Análise de sobrevivência com dados de diálise - Primeira Avaliação

Universidade Federal da Paraíba - CCEN

Gabriel de Jesus Pereira

18 de agosto de 2024

Questão 1: Considere as seguintes funções $S(t)$ apresentadas abaixo e responda o que se pede.

1. $S_1(t) = e^{-t/5}$, em que $t \geq 0$
2. $S_2(t) = \frac{1}{1+t}$, em que $t \geq 0$.
3. $S_3(t) = 1 - \frac{t}{2}$, em que $t \geq 0$
4. $S_4(t) = 2e^{-t/2}$, em que $t \geq 0$

(a) Considerando as condições, vistas em sala de aula, para que $S(t)$ seja uma função de sobrevivência, identifique quais das funções apresentadas são realmente funções de sobrevivência.

Resolução (a)

Para que $S(t)$ seja uma função de sobrevivência, deve satisfazer as seguintes condições: $S(t) \to 0$ para $t \to \infty$ e $S(t) \to 1$ para $t \to 0$. Assim, temos os seguintes resultados:

1. $\lim_{t\to\infty} e^{-t/5} = 0$ e $\lim_{t\to 0} e^{-t/5} = 1$
2. $\lim_{t\to\infty} \frac{1}{1+t} = 0$ e $\lim_{t\to 0} \frac{1}{1+t} = 1$
3. $\lim_{t\to\infty} (1 - \frac{t}{2}) = -\infty$ e $\lim_{t\to 0} (1 - \frac{t}{2}) = 1$
4. $\lim_{t\to\infty} 2e^{-t/2} = 0$ e $\lim_{t\to 0} 2e^{-t/2} = 2$

Dessa forma, pelas condições necessárias, vemos que apenas $S_1(t)$ e $S_2(t)$ validam a condição para ser uma função de sobrevivência.

```
library(flexsurv)
library(survminer)
library(discSurv)
library(survival)
library(tidyverse)
library(vroom)
library(ggsurvfit)
library(mice)

df <- read_delim("sobrevivencia/primeira_avaliacao/includes/dialcompete.txt", delim =
  mutate(
    intervalo = cut(
      tempo,
```

```
      breaks = 1:44,
      labels = paste0("[", 1:43, ",",  2:44, ")"),
      right = FALSE
    )
  )
```

Questão 2:

Escolha um dos bancos de dados disponíveis no seguinte endereço eletrônico: **http://sobrevida.fiocruz.br/dados.html**. A partir do banco de dados escolhido por você, faça o que se pede a seguir

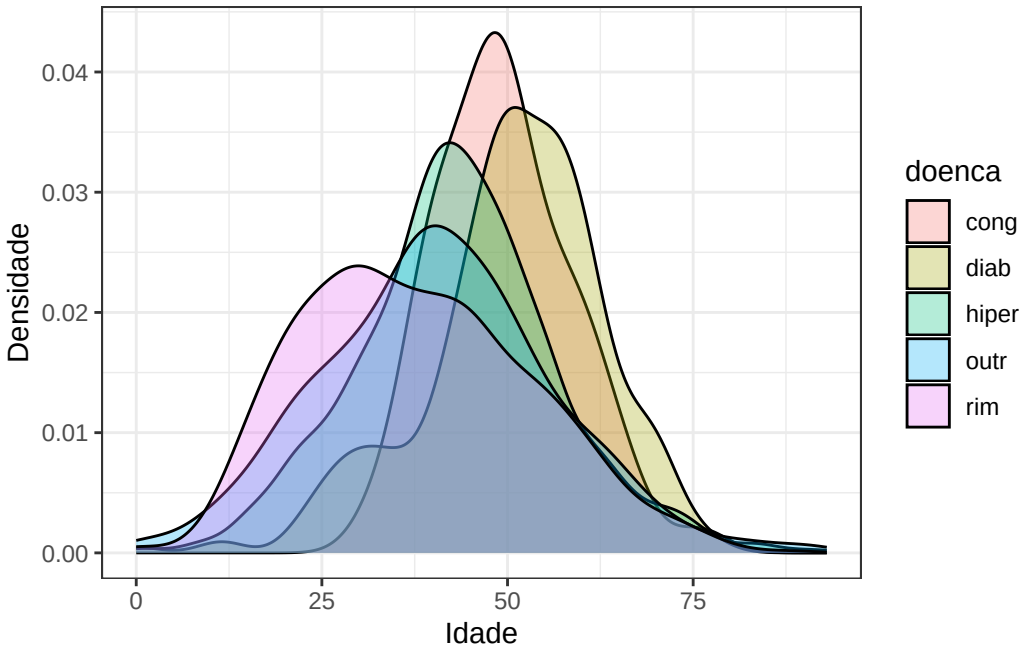
2a)

Faça uma análise exploratória do banco de dados e forneça interpretações pausíveis acerca das variáveis que encontram-se disponíveis.

O banco de dados intitulado "Diálise – eventos competitivos (SUS)" tem como objetivo analisar eventos aos quais pacientes em diálise estão expostos, como transplante, óbito por causa renal e óbito por outras causas. Esse banco contém 6 variáveis diferentes, incluindo o ID do paciente, a idade ao iniciar a diálise, o motivo do óbito (seja por causa renal, por outra causa ou transplante, censura). Além disso, há informações sobre a doença do paciente, que pode ser hipertensão, diabetes, problemas renais, condições congênitas ou outras. O banco também inclui o status, indicando se ocorreu falha ou censura, e o tempo até a ocorrência de um dos possíveis eventos.

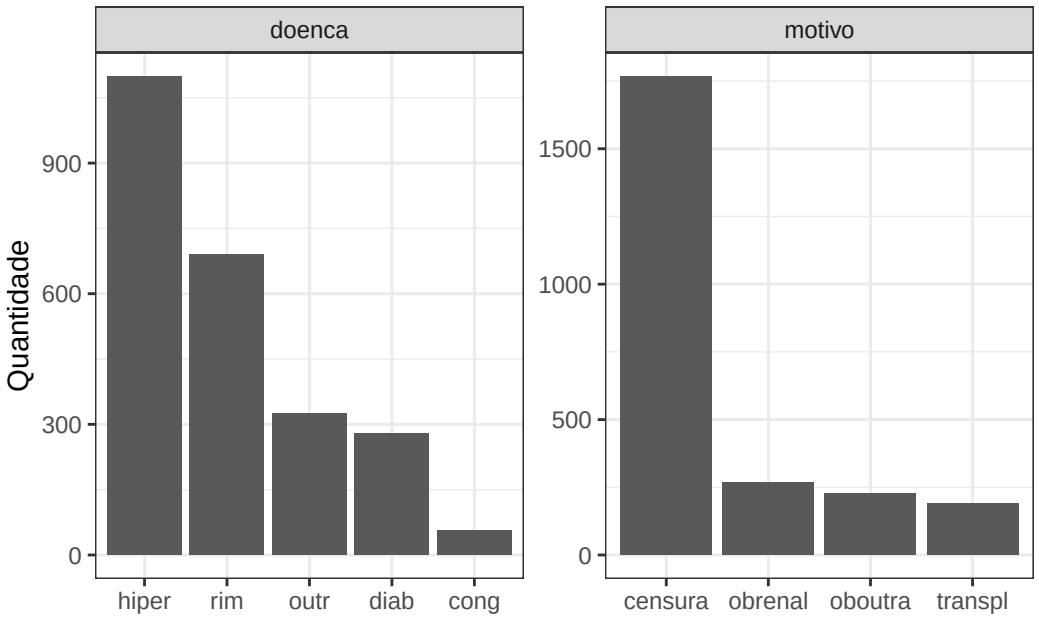
Conforme o gráfico abaixo, observa-se que os pacientes que iniciam o tratamento de diálise mais tarde tendem a ser aqueles com doenças congênitas ou diabetes. Em contrapartida, pessoas com doenças renais geralmente começam o tratamento mais cedo. É importante notar que a idade ao iniciar a diálise pode influenciar sua eficácia.

```
df |>
  select(-intervalo) |>
  ggplot(aes(x = idade, fill = doenca)) +
  geom_density(alpha = 0.3) +
  theme_bw() +
  labs(y = "Densidade", x = "Idade")
```



