Análise de Sobrevivência

Modelos Paramétricos

Ricardo Accioly

Modelos Paramétricos

Neste exemplo são considerados os tempos de reincidência, em meses, de um grupo de 20 pacientes com cancer de bexiga que foram submetidos a um procedimento cirurgico feito por laser.

Vamos ver passo a passo como ajustar os modelos exponencial, Weibull e lognormal.

Modelos Paramétricos

Neste exemplo são considerados os tempos de reincidência, em meses, de um grupo de 20 pacientes com cancer de bexiga que foram submetidos a um procedimento cirurgico feito por laser.

Vamos ver passo a passo como ajustar os modelos exponencial, Weibull e lognormal.

Aqui vamos utilizar as funções existentes no pacote `survival`.

Modelos Paramétricos

Neste exemplo são considerados os tempos de reincidência, em meses, de um grupo de 20 pacientes com cancer de bexiga que foram submetidos a um procedimento cirurgico feito por laser.

Vamos ver passo a passo como ajustar os modelos exponencial, Weibull e lognormal.

Aqui vamos utilizar as funções existentes no pacote `survival`.

Para ajustar um modelo paramétrico usamo a função `survreg`.

Modelos Não Paramétricos - KM

```r
library(survival)
```

Modelos Não Paramétricos - KM

```r
library(survival)
tempos<-c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
```

Modelos Não Paramétricos - KM

```
library(survival)
tempos←c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
cens←c(1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0)
```

Modelos Não Paramétricos - KM

```r
library(survival)
tempos<-c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
cens<-c(1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0)
dados <- data.frame(tempos=tempos, status=cens)
```

Modelos Não Paramétricos - KM

```
library(survival)
tempos←c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
cens←c(1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0)
dados ← data.frame(tempos=tempos, status=cens)
ekm ← survfit(Surv(tempos,status)~1, data=dados)
```

Modelos Não Paramétricos - KM

```r
library(survival)
tempos←c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
cens←c(1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0)
dados ← data.frame(tempos=tempos, status=cens)
ekm ← survfit(Surv(tempos,status)~1, data=dados)
summary(ekm)
```

```
Call: survfit(formula = Surv(tempos, status) ~ 1, data = dados)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    3     20       1   0.9500  0.0487      0.85913        1.000
    5     19       1   0.9000  0.0671      0.77767        1.000
    6     18       1   0.8500  0.0798      0.70707        1.000
    7     17       1   0.8000  0.0894      0.64257        0.996
    8     16       1   0.7500  0.0968      0.58233        0.966
    9     15       1   0.7000  0.1025      0.52541        0.933
   10     14       1   0.6500  0.1067      0.47124        0.897
   12     12       1   0.5958  0.1107      0.41402        0.857
   15     11       1   0.5417  0.1131      0.35976        0.816
   18      9       1   0.4815  0.1154      0.30096        0.770
   19      8       1   0.4213  0.1156      0.24601        0.721
   20      7       1   0.3611  0.1137      0.19481        0.669
   22      6       1   0.3009  0.1095      0.14745        0.614
   25      5       1   0.2407  0.1028      0.10422        0.556
   28      4       1   0.1806  0.0931      0.06573        0.496
   30      3       1   0.1204  0.0792      0.03317        0.437
   40      2       1   0.0602  0.0581      0.00907        0.399
```

Modelos Não Paramétricos - KM

```
tempos<-c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
```

Modelos Não Paramétricos - KM

```
tempos←c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
cens←c(1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0)
```

Modelos Não Paramétricos - KM

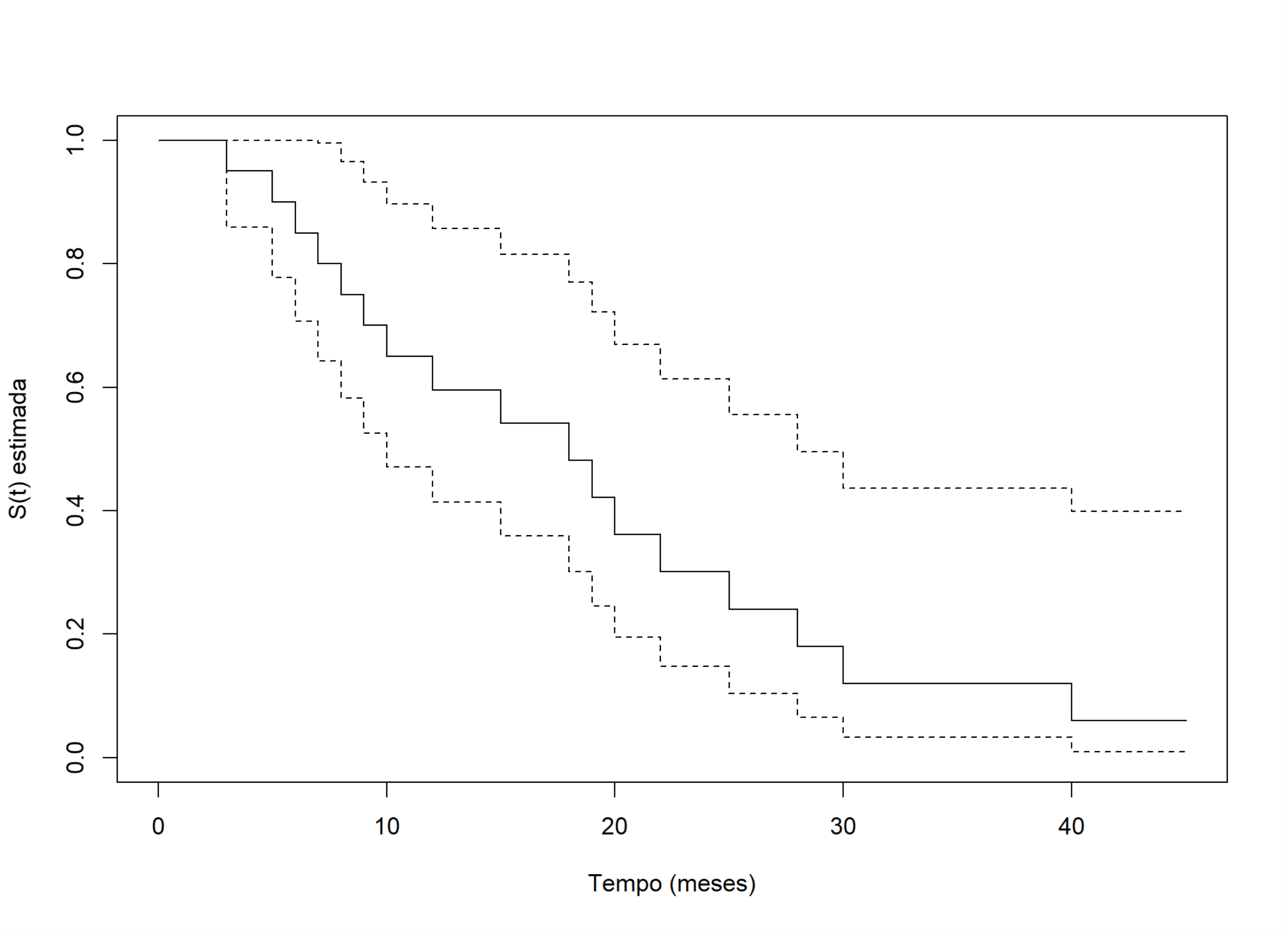
```
tempos←c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
cens←c(1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0)
dados ← data.frame(tempos=tempos, status=cens)
```

Modelos Não Paramétricos - KM

```
tempos←c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
cens←c(1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0)
dados ← data.frame(tempos=tempos, status=cens)
ekm ← survfit(Surv(tempos,status)~1, data=dados)
```

Modelos Não Paramétricos - KM

```
tempos←c(3,5,6,7,8,9,10,10,12,15,15,18,19,20,22,25,28,3
cens←c(1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,0)
dados ← data.frame(tempos=tempos, status=cens)
ekm ← survfit(Surv(tempos,status)~1, data=dados)
plot(ekm, xlab="Tempo (meses)",ylab="S(t) estimada")
```



Função taxa de falhas não paramétrica

```r
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
```

Função taxa de falhas não paramétrica

```r
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum <- summary(ajusteKM)
```

Função taxa de falhas não paramétrica

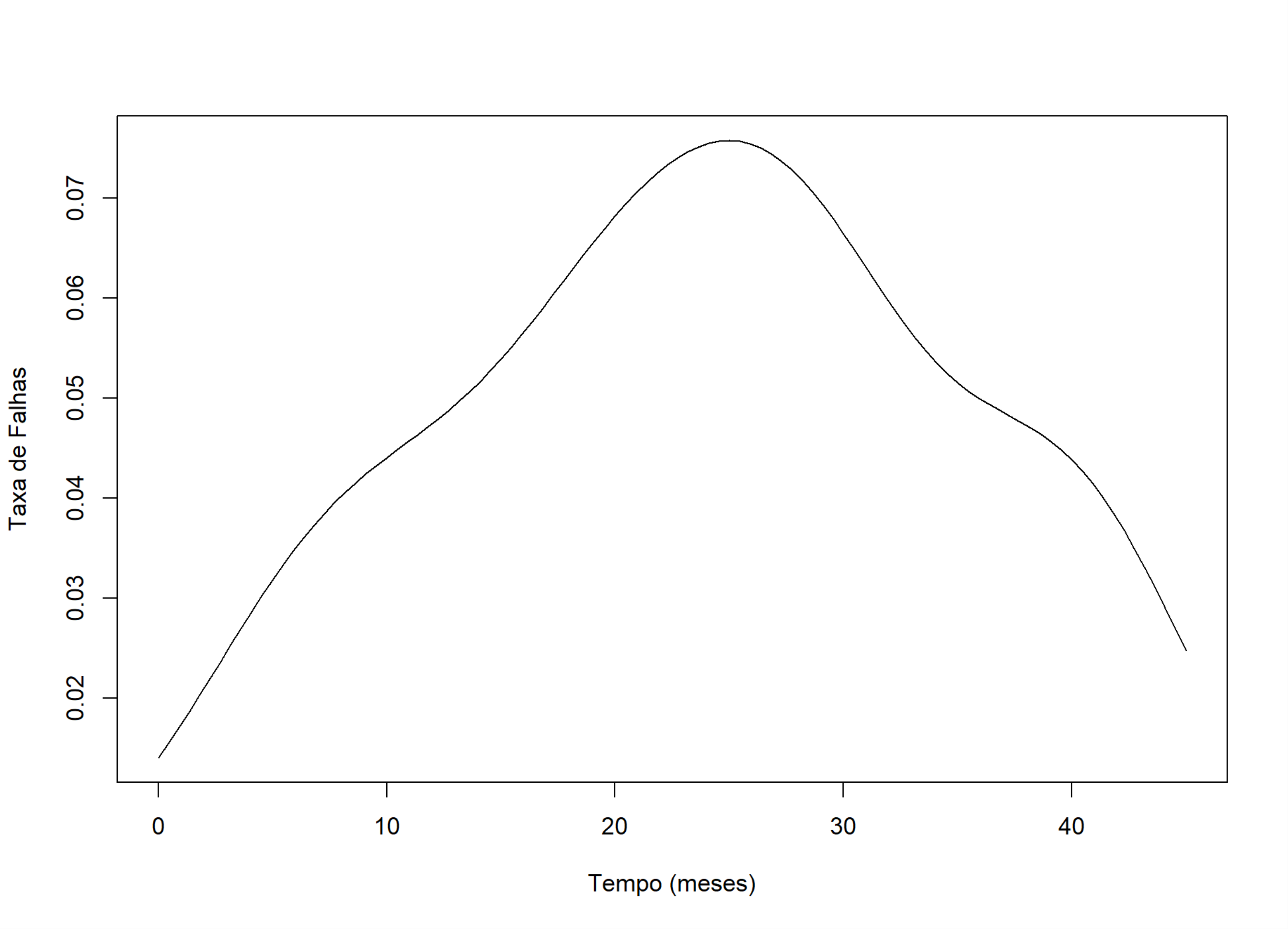
```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
```

Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
             kernel = "gaussian", n = 100, from = 0, to
```

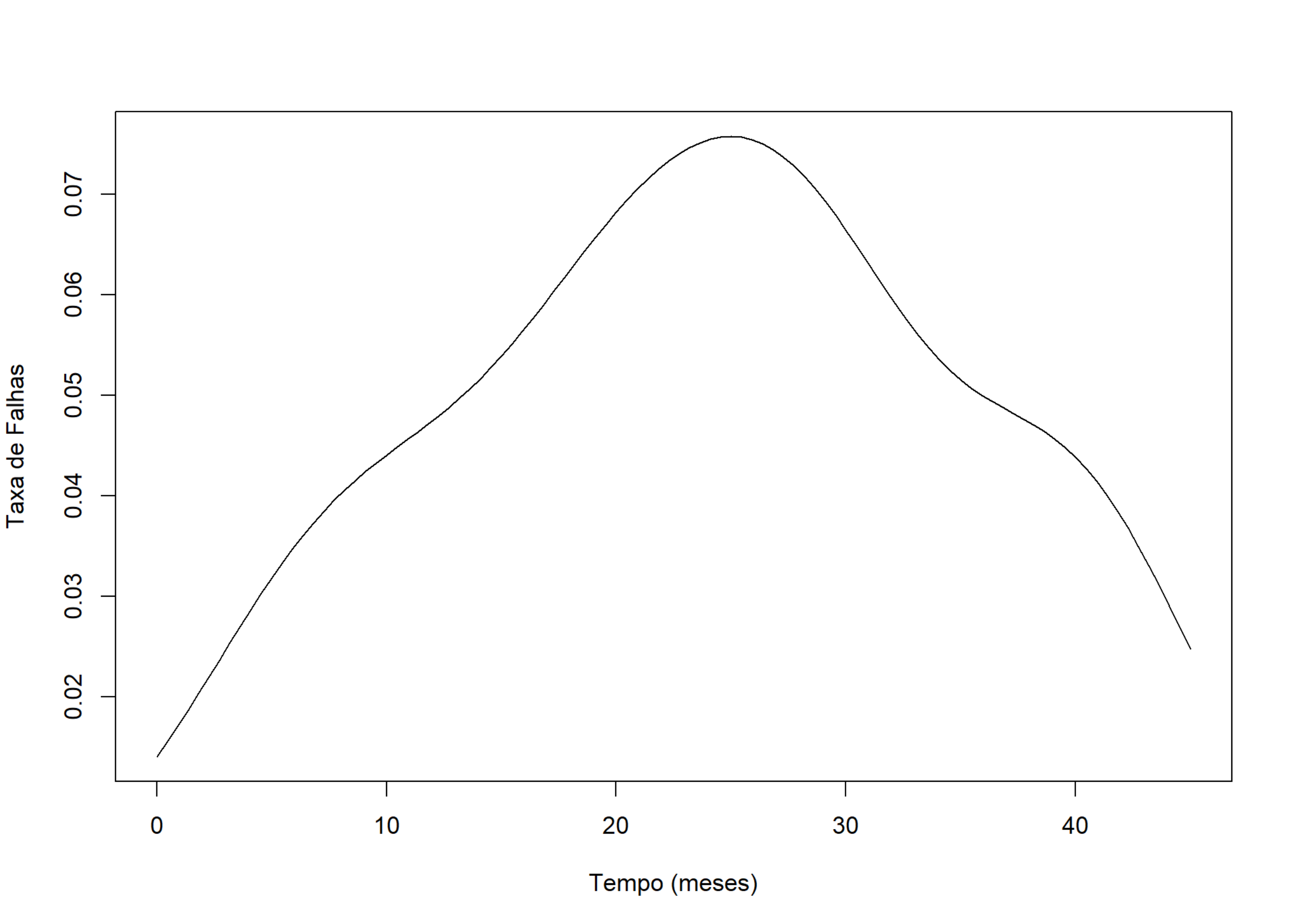
Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
```



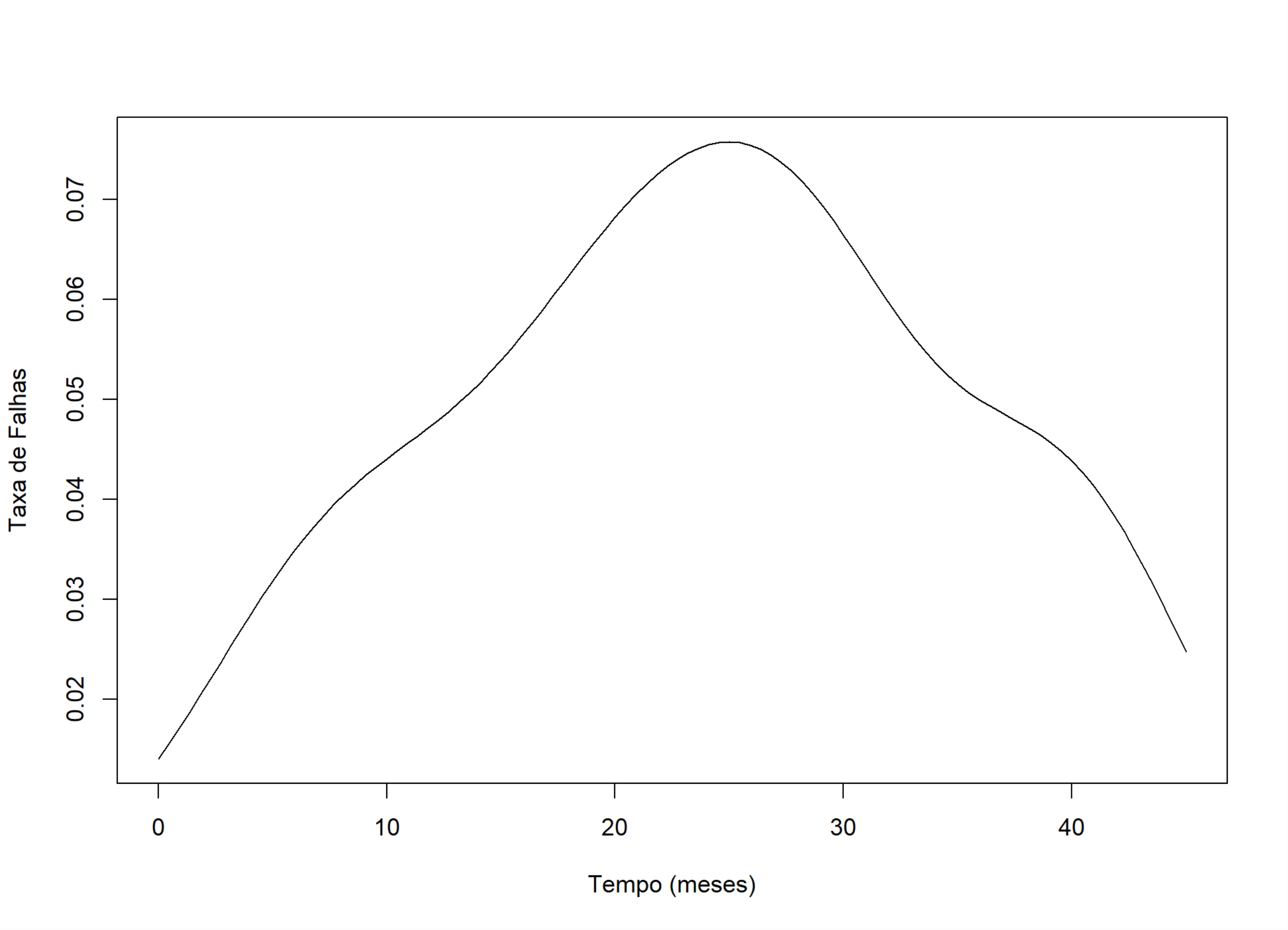
Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab

library(survey)
```



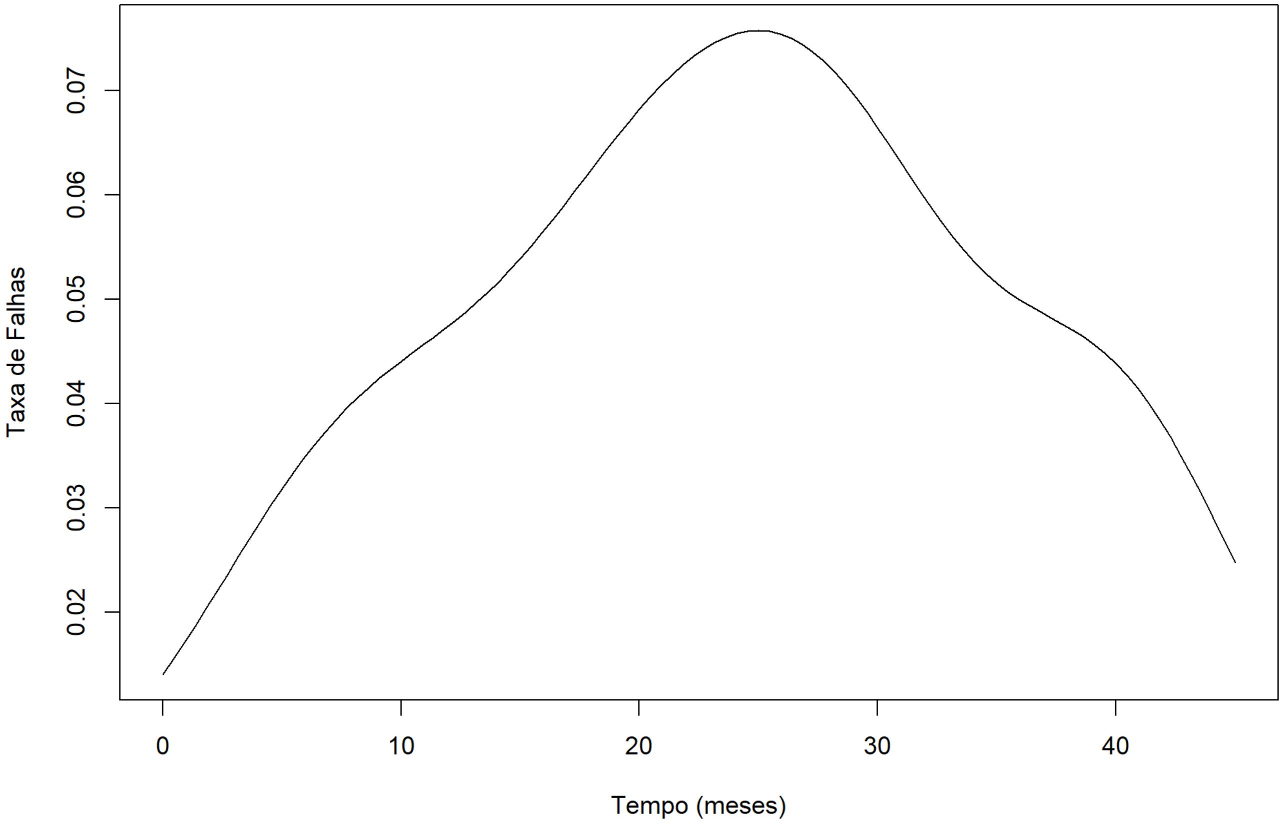
Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab

library(survey)
dados_p ← svydesign(ids = ~1, data=dados)
```



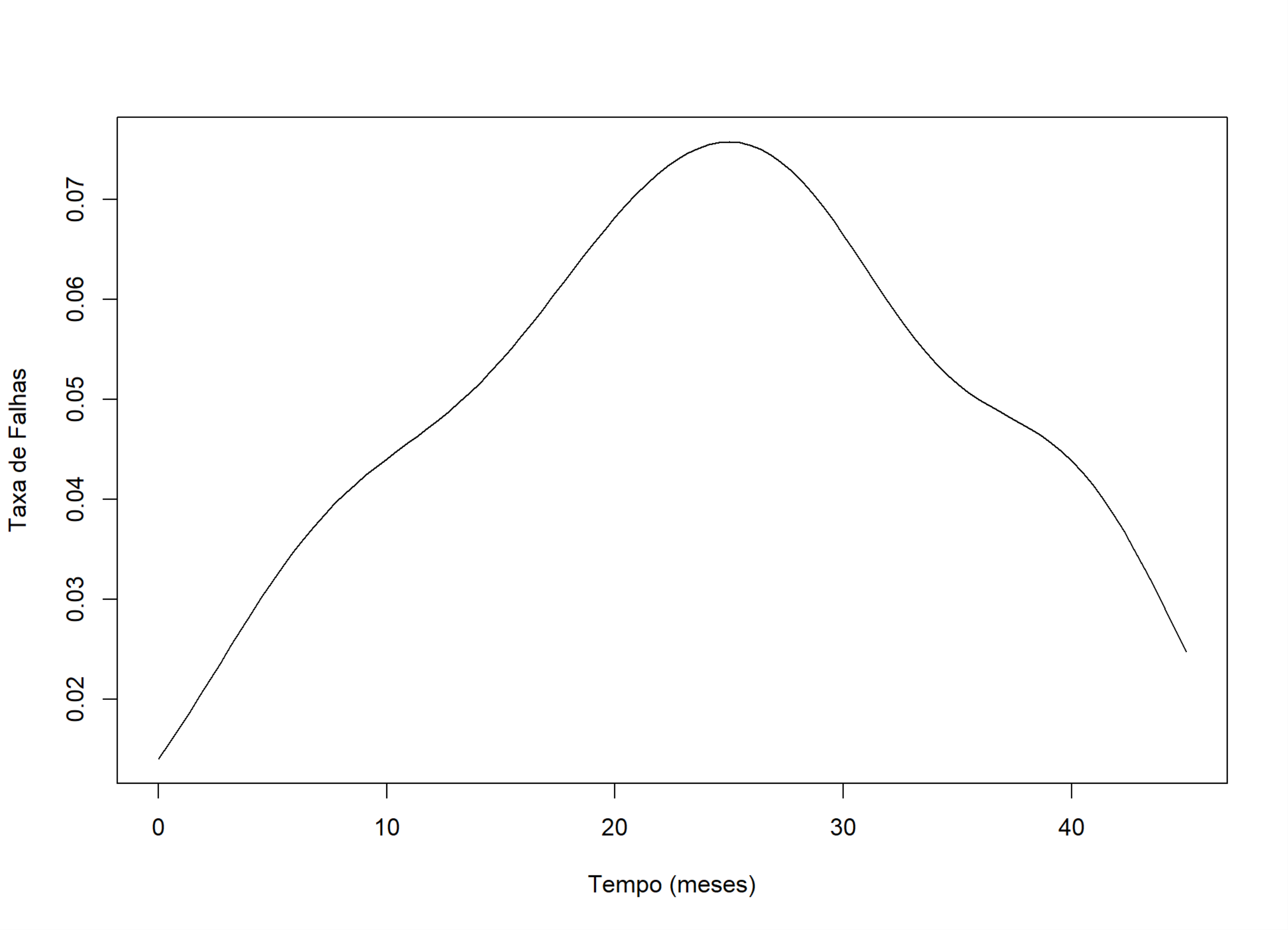
Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab

library(survey)
dados_p ← svydesign(ids = ~1, data=dados)
dados_svykm ← svykm(Surv(tempos, status) ~ 1, dados_p,
```



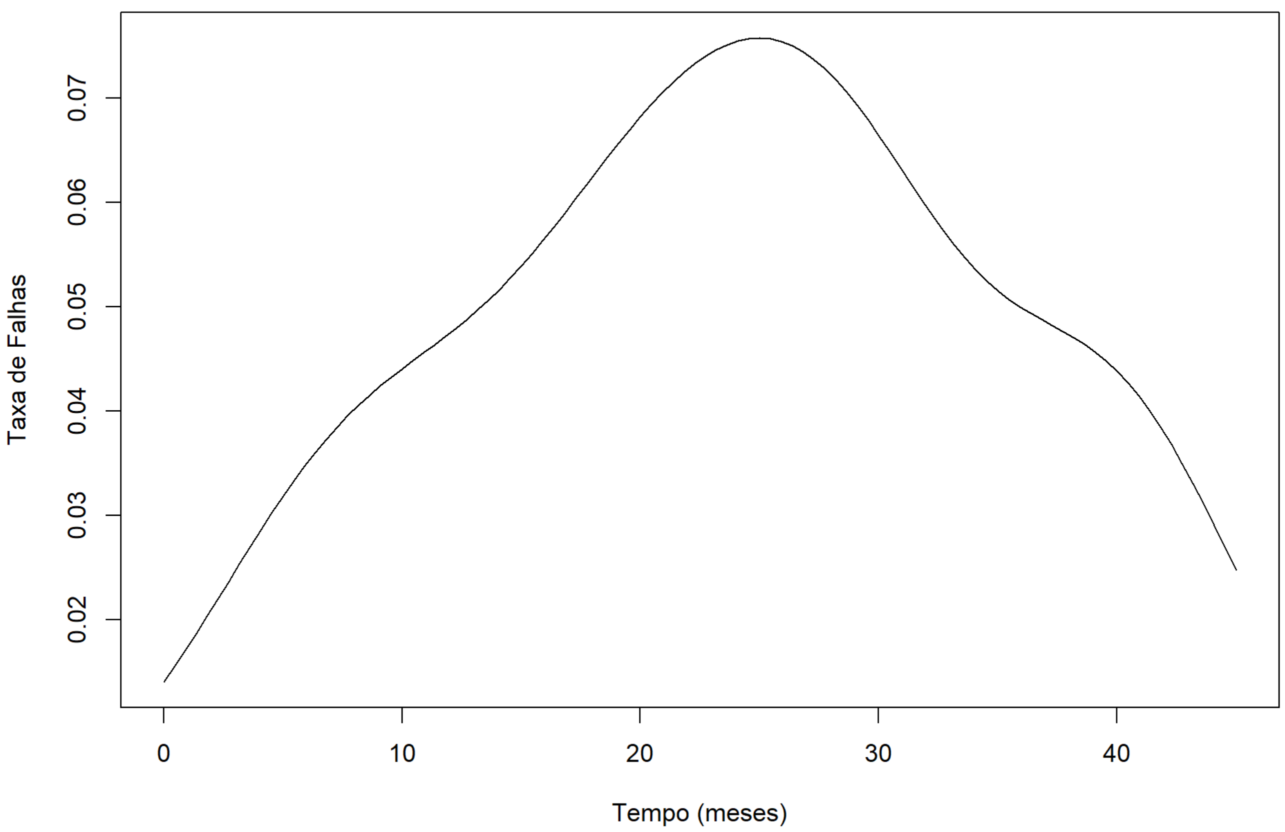
Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab

library(survey)
dados_p ← svydesign(ids = ~1, data=dados)
dados_svykm ← svykm(Surv(tempos, status) ~ 1, dados_p,
mediana.km ← quantile(dados_svykm, 0.5, ci=TRUE)
```



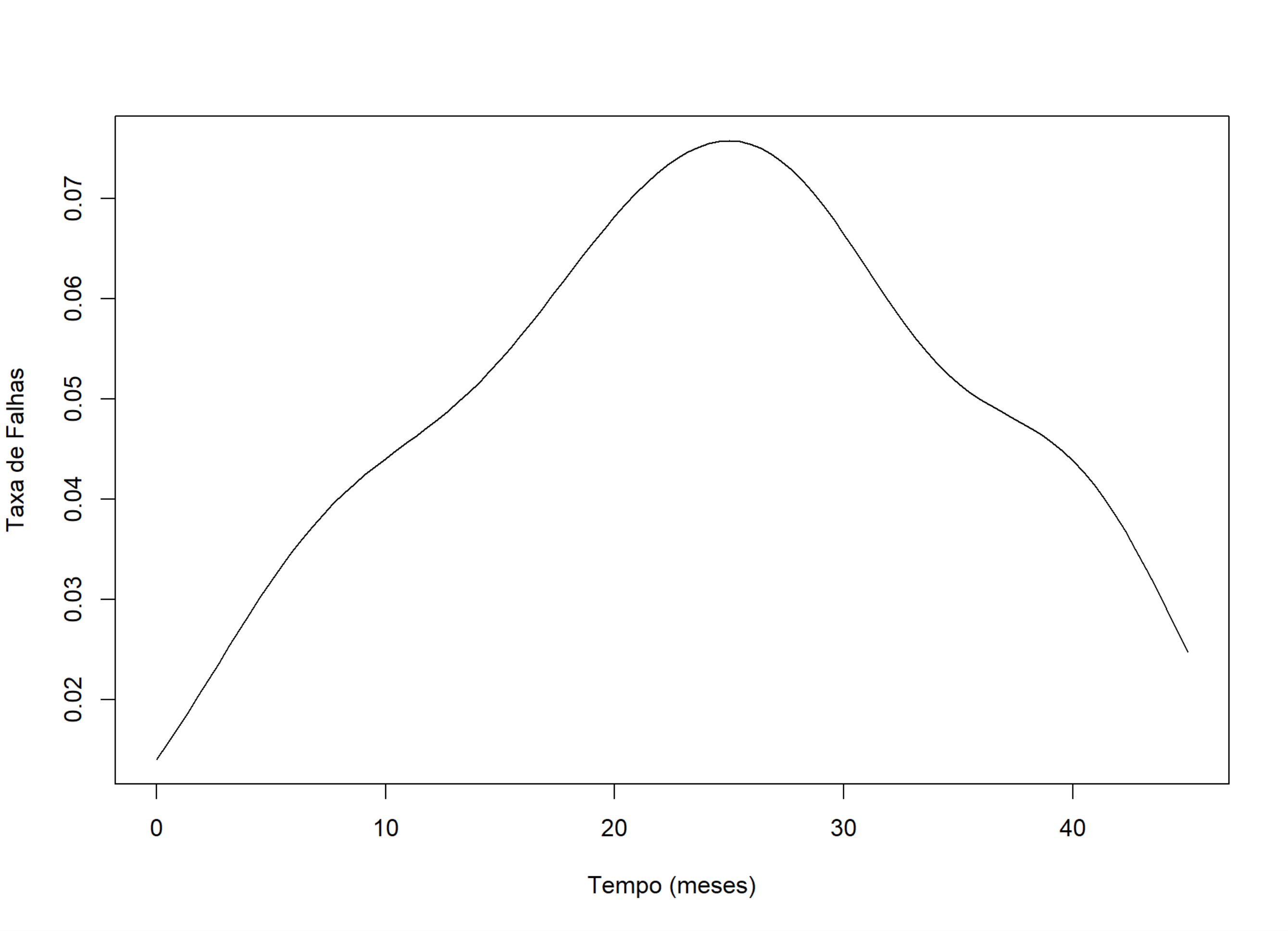
Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab

library(survey)
dados_p ← svydesign(ids = ~1, data=dados)
dados_svykm ← svykm(Surv(tempos, status) ~ 1, dados_p,
mediana.km ← quantile(dados_svykm, 0.5, ci=TRUE)
med.km ← c(mediana.km[1], attr(mediana.km, "ci")[1], at
```



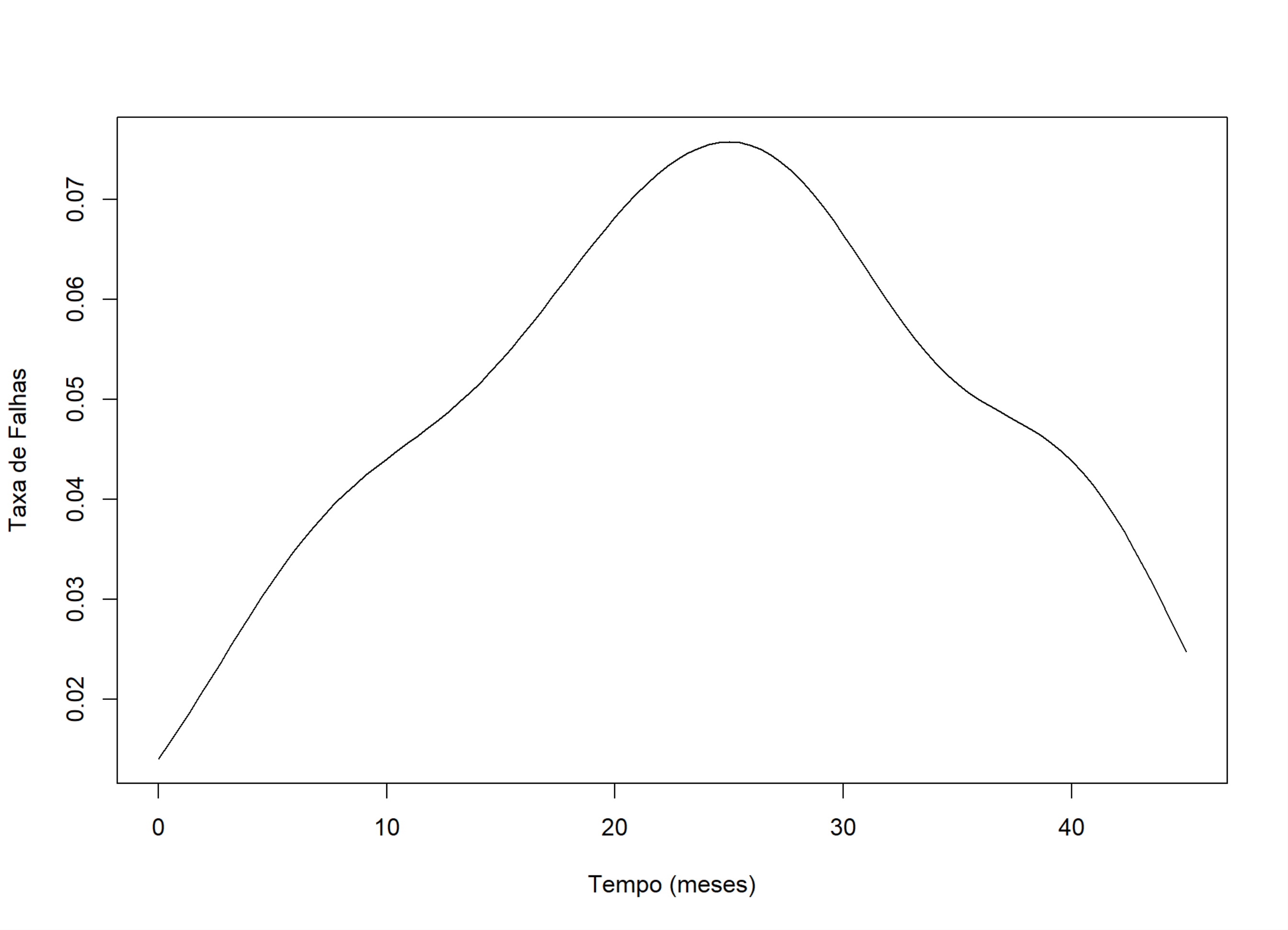
Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
             kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab

library(survey)
dados_p ← svydesign(ids = ~1, data=dados)
dados_svykm ← svykm(Surv(tempos, status) ~ 1, dados_p,
mediana.km ← quantile(dados_svykm, 0.5, ci=TRUE)
med.km ← c(mediana.km[1], attr(mediana.km, "ci")[1], at
names(med.km) ← c("mediana KM","ICI","ICS")
```



