Lista 2 - Análise de Sobrevivência

Augusto Cesar Nunes - 13/0103004

Contents

Exercício 1

Item (a)

$P(T > 1) = 1 - P(T < 1) = 1 - \int_0^1 2t * exp\{-t^2\}dt = 1 - (1 - exp\{-1\}) \approx 0,3679$

Item (b)

$h(t) = \frac{f(t)}{S(t)} = \frac{2t*exp\{-t^2\}}{exp\{-t^2\}} \Rightarrow \lim_{t\to\infty} h(t) = \infty$

Exercício 2

Item (a)

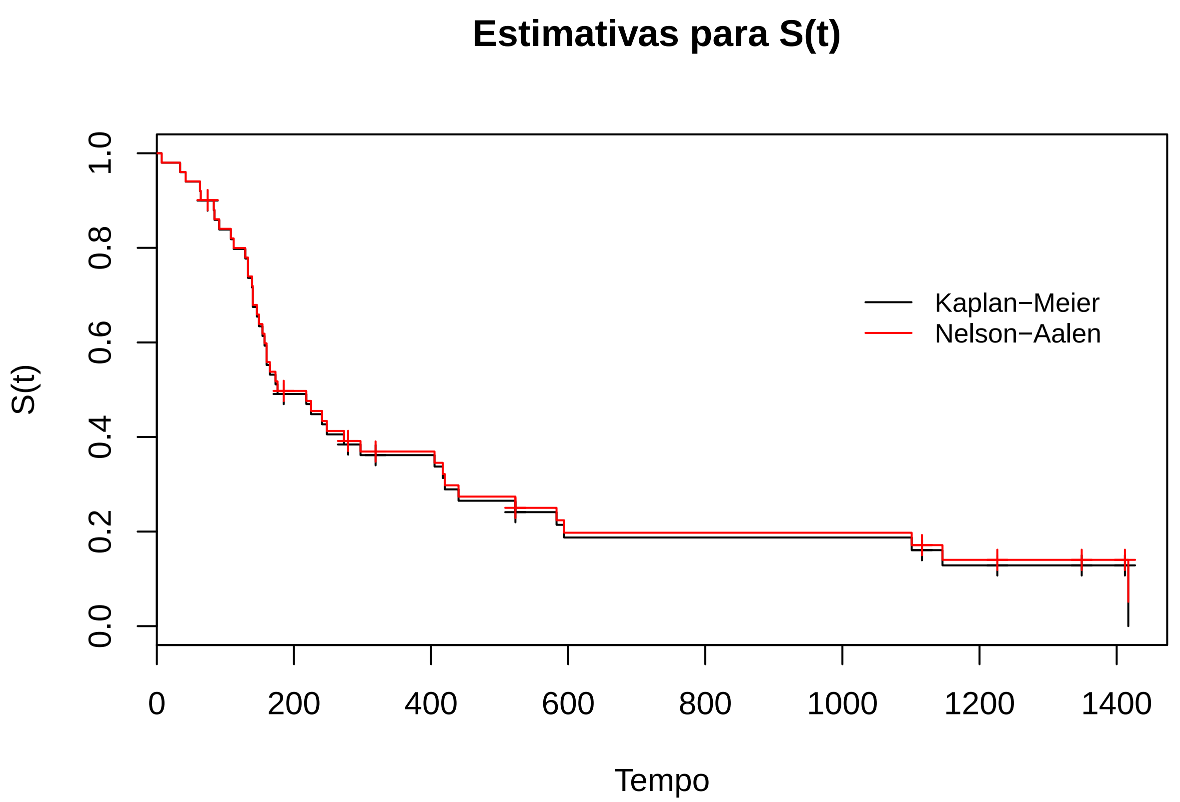
```
## Loading required package: survival

## Call: survfit(formula = Surv(time = tempos.ex2, event = censuras.ex2) ~
##     1, conf.int = F)
##
##  time n.risk n.event survival std.err
##     7     50       1    0.980  0.0198
##    34     49       1    0.960  0.0277
##    42     48       1    0.940  0.0336
##    63     47       1    0.920  0.0384
##    64     46       1    0.900  0.0424
##    83     44       1    0.880  0.0461
##    84     43       1    0.859  0.0494
##    91     42       1    0.839  0.0523
##   108     41       1    0.818  0.0549
##   112     40       1    0.798  0.0572
##   129     39       1    0.777  0.0593
##   133     38       2    0.736  0.0628
##   139     36       1    0.716  0.0643
##   140     35       2    0.675  0.0668
##   146     33       1    0.655  0.0678
##   149     32       1    0.634  0.0687
##   154     31       1    0.614  0.0695
##   157     30       1    0.593  0.0701
##   160     29       2    0.552  0.0710
##   165     27       1    0.532  0.0713
##   173     26       1    0.511  0.0714
##   176     25       1    0.491  0.0714
##   218     23       1    0.470  0.0714
##   225     22       1    0.448  0.0713
##   241     21       1    0.427  0.0710
##   248     20       1    0.406  0.0706
##   273     19       1    0.384  0.0700
##   297     17       1    0.362  0.0695
##   405     15       1    0.337  0.0689
##   417     14       1    0.313  0.0681
##   420     13       1    0.289  0.0670
##   440     12       1    0.265  0.0656
##   523     11       1    0.241  0.0639
##   583      9       1    0.214  0.0622
##   594      8       1    0.187  0.0599
##  1101      7       1    0.161  0.0570
##  1146      5       1    0.129  0.0539
##  1417      1       1    0.000     NaN

## Call: survfit(formula = coxph(Surv(tempos.ex2, censuras.ex2) ~ 1, method = "breslow"))
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     7     50       1   0.9802  0.0196      0.94252        1.000
##    34     49       1   0.9604  0.0274      0.90809        1.000
```

```
##    42      48        1   0.9406  0.0333      0.87761         1.000
##    63      47        1   0.9208  0.0380      0.84925         0.998
##    64      46        1   0.9010  0.0420      0.82227         0.987
##    83      44        1   0.8807  0.0457      0.79558         0.975
##    84      43        1   0.8605  0.0489      0.76975         0.962
##    91      42        1   0.8403  0.0518      0.74463         0.948
##   108      41        1   0.8200  0.0544      0.72009         0.934
##   112      40        1   0.7998  0.0567      0.69607         0.919
##   129      39        1   0.7795  0.0587      0.67249         0.904
##   133      38        2   0.7396  0.0622      0.62724         0.872
##   139      36        1   0.7193  0.0637      0.60473         0.856
##   140      35        2   0.6793  0.0661      0.56139         0.822
##   146      33        1   0.6591  0.0672      0.53974         0.805
##   149      32        1   0.6388  0.0681      0.51835         0.787
##   154      31        1   0.6185  0.0689      0.49722         0.769
##   157      30        1   0.5982  0.0695      0.47634         0.751
##   160      29        2   0.5584  0.0704      0.43613         0.715
##   165      27        1   0.5381  0.0707      0.41591         0.696
##   173      26        1   0.5178  0.0709      0.39591         0.677
##   176      25        1   0.4975  0.0710      0.37614         0.658
##   218      23        1   0.4763  0.0710      0.35559         0.638
##   225      22        1   0.4551  0.0709      0.33531         0.618
##   241      21        1   0.4340  0.0707      0.31529         0.597
##   248      20        1   0.4128  0.0704      0.29554         0.577
##   273      19        1   0.3916  0.0699      0.27606         0.556
##   297      17        1   0.3693  0.0694      0.25551         0.534
##   405      15        1   0.3454  0.0689      0.23372         0.511
##   417      14        1   0.3216  0.0681      0.21238         0.487
##   420      13        1   0.2978  0.0671      0.19150         0.463
##   440      12        1   0.2740  0.0658      0.17111         0.439
##   523      11        1   0.2502  0.0643      0.15124         0.414
##   583       9        1   0.2239  0.0627      0.12937         0.387
##   594       8        1   0.1976  0.0606      0.10836         0.360
##  1101       7        1   0.1713  0.0579      0.08828         0.332
##  1146       5        1   0.1402  0.0551      0.06493         0.303
##  1417       1        1   0.0516  0.0554      0.00628         0.424
```