Pelo gráfico abaixo vemos que a maior parte dos pacientes são hipertensos, enquanto a menor parte tem doenças congênitas. Além disso, óbito por causa renal é a mais frequente. Por último, os dados de sobrevivência tem 1767 censura e apenas 686 falhas.

```
df |>
  select(-intervalo) |>
  pivot_longer(
    cols = c(doenca, motivo),
    names_to = "Variável",
    values_to = "teste"
  ) |>
  ggplot(aes(x = fct_infreq(teste))) +
  geom_bar() +
  facet_wrap(vars(Variável), scales = "free") +
  theme_bw() +
  labs(y = "Quantidade", x = "")
```



2b)

Considerando a variável de tempo até ocorrência do evento de interesse na base de dados escolhida, forneça as seguintes informações:

i) É possível montar uma tabela para descrever os dados de acordo com o número de ocorrências do evento de interesse registradas em intervalos de tempo da pesquisa? Se sim, apresente-a.

```
tabela_eventos <- df |>
  group_by(intervalo) |>
  mutate(
    falha = sum(status, na.rm = TRUE),
    censura = sum(status == 0, na.rm = TRUE),
    amplitude = (\(x) {
      inter = as.character(intervalo) |>
        gsub(pattern = "\\[|\\(|\\)", replacement = "") |>
        strsplit(",")

      sapply(
        X = inter,
        FUN = \(x) {
          as.numeric(x[2]) - as.numeric(x[1])
```

```
        })
    })(),
  ) |>
  ungroup() |>
  arrange(intervalo) |>
  filter(status != 0) |>
  select(-c(
    status, id, idade,
    doenca, motivo)
  ) |>
  distinct(intervalo, .keep_all = TRUE)

nj <- nrow(df)
for (x in 2:nrow(tabela_eventos)) {
  nj[x] <-  nj[x - 1] - (tabela_eventos$falha[x - 1] + tabela_eventos$censura[x - 1])
}

tabela_eventos |>
  mutate(risco = nj) |>
  knitr::kable()
```

tempo	intervalo	falha	censura	amplitude	risco
1	[1,2)	47	66	1	2453
2	[2,3)	41	47	1	2340
3	[3,4)	32	33	1	2252
4	[4,5)	30	33	1	2187
5	[5,6)	29	37	1	2124
6	[6,7)	22	25	1	2058
7	[7,8)	25	40	1	2011
8	[8,9)	18	39	1	1946
9	[9,10)	32	43	1	1889
10	[10,11)	19	36	1	1814
11	[11,12)	31	35	1	1759
12	[12,13)	27	47	1	1693
13	[13,14)	22	36	1	1619
14	[14,15)	13	27	1	1561
15	[15,16)	25	23	1	1521
16	[16,17)	10	24	1	1473
17	[17,18)	11	38	1	1439

tempo	intervalo	falha	censura	amplitude	risco
18	[18,19)	11	28	1	1390
19	[19,20)	12	24	1	1351
20	[20,21)	15	14	1	1315
21	[21,22)	15	31	1	1286
22	[22,23)	14	31	1	1240
23	[23,24)	11	30	1	1195
24	[24,25)	7	33	1	1154
25	[25,26)	13	28	1	1114
26	[26,27)	7	16	1	1073
27	[27,28)	14	14	1	1050
28	[28,29)	12	17	1	1022
29	[29,30)	10	19	1	993
30	[30,31)	14	9	1	964
31	[31,32)	10	12	1	941
32	[32,33)	11	29	1	919
33	[33,34)	7	18	1	879
34	[34,35)	4	18	1	854
35	[35,36)	8	20	1	832
36	[36,37)	15	25	1	804
37	[37,38)	8	22	1	764
38	[38,39)	11	18	1	734
39	[39,40)	5	13	1	705
40	[40,41)	8	19	1	687
41	[41,42)	2	16	1	660
42	[42,43)	3	34	1	642
43	[43,44)	5	600	1	605

ii. Apresente o cálculo de sobrevivência empírica (pela definição apresentada na aula 1). Apresente também as estimativas empíricas das seguintes quantidades: função densidade, função de risco, função de risco acumulada.

Os estimadores da função empírica consideram o caso em que não há censura nos dados. Os seus estimadores de função de densidade, risco, sobrevivência e risco acumulado são definidos da seguintes forma:

$$\hat{f}(t) = \frac{\text{nº falhas no intervalo começando em t}}{(\text{nº total de indivíduos no estudo})\,(\text{Amplitude do intervalo})}$$

$$\hat{S}(t) = \frac{\text{nº de indivíduos sob risco até o tempo t}}{\text{nº total de indivíduos no estudo}}$$

$$\hat{h}\,(t) = \frac{\text{nº de falha no intervalo iniciado em t}}{\text{nº de indivíduos sob risco em t}\,(\text{Amplitude do intervalo})}$$

Para o cálculo da taxa de risco acumulado, considerou-se a relação entre $H(t)$ e a função de sobrevivência, definida como $H(t) = -\log S(t)$. Além disso, para calcular a variância da função de sobrevivência e seu intervalo de confiança, utilizou-se a mesma expressão do estimador de Kaplan-Meier, que, como será discutido adiante, é uma adaptação da função empírica para o caso em que há censura nos dados de sobrevivência. Assim, definimos a seguinte expressão de variância:

$$\hat{Var}\left(\hat{S}\,(t)\right) = \left[\hat{S}\,(t)\right]^2 \sum_{j:t_j<t} \frac{d_j}{n_j\,(n_j - d_j)} \tag{0.1}$$

Dessa forma, temos a tabela com as estimativas da empírica:

```
# empirica

tabela_empirica <- tabela_eventos |>
  mutate(
    risco = nj,
    `S(t)` = nj / nrow(df),
    `f(t)` = falha / (amplitude * nrow(df)),
     `h(t)` = falha / (risco * amplitude),
     `H(t)` = - log(`S(t)`),
     seS = `S(t)` ^ 2 * cumsum(falha / (risco * (risco - falha))),
     across(`S(t)`:`H(t)`, ~ round(.x, 4)),
     lower = `S(t)` - qnorm(1 - 0.05 / 2) * sqrt(seS),
     upper = `S(t)` + qnorm(1 - 0.05 / 2) * sqrt(seS)
  ) |>
  mutate(upper = ifelse(upper > 1, 1, upper))

tabela_empirica |>
  select(-amplitude, -tempo, -seS) |>
  relocate(lower, .before = `f(t)`) |>
  relocate(upper, .after = lower) |>
  mutate(across(where(is.double), \(x) round(x, 3))) |>
  knitr::kable()
```