Função taxa de falhas não paramétrica

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab

library(survey)
dados_p ← svydesign(ids = ~1, data=dados)
dados_svykm ← svykm(Surv(tempos, status) ~ 1, dados_p,
mediana.km ← quantile(dados_svykm, 0.5, ci=TRUE)
med.km ← c(mediana.km[1], attr(mediana.km, "ci")[1], at
names(med.km) ← c("mediana KM","ICI","ICS")
med.km
```



```
mediana KM         ICI         ICS
        18          10          28
```

Modelos Paramétricos - Exponencial

```
ajustExp ← survreg(Surv(tempos,status)~1,
                    data=dados, dist='exponential')
```

Modelos Paramétricos - Exponencial

```r
ajustExp <- survreg(Surv(tempos,status)~1,
                    data=dados, dist='exponential')
ajustExp
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "exponential")

Coefficients:
(Intercept)
   3.016111

Scale fixed at 1

Loglik(model)= -68.3   Loglik(intercept only)= -68.3
n= 20
```

Modelos Paramétricos - Exponencial

```r
ajustExp <- survreg(Surv(tempos,status)~1,
                    data=dados, dist='exponential')
ajustExp
alfa <- exp(ajustExp$coefficients[1])
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "exponential")

Coefficients:
(Intercept)
   3.016111

Scale fixed at 1

Loglik(model)= -68.3   Loglik(intercept only)= -68.3
n= 20
```

Modelos Paramétricos - Exponencial

```r
ajustExp <- survreg(Surv(tempos,status)~1,
                    data=dados, dist='exponential')
ajustExp
alfa <- exp(ajustExp$coefficients[1])
alfa
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "exponential")

Coefficients:
(Intercept)
   3.016111

Scale fixed at 1

Loglik(model)= -68.3   Loglik(intercept only)= -68.3
n= 20

(Intercept)
   20.41176
```

Modelos Paramétricos - Exponencial

```
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
```

Modelos Paramétricos - Exponencial

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
```

Modelos Paramétricos - Exponencial

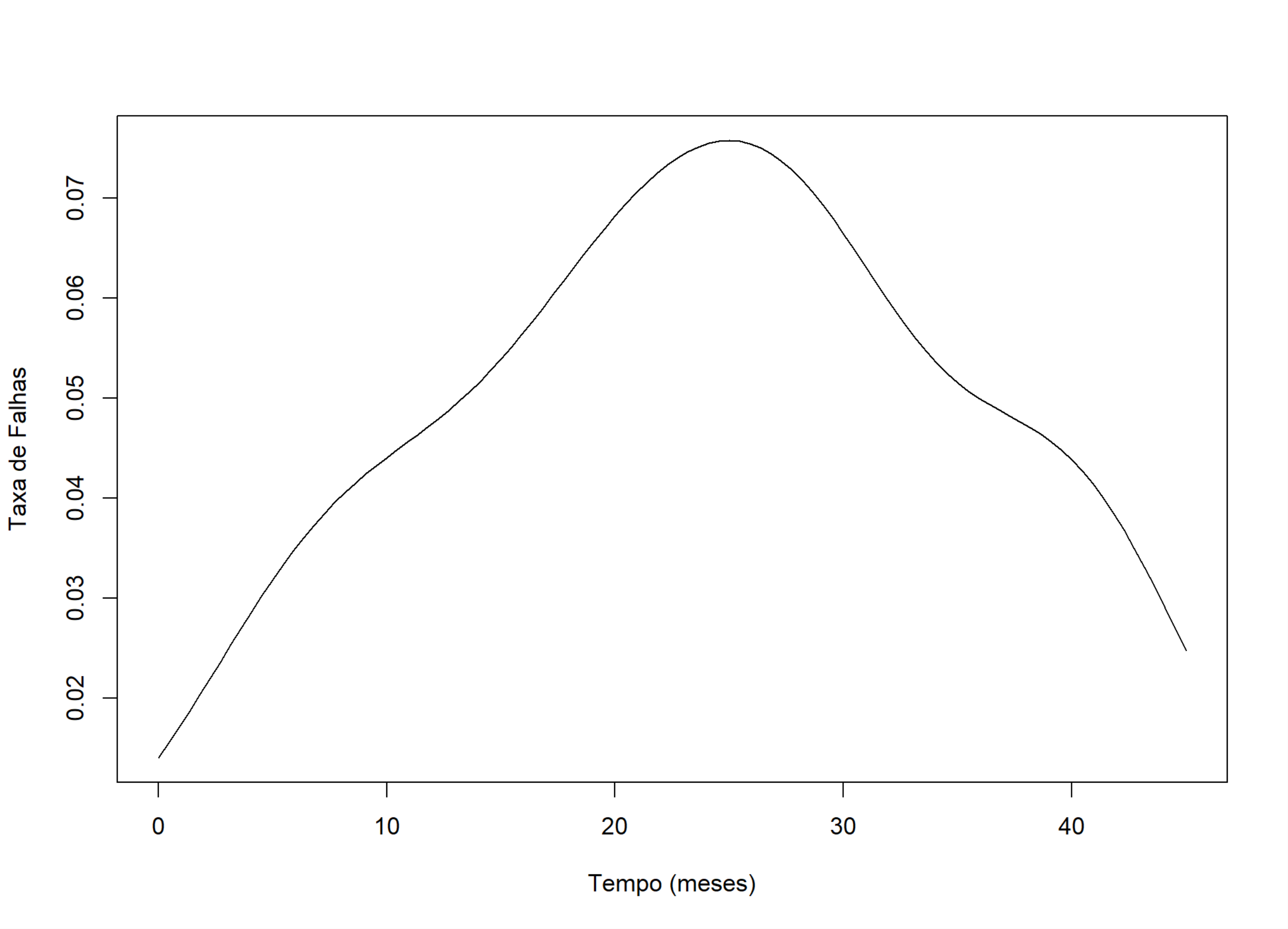
```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
```

Modelos Paramétricos - Exponencial

```r
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum <- summary(ajusteKM)
pesos <- ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h <- density(ajusteKM_sum$time, weights = pesos,
             kernel = "gaussian", n = 100, from = 0, to
```

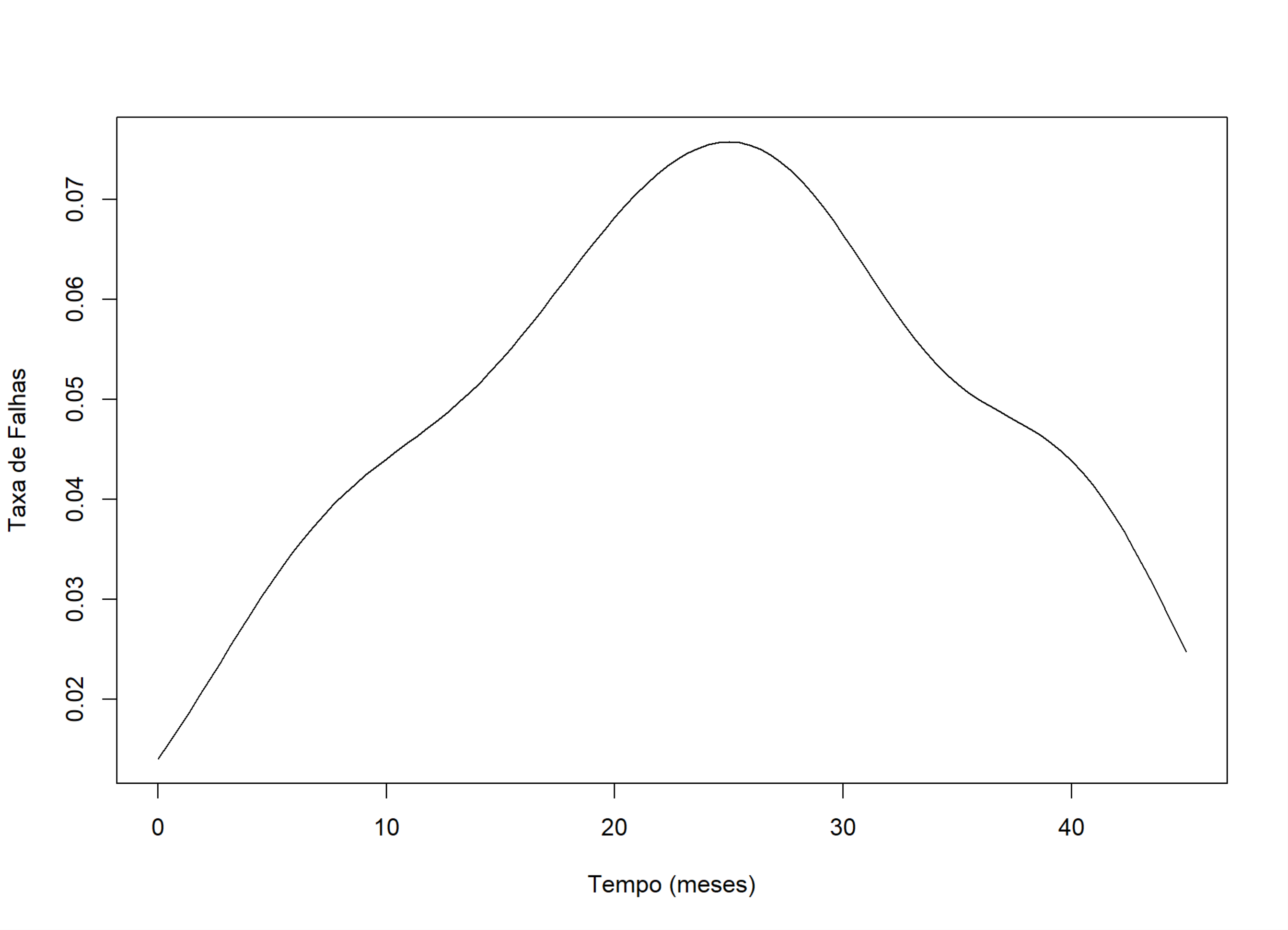
Modelos Paramétricos - Exponencial

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
```



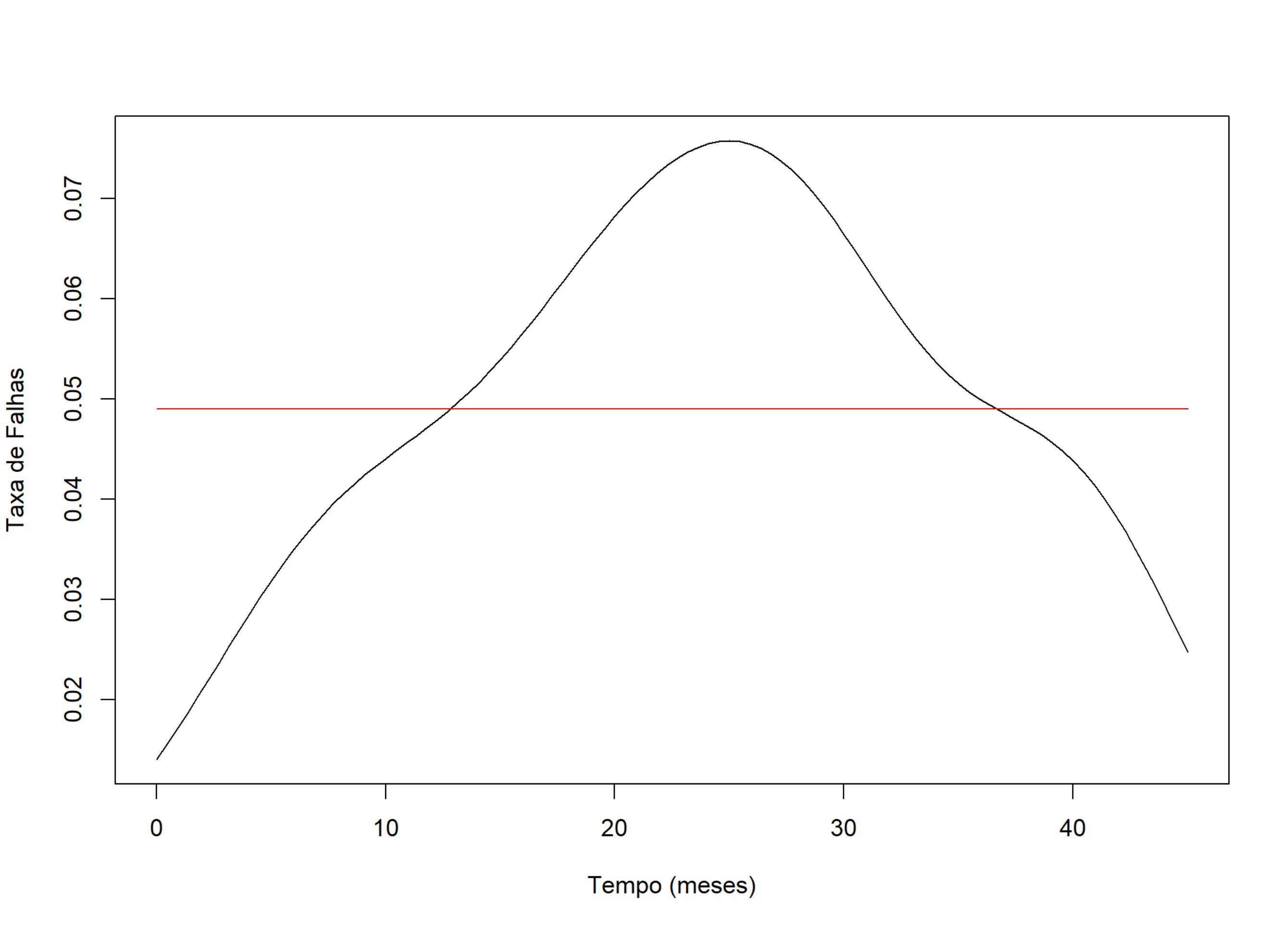
Modelos Paramétricos - Exponencial

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
expHaz ← function(x, rate) {
  dexp(x, rate=rate)/
    pexp(x, rate=rate, lower.tail=F)
}
```