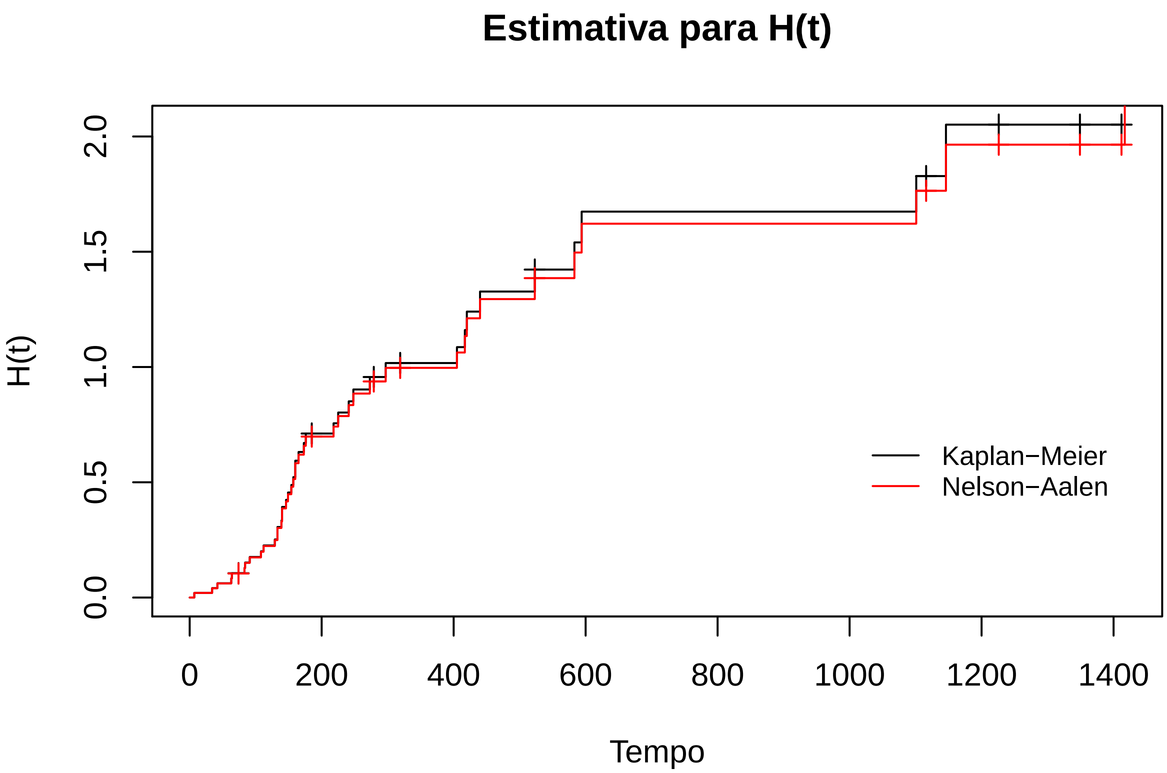
Item (b)

```r
# plot(
# KM.ex2,
# conf.int = FALSE,
# xlab = "Tempo",
# ylab = "S(t)",
# main = "Estimativa para S(t) \n Método de Kaplan-Meier"
# )
# plot(
# NA.ex2,
# conf.int = FALSE,
# xlab = "Tempo",
# ylab = "S(t)",
# main = "Estimativa para S(t) \n Método de Nelson-Aalen"
```

```r
# )
# par(mfrow=c(1,2))
# plot(
# KM.ex2,
# conf.int = FALSE,
# xlab = "Tempo",
# ylab = "S(t)",
# main = "Estimativa para S(t) \n Método de Kaplan-Meier"
# )
# plot(
# NA.ex2,
# conf.int = FALSE,
# xlab = "Tempo",
# ylab = "S(t)",
# main = "Estimativa para S(t) \n Método de Nelson-Aalen"
# )
par(mfrow = c(1,1))
plot(
KM.ex2,
conf.int = FALSE,
xlab = "Tempo",
ylab = "S(t)",
main = "Estimativas para S(t) \n ",
mark.time = T
)
lines(NA.ex2, col = 2, conf.int = F, mark.time = T)
legend(1000,0.75,lty=c(1,1),c("Kaplan-Meier","Nelson-Aalen"),bty="n",cex=0.8,col=c(1,2))
```



```r
# plot(KM.ex2, fun = "cumhaz", xlab = "Tempo", ylab = "H(t)", main = "Estimativa para H(t) \n Método de
# plot(NA.ex2, conf.int = F, fun = "cumhaz", xlab = "Tempo", ylab = "H(t)", main = "Estimativa para H(t


plot(KM.ex2, fun = "cumhaz", xlab = "Tempo", ylab = "H(t)", main = "Estimativa para H(t)", mark.time = 
lines(NA.ex2, col =2, fun = "cumhaz", conf.int = F, mark.time = T)
legend(1000,0.75,lty=c(1,1),c("Kaplan-Meier","Nelson-Aalen"),bty="n",cex=0.8,col=c(1,2))
```



Item (c)

Usando Interpolação:

$$\frac{176-173}{0,491-0,511} = \frac{\hat{t}_{MD,K-M}-173}{0,5-0,511} \Rightarrow \hat{t}_{MD,K-M} \approx 175(174,65)$$

$$\frac{176-173}{0,4975-0,5178} = \frac{\hat{t}_{MD,N-A}-173}{0,5-0,5178} \Rightarrow \hat{t}_{MD,N-A} \approx 175(174,993)$$

Item (d)

1. $\frac{42-34}{0,940-0,960} = \frac{40-34}{\hat{S}(40)_{K-M}-0,960} \Rightarrow \hat{S}(40)_{K-M} = 0,945$
2. $\frac{108-91}{0,818-0,839} = \frac{100-91}{\hat{S}(100)_{K-M}-0,839} \Rightarrow \hat{S}(100)_{K-M} = 0,827882$
3. $\frac{405-294}{0,337-0,362} = \frac{300-294}{\hat{S}(300)_{K-M}-0,362} \Rightarrow \hat{S}(300)_{K-M} = 0,360649$

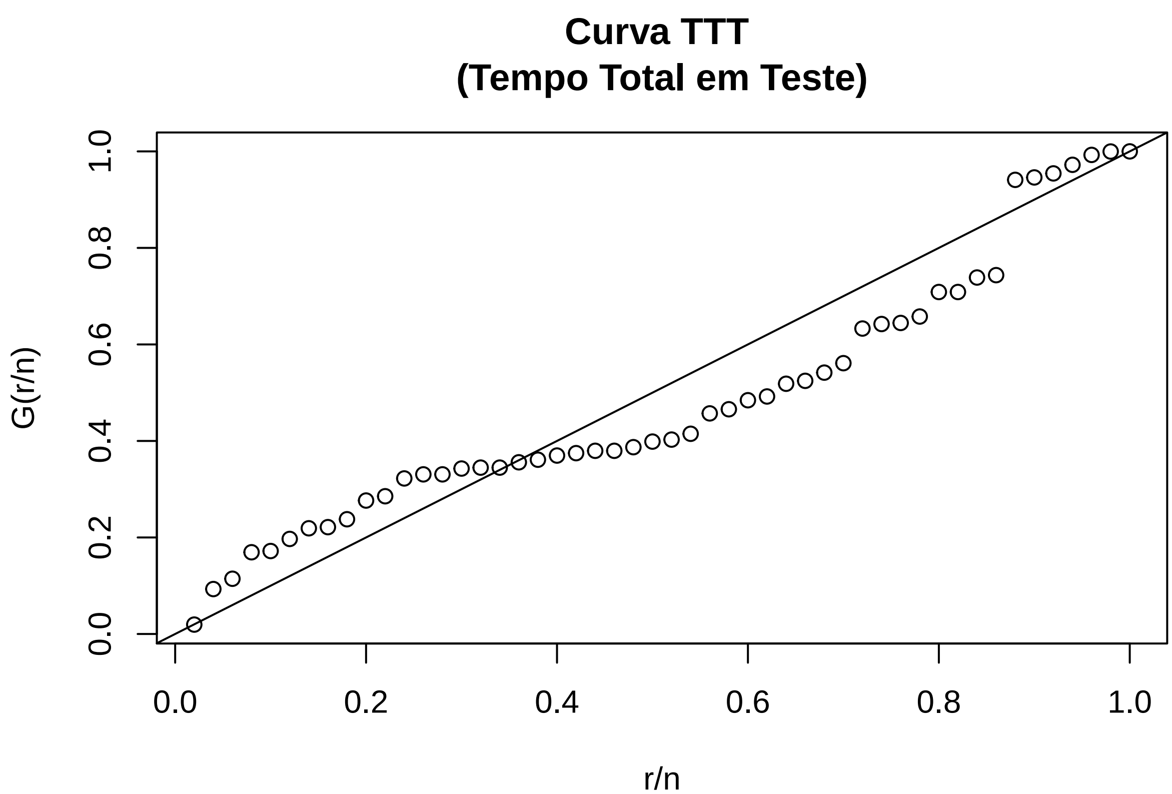
4. $\frac{1101-594}{0{,}161-0{,}187} = \frac{1000-594}{S(\hat{1000})_{K-M}-0{,}187} \Rightarrow \hat{S}(1000)_{K-M} = 0,166179$