Tabela 2: Estimativas empíricas

intervalo	falha	censura	risco	S(t)	lower	upper	f(t)	h(t)	H(t)
[1,2)	47	66	2453	1.000	0.994	1.000	0.019	0.019	0.000
[2,3)	41	47	2340	0.954	0.947	0.961	0.017	0.018	0.047
[3,4)	32	33	2252	0.918	0.910	0.927	0.013	0.014	0.086
[4,5)	30	33	2187	0.892	0.882	0.901	0.012	0.014	0.115
[5,6)	29	37	2124	0.866	0.856	0.876	0.012	0.014	0.144
[6,7)	22	25	2058	0.839	0.829	0.849	0.009	0.011	0.176
[7,8)	25	40	2011	0.820	0.809	0.831	0.010	0.012	0.199
[8,9)	18	39	1946	0.793	0.782	0.804	0.007	0.009	0.232
[9,10)	32	43	1889	0.770	0.758	0.782	0.013	0.017	0.261
[10,11)	19	36	1814	0.740	0.728	0.751	0.008	0.011	0.302
[11,12)	31	35	1759	0.717	0.705	0.729	0.013	0.018	0.333
[12,13)	27	47	1693	0.690	0.678	0.703	0.011	0.016	0.371
[13,14)	22	36	1619	0.660	0.647	0.673	0.009	0.014	0.416
[14,15)	13	27	1561	0.636	0.624	0.649	0.005	0.008	0.452
[15,16)	25	23	1521	0.620	0.607	0.633	0.010	0.016	0.478
[16,17)	10	24	1473	0.601	0.588	0.613	0.004	0.007	0.510
[17,18)	11	38	1439	0.587	0.574	0.599	0.004	0.008	0.533
[18,19)	11	28	1390	0.567	0.554	0.579	0.004	0.008	0.568
[19,20)	12	24	1351	0.551	0.538	0.563	0.005	0.009	0.597
[20,21)	15	14	1315	0.536	0.524	0.549	0.006	0.011	0.624
[21,22)	15	31	1286	0.524	0.512	0.537	0.006	0.012	0.646
[22,23)	14	31	1240	0.505	0.493	0.518	0.006	0.011	0.682
[23,24)	11	30	1195	0.487	0.475	0.500	0.004	0.009	0.719
[24,25)	7	33	1154	0.470	0.458	0.483	0.003	0.006	0.754
[25,26)	13	28	1114	0.454	0.442	0.466	0.005	0.012	0.789
[26,27)	7	16	1073	0.437	0.426	0.449	0.003	0.006	0.827
[27,28)	14	14	1050	0.428	0.416	0.440	0.006	0.013	0.849
[28,29)	12	17	1022	0.417	0.405	0.429	0.005	0.012	0.876
[29,30)	10	19	993	0.405	0.393	0.417	0.004	0.010	0.904
[30,31)	14	9	964	0.393	0.381	0.405	0.006	0.015	0.934
[31,32)	10	12	941	0.384	0.372	0.396	0.004	0.011	0.958
[32,33)	11	29	919	0.375	0.363	0.387	0.004	0.012	0.982
[33,34)	7	18	879	0.358	0.347	0.370	0.003	0.008	1.026
[34,35)	4	18	854	0.348	0.337	0.360	0.002	0.005	1.055
[35,36)	8	20	832	0.339	0.328	0.351	0.003	0.010	1.081
[36,37)	15	25	804	0.328	0.316	0.339	0.006	0.019	1.115
[37,38)	8	22	764	0.312	0.300	0.323	0.003	0.011	1.167

Tabela 2: Estimativas empíricas

intervalo	falha	censura	risco	S(t)	lower	upper	f(t)	h(t)	H(t)
[38,39)	11	18	734	0.299	0.288	0.310	0.004	0.015	1.207
[39,40)	5	13	705	0.287	0.277	0.298	0.002	0.007	1.247
[40,41)	8	19	687	0.280	0.269	0.291	0.003	0.012	1.273
[41,42)	2	16	660	0.269	0.259	0.279	0.001	0.003	1.313
[42,43)	3	34	642	0.262	0.252	0.272	0.001	0.005	1.340
[43,44)	5	600	605	0.247	0.237	0.256	0.002	0.008	1.400

iii. Apresente o cálculo da função de sobrevivência $S(t)$ considerando os seguintes estimadores: Kaplan-Meier, Nelson-Aalen e Tabela de Vida. Para cada versão desses estimadores, apresente também as estimativas das seguintes quantidades: função densidade, função de risco, função de risco acumulada. Interprete os resultados.

O primeiro estimador a ser calculado foi o de tabela de vida, que consiste em dividir o eixo do tempo em um certo número de intervalos. Ele define duas quantidades d_j e n_j:

$$d_j = \text{ nº de falhas no intervalo } [t_{j-1}, t_j) \text{ e}$$

$$n_j = \left[\text{nº sob risco em } t_{j-1}\right] - \left[\frac{1}{2}\text{x nº de censuras em } [t_{j_1}, t_j)\right]$$

A partir dessas duas quantidades é possível obter a estimativa para $S(t)$. Assim, a estimativa é definida da seguinte forma

$$\hat{S}(t) = \prod_{i=1}^{j} \left(1 - \frac{d_{i-1}}{n_{i-1}}\right)$$

O cálculo das funções de risco, densidade e sobrevivência para o estimador de tabela de vida foi realizado com o pacote **discSurv** e a sua função **lifeTable**. Essa função também entrega a variância (com expressão semelhante aquela da Equação 0.1) da função de sobrevivência, o que nos permite fazer o cálculo do intervalo de confiança. Assim, a tabela a seguir contem as quantidades para o estimador de tabela de vida:

```
tabela_de_vida <- lifeTable(
  as.data.frame(df),
  timeColumn = "tempo",
  eventColumn = "status"
  )$Output |>
  as_tibble() |>
  rename(
    risco = n,
    falha = events,
    censura = dropouts,
    `S(t)` = S,
    `h(t)` = hazard,
    `H(t)` = cumHazard
  ) |>
  mutate(
    intervalo = tabela_empirica$intervalo,
```

```
    upper = `S(t)` + qnorm(1 - 0.05/2) * seS,
    lower = `S(t)` - qnorm(1 - 0.05/2) * seS,
    `f(t)` = `h(t)` * `S(t)`, tempo = tabela_empirica$tempo,
    estimador = "Tabela de Vida"
  ) |>
  relocate(tempo, .before = risco) |>
  relocate(lower, .before = `h(t)`) |>
  relocate(upper, .after = lower) |>
  relocate(intervalo, .after = tempo) |>
  relocate(`S(t)`, .before = lower) |>
  select(-c(atRisk, seHazard, seS, seCumHazard, margProb))

tabela_de_vida |>
  select(-tempo, -estimador) |>
  mutate(across(where(is.double), \(x) round(x, 3))) |>
  knitr::kable()
```

Tabela 3: Estimativas tabela de vida

intervalo	risco	falha	censura	S(t)	lower	upper	h(t)	H(t)	f(t)
[1,2)	2453	47	66	0.981	0.975	0.986	0.019	0.019	0.019
[2,3)	2340	41	47	0.963	0.956	0.971	0.018	0.037	0.017
[3,4)	2252	32	33	0.949	0.941	0.958	0.014	0.051	0.014
[4,5)	2187	30	33	0.936	0.926	0.946	0.014	0.065	0.013
[5,6)	2124	29	37	0.923	0.913	0.934	0.014	0.079	0.013
[6,7)	2058	22	25	0.913	0.902	0.925	0.011	0.090	0.010
[7,8)	2011	25	40	0.902	0.890	0.914	0.013	0.102	0.011
[8,9)	1946	18	39	0.894	0.881	0.906	0.009	0.112	0.008
[9,10)	1889	32	43	0.878	0.865	0.892	0.017	0.129	0.015
[10,11)	1814	19	36	0.869	0.855	0.883	0.011	0.139	0.009
[11,12)	1759	31	35	0.854	0.839	0.868	0.018	0.157	0.015
[12,13)	1693	27	47	0.840	0.824	0.855	0.016	0.173	0.014
[13,14)	1619	22	36	0.828	0.812	0.844	0.014	0.187	0.011
[14,15)	1561	13	27	0.821	0.805	0.837	0.008	0.196	0.007
[15,16)	1521	25	23	0.808	0.791	0.824	0.017	0.212	0.013
[16,17)	1473	10	24	0.802	0.785	0.819	0.007	0.219	0.005
[17,18)	1439	11	38	0.796	0.779	0.813	0.008	0.227	0.006
[18,19)	1390	11	28	0.790	0.772	0.807	0.008	0.235	0.006
[19,20)	1351	12	24	0.782	0.765	0.800	0.009	0.244	0.007
[20,21)	1315	15	14	0.773	0.755	0.792	0.011	0.255	0.009