Modelos Paramétricos - Exponencial

```r
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
expHaz ← function(x, rate) {
  dexp(x, rate=rate)/
    pexp(x, rate=rate, lower.tail=F)
}
curve(expHaz(x, rate=1/alfa), from=0, to=45,
      ylab="h(t)", xlab="Tempo",
      col="red", ylim=c(0,1),
      add=TRUE)
```



Modelos Paramétricos - Exponencial

```
coef ← ajustExp$coefficients[1]
```

Modelos Paramétricos - Exponencial

```
coef ← ajustExp$coefficients[1]
var ← ajustExp$var
```

Modelos Paramétricos - Exponencial

```
coef ← ajustExp$coefficients[1]
var ← ajustExp$var
IC.exp ← c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
```

Modelos Paramétricos - Exponencial

```r
coef <- ajustExp$coefficients[1]
var <- ajustExp$var
IC.exp <- c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) <- c("media","ICI","ICS")
```

Modelos Paramétricos - Exponencial

```r
coef <- ajustExp$coefficients[1]
var <- ajustExp$var
IC.exp <- c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) <- c("media","ICI","ICS")
IC.exp
```

```
   media      ICI      ICS
20.41176 12.68908 32.83455
```

Modelos Paramétricos - Exponencial

```r
coef <- ajustExp$coefficients[1]
var <- ajustExp$var
IC.exp <- c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) <- c("media","ICI","ICS")
IC.exp

est_med <- predict(ajustExp,type="uquantile",p=0.5,se.fi
```

```
   media      ICI      ICS 
20.41176 12.68908 32.83455 
```

Modelos Paramétricos - Exponencial

```r
coef <- ajustExp$coefficients[1]
var <- ajustExp$var
IC.exp <- c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) <- c("media","ICI","ICS")
IC.exp

est_med <- predict(ajustExp,type="uquantile",p=0.5,se.fi
est_med1 <- est_med$fit[1]
```

```
   media      ICI      ICS 
20.41176 12.68908 32.83455 
```

Modelos Paramétricos - Exponencial

```r
coef ← ajustExp$coefficients[1]
var ← ajustExp$var
IC.exp ← c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) ← c("media","ICI","ICS")
IC.exp

est_med ← predict(ajustExp,type="uquantile",p=0.5,se.fi
est_med1 ← est_med$fit[1]
est_med1.ep ← est_med$se.fit[1]
```

```
   media      ICI      ICS 
20.41176 12.68908 32.83455 
```

Modelos Paramétricos - Exponencial

```r
coef ← ajustExp$coefficients[1]
var ← ajustExp$var
IC.exp ← c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) ← c("media","ICI","ICS")
IC.exp

est_med ← predict(ajustExp,type="uquantile",p=0.5,se.fi
est_med1 ← est_med$fit[1]
est_med1.ep ← est_med$se.fit[1]
exp(est_med1)
```

```
   media      ICI      ICS 
20.41176 12.68908 32.83455 

[1] 14.14836
```

Modelos Paramétricos - Exponencial

```r
coef <- ajustExp$coefficients[1]
var <- ajustExp$var
IC.exp <- c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) <- c("media","ICI","ICS")
IC.exp

est_med <- predict(ajustExp,type="uquantile",p=0.5,se.fi
est_med1 <- est_med$fit[1]
est_med1.ep <- est_med$se.fit[1]
exp(est_med1)
IC.mediana_exp <- c(exp(est_med1),exp(est_med1-1.96*est_
exp(est_med1+1.96*est_med1.ep))
```

```
   media      ICI      ICS 
20.41176 12.68908 32.83455 

[1] 14.14836
```

Modelos Paramétricos - Exponencial

```
coef ← ajustExp$coefficients[1]
var ← ajustExp$var
IC.exp ← c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) ← c("media","ICI","ICS")
IC.exp

est_med ← predict(ajustExp,type="uquantile",p=0.5,se.fi
est_med1 ← est_med$fit[1]
est_med1.ep ← est_med$se.fit[1]
exp(est_med1)
IC.mediana_exp ← c(exp(est_med1),exp(est_med1-1.96*est_
exp(est_med1+1.96*est_med1.ep))
names(IC.mediana_exp) ← c("mediana Exp","ICI","ICS")
```

```
   media      ICI      ICS
20.41176 12.68908 32.83455

[1] 14.14836
```

Modelos Paramétricos - Exponencial

```
coef ← ajustExp$coefficients[1]
var ← ajustExp$var
IC.exp ← c(alfa,exp(coef-1.96*sqrt(var)),
            exp(coef+1.96*sqrt(var)))
names(IC.exp) ← c("media","ICI","ICS")
IC.exp

est_med ← predict(ajustExp,type="uquantile",p=0.5,se.fi
est_med1 ← est_med$fit[1]
est_med1.ep ← est_med$se.fit[1]
exp(est_med1)
IC.mediana_exp ← c(exp(est_med1),exp(est_med1-1.96*est_
exp(est_med1+1.96*est_med1.ep))
names(IC.mediana_exp) ← c("mediana Exp","ICI","ICS")
IC.mediana_exp
```

```
   media      ICI      ICS
20.41176 12.68908 32.83455

[1] 14.14836

mediana Exp         ICI         ICS
  14.148357    8.795399   22.759175
```

Modelos Paramétricos - Exponencial

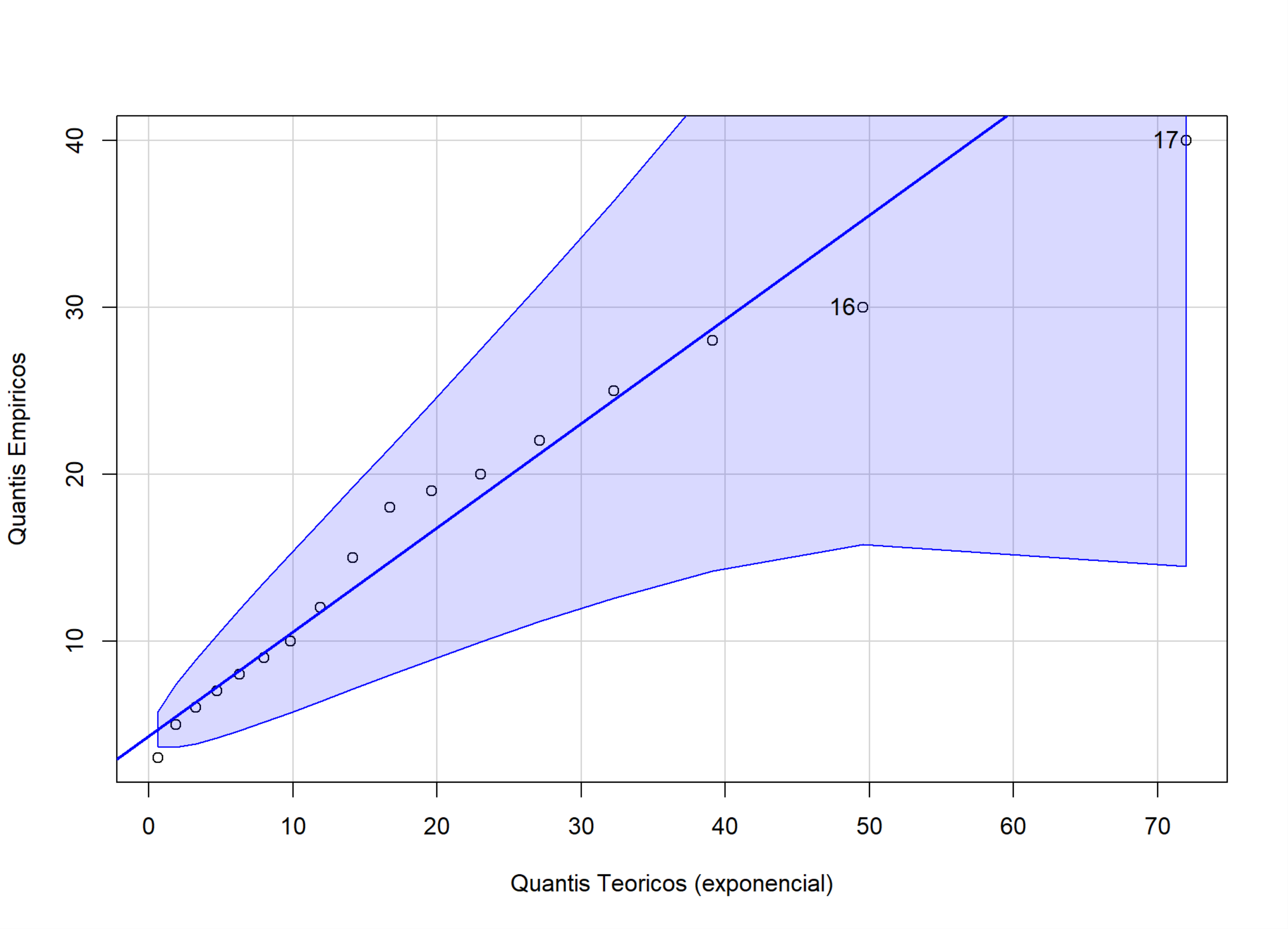
```r
library(car)
```

Modelos Paramétricos - Exponencial

```r
library(car)
n.censurado <- dados$tempos[dados$status == 1]
```

Modelos Paramétricos - Exponencial

```r
library(car)
n.censurado ← dados$tempos[dados$status == 1]
qqPlot(n.censurado,
dist = "exp",
rate = 1/alfa,
xlab = "Quantis Teoricos (exponencial)",
ylab = "Quantis Empiricos")
```



```
[1] 17 16
```

Modelos Paramétricos - Weibull

```
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
```

Modelos Paramétricos - Weibull

```
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
ajustWei
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "weibull")

Coefficients:
(Intercept)
   3.060529

Scale= 0.647922

Loglik(model)= -66.1   Loglik(intercept only)= -66.1
n= 20
```

Modelos Paramétricos - Weibull

```
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
ajustWei
alfaw ← exp(ajustWei$coefficients[1])
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "weibull")

Coefficients:
(Intercept)
   3.060529

Scale= 0.647922

Loglik(model)= -66.1   Loglik(intercept only)= -66.1
n= 20
```

Modelos Paramétricos - Weibull

```r
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
ajustWei
alfaw ← exp(ajustWei$coefficients[1])
alfaw
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "weibull")

Coefficients:
(Intercept)
   3.060529

Scale= 0.647922

Loglik(model)= -66.1   Loglik(intercept only)= -66.1
n= 20

(Intercept)
   21.33885
```

Modelos Paramétricos - Weibull

```r
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
ajustWei
alfaw ← exp(ajustWei$coefficients[1])
alfaw
betaw ← 1/ajustWei$scale
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "weibull")

Coefficients:
(Intercept)
   3.060529

Scale= 0.647922

Loglik(model)= -66.1   Loglik(intercept only)= -66.1
n= 20

(Intercept)
   21.33885
```

Modelos Paramétricos - Weibull

```r
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
ajustWei
alfaw ← exp(ajustWei$coefficients[1])
alfaw
betaw ← 1/ajustWei$scale
betaw
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "weibull")

Coefficients:
(Intercept)
   3.060529

Scale= 0.647922

Loglik(model)= -66.1   Loglik(intercept only)= -66.1
n= 20

(Intercept)
   21.33885

[1] 1.543396
```

Modelos Paramétricos - Weibull

```r
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
```

Modelos Paramétricos - Weibull

```
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum <- summary(ajusteKM)
```