Item (e)

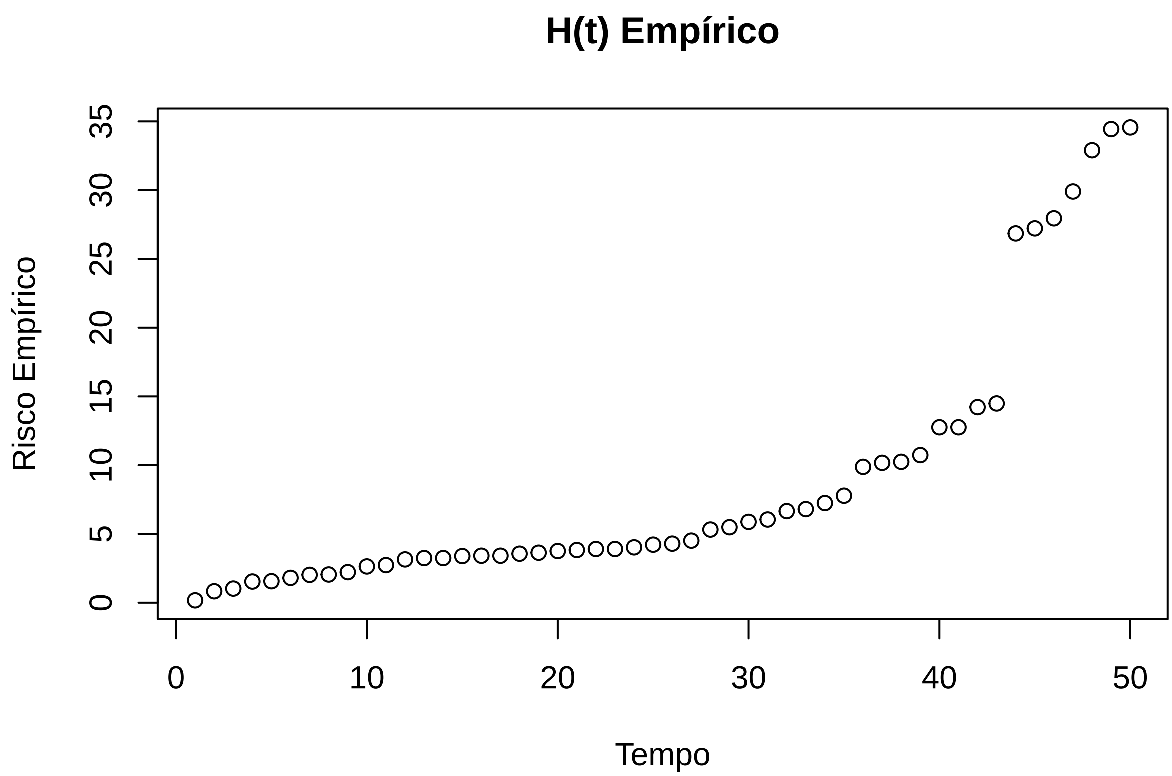
O comportamento das estimativas para a função de sobrevivência tanto usando o Estimador de Kaplan-Meier quanto utilizando o de Nelson-Aalen sugerem que este não é um problema com fração de cura, já que a mesma *tende a zero* com o passar do tempo.

Item (f)

```
curvaTTT <- function(tempo) {
  G_rn <- numeric()
  for (i in 1:length(tempo)) {
    G_rn[i] <-
      (sum(sort(tempo)[1:i]) + (length(tempo) - i) * sort(tempo)[i]) / (sum(tempo))
  }
  rn <- seq_along(tempo)/length(tempo)
  plot(G_rn ~ rn,
       main = "Curva TTT \n(Tempo Total em Teste)",
       ylab = "G(r/n)",
       xlab = "r/n")
  abline(0,1)
}

curvaTTT(tempos.ex2)
```



```r
plot(tempos.ex2/sum(censuras.ex2),
     ylab = "Risco Empírico",
     xlab = "Tempo",
     main = "H(t) Empírico")
```



O Gráfico da Curva TTT sugere uma distribuição paramétrica unimodal, como a Gama Generalizada, ou a Weibull Exponencializada (com $\gamma < 1$ e $\gamma a > 1$).

Exercício 3

```r
tempos.g.ex3 <- c(28, 89, 175, 195, 309, 377, 393, 421, 447, 462, 709, 744, 770, 1106, 1206)
tempos.p.ex3 <- c(34, 88, 137, 199, 280, 291, 299, 300, 309, 351,358, 369, 369, 370, 375, 382, 392, 429
censuras.g.ex3 <- c(rep(1,5), rep(0,4), 1, rep(0,5))
censuras.p.ex3 <- c(rep(1,6), rep(0,2), rep(1,9), 0, 1, 0)
```

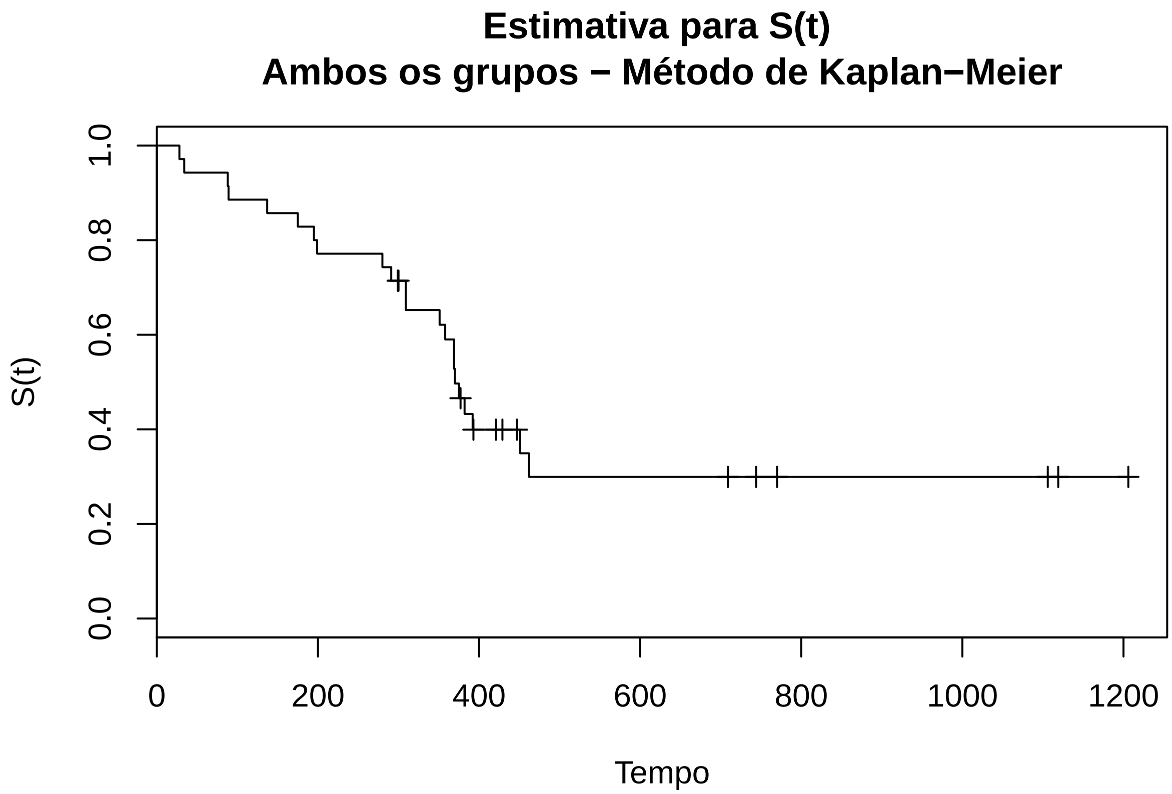
Item (a)

```r
ex3 <-
  survfit(Surv(time = append(tempos.g.ex3, tempos.p.ex3), event = append(censuras.g.ex3, censuras.p.ex3
summary(ex3)
```

```
## Call: survfit(formula = Surv(time = append(tempos.g.ex3, tempos.p.ex3),
##     event = append(censuras.g.ex3, censuras.p.ex3)) ~ 1, conf.int = F)
##
##  time n.risk n.event survival std.err
##    28     35       1    0.971  0.0282
```

```
##    34      34        1     0.943  0.0392
##    88      33        1     0.914  0.0473
##    89      32        1     0.886  0.0538
##   137      31        1     0.857  0.0591
##   175      30        1     0.829  0.0637
##   195      29        1     0.800  0.0676
##   199      28        1     0.771  0.0710
##   280      27        1     0.743  0.0739
##   291      26        1     0.714  0.0764
##   309      23        2     0.652  0.0814
##   351      21        1     0.621  0.0832
##   358      20        1     0.590  0.0847
##   369      19        2     0.528  0.0864
##   370      17        1     0.497  0.0867
##   375      16        1     0.466  0.0867
##   382      14        1     0.433  0.0866
##   392      13        1     0.399  0.0861
##   451       8        1     0.349  0.0886
##   462       7        1     0.299  0.0889
```

```r
plot(ex3, conf.int = FALSE, xlab = "Tempo", ylab = "S(t)", main = "Estimativa para S(t) \n Ambos os gru
```