intervalo	risco	falha	censura	S(t)	lower	upper	h(t)	H(t)	f(t)
[21,22)	1286	15	31	0.764	0.746	0.783	0.012	0.267	0.009
[22,23)	1240	14	31	0.756	0.737	0.775	0.011	0.278	0.009
[23,24)	1195	11	30	0.749	0.729	0.768	0.009	0.288	0.007
[24,25)	1154	7	33	0.744	0.725	0.763	0.006	0.294	0.005
[25,26)	1114	13	28	0.735	0.715	0.755	0.012	0.306	0.009
[26,27)	1073	7	16	0.730	0.710	0.750	0.007	0.312	0.005
[27,28)	1050	14	14	0.721	0.700	0.741	0.013	0.326	0.010
[28,29)	1022	12	17	0.712	0.691	0.733	0.012	0.337	0.008
[29,30)	993	10	19	0.705	0.684	0.726	0.010	0.348	0.007
[30,31)	964	14	9	0.694	0.673	0.716	0.015	0.362	0.010
[31,32)	941	10	12	0.687	0.665	0.709	0.011	0.373	0.007
[32,33)	919	11	29	0.679	0.657	0.701	0.012	0.385	0.008
[33,34)	879	7	18	0.673	0.651	0.695	0.008	0.393	0.005
[34,35)	854	4	18	0.670	0.648	0.692	0.005	0.398	0.003
[35,36)	832	8	20	0.664	0.641	0.686	0.010	0.408	0.006
[36,37)	804	15	25	0.651	0.628	0.674	0.019	0.427	0.012
[37,38)	764	8	22	0.644	0.621	0.667	0.011	0.437	0.007
[38,39)	734	11	18	0.634	0.611	0.658	0.015	0.452	0.010
[39,40)	705	5	13	0.630	0.606	0.653	0.007	0.459	0.005
[40,41)	687	8	19	0.622	0.598	0.646	0.012	0.471	0.007
[41,42)	660	2	16	0.620	0.596	0.644	0.003	0.474	0.002
[42,43)	642	3	34	0.617	0.593	0.642	0.005	0.479	0.003
[43,44)	605	5	600	0.607	0.582	0.633	0.016	0.496	0.010

As estimativas de Kaplan-Meier podem ser obtidas de forma semelhantes àquelas da tabela de vida. Da mesma forma, defini-se duas quantidades n_j e d_j. n_j é o número de indivíduos sob risco em t_j, ou seja, os indivíduos que não falharam e não foram censurados até o instante imediatamente anterior a t_j. Assim, o estimador de Kaplan-Meier é definido como:

$$\hat{S}(t) = \prod_{j:t_j<t} \left(1 - \frac{d_j}{n_j}\right)$$

A variância da sobrevivência é a mesma mostrada anteriormente na Equação 0.1. A taxa de risco é calculada como:

$$\hat{h}_{KM} = \frac{1}{h}\left(1 - \frac{n_j - d_j}{n_j}\right)$$

em que h é a amplitude do intervalo. Não obstante, para a obtenção da densidade, foi

simplesmente utilizado a expressão $f(t) = S(t)\,h(t)$. Assim, temos a seguinte tabela com as estimativas do estimador de Kaplan-Meier:

```
# kaplan meier
ekm <- survfit(
  Surv(df$tempo, df$status) ~ 1,
  conf.type = "plain"
)

tabela_ekm <- ekm |>
  summary() |>
  (\(x) {
    tibble(
      tempo = x$time,
      risco = x$n.risk,
      falha = x$n.event,
      censura = x$n.censor,
      `S(t)` = x$surv,
      upper = x$upper,
      lower = x$lower,
      `H(t)` = x$cumhaz,
      amplitude = tabela_eventos$amplitude
    ) |>
    mutate(
      intervalo = tabela_empirica$intervalo,
      `h(t)` = (1 - (risco - falha) / risco) / amplitude,
      `f(t)` = `h(t)` * `S(t)`,
      estimador = "Kaplan-Meier"
    ) |>
    select(-amplitude) |>
    relocate(intervalo, .after = tempo)
  })()

tabela_ekm |>
  select(-tempo, -estimador) |>
  mutate(across(where(is.double), \(x) round(x, 3))) |>
  relocate(intervalo, .before = risco) |>
  knitr::kable()
```

Tabela 4: Etimativas Kaplan-Meier

intervalo	risco	falha	censura	S(t)	upper	lower	H(t)	h(t)	f(t)
[1,2)	2453	47	66	0.981	0.986	0.975	0.019	0.019	0.019
[2,3)	2340	41	47	0.964	0.971	0.956	0.037	0.018	0.017
[3,4)	2252	32	33	0.950	0.959	0.941	0.051	0.014	0.013
[4,5)	2187	30	33	0.937	0.947	0.927	0.065	0.014	0.013
[5,6)	2124	29	37	0.924	0.935	0.913	0.078	0.014	0.013
[6,7)	2058	22	25	0.914	0.926	0.903	0.089	0.011	0.010
[7,8)	2011	25	40	0.903	0.915	0.891	0.101	0.012	0.011
[8,9)	1946	18	39	0.895	0.907	0.882	0.111	0.009	0.008
[9,10)	1889	32	43	0.879	0.893	0.866	0.128	0.017	0.015
[10,11)	1814	19	36	0.870	0.884	0.856	0.138	0.010	0.009
[11,12)	1759	31	35	0.855	0.869	0.840	0.156	0.018	0.015
[12,13)	1693	27	47	0.841	0.856	0.826	0.172	0.016	0.013
[13,14)	1619	22	36	0.830	0.846	0.814	0.185	0.014	0.011
[14,15)	1561	13	27	0.823	0.839	0.807	0.194	0.008	0.007
[15,16)	1521	25	23	0.809	0.826	0.793	0.210	0.016	0.013
[16,17)	1473	10	24	0.804	0.821	0.787	0.217	0.007	0.005
[17,18)	1439	11	38	0.798	0.815	0.781	0.224	0.008	0.006
[18,19)	1390	11	28	0.791	0.809	0.774	0.232	0.008	0.006
[19,20)	1351	12	24	0.784	0.802	0.767	0.241	0.009	0.007
[20,21)	1315	15	14	0.775	0.794	0.757	0.253	0.011	0.009
[21,22)	1286	15	31	0.766	0.785	0.748	0.264	0.012	0.009
[22,23)	1240	14	31	0.758	0.777	0.739	0.276	0.011	0.009
[23,24)	1195	11	30	0.751	0.770	0.732	0.285	0.009	0.007
[24,25)	1154	7	33	0.746	0.765	0.727	0.291	0.006	0.005
[25,26)	1114	13	28	0.737	0.757	0.718	0.303	0.012	0.009
[26,27)	1073	7	16	0.733	0.752	0.713	0.309	0.007	0.005
[27,28)	1050	14	14	0.723	0.743	0.703	0.322	0.013	0.010
[28,29)	1022	12	17	0.714	0.735	0.694	0.334	0.012	0.008
[29,30)	993	10	19	0.707	0.728	0.686	0.344	0.010	0.007
[30,31)	964	14	9	0.697	0.718	0.676	0.359	0.015	0.010
[31,32)	941	10	12	0.690	0.711	0.668	0.369	0.011	0.007
[32,33)	919	11	29	0.681	0.703	0.660	0.381	0.012	0.008
[33,34)	879	7	18	0.676	0.698	0.654	0.389	0.008	0.005
[34,35)	854	4	18	0.673	0.695	0.651	0.394	0.005	0.003
[35,36)	832	8	20	0.666	0.689	0.644	0.404	0.010	0.006
[36,37)	804	15	25	0.654	0.677	0.631	0.422	0.019	0.012
[37,38)	764	8	22	0.647	0.670	0.624	0.433	0.010	0.007

intervalo	risco	falha	censura	S(t)	upper	lower	H(t)	h(t)	f(t)
[38,39)	734	11	18	0.637	0.661	0.614	0.448	0.015	0.010
[39,40)	705	5	13	0.633	0.656	0.609	0.455	0.007	0.004
[40,41)	687	8	19	0.625	0.649	0.602	0.466	0.012	0.007
[41,42)	660	2	16	0.623	0.647	0.600	0.469	0.003	0.002
[42,43)	642	3	34	0.621	0.645	0.597	0.474	0.005	0.003
[43,44)	605	5	600	0.615	0.640	0.591	0.482	0.008	0.005

O Estimador de Nelson-Aalen se baseia na função de sobrevivência expressa por:

$$S\left(t\right) = exp\{-H\left(t\right)\}$$

O estimador de Nelson-Aalen estima a função acumulada da taxa de risco para só depois estimar a função de sobrevivência. Portanto, temos o seguinte estimador de Nelson-Aalen:

$$\tilde{H}\left(t\right) = \sum_{j:t_j<t} \left(\frac{d_j}{n_j}\right)$$

em que n_j e d_j são definidos como no estimador de Kaplan-Meier. A variância da sobrevivência pode ser obtido substituindo $\tilde{S}\left(t\right)$ em $\hat{S}\left(t\right)$ da variância de Kaplan-Meier:

$$\hat{Var}\left(\tilde{S}\left(t\right)\right) = \left[\tilde{S}\left(t\right)\right]^2 \sum_{j:t_j<t} \left(\frac{d_j}{n_j^2}\right)$$