Modelos Paramétricos - Weibull

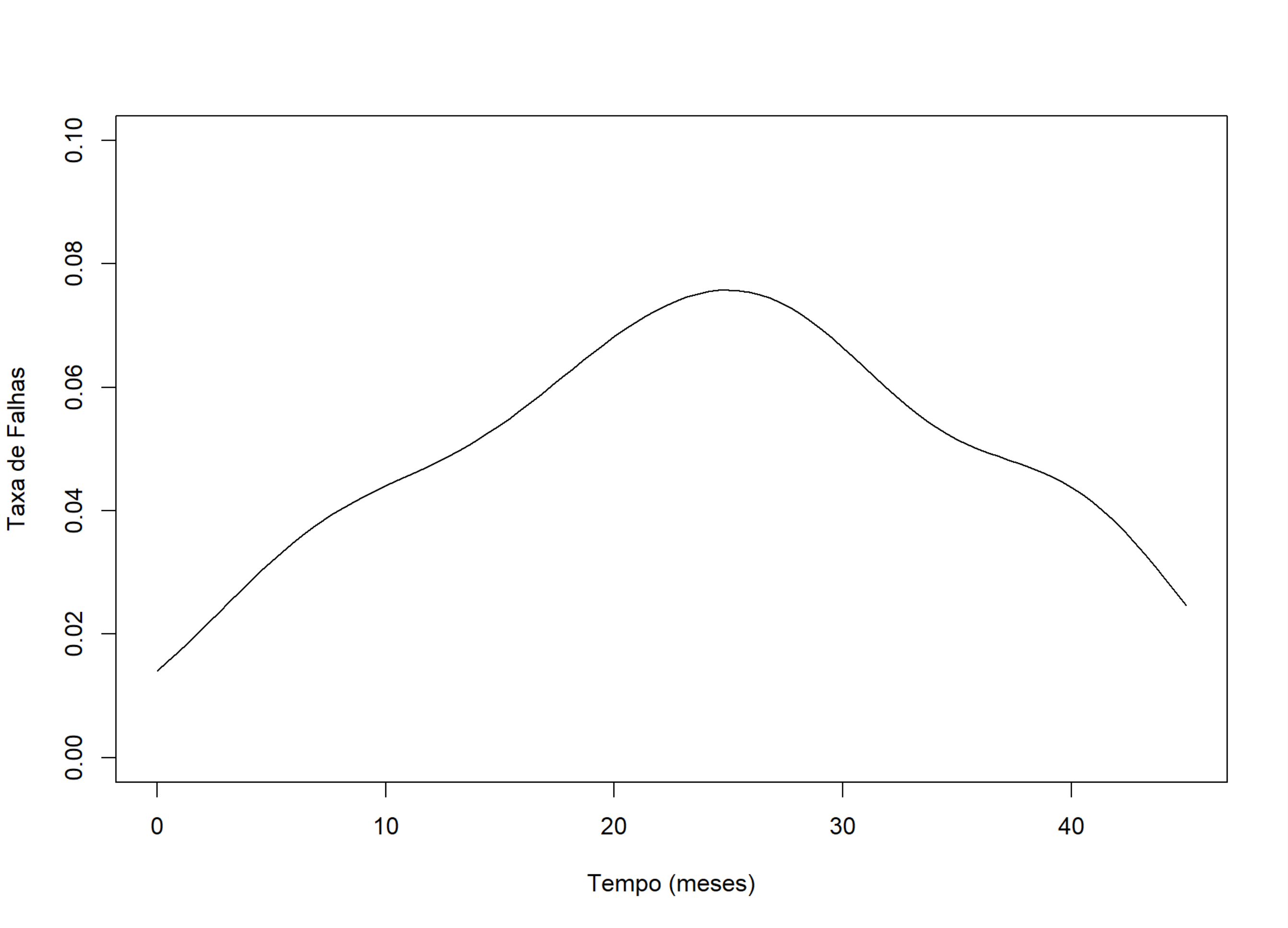
```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
```

Modelos Paramétricos - Weibull

```r
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
             kernel = "gaussian", n = 100, from = 0, to
```

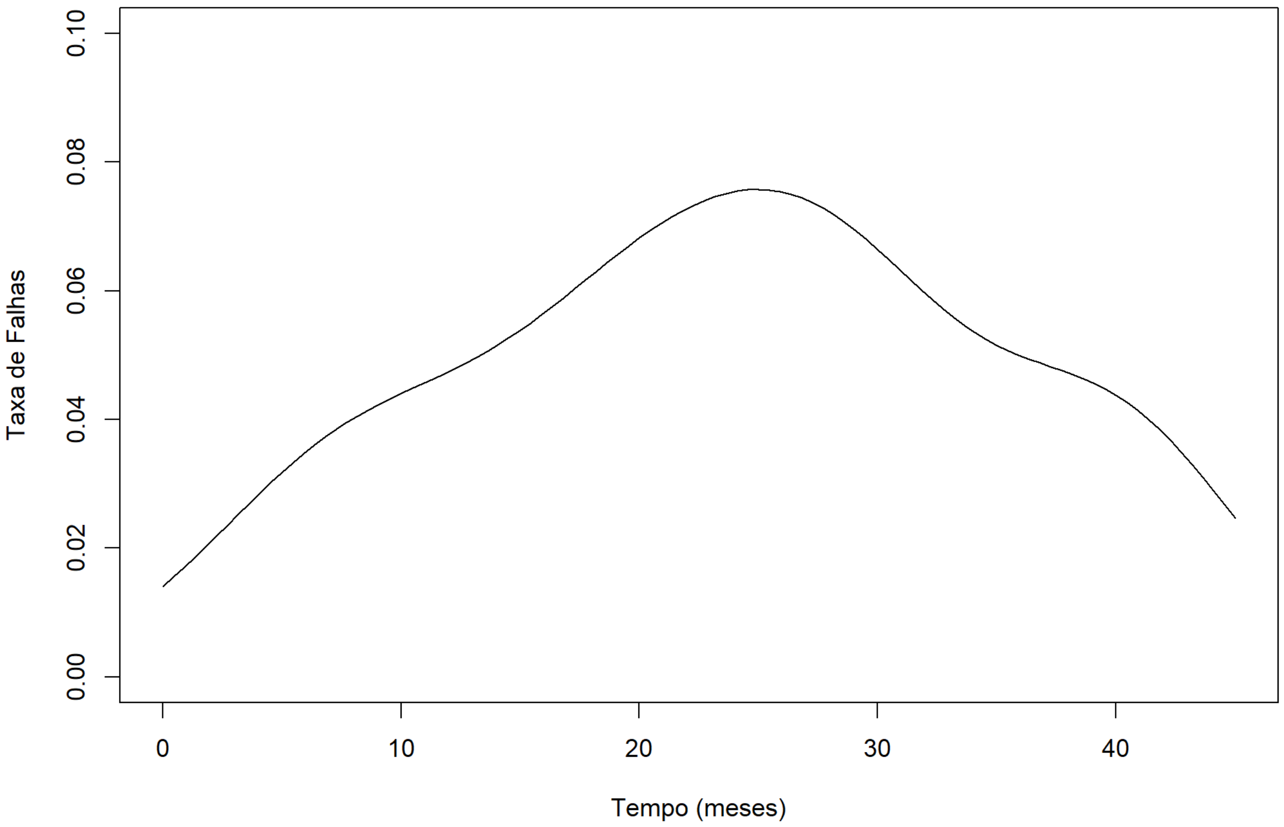
Modelos Paramétricos - Weibull

```
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum <- summary(ajusteKM)
pesos <- ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h <- density(ajusteKM_sum$time, weights = pesos,
          kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.10))
```



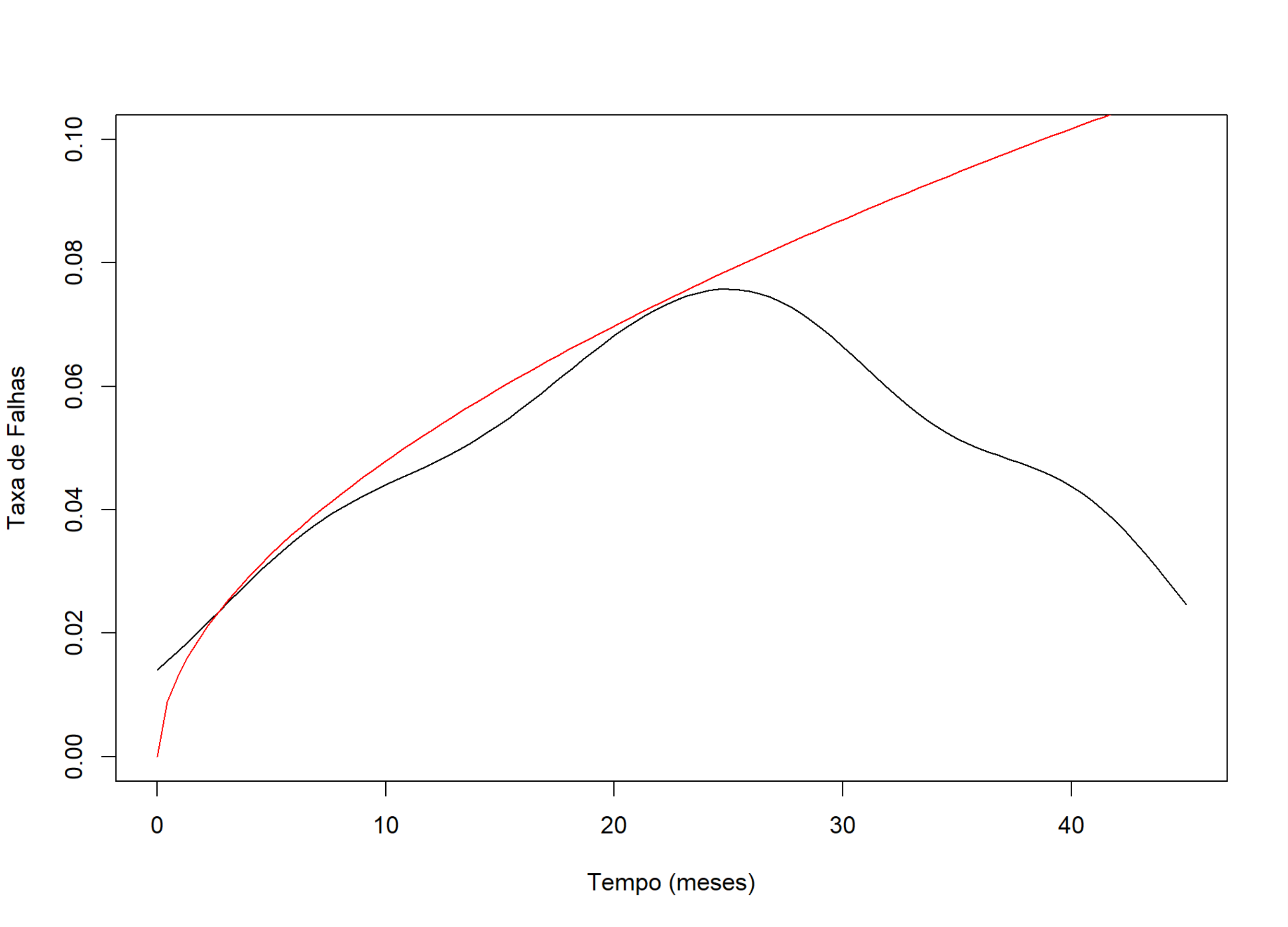
Modelos Paramétricos - Weibull

```r
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum <- summary(ajusteKM)
pesos <- ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h <- density(ajusteKM_sum$time, weights = pesos,
                kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.10))
weibHaz <- function(x, shape, scale) {
  dweibull(x, shape=shape, scale=scale)/
  pweibull(x, shape=shape, scale=scale,
           lower.tail=F)
}
```