- Este gráfico de sobrevivência sugere a presença de fração de cura na amostra. Há suspeita de que este comportamento pode ser explicado pela perda de informação ao agregarmos os dois grupos em um só, ou seja, pode ser um comportamento espúrio.

Item (b)

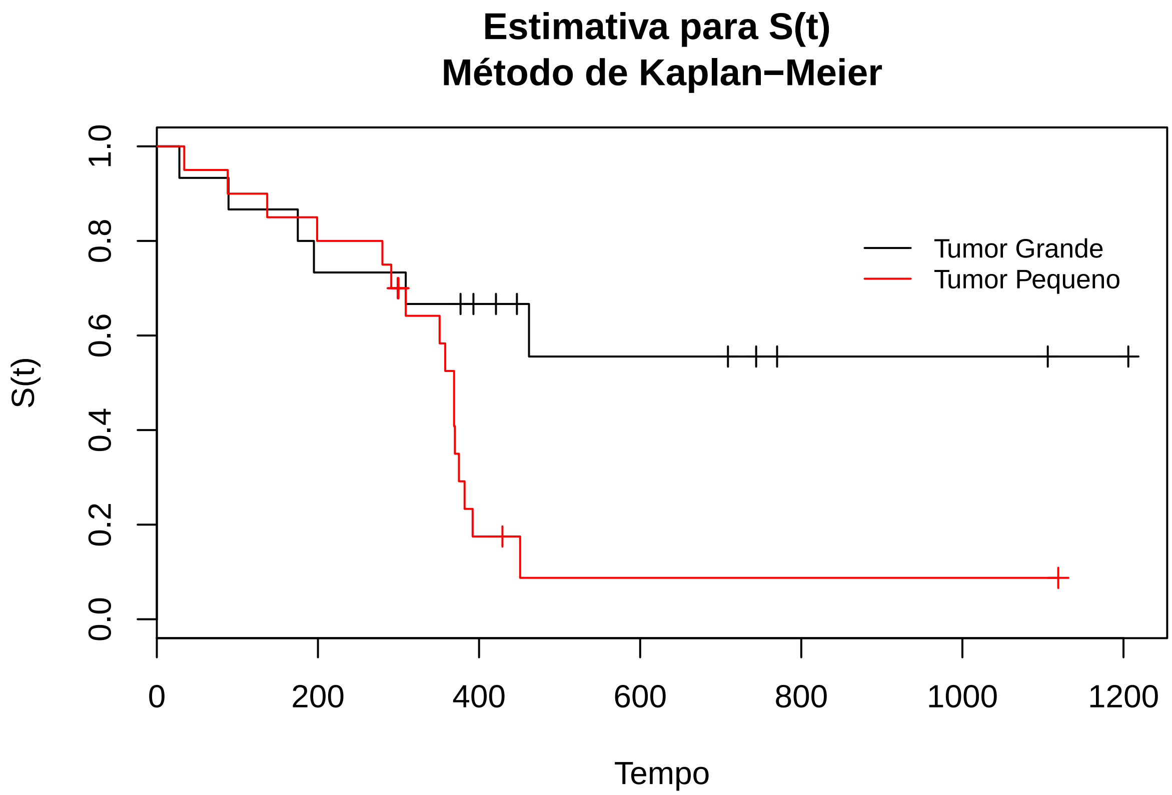
```r
require(survival)
g.ex3 <-
  survfit(Surv(time = tempos.g.ex3, event = censuras.g.ex3) ~ 1, conf.int = F)
summary(g.ex3)
```

```
## Call: survfit(formula = Surv(time = tempos.g.ex3, event = censuras.g.ex3) ~
##     1, conf.int = F)
##
##  time n.risk n.event survival std.err
##    28     15       1    0.933  0.0644
##    89     14       1    0.867  0.0878
##   175     13       1    0.800  0.1033
##   195     12       1    0.733  0.1142
##   309     11       1    0.667  0.1217
##   462      6       1    0.556  0.1434
```

```r
p.ex3 <-
  survfit(Surv(time = tempos.p.ex3, event = censuras.p.ex3) ~ 1, conf.int = F)
summary(p.ex3)
```

```
## Call: survfit(formula = Surv(time = tempos.p.ex3, event = censuras.p.ex3) ~
##     1, conf.int = F)
##
##  time n.risk n.event survival std.err
##    34     20       1   0.9500  0.0487
##    88     19       1   0.9000  0.0671
##   137     18       1   0.8500  0.0798
##   199     17       1   0.8000  0.0894
##   280     16       1   0.7500  0.0968
##   291     15       1   0.7000  0.1025
##   309     12       1   0.6417  0.1093
##   351     11       1   0.5833  0.1139
##   358     10       1   0.5250  0.1165
##   369      9       2   0.4083  0.1162
##   370      7       1   0.3500  0.1133
##   375      6       1   0.2917  0.1084
##   382      5       1   0.2333  0.1012
##   392      4       1   0.1750  0.0912
##   451      2       1   0.0875  0.0769
```

```r
plot(g.ex3, conf.int = FALSE, xlab = "Tempo", ylab = "S(t)", main = "Estimativa para S(t) \n Método de K
lines(p.ex3, col = 2, conf.int = F, mark.time = T)
legend(850,0.85,lty=c(1,1),c("Tumor Grande","Tumor Pequeno"),bty="n",cex=0.8,col=c(1,2))
```



- Como citado na interpretação do item anterior, nota-se uma clara diferença entre as estimativas da função de sobrevivência entre os grupos "Tumor Grande" e "Tumor Pequeno": no primeiro temos a presença de fração de cura.
- A razão entre as funções de sobrevivência (e consequentemente de risco) estimadas para os dois grupos não é (aproximadamente) proporcional, o que inviabiliza o uso do Teste de logRank para a igualdade das curvas de sobrevivência. O que de certa forma não é grande prejuízo já que, por inspeção, podemos considerar que as curvas são *suficientemente* diferentes para que o teste não tenha tanta relevância prática.

