(Inf,length(vpar1))))
EMVp=fitp$par
covp=solve(-fitp$hessian)
EPp=sqrt(diag(covp))

Estp=EMVp/EPp
p_valorp=2*(1-pnorm(abs(Estp)))

```



```

Saidap=cbind(EMVp, EPp,p_valorp)
#print(Saidap, digits = 3)
AIC_p=-2*fitp$value+2*length(EMVp)
library(xtable)
LogL <- c(LogL,Log_vero_p_ll(vpar1))
AIC <- c(AIC, AIC_p)
xtable(Saidap,digits=3)

## % latex table generated in R 4.0.3 by xtable 1.8-4 package
## % Sun Jul 18 15:26:50 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrr}
## \hline
## & EMVp & EPp & p\_valorp \\
## \hline
## 1 & 0.789 & 0.097 & 0.000 \\
## 2 & 0.468 & 0.040 & 0.000 \\
## 3 & -0.935 & 0.266 & 0.000 \\
## 4 & 0.066 & 0.150 & 0.659 \\
## 5 & 0.105 & 0.152 & 0.489 \\
## 6 & 0.410 & 0.236 & 0.082 \\
## 7 & 0.643 & 0.237 & 0.007 \\
## 8 & 1.174 & 0.225 & 0.000 \\
## 9 & -0.189 & 0.156 & 0.226 \\
## 10 & 0.145 & 0.211 & 0.491 \\
## 11 & 0.023 & 0.023 & 0.326 \\
## \hline
## \end{tabular}
## \end{table}

```

Modelo de Tempo de Promoção com Dispersão

Abaixo, ajustam-se um modelo de Tempo de Promoção com Dispersão com a função lognormal

Lognormal

```

vpar1 = c(0.5,1,log(mean(y)),rep(1,nc))
fitb=optim(vpar1,Log_vero_b_ln,control=list(fnscale=-1),method="L-BFGS-B",
          hessian = T,
          lower=c(0.0001,0.0001,-Inf,rep(-Inf,(nc))),
          upper=c(rep(Inf,length(vpar1))))

EMVb=fitb$par
covb=solve(-fitb$hessian)
EPb=sqrt(diag(covb))

Estb=EMVb/EPb
p_valorb=2*(1-pnorm(abs(Estb)))
Saidab=cbind(EMVb, EPb,p_valorb)

```

```

rownames(Saidab)=c("eta","phi_1","phi_2",paste("beta_", 0:(nc-1), sep = ""))
#print(Saidab, digits = 3)
AIC_b=-2*fitb$value+2*length(EMVb)

library(xtable)
LogL <- c(LogL,Log_vero_b_ln(vpar1))
AIC <- c(AIC, AIC_b)
xtable(Saidab,digits=3)

```

```

## % latex table generated in R 4.0.3 by xtable 1.8-4 package
## % Sun Jul 18 15:26:52 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrr}
## \hline
## & EMVb & EPb & p\_valorb \\
## \hline
## eta & 2.522 & 2.292 & 0.271 \\
## phi\_1 & 2.163 & 1.700 & 0.203 \\
## phi\_2 & 1.094 & 0.282 & 0.000 \\
## beta\_0 & 0.651 & 2.346 & 0.782 \\
## beta\_1 & -0.192 & 0.384 & 0.617 \\
## beta\_2 & 0.161 & 0.251 & 0.521 \\
## beta\_3 & 0.776 & 0.493 & 0.115 \\
## beta\_4 & 1.312 & 0.721 & 0.069 \\
## beta\_5 & 2.137 & 0.801 & 0.008 \\
## beta\_6 & -0.113 & 0.308 & 0.713 \\
## beta\_7 & 0.282 & 0.399 & 0.479 \\
## beta\_8 & 0.038 & 0.041 & 0.363 \\
## \hline
## \end{tabular}
## \end{table}

```

Log-logística

```

vpar1 = c(0.5,1,log(mean(y)),rep(1,nc))
fitb=optim(vpar1,Log_vero_b_ll,control=list(fnscale=-1),method="L-BFGS-B",
          hessian = T,
          lower=c(0.0001,0.0001,-Inf,rep(-Inf,(nc))),
          upper=c(rep(Inf,length(vpar1))))

EMVb=fitb$par
covb=solve(-fitb$hessian)
EPb=sqrt(diag(covb))

Estb=EMVb/EPb
p_valorb=2*(1-pnorm(abs(Estb)))
Saidab=cbind(EMVb, EPb,p_valorb)
rownames(Saidab)=c("eta","phi_1","phi_2",paste("beta_", 0:(nc-1), sep = ""))
#print(Saidab, digits = 3)
AIC_b=-2*fitb$value+2*length(EMVb)

```

```

library(xtable)
LogL <- c(LogL, Log_vero_b_ll(vpar1))
AIC <- c(AIC, AIC_b)
xtable(Saidab, digits=3)

## % latex table generated in R 4.0.3 by xtable 1.8-4 package
## % Sun Jul 18 15:26:54 2021
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrr}
## \hline
## & EMVb & EPb & p\_valorb \\
## \hline
## eta & 3.111 & 2.310 & 0.178 \\
## phi\_1 & 1.312 & 0.342 & 0.000 \\
## phi\_2 & 0.412 & 0.062 & 0.000 \\
## beta\_0 & 0.034 & 1.071 & 0.974 \\
## beta\_1 & -0.272 & 0.412 & 0.509 \\
## beta\_2 & 0.158 & 0.273 & 0.564 \\
## beta\_3 & 0.844 & 0.514 & 0.101 \\
## beta\_4 & 1.457 & 0.731 & 0.046 \\
## beta\_5 & 2.340 & 0.777 & 0.003 \\
## beta\_6 & -0.074 & 0.344 & 0.831 \\
## beta\_7 & 0.337 & 0.435 & 0.438 \\
## beta\_8 & 0.034 & 0.045 & 0.446 \\
## \hline
## \end{tabular}
## \end{table}

```

```
cat("\nAIC:", AIC)
```

```

##
## AIC: 1042.599 1041.069 1039.642 1037.433 1037.626 1033.903

```

```
cat("\nLo Likelihood:", LogL)
```

```

##
## Lo Likelihood: -586.8452 -624.0537 -635.1584 -674.0988 -4669.18 -4847.102

```

Considerando-se os valores de AIC observados, o modelo de tempo promoção com dispersão com função log-logística foi o que melhor se ajustou aos dados (AIC = 1037.233). Agora, considerando-se a função log verossimilhança, o modelo de mistura padrão com função lognormal mostrou-se o mais adequado (LogLikelihood=-586,84). Considerando-se a proximidade da AIC para todos os modelos, o modelo mais adequado dos aqui estudados é o modelo de mistura padrão com função lognormal.

Observação: o código em R utilizado nesse exercício foi disponibilizado pelo professor responsável pela disciplina.

Bibliografia:

- [1] COLOSIMO, Enrico Antonio; GIOLO, Suely Ruiz. Análise de sobrevivência aplicada. Editora Blucher, 2006.
- [2] P. Grambsch and T. Therneau (1994), Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*, 81, 515-26. (Informação obtida na documentação R da função *cox.zph* do pacote *survival*)
- [3] <http://www.sthda.com/english/articles/38-regression-model-validation/158-regression-model-accuracy-metrics-r-square-aic-bic-cp-and-more/>