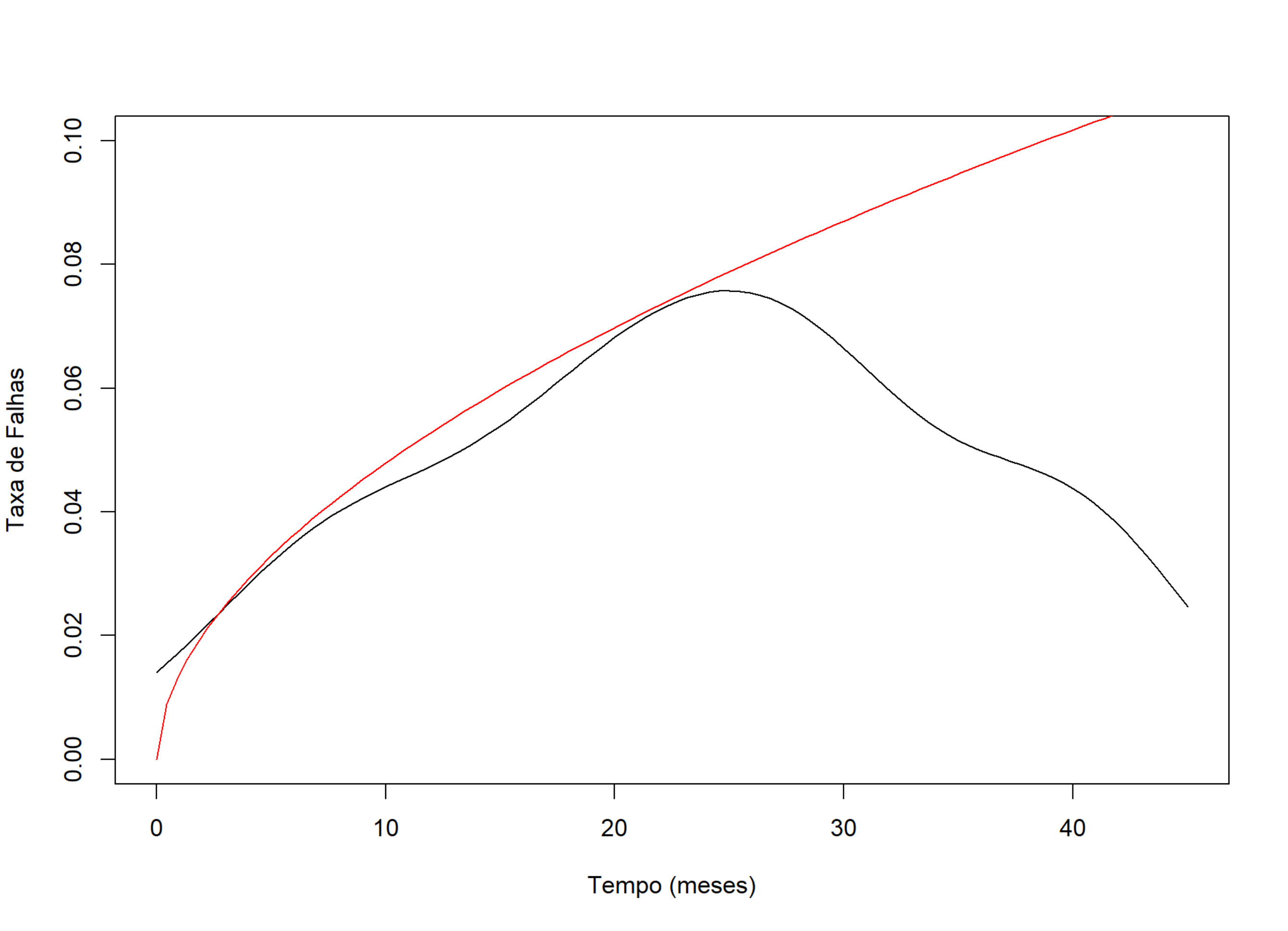
Modelos Paramétricos - Weibull

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
                kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
      ylim=c(0,0.10))
weibHaz ← function(x, shape, scale) {
  dweibull(x, shape=shape, scale=scale)/
  pweibull(x, shape=shape, scale=scale,
             lower.tail=F)
}
curve(weibHaz(x, shape=betaw, scale=alfaw),
       from=0, to=45,
       ylab="h(t)", xlab="Tempo", col="red",
       add=TRUE)
```



Modelos Paramétricos - Weibull

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.10))
weibHaz ← function(x, shape, scale) {
  dweibull(x, shape=shape, scale=scale)/
  pweibull(x, shape=shape, scale=scale,
           lower.tail=F)
}
curve(weibHaz(x, shape=betaw, scale=alfaw),
      from=0, to=45,
      ylab="h(t)", xlab="Tempo", col="red",
      add=TRUE)
```



Modelos Paramétricos - Weibull

```
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
```

Modelos Paramétricos - Weibull

```
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
estmedw ← predict(ajustWei,type="uquantile",p=0.5,se.fi
```

Modelos Paramétricos - Weibull

```
ajustWei <- survreg(Surv(tempos,status)~1, data=dados, d
estmedw <- predict(ajustWei,type="uquantile",p=0.5,se.fi
estmedw1 <- estmedw$fit[1]
```

Modelos Paramétricos - Weibull

```
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
estmedw ← predict(ajustWei,type="uquantile",p=0.5,se.fi
estmedw1 ← estmedw$fit[1]
estmedw1.ep ← estmedw$se.fit[1]
```

Modelos Paramétricos - Weibull

```
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
estmedw ← predict(ajustWei,type="uquantile",p=0.5,se.fi
estmedw1 ← estmedw$fit[1]
estmedw1.ep ← estmedw$se.fit[1]
exp(estmedw1)
```

```
[1] 16.82823
```

Modelos Paramétricos - Weibull

```
ajustWei ← survreg(Surv(tempos,status)~1, data=dados, d
estmedw ← predict(ajustWei,type="uquantile",p=0.5,se.fi
estmedw1 ← estmedw$fit[1]
estmedw1.ep ← estmedw$se.fit[1]
exp(estmedw1)
IC.mediana_w ← c(exp(estmedw1),exp(estmedw1-1.96*estmed
exp(estmedw1+1.96*estmedw1.ep))
```

```
[1] 16.82823
```

Modelos Paramétricos - Weibull

```r
ajustWei <- survreg(Surv(tempos,status)~1, data=dados, d
estmedw <- predict(ajustWei,type="uquantile",p=0.5,se.fi
estmedw1 <- estmedw$fit[1]
estmedw1.ep <- estmedw$se.fit[1]
exp(estmedw1)
IC.mediana_w <- c(exp(estmedw1),exp(estmedw1-1.96*estmed
exp(estmedw1+1.96*estmedw1.ep))
names(IC.mediana_w) <- c("mediana Wei","ICI","ICS")
```

```
[1] 16.82823
```

Modelos Paramétricos - Weibull

```r
ajustWei <- survreg(Surv(tempos,status)~1, data=dados, d
estmedw <- predict(ajustWei,type="uquantile",p=0.5,se.fi
estmedw1 <- estmedw$fit[1]
estmedw1.ep <- estmedw$se.fit[1]
exp(estmedw1)
IC.mediana_w <- c(exp(estmedw1),exp(estmedw1-1.96*estmed
exp(estmedw1+1.96*estmedw1.ep))
names(IC.mediana_w) <- c("mediana Wei","ICI","ICS")
IC.mediana_w
```

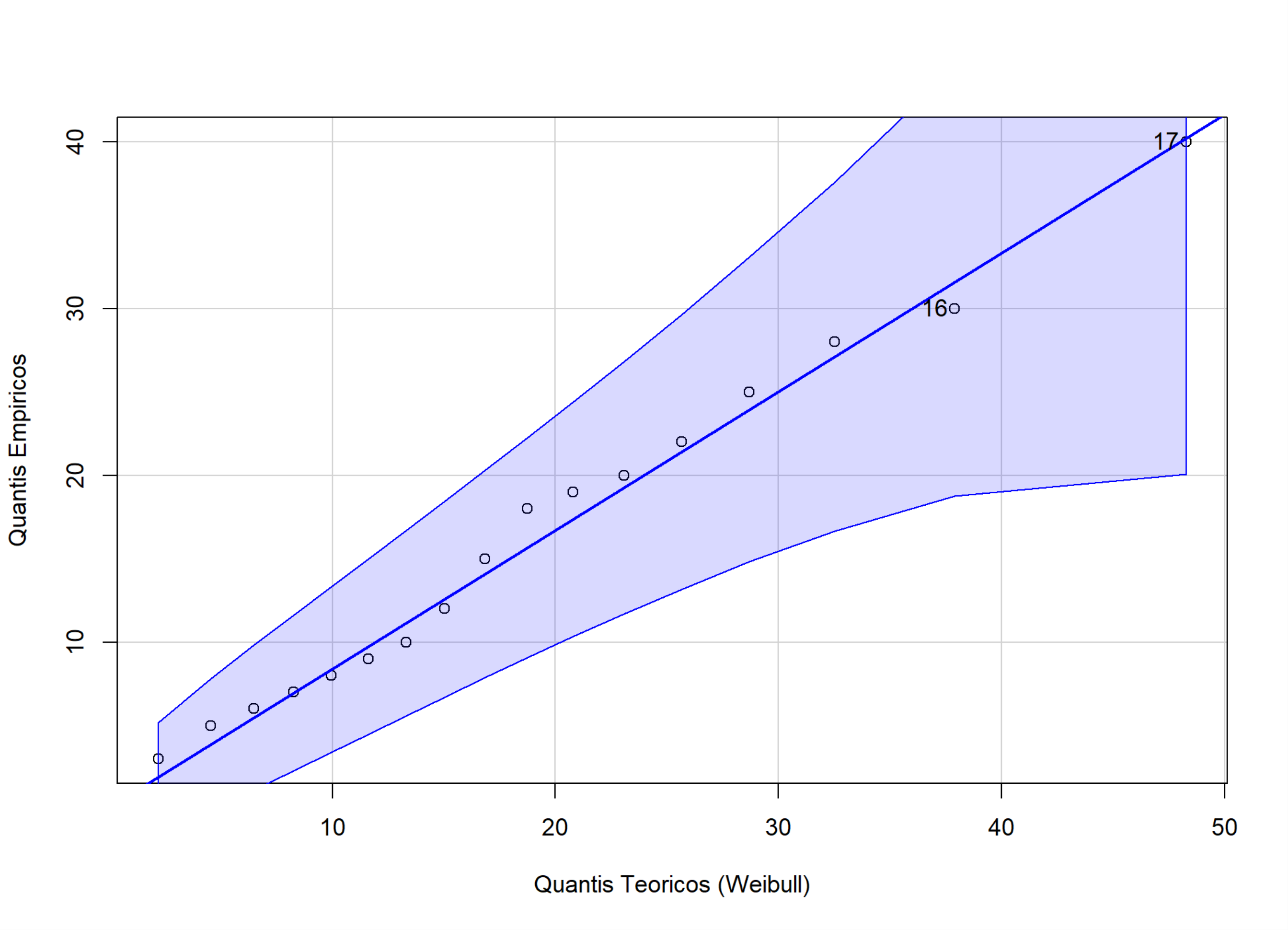
```
[1] 16.82823

mediana Wei         ICI         ICS 
   16.82823    11.95670    23.68456 
```

Modelos Paramétricos - Weibull

```r
n.censurado <- dados$tempos[dados$status == 1]
```

Modelos Paramétricos - Weibull

```
n.censurado <- dados$tempos[dados$status == 1]
qqPlot(n.censurado,
dist = "weibull",
shape = betaw,
scale = alfaw,
xlab = "Quantis Teoricos (Weibull)",
ylab = "Quantis Empiricos")
```



```
[1] 17 16
```

Modelos Paramétricos - Lognormal

```
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
```

Modelos Paramétricos - Lognormal

```
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLog
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "lognorm")

Coefficients:
(Intercept) 
   2.717176 

Scale= 0.7648167 

Loglik(model)= -65.7   Loglik(intercept only)= -65.7
n= 20 
```

Modelos Paramétricos - Lognormal

```r
ajustLog <- survreg(Surv(tempos, status)~1, data=dados,
ajustLog
mulog <- ajustLog$icoef[1]
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "lognorm")

Coefficients:
(Intercept)
   2.717176

Scale= 0.7648167

Loglik(model)= -65.7   Loglik(intercept only)= -65.7
n= 20
```

Modelos Paramétricos - Lognormal

```r
ajustLog <- survreg(Surv(tempos, status)~1, data=dados,
ajustLog
mulog <- ajustLog$icoef[1]
mulog
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "lognorm")

Coefficients:
(Intercept)
   2.717176

Scale= 0.7648167

Loglik(model)= -65.7   Loglik(intercept only)= -65.7
n= 20

(Intercept)
   2.717176
```

Modelos Paramétricos - Lognormal

```r
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLog
mulog ← ajustLog$icoef[1]
mulog
sigmalog ← ajustLog$scale
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "lognorm")

Coefficients:
(Intercept)
   2.717176

Scale= 0.7648167

Loglik(model)= -65.7   Loglik(intercept only)= -65.7
n= 20

(Intercept)
   2.717176
```

Modelos Paramétricos - Lognormal

```r
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLog
mulog ← ajustLog$icoef[1]
mulog
sigmalog ← ajustLog$scale
sigmalog
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "lognorm")

Coefficients:
(Intercept)
   2.717176

Scale= 0.7648167

Loglik(model)= -65.7   Loglik(intercept only)= -65.7
n= 20

(Intercept)
   2.717176

[1] 0.7648167
```

Modelos Paramétricos - Lognormal

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
```

Modelos Paramétricos - Lognormal

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
```