Item (c)

```
grupo <- c(rep(1,15), rep(2,20))
survdiff(Surv(time = append(tempos.g.ex3, tempos.p.ex3), event = append(censuras.g.ex3, censuras.p.ex3))
```

```
## Call:
## survdiff(formula = Surv(time = append(tempos.g.ex3, tempos.p.ex3),
##     event = append(censuras.g.ex3, censuras.p.ex3)) ~ grupo,
##     rho = 0)
##
##          N Observed Expected (O-E)^2/E (O-E)^2/V
## grupo=1 15        6     11.3      2.51      5.57
## grupo=2 20       16     10.7      2.67      5.57
##
##  Chisq= 5.6  on 1 degrees of freedom, p= 0.0183
```

```
survdiff(Surv(time = append(tempos.g.ex3, tempos.p.ex3), event = append(censuras.g.ex3, censuras.p.ex3))
```

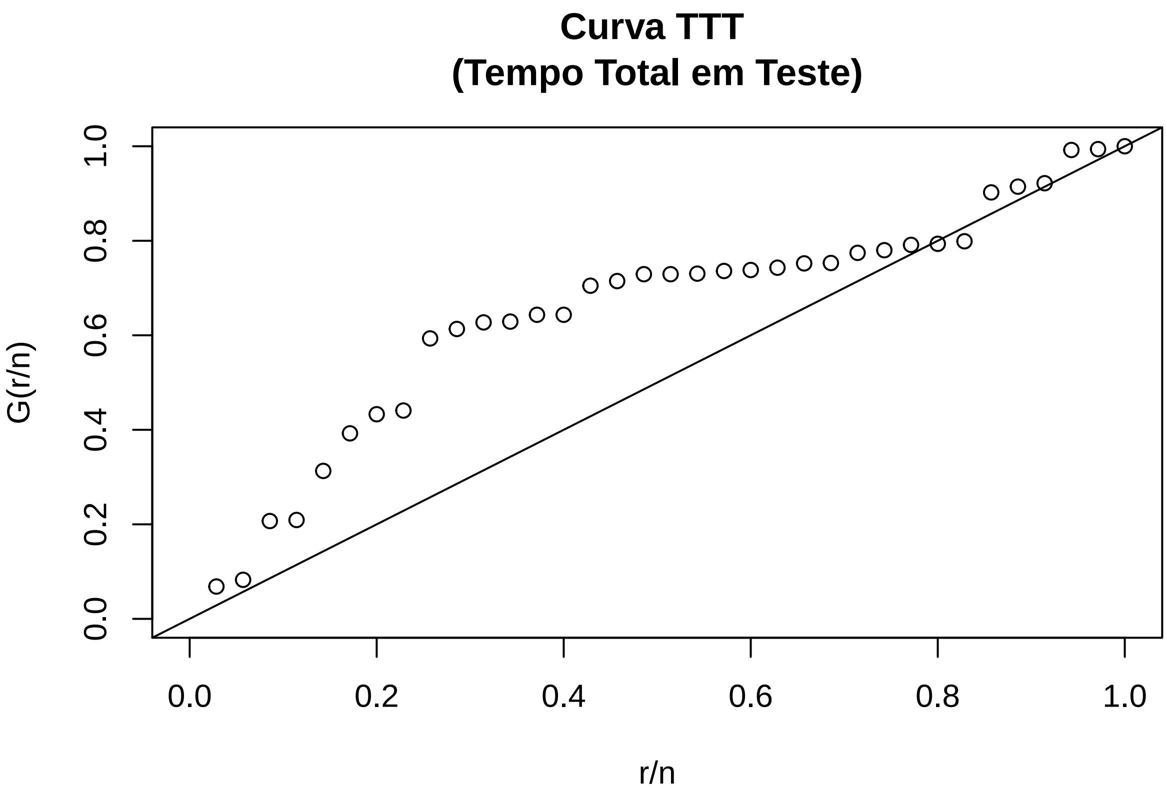
```
## Call:
```

```
## survdiff(formula = Surv(time = append(tempos.g.ex3, tempos.p.ex3),
##     event = append(censuras.g.ex3, censuras.p.ex3)) ~ grupo,
##     rho = 1)
##
##          N Observed Expected (O-E)^2/E (O-E)^2/V
## grupo=1 15     4.66     7.40     1.013      2.74
## grupo=2 20    10.60     7.87     0.953      2.74
##
##  Chisq= 2.7  on 1 degrees of freedom, p= 0.0978
```

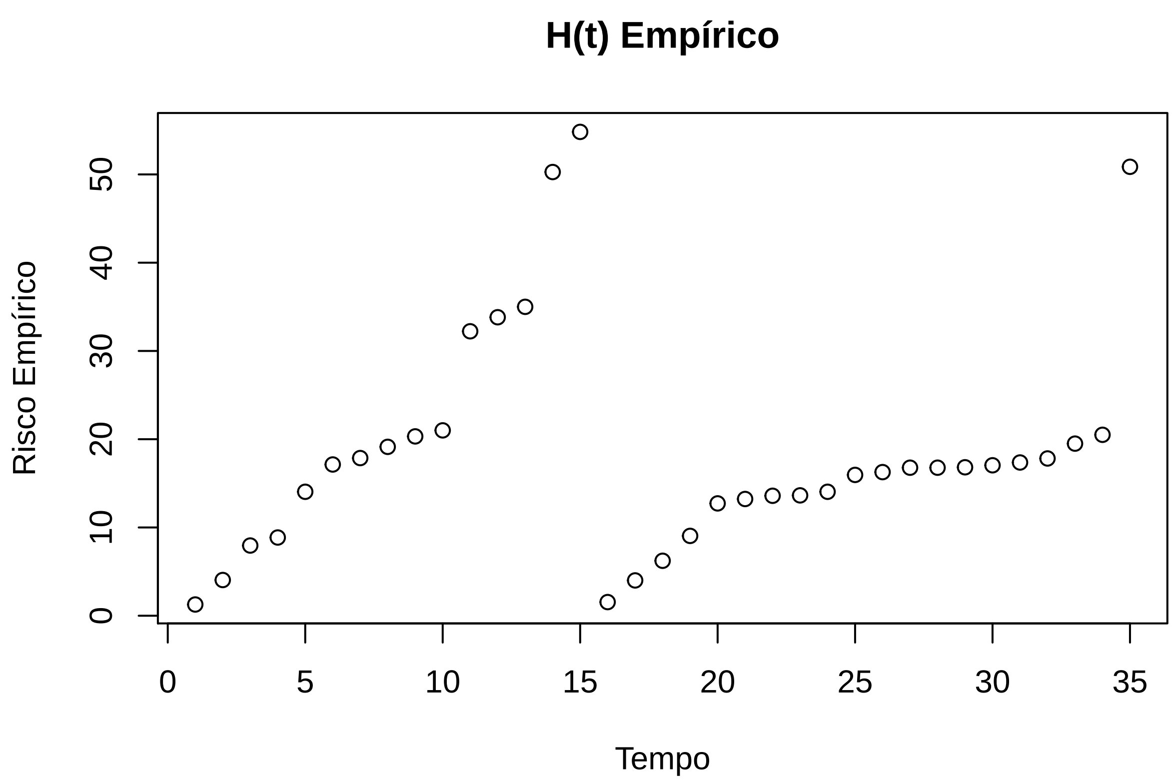
- O Teste _log_Rank rejeita a hipótese nula (de igualdade entre as funções de sobrevivência) a um p-valor ≈ 0,02, ao contrário do Teste de Wilcoxon, que rejeita a igualdade com p-valor inferior, que não seria adequado caso considerássemos $\alpha = 0,05$ como critério.
- O motivo desta discrepância é o achado citado no item anterior, a suposição de razão constante entre as funções de sobrevivência, exigida pelo Teste de _log_Rank, não é verificada.
- O Teste de Wilcoxon é robusto à falta desta suposição, então **não rejeitamos a igualdade entre as funções de sobrevivência a um nível de significância de cerca de 9%**.