Para obter a estimativa para obter a estimativa da taxa de risco $h\left(t\right)$ foi utilizada a função **diff** do R para obter as diferenças sucessivas. Com as estimativas da função da taxa de risco obtidas, agora é possível estimar a densidade utilizando as relações entre as funções. Assim, temos as estimativas na tabela a seguir:

```
# nelson alen

alen <- survfit(
  Surv(tempo, status) ~ 1,
  type = "fleming-harrington",
  data = df,
  conf.type = "plain"
)

tabela_alen <- alen |>
  summary() |>
```

```
  (\(x) {
    tab <- tibble(
      tempo = x$time,
      risco = x$n.risk,
      falha = x$n.event,
      censura = x$n.censor,
      `S(t)` = x$surv,
      lower = x$lower,
      upper = x$upper,
      `H(t)` = x$cumhaz,
      amplitude = tabela_eventos$amplitude
    )
    ht <- c(tab$`H(t)`[1], diff(tab$`H(t)`))

    tab |>
      mutate(
        `h(t)` = c(`H(t)`[1], diff(`H(t)`)),
        `f(t)` = `h(t)` * `S(t)`,
        intervalo = tabela_ekm$intervalo,
        estimador = "Aalen"
      ) |>
      select(-amplitude) |>
      relocate(intervalo, .before = risco)
  })()

tabela_alen |>
  select(-tempo, -estimador) |>
  mutate(across(where(is.double), \(x) round(x, 3))) |>
  knitr::kable()
```

Tabela 5: Estimativas Nelson-Aalen

intervalo	risco	falha	censura	S(t)	lower	upper	H(t)	h(t)	f(t)
[1,2)	2453	47	66	0.981	0.976	0.986	0.019	0.019	0.019
[2,3)	2340	41	47	0.964	0.957	0.971	0.037	0.018	0.017
[3,4)	2252	32	33	0.950	0.942	0.959	0.051	0.014	0.014
[4,5)	2187	30	33	0.937	0.928	0.947	0.065	0.014	0.013
[5,6)	2124	29	37	0.925	0.914	0.935	0.078	0.014	0.013
[6,7)	2058	22	25	0.915	0.904	0.926	0.089	0.011	0.010
[7,8)	2011	25	40	0.904	0.892	0.916	0.101	0.012	0.011

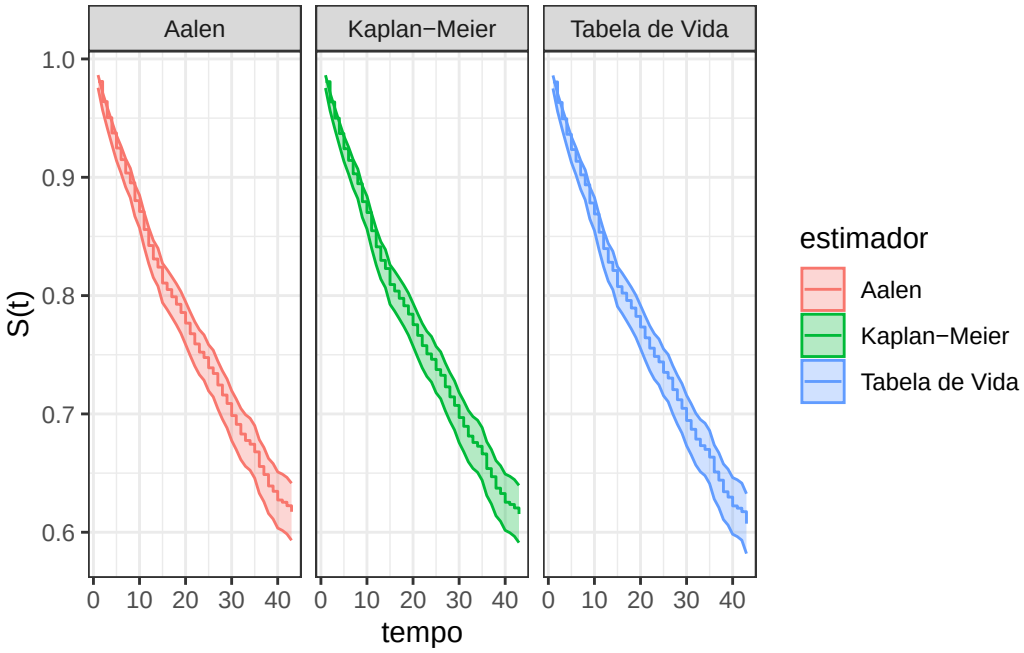
intervalo	risco	falha	censura	S(t)	lower	upper	H(t)	h(t)	f(t)
[8,9)	1946	18	39	0.895	0.883	0.908	0.111	0.009	0.008
[9,10)	1889	32	43	0.880	0.867	0.894	0.128	0.017	0.015
[10,11)	1814	19	36	0.871	0.857	0.885	0.138	0.010	0.009
[11,12)	1759	31	35	0.856	0.841	0.870	0.156	0.018	0.015
[12,13)	1693	27	47	0.842	0.827	0.857	0.172	0.016	0.013
[13,14)	1619	22	36	0.831	0.815	0.847	0.185	0.014	0.011
[14,15)	1561	13	27	0.824	0.808	0.840	0.194	0.008	0.007
[15,16)	1521	25	23	0.811	0.794	0.827	0.210	0.016	0.013
[16,17)	1473	10	24	0.805	0.788	0.822	0.217	0.007	0.005
[17,18)	1439	11	38	0.799	0.782	0.816	0.224	0.008	0.006
[18,19)	1390	11	28	0.793	0.775	0.810	0.232	0.008	0.006
[19,20)	1351	12	24	0.786	0.768	0.803	0.241	0.009	0.007
[20,21)	1315	15	14	0.777	0.759	0.795	0.253	0.011	0.009
[21,22)	1286	15	31	0.768	0.749	0.786	0.264	0.012	0.009
[22,23)	1240	14	31	0.759	0.740	0.778	0.276	0.011	0.009
[23,24)	1195	11	30	0.752	0.733	0.771	0.285	0.009	0.007
[24,25)	1154	7	33	0.748	0.728	0.767	0.291	0.006	0.005
[25,26)	1114	13	28	0.739	0.719	0.758	0.303	0.012	0.009
[26,27)	1073	7	16	0.734	0.714	0.754	0.309	0.007	0.005
[27,28)	1050	14	14	0.724	0.704	0.745	0.322	0.013	0.010
[28,29)	1022	12	17	0.716	0.696	0.736	0.334	0.012	0.008
[29,30)	993	10	19	0.709	0.688	0.730	0.344	0.010	0.007
[30,31)	964	14	9	0.699	0.678	0.720	0.359	0.015	0.010
[31,32)	941	10	12	0.691	0.670	0.713	0.369	0.011	0.007
[32,33)	919	11	29	0.683	0.661	0.705	0.381	0.012	0.008
[33,34)	879	7	18	0.678	0.656	0.699	0.389	0.008	0.005
[34,35)	854	4	18	0.674	0.652	0.696	0.394	0.005	0.003
[35,36)	832	8	20	0.668	0.646	0.690	0.404	0.010	0.006
[36,37)	804	15	25	0.656	0.633	0.678	0.422	0.019	0.012
[37,38)	764	8	22	0.649	0.626	0.672	0.433	0.010	0.007
[38,39)	734	11	18	0.639	0.616	0.662	0.448	0.015	0.010
[39,40)	705	5	13	0.635	0.611	0.658	0.455	0.007	0.005
[40,41)	687	8	19	0.627	0.604	0.651	0.466	0.012	0.007
[41,42)	660	2	16	0.625	0.602	0.649	0.469	0.003	0.002
[42,43)	642	3	34	0.622	0.599	0.646	0.474	0.005	0.003
[43,44)	605	5	600	0.617	0.593	0.641	0.482	0.008	0.005

Observando o gráfico abaixo das curvas dos três estimadores, vemos que eles não apresentam uma diferença muito grande na estimação da função de sobrevivência. A

única que subestima um pouco a sobrevivência é a tabela de vida, pois considera também a quantidade de censura presente em cada intervalo.