Modelos Paramétricos - Lognormal

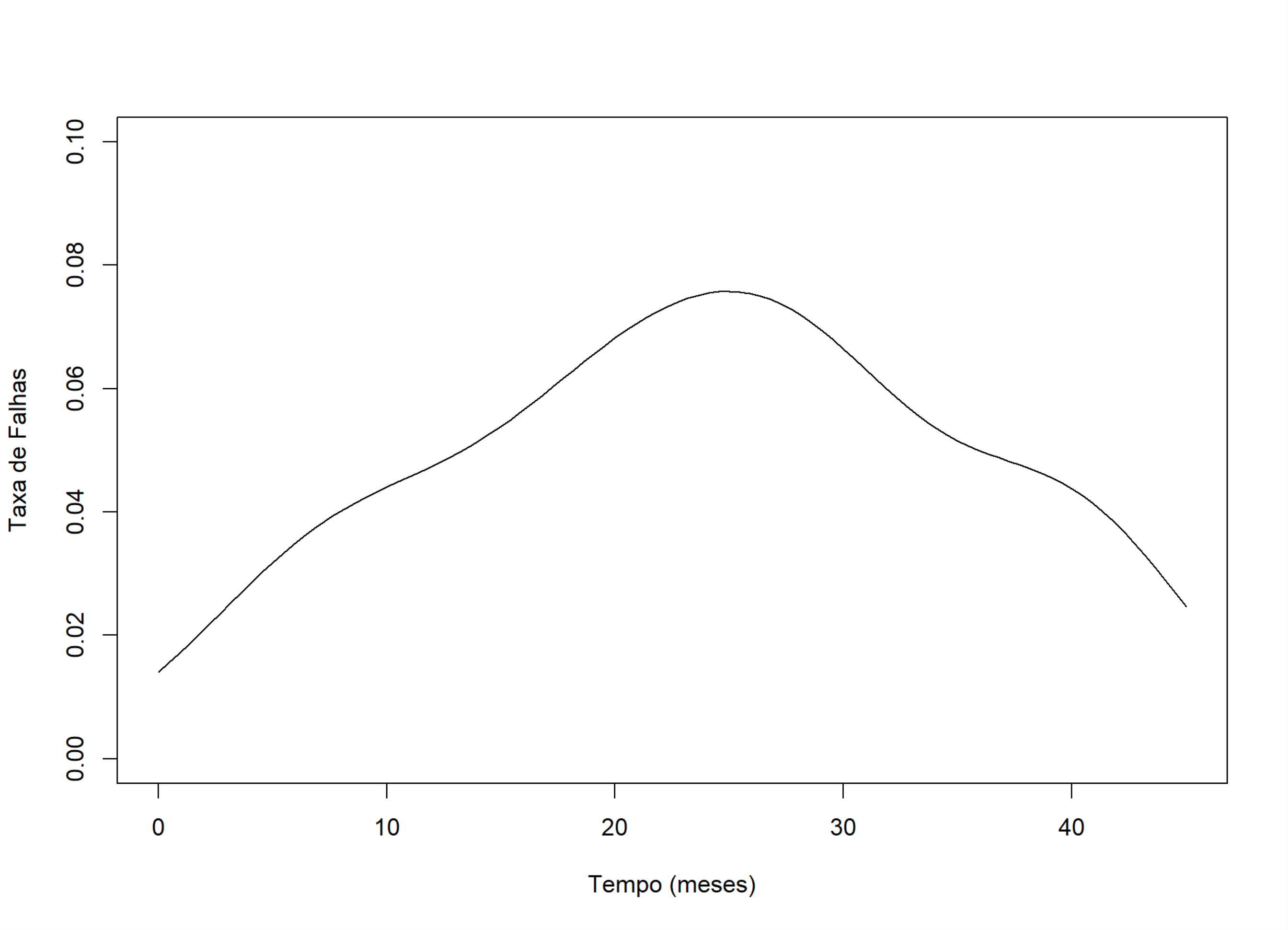
```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
```

Modelos Paramétricos - Lognormal

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
             kernel = "gaussian", n = 100, from = 0, to
```

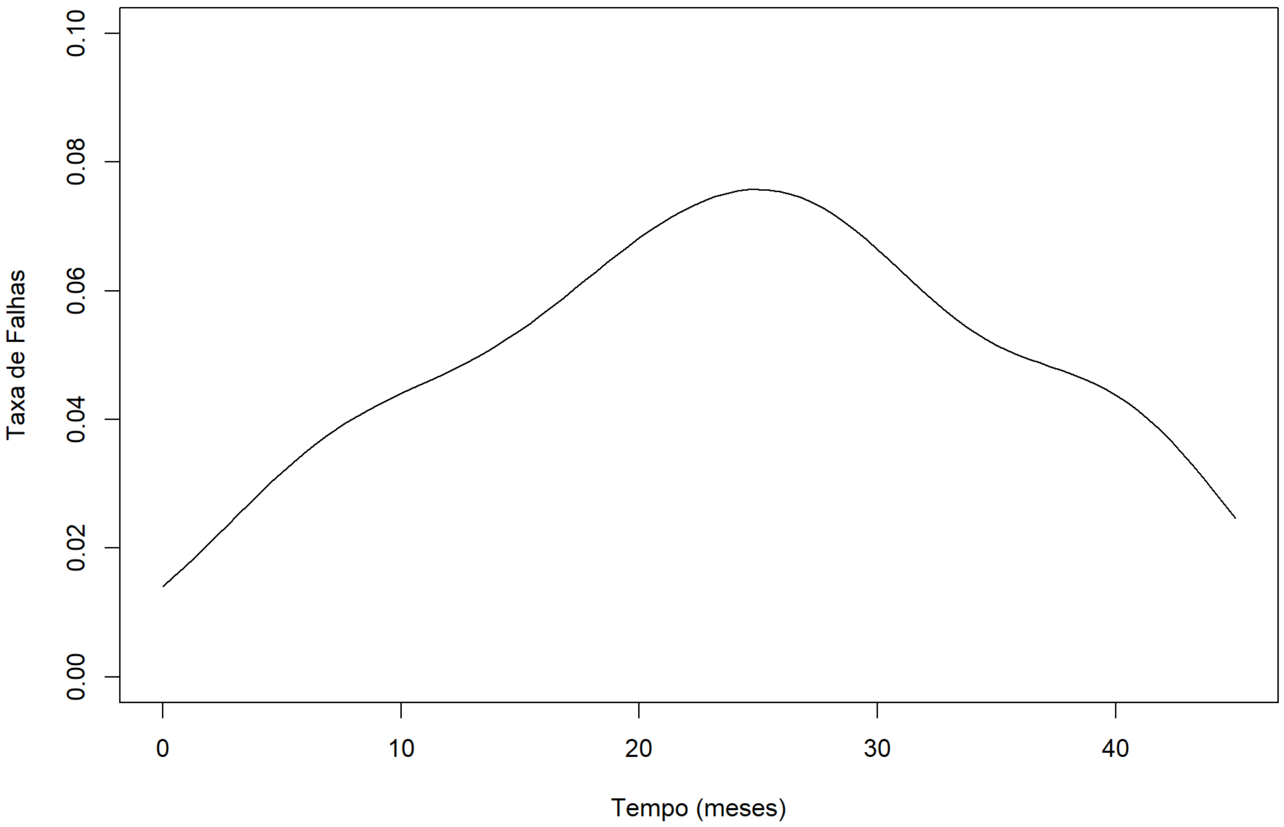
Modelos Paramétricos - Lognormal

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.10))
```



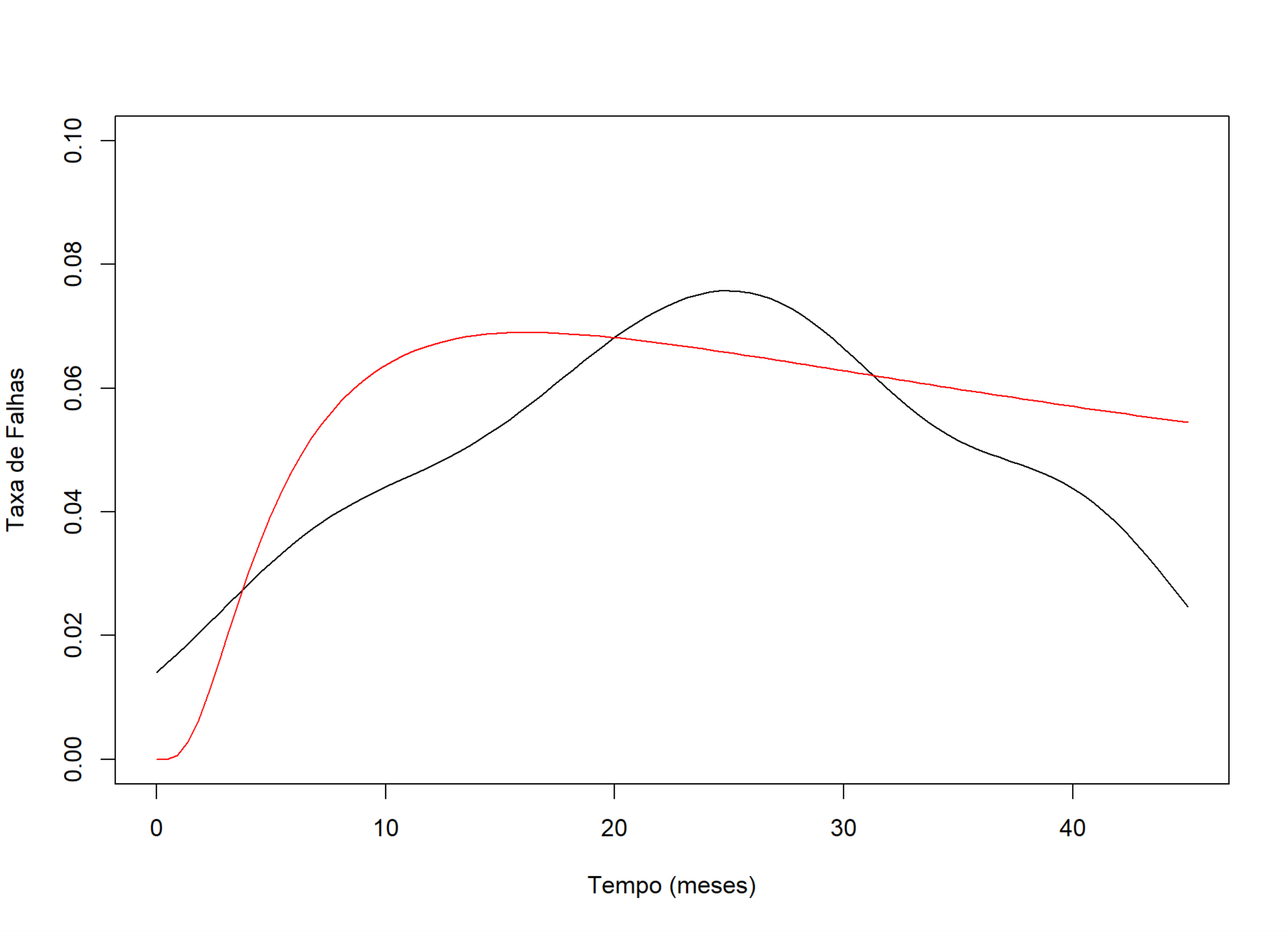
Modelos Paramétricos - Lognormal

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.10))
logHaz ← function(x, meanlog, sdlog) {
  dlnorm(x, meanlog=meanlog, sdlog=sdlog)/
  plnorm(x, meanlog=meanlog, sdlog=sdlog,
         lower.tail=F)
}
```



Modelos Paramétricos - Lognormal

```r
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.10))
logHaz ← function(x, meanlog, sdlog) {
  dlnorm(x, meanlog=meanlog, sdlog=sdlog)/
  plnorm(x, meanlog=meanlog, sdlog=sdlog,
         lower.tail=F)
}
curve(logHaz(x, meanlog=mulog, sdlog=sigmalog), from=0,
      ylab="h(t)", xlab="Tempo",
      col="red",
      add=TRUE)
```



Modelos Paramétricos - Lognormal

```
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
```

Modelos Paramétricos - Lognormal

```
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
estmedl ← predict(ajustLog,type="uquantile",p=0.5,se.fi
```

Modelos Paramétricos - Lognormal

```
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
estmedl ← predict(ajustLog,type="uquantile",p=0.5,se.fi
estmedl1 ← estmedl$fit[1]
```

Modelos Paramétricos - Lognormal

```r
ajustLog <- survreg(Surv(tempos, status)~1, data=dados,
estmedl <- predict(ajustLog,type="uquantile",p=0.5,se.fi
estmedl1 <- estmedl$fit[1]
estmedl1.ep <- estmedl$se.fit[1]
```

Modelos Paramétricos - Lognormal

```r
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
estmedl ← predict(ajustLog,type="uquantile",p=0.5,se.fi
estmedl1 ← estmedl$fit[1]
estmedl1.ep ← estmedl$se.fit[1]
exp(estmedl1)
```

```
[1] 15.13751
```

Modelos Paramétricos - Lognormal

```r
ajustLog <- survreg(Surv(tempos, status)~1, data=dados,
estmedl <- predict(ajustLog,type="uquantile",p=0.5,se.fi
estmedl1 <- estmedl$fit[1]
estmedl1.ep <- estmedl$se.fit[1]
exp(estmedl1)
IC.mediana_log <- c(exp(estmedl1),exp(estmedl1-1.96*estm
exp(estmedl1+1.96*estmedl1.ep))
```

```
[1] 15.13751
```

Modelos Paramétricos - Lognormal

```
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
estmedl ← predict(ajustLog,type="uquantile",p=0.5,se.fi
estmedl1 ← estmedl$fit[1]
estmedl1.ep ← estmedl$se.fit[1]
exp(estmedl1)
IC.mediana_log ← c(exp(estmedl1),exp(estmedl1-1.96*estm
exp(estmedl1+1.96*estmedl1.ep))
names(IC.mediana_log) ← c("mediana Logn","ICI","ICS")
```

```
[1] 15.13751
```

Modelos Paramétricos - Lognormal

```
ajustLog ← survreg(Surv(tempos, status)~1, data=dados,
estmedl ← predict(ajustLog,type="uquantile",p=0.5,se.fi
estmedl1 ← estmedl$fit[1]
estmedl1.ep ← estmedl$se.fit[1]
exp(estmedl1)
IC.mediana_log ← c(exp(estmedl1),