Item (d)

```r
curvaTTT <- function(tempo) {
  G_rn <- numeric()
  for (i in 1:length(tempo)) {
    G_rn[i] <-
      (sum(sort(tempo)[1:i]) + (length(tempo) - i) * sort(tempo)[i]) / (sum(tempo))
  }
  rn <- seq_along(tempo)/length(tempo)
  plot(G_rn ~ rn,
       main = "Curva TTT \n(Tempo Total em Teste)",
       ylab = "G(r/n)",
       xlab = "r/n",
       xlim = c(0, 1),
       ylim = c(0, max(rn))
       )
  abline(0,1)
}

curvaTTT(append(tempos.g.ex3, tempos.p.ex3))
```




```r
plot(append(tempos.g.ex3, tempos.p.ex3)/sum(append(censuras.g.ex3, censuras.p.ex3)),
     ylab = "Risco Empírico",
     xlab = "Tempo",
     main = "H(t) Empírico")
```



A Curva TTT indica que a função Taxa de Falha é monotonicamente crescente, ou seja, podemos utilizar a distribuição Weibull com $\gamma > 1$.

Exercício 4

```
tempos.1.ex4 <- c(140, 177, 50, 65, 86, 153, 181, 191, 77, 84, 87, 56, 66, 73, 119, 140, 200, 200,200,20
censuras.1.ex4 <- c(rep(1,15), rep(0,15))

tempos.2.ex4 <- c(124, 58, 56, 68, 79, 89, 107, 86, 142, 110, 96, 142, 86, 75, 117, 98, 105, 126, 43, 4
censuras.2.ex4 <- c(rep(1,23), rep(0,7))

tempos.3.ex4 <- c(112, 68, 84, 109, 153, 143, 60, 70, 98, 164, 63, 63, 77, 91, 91, 66, 70, 77, 63, 66,
censuras.3.ex4 <- c(rep(1,30))
```

Item (a)

```
ex4 <- survfit(Surv(time = append(tempos.1.ex4, tempos.2.ex4, tempos.3.ex4), event = append(censuras.1.
```

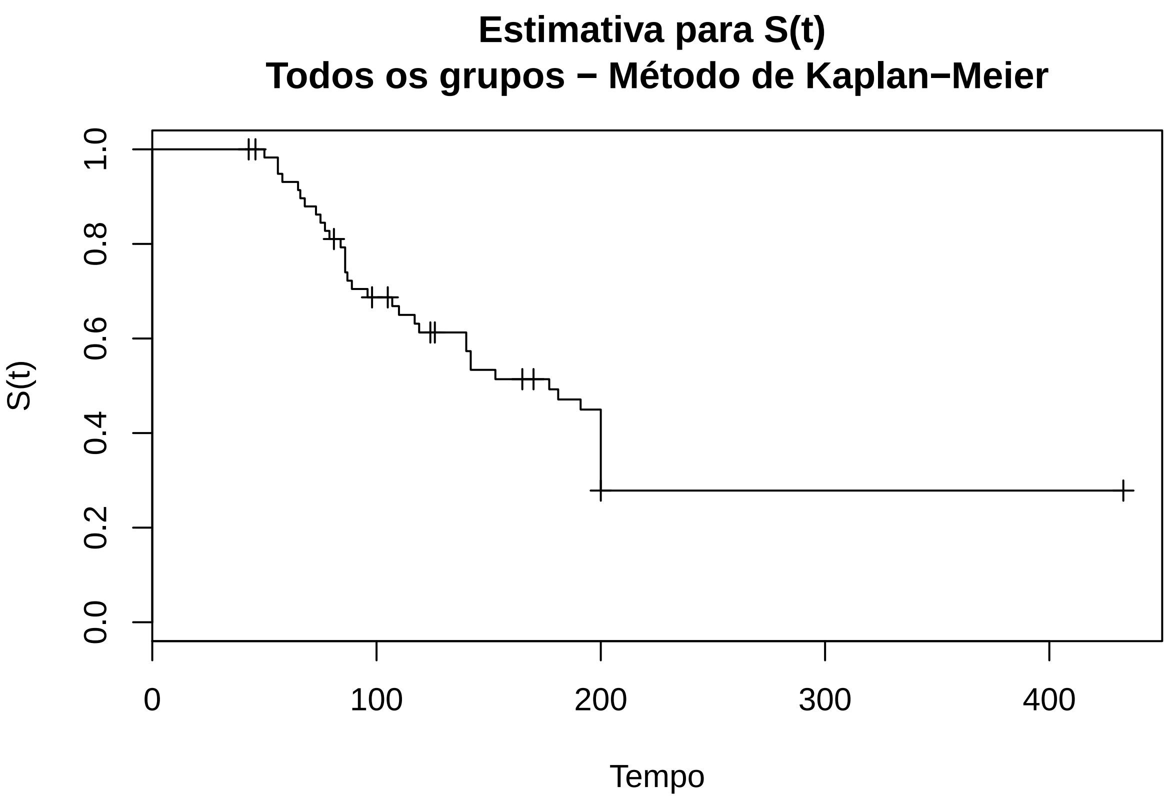
```
## Warning in if (!after) c(values, x) else if (after >= lengx) c(x, values)
## else c(x[1L:after], : the condition has length > 1 and only the first
## element will be used

## Warning in if (after >= lengx) c(x, values) else c(x[1L:after], values, :
## the condition has length > 1 and only the first element will be used

## Warning in if (!after) c(values, x) else if (after >= lengx) c(x, values)
## else c(x[1L:after], : the condition has length > 1 and only the first
## element will be used

## Warning in if (after >= lengx) c(x, values) else c(x[1L:after], values, :
## the condition has length > 1 and only the first element will be used

## Warning in 1L:after: numerical expression has 30 elements: only the first
## used

## Warning in (after + 1L):lengx: numerical expression has 30 elements: only
## the first used
```

```
summary(ex4)
```

```
## Call: survfit(formula = Surv(time = append(tempos.1.ex4, tempos.2.ex4,
##     tempos.3.ex4), event = append(censuras.1.ex4, censuras.2.ex4,
##     censuras.3.ex4)) ~ 1, conf.int = F)
##
##  time n.risk n.event survival std.err
##    50     58       1    0.983  0.0171
##    56     57       2    0.948  0.0291
##    58     55       1    0.931  0.0333
##    65     54       1    0.914  0.0369
##    66     53       1    0.897  0.0400
##    68     52       1    0.879  0.0428
##    73     51       1    0.862  0.0453
##    75     50       1    0.845  0.0475
##    77     49       1    0.828  0.0496
```

```
##     79      48        1     0.810  0.0515
##     84      46        1     0.793  0.0533
##     86      45        3     0.740  0.0578
##     87      42        1     0.722  0.0591
##     89      41        1     0.705  0.0602
##     96      40        1     0.687  0.0612
##    107      37        1     0.668  0.0623
##    110      36        1     0.650  0.0633
##    117      35        1     0.631  0.0641
##    119      34        1     0.613  0.0649
##    140      31        2     0.573  0.0664
##    142      29        2     0.534  0.0675
##    153      27        1     0.514  0.0678
##    177      24        1     0.493  0.0683
##    181      23        1     0.471  0.0686
##    191      22        1     0.450  0.0687
##    200      21        8     0.278  0.0639
```

```
plot(ex4, conf.int = FALSE, xlab = "Tempo", ylab = "S(t)", main = "Estimativa para S(t) \n Todos os gru
```



O gráfico sugere a presença de fração de cura. Mas este efeito pode ser espúrio, explicado pela mesma razão do item 3.a: a perda de informação quando desconsideramos os grupos.