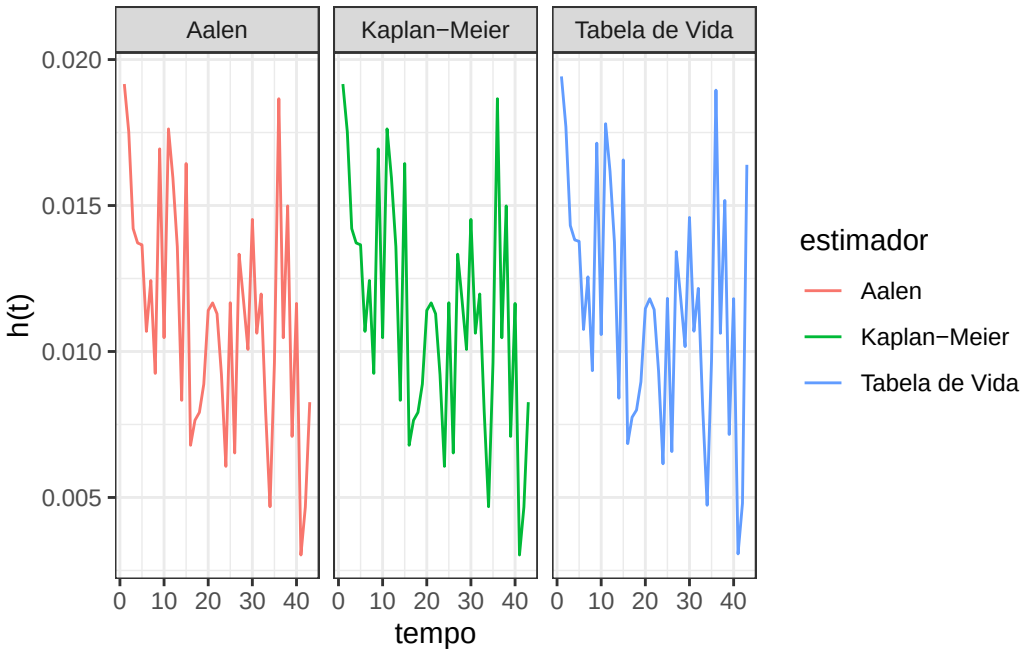
```
juntos <- select(tabela_alen, - intervalo) |>
  rbind(tabela_ekm |> select(-intervalo)) |>
  rbind(tabela_de_vida |> select(-intervalo))

juntos |>
  ggplot(aes(x = tempo, y = `S(t)`, color = estimador)) +
  geom_step() +
  geom_ribbon(
    aes(ymin = lower, ymax = upper, fill = estimador), alpha = 0.3
  ) +
  facet_wrap(vars(estimador)) +
  theme_bw()
```



Com base na função de taxa de falha, observamos que as estimativas dos diferentes estimadores são bastante semelhantes. Além disso, nota-se que a taxa de falha diminui de forma quase contínua ao longo do estudo, aumentando novamente apenas no final.

```
juntos |>
  ggplot(aes(x = tempo, y = `h(t)`, color = estimador)) +
  geom_line() +
  facet_wrap(vars(estimador)) +
  theme_bw()
```



iv. Explique como o teste de LogRank deve ser aplicado. Escolha uma variável qualitativa de sua base e realize o teste de comparação de curvas de sobrevivência. Interprete adequadamente os resultados.

O teste de LogRank serve para comparação de grupos. A hipótese nula é se as curvas de sobrevivência são iguais ($H_0 : S_1(t) = S_2(t)$). Para isso, é utilizada a seguinte estatística de teste

$$T = \frac{\left[\sum_{j=1}^{k} \left(d_{2j} - w_{2j}\right)\right]}{\sum_{j=1}^{k} \left(V_j\right)_2}$$

que para grandes amostras segue uma distribuição qui-quadrado com 1 grau de liberdade. Aqui d representa a quantidade de falhas, w é a média das falhas do grupo e V é a variância da falha do grupo.

Para aplicar o teste nos dados de sobrevivência foi considerada a variável da doença de cada paciente. Assim, foram comparados aqueles pacientes com hipertensão, diabetes, renal, congênita ou outras doenças. O teste foi feito comparando os grupos 2 a 2, gerando a seguinte tabela:

```
grupos <- list("outr", "hiper", "diab", "rim", "cong")
results <- list()
results_tibble <- tibble::tibble(
  comparação = character(),
```

```
  estatística = numeric(),
  `p-valor` = numeric()
)

for (i in 1:(length(grupos) - 1)) {
  for (j in (i + 1):length(grupos)) {
    grupo1 <- grupos[[i]]
    grupo2 <- grupos[[j]]
    data_subset <- df[df$doenca %in% c(grupo1, grupo2),]

    test_result <- survdiff(
      Surv(data_subset$tempo, data_subset$status) ~ data_subset$doenca
    )
    statistic <- test_result$chisq
    pvalue <- test_result$pvalue
    result_name <- paste(grupo1, grupo2, sep = " vs ")

    results_tibble <- results_tibble |>
      add_row(
        comparação = result_name,
        estatística = statistic,
        `p-valor` = pvalue
      )
  }
}
results_tibble |>
  arrange(`p-valor`) |>
  knitr::kable()
```

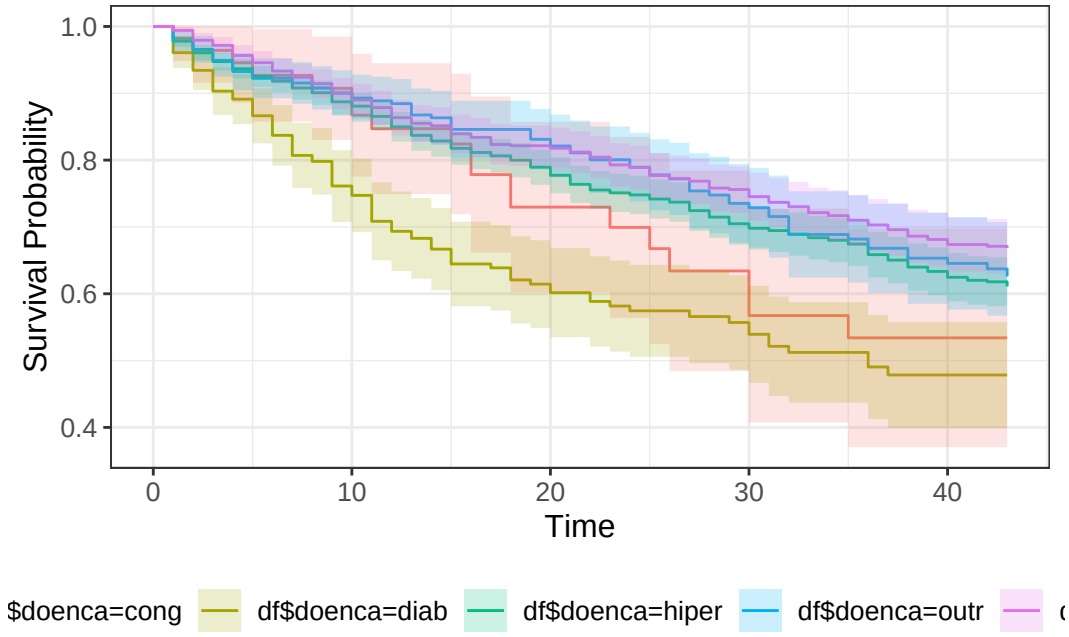
comparação	estatística	p-valor
diab vs rim	39.1219659	0.0000000
hiper vs diab	26.9880266	0.0000002
outr vs diab	20.9698863	0.0000047
hiper vs rim	4.2081902	0.0402292
rim vs cong	3.1526686	0.0758029
diab vs cong	1.9015330	0.1679068
outr vs cong	1.8960397	0.1685223
hiper vs cong	1.0446274	0.3067477
outr vs hiper	0.6326518	0.4263843

comparação	estatística	p-valor
outr vs rim	0.5012909	0.4789335

Pela tabela, podemos ver que os únicos grupos diferentes são os pacientes hipertensão e doença renal, diabetes e outras doenças, hipertensão e diabetes, e diabetes e renal.

De fato, observando os grupos pelo gráfico abaixo, podemos ver claramente os resultados do teste. Além disso, é possível ver claramente que a sobrevivência do grupo com diabetes é aquela que mais descresce rapidamente, e as pessoas com doença congênitas é a segunda que mais decresce rapidamente. Como foi visto anteriormente na análise exploratória, os pacientes com diabetes e doenças congênitas são aqueles que mais demoram para iniciar o tratamento de diálise, o que pode ser uma das causas dessa diminuição acelerada. No entanto, o grupo de congênitos tem muitas observações censuradas, o que é preciso ter cuidado ao se analisar sua curva.

```
ekm_gp <- survfit(
  Surv(df$tempo, df$status) ~ df$doenca, conf.type = "plain"
)
ekm_gp |>
 ggsurvfit() +
 add_confidence_interval()
```



v. Apresente o cálculo da função de sobrevivência $S(t)$ considerando as seguintes distribuições de probabilidade: Exponencial, Weibull, Gama, Log-Normal, Gama Generalizada e as duas distribuições da questão 1 que você identificou. Apresente os valores do AIC e BIC apenas para os ajustes baseados nas distribuições Exponencial, Weibull, Gama, Log-Normal e gama Generalizada. Como você pode comparar as estimativas geradas por essas distribuições a partir do teste de razão de verossimilhanças? Interprete os resultados.

```
expo_fit <- flexsurvreg(
  Surv(df$tempo, df$status) ~ 1,
  dist = "exponential")

weib_fit <- flexsurvreg(
  Surv(df$tempo, df$status) ~ 1,
  dist = "weibull")

gengamma_fit <- flexsurvreg(
  Surv(df$tempo, df$status) ~ 1,
  dist = "gengamma")

lognormal_fit <- flexsurvreg(
  Surv(df$tempo, df$status) ~ 1,
  dist = "lognormal")

gamma_fit <- flexsurvreg(
  Surv(df$tempo, df$status) ~ 1,
  dist = "gamma")

dists <- list(
  expo_fit, weib_fit,
  gengamma_fit, lognormal_fit,
  gamma_fit)

alltho <- lapply(dists, \(x) {
  name <- x$call$dist
  x |>
  summary() |>
  (\(y) {
    y <- y[[1]]
    tibble(
```

```
      tempo = y$time,
      `S(t)` = y$est,
      lower = y$lcl,
      upper = y$ucl,
      estimador = name
    )
    })()
  }) |>
  Reduce(f = rbind)
```

Pelo teste de razão de verossimilhanças, vemos que a log-normal, ao nível de 5% de significância, apresenta um bom ajuste nos dados de sobrevivência.

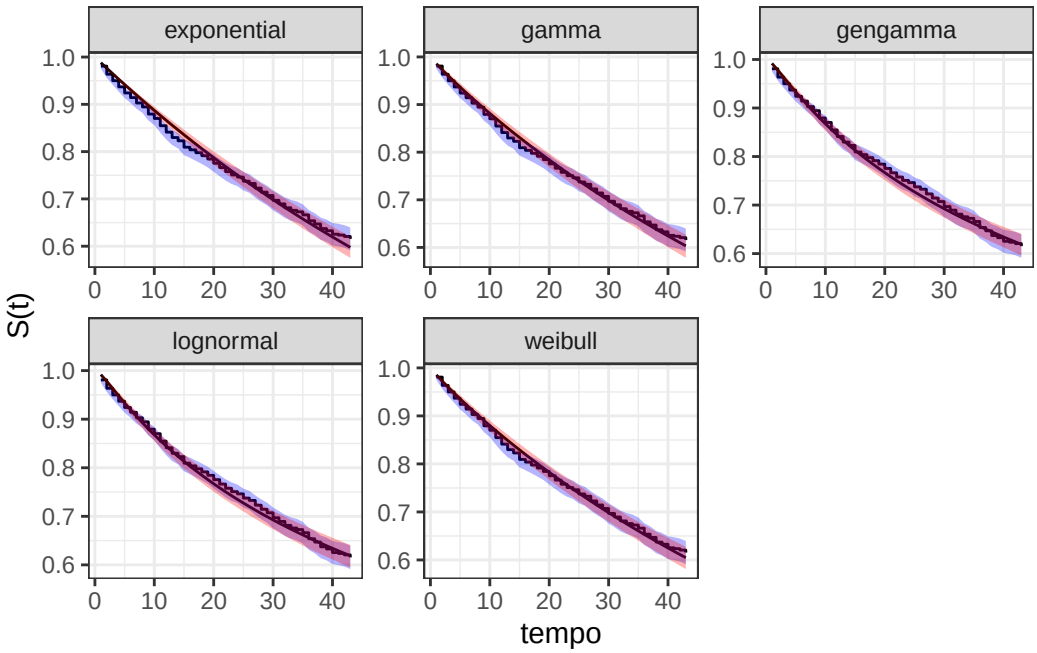
No entanto, pelo gráfico, as outras distribuições também parecem se ajustar bem aos dados de sobrevivência. Não obstante, a distribuição log-normal foi aquela também que apresentou o menor AIC e BIC.

```
tibble(
  Modelo = c(
    "Gama Generalizado", "Exponencial",
    "Weibull", "Log-Normal", "Gama"
    ),
  logvero = c(
    logLik(gengamma_fit), logLik(expo_fit),
    logLik(weib_fit), logLik(lognormal_fit),
    logLik(gamma_fit)
    ),
  AIC = c(
    AIC(gengamma_fit), AIC(expo_fit),
    AIC(weib_fit), AIC(lognormal_fit),
    AIC(gamma_fit)
    ),
  BIC = c(
    BIC(gengamma_fit), BIC(expo_fit),
    BIC(weib_fit), BIC(lognormal_fit),
    BIC(gamma_fit)
    )
  ) |>
  mutate(
    TRV = 2 * (logvero[1] - logvero[1:5]),
    `p-valor` = 1 - pchisq(TRV[1:5], df = c(0, 2, 1, 1, 1))
```

```
) |>
knitr::kable()
```

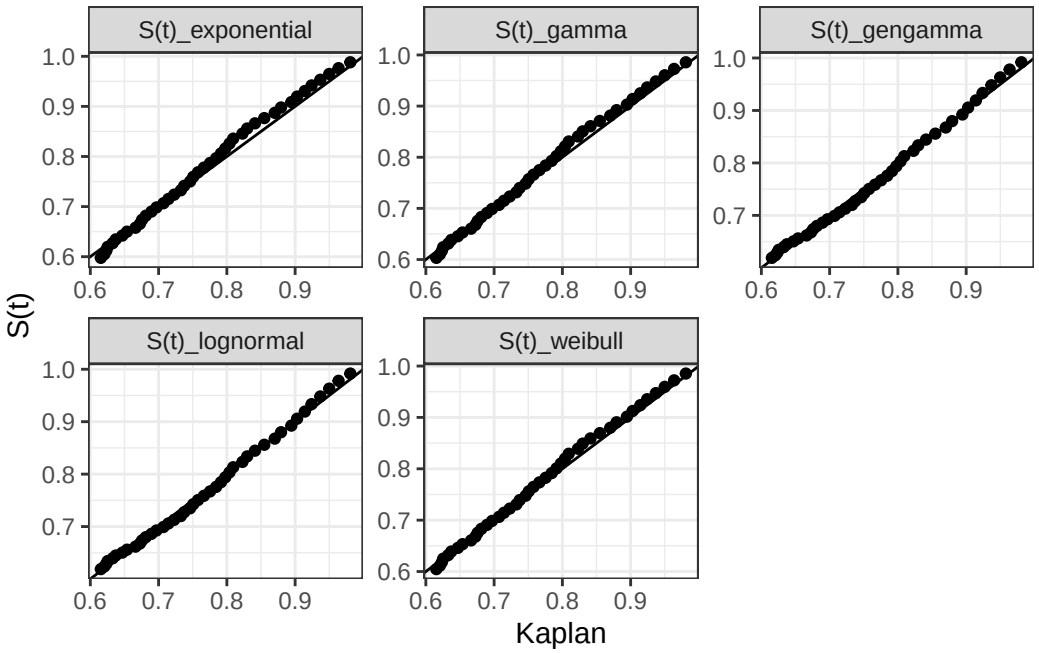
Modelo	logvero	AIC	BIC	TRV	p-valor
Gama Generalizado	-3708.408	7422.817	7440.232	0.0000000	1.0000000
Exponencial	-3721.816	7445.632	7451.437	26.8150245	0.0000015
Weibull	-3719.989	7443.979	7455.589	23.1616870	0.0000015
Log-Normal	-3708.411	7420.822	7432.432	0.0052486	0.9422462
Gama	-3720.558	7445.115	7456.725	24.2983908	0.0000008

```
alltho |>
  ggplot(aes(x = tempo, y = `S(t)`)) +
  geom_line() +
  geom_step(
    data = select(tabela_ekm, -estimador),
    aes(x = tempo, y = `S(t)`)
    ) +
  geom_ribbon(
    data = select(tabela_ekm, -estimador),
    aes(ymin = lower, ymax = upper, fill = "Kaplan-Meier"),
    alpha = 0.3, fill = "blue"
    ) +
  geom_ribbon(
    data = alltho,
    aes(ymin = lower, ymax = upper, fill = "Estimador"),
    alpha = 0.3, fill = "red"
    ) +
  facet_wrap(vars(estimador), scales = "free") +
  theme_bw()
```



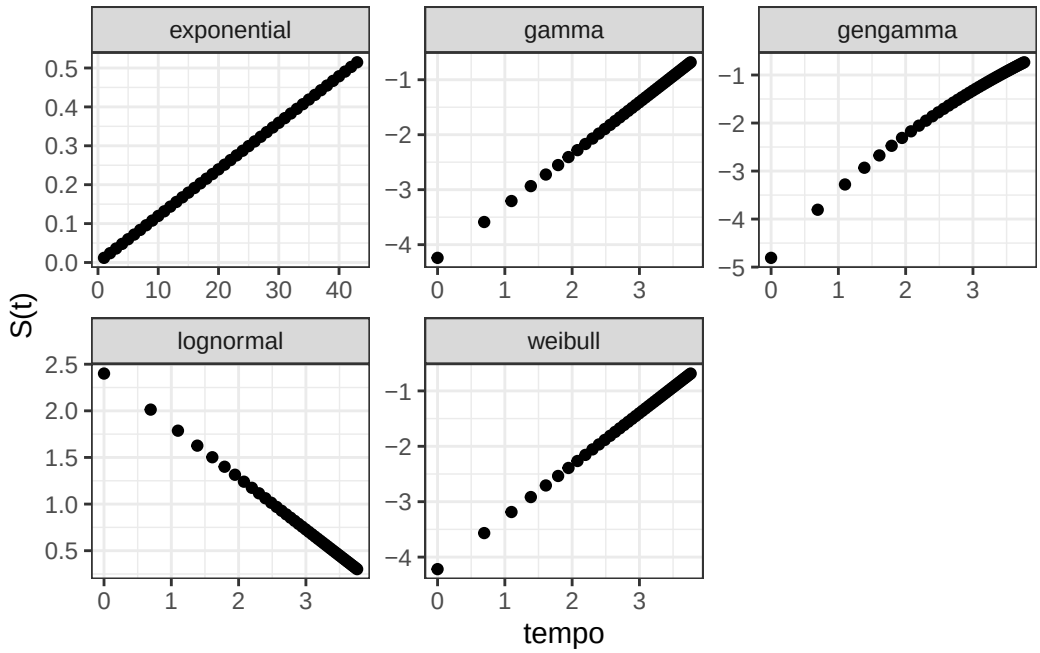
O scatterplot abaixo nos ajuda a ver melhor a diferença entre o ajuste das curvas de cada distribuição. Abaixo vemos claramente que a lognormal é aquela que os pontos estão mais próximos da reta. Portanto, como a log normal é aquela que menos demostra distanciamentos marcantes da reta, ela é a que melhor se ajusta aos dados de sobrevivência.

```
alltho |>
  pivot_wider(
    names_from = estimador,
    values_from = -tempo
  ) |>
  mutate(Kaplan = tabela_ekm$`S(t)`) |>
  pivot_longer(
    cols = starts_with("S(t)"),
    names_to = "Estimador",
    values_to = "S(t)"
  ) |>
  ggplot(aes(x = Kaplan, y = `S(t)`)) +
  geom_point() +
  geom_abline() +
  facet_wrap(vars(Estimador), scales = "free") +
  theme_bw()
```



Pela método de linearização, conseguimos também análisar qual consegue se ajustar melhor aos dados. No entanto, o que consegue mostrar as diferenças foi o segundo método, que de fato é possível ver que a que menos tem desvio da reta é a lognormal.

```
alltho |>
  mutate(
    `S(t)` = case_when(
      estimador == "exponential" ~ -log(`S(t)`),
      estimador == "weibull" ~ log(-log(`S(t)`)),
      estimador == "lognormal" ~ qnorm(`S(t)`),
      estimador == "gamma" ~ log(-log(`S(t)`)),
      estimador == "gengamma" ~ log(-log(`S(t)`)),
      .default = `S(t)`
    ),
    tempo = ifelse(estimador == "exponential", tempo, log(tempo))
  ) |>
  ggplot(aes(x = tempo, y = `S(t)`)) +
  geom_point() +
  facet_wrap(vars(estimador), scales = "free") +
  theme_bw()
```



Abaixo tem as estimativas das funções de sobrevivência da questão 1.

```
st1 <- \(t) exp(-t/5)
st2 <- \(t) 1 / (1 + t)

tibble(
  tempo = 1:43,
  `S1(t)` = st1(tempo),
  `S2(t)` = st2(tempo)
) |>
  knitr::kable()
```

tempo	S1(t)	S2(t)
1	0.8187308	0.5000000
2	0.6703200	0.3333333
3	0.5488116	0.2500000
4	0.4493290	0.2000000
5	0.3678794	0.1666667
6	0.3011942	0.1428571
7	0.2465970	0.1250000
8	0.2018965	0.1111111
9	0.1652989	0.1000000

tempo	S1(t)	S2(t)
10	0.1353353	0.0909091
11	0.1108032	0.0833333
12	0.0907180	0.0769231
13	0.0742736	0.0714286
14	0.0608101	0.0666667
15	0.0497871	0.0625000
16	0.0407622	0.0588235
17	0.0333733	0.0555556
18	0.0273237	0.0526316
19	0.0223708	0.0500000
20	0.0183156	0.0476190
21	0.0149956	0.0454545
22	0.0122773	0.0434783
23	0.0100518	0.0416667
24	0.0082297	0.0400000
25	0.0067379	0.0384615
26	0.0055166	0.0370370
27	0.0045166	0.0357143
28	0.0036979	0.0344828
29	0.0030276	0.0333333
30	0.0024788	0.0322581
31	0.0020294	0.0312500
32	0.0016616	0.0303030
33	0.0013604	0.0294118
34	0.0011138	0.0285714
35	0.0009119	0.0277778
36	0.0007466	0.0270270
37	0.0006113	0.0263158
38	0.0005005	0.0256410
39	0.0004097	0.0250000
40	0.0003355	0.0243902
41	0.0002747	0.0238095
42	0.0002249	0.0232558
43	0.0001841	0.0227273

viii Considerando os resultados, o que é possível concluir sobre a sobrevivência dos pacientes na sua base de dados.

Analisando a base de dados de diálise, constatou-se que os pacientes que iniciam o

tratamento com antecedência apresentam uma função de sobrevivência que decresce mais lentamente. Em contrapartida, aqueles que iniciam o tratamento tardiamente têm uma probabilidade de sobrevivência que diminui em um período mais curto. Além disso, pacientes com doenças congênitas e diabetes tendem a falecer mais rapidamente

Foi possível ver também que a função de sobrevivência que melhor se ajustou aos dados foi a log-normal. Além disso, uma das maiores causas de óbito é de causa renal.

Os itens vi e vii foram omitidos pois já foram sendo feitos durante a solução dos outros exercícios.