Item (b)

```
grupo <- c(rep(1,30), rep(2,30), rep(3,30))
g1.ex4 <-
  survfit(Surv(time = tempos.1.ex4, event = censuras.1.ex4) ~ 1, conf.int = F)
```

```r
summary(g1.ex4)
```

```
## Call: survfit(formula = Surv(time = tempos.1.ex4, event = censuras.1.ex4) ~
##     1, conf.int = F)
##
##  time n.risk n.event survival std.err
##    50     30       1    0.967  0.0328
##    56     29       1    0.933  0.0455
##    65     28       1    0.900  0.0548
##    66     27       1    0.867  0.0621
##    73     26       1    0.833  0.0680
##    77     25       1    0.800  0.0730
##    84     24       1    0.767  0.0772
##    86     23       1    0.733  0.0807
##    87     22       1    0.700  0.0837
##   119     21       1    0.667  0.0861
##   140     20       1    0.633  0.0880
##   153     18       1    0.598  0.0899
##   177     17       1    0.563  0.0912
##   181     16       1    0.528  0.0920
##   191     15       1    0.493  0.0924
```

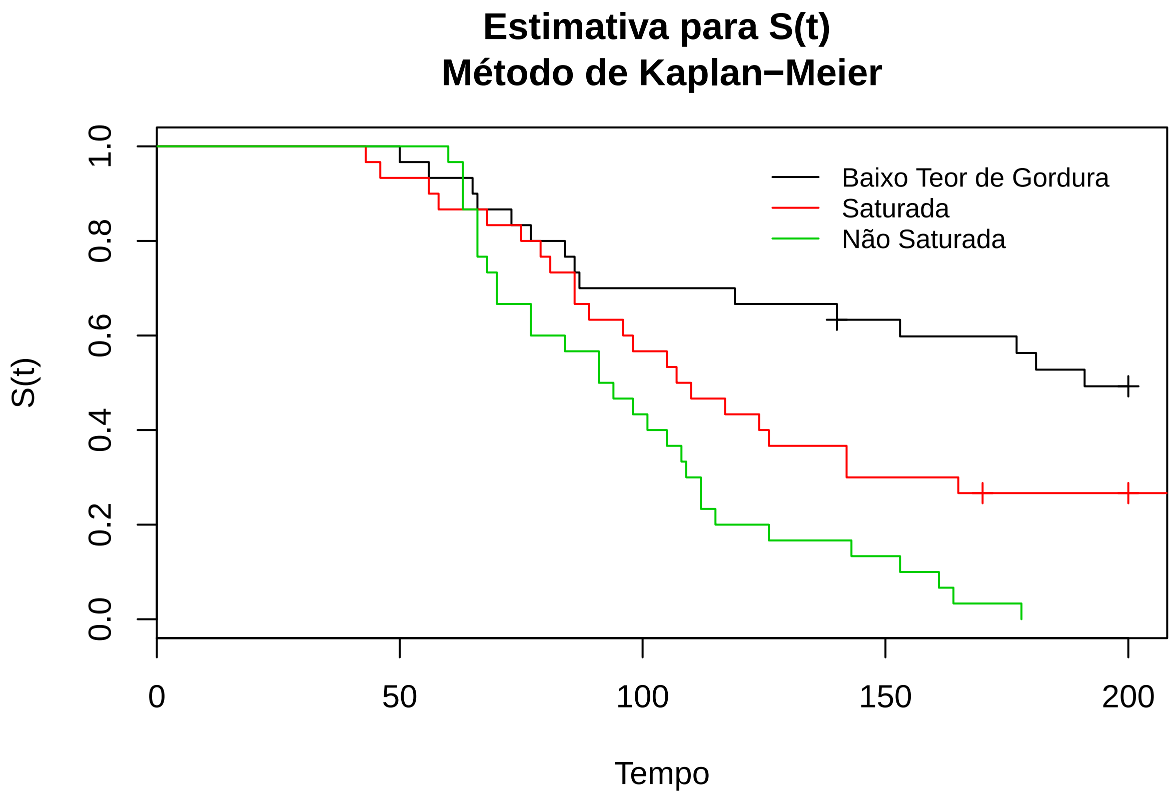
```r
g2.ex4 <-
  survfit(Surv(time = tempos.2.ex4, event = censuras.2.ex4) ~ 1, conf.int = F)
summary(g2.ex4)
```

```
## Call: survfit(formula = Surv(time = tempos.2.ex4, event = censuras.2.ex4) ~
##     1, conf.int = F)
##
##  time n.risk n.event survival std.err
##    43     30       1    0.967  0.0328
##    46     29       1    0.933  0.0455
##    56     28       1    0.900  0.0548
##    58     27       1    0.867  0.0621
##    68     26       1    0.833  0.0680
##    75     25       1    0.800  0.0730
##    79     24       1    0.767  0.0772
##    81     23       1    0.733  0.0807
##    86     22       2    0.667  0.0861
##    89     20       1    0.633  0.0880
##    96     19       1    0.600  0.0894
##    98     18       1    0.567  0.0905
##   105     17       1    0.533  0.0911
##   107     16       1    0.500  0.0913
##   110     15       1    0.467  0.0911
##   117     14       1    0.433  0.0905
##   124     13       1    0.400  0.0894
##   126     12       1    0.367  0.0880
##   142     11       2    0.300  0.0837
##   165      9       1    0.267  0.0807
##   433      1       1    0.000     NaN
```

```r
g3.ex4 <-
  survfit(Surv(time = tempos.3.ex4, event = censuras.3.ex4) ~ 1, conf.int = F)
summary(g3.ex4)
```

```
## Call: survfit(formula = Surv(time = tempos.3.ex4, event = censuras.3.ex4) ~
##     1, conf.int = F)
##
##  time n.risk n.event survival std.err
##    60     30       1   0.9667  0.0328
##    63     29       3   0.8667  0.0621
##    66     26       3   0.7667  0.0772
##    68     23       1   0.7333  0.0807
##    70     22       2   0.6667  0.0861
##    77     20       2   0.6000  0.0894
##    84     18       1   0.5667  0.0905
##    91     17       2   0.5000  0.0913
##    94     15       1   0.4667  0.0911
##    98     14       1   0.4333  0.0905
##   101     13       1   0.4000  0.0894
##   105     12       1   0.3667  0.0880
##   108     11       1   0.3333  0.0861
##   109     10       1   0.3000  0.0837
##   112      9       2   0.2333  0.0772
##   115      7       1   0.2000  0.0730
##   126      6       1   0.1667  0.0680
##   143      5       1   0.1333  0.0621
##   153      4       1   0.1000  0.0548
##   161      3       1   0.0667  0.0455
##   164      2       1   0.0333  0.0328
##   178      1       1   0.0000     NaN
```

```r
plot(g1.ex4, conf.int = FALSE, xlab = "Tempo", ylab = "S(t)", main = "Estimativa para S(t) \n Método de
lines(g2.ex4, col = 2, conf.int = F, mark.time = T)
lines(g3.ex4, col = 3, conf.int = F, mark.time = T)
legend(122,1,lty=c(1,1),c("Baixo Teor de Gordura","Saturada", "Não Saturada"),bty="n",cex=0.8,col=c(1,2
```



Item (c)

```
grupos <- c(rep(1,30), rep(2,30), rep(3,30))
survdiff(Surv(time = append(tempos.1.ex4, append(tempos.2.ex4, tempos.3.ex4)), event = append(censuras.
```

```
## Call:
## survdiff(formula = Surv(time = append(tempos.1.ex4, append(tempos.2.ex4,
##     tempos.3.ex4)), event = append(censuras.1.ex4, append(censuras.2.ex4,
##     censuras.3.ex4))) ~ grupos, rho = 0)
##
##             N Observed Expected (O-E)^2/E (O-E)^2/V
## grupos=1 30       15     28.9  6.70e+00  1.25e+01
## grupos=2 30       23     23.1  1.41e-04  2.23e-04
## grupos=3 30       30     16.0  1.22e+01  1.71e+01
##
##  Chisq= 20.4  on 2 degrees of freedom, p= 3.7e-05
```

```
survdiff(Surv(time = append(tempos.1.ex4, append(tempos.2.ex4, tempos.3.ex4)), event = append(censuras.
```

```
## Call:
## survdiff(formula = Surv(time = append(tempos.1.ex4, append(tempos.2.ex4,
##     tempos.3.ex4)), event = append(censuras.1.ex4, append(censuras.2.ex4,
##     censuras.3.ex4))) ~ grupo, rho = 1)
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## grupo=1 30     9.35     17.0  3.47e+00  8.56e+00
## grupo=2 30    14.41     14.4  2.91e-06  6.31e-06
## grupo=3 30    19.18     11.5  5.13e+00  9.75e+00
```

```
##
##  Chisq= 12.3  on 2 degrees of freedom, p= 0.00211
```

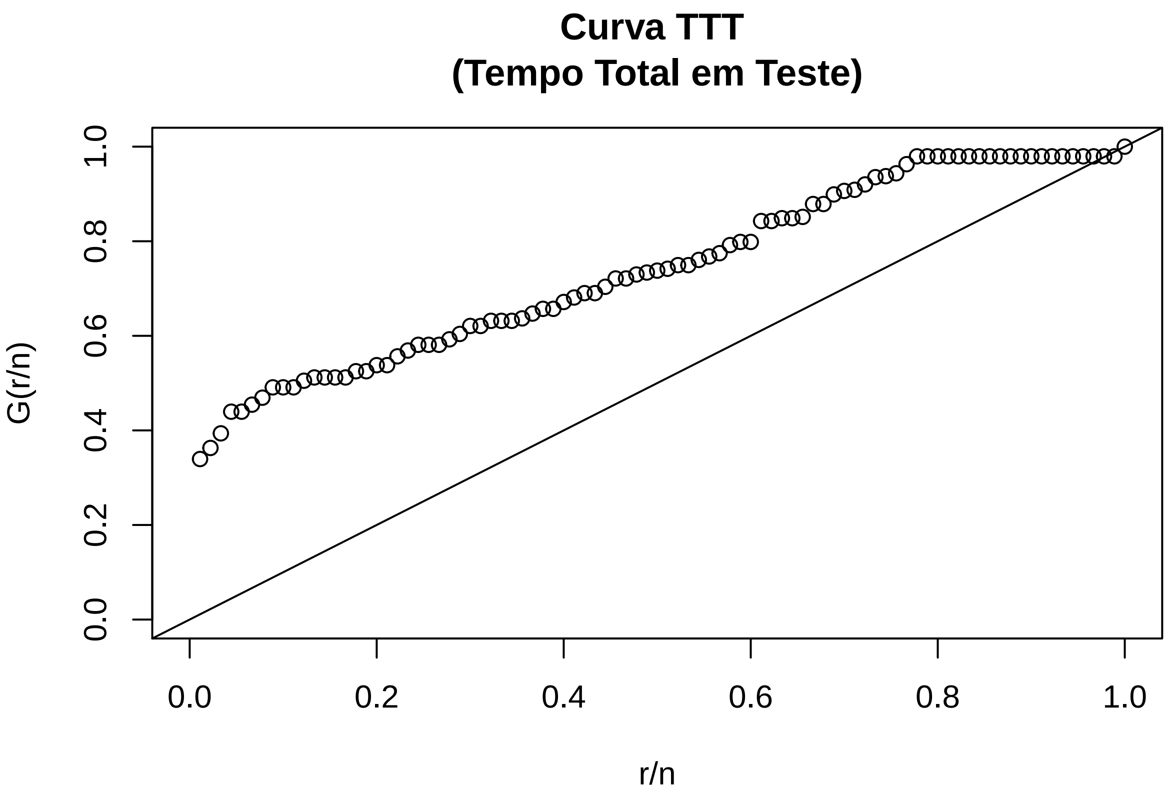
Como no item (c) da questão anterior, utilizamos aqui o Teste _log_Rank e o Teste de Wilcoxon, respectivamente.

O Teste _log_Rank supõe a razão entre as curvas de sobrevivência estimadas como constante, o que claramente não se verifica de acordo com o item (b). De qualquer maneira, este Teste rejeitou a hipótese de igualdade entre as curvas a um alto nível de significância, da ordem de 10^{-5}.

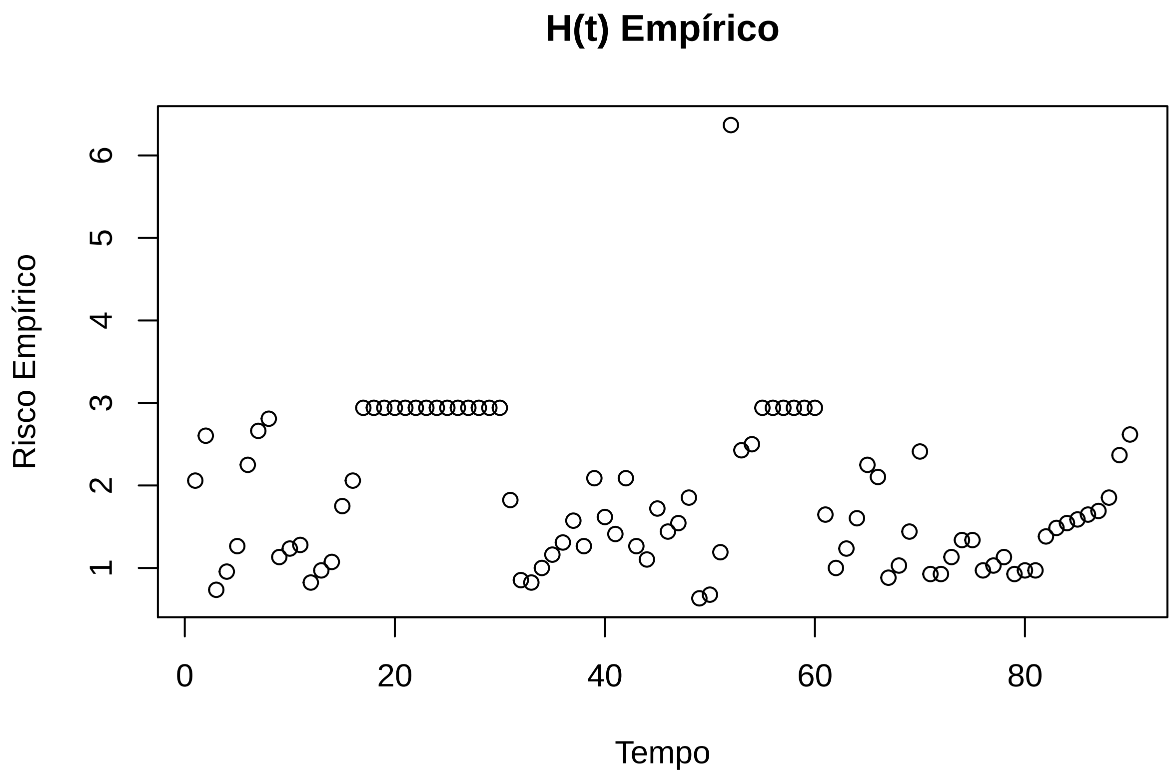
O Teste de Wilcoxon não necessita da suposição do Teste _log_Rank, então ele é perfeitamente aplicável à nossa situação. E de fato ele rejeita a hipótese nula também com nível de significância considerável, da ordem de 10^{-3}. Aliás a significância deste teste é observada mesmo considerando a correção de Bonferroni para comparações múltiplas: já que estamos comparando três grupos, a um nível $\alpha_g = 0,05$ para as comparações dois-a-dois devemos ter um nível de significância $alpha_b \approx 0,0167$ para as comparações de três grupos.

Item (d)

```
curvaTTT(append(tempos.1.ex4, append(tempos.2.ex4, tempos.3.ex4)))
```




```
plot(append(tempos.1.ex4, append(tempos.2.ex4, tempos.3.ex4))/sum(append(censuras.1.ex4, append(censuras
     ylab = "Risco Empírico",
     xlab = "Tempo",
     main = "H(t) Empírico")
```



A Curva TTT sugere uma distribuição crescente para a função de risco, ou seja, podemos utilizar a Distribuição Weibull com $\gamma > 1$.

Exercício 5

Item (a)

Entendo que o **Tratamento B** deve ser o mais eficiente dos três, pois sua curva de sobrevivência decresce com certa regularidade a partir de $t \approx 30$. Como o evento de interesse aqui é a cura do paciente, isso indica que o Tratamento B cura pacientes a partir de um tempo inferior ao do Tratamento C, mesmo que este tenha um desempenho (o que pode ser chamado de eficiência?) relativamente maior em $t \approx 45$.

Entendo que esta seria uma conclusão um tanto *grosseira*, já que não leva em conta outros aspectos dos tratamentos em questão como seus custos, efeitos colaterais etc.

É interessante também notar a inexistência de marcações de censura em ambas as curvas de sobrevivência.

Item (b)

Difícil dizer sem verificar que a hipótese de igualdade entre ambas as curvas, e sem levar em consideração outros aspectos dos dois tratamentos em questão, já que ambos observam o mesmo valor em $t = 50$.

Item (c)

Aqui o valor da função de sobrevivência do **Tratamento B** é claramente superior, então supondo que pudéssemos estabelecer a diferença entre ambas as curvas, este é o tratamento mais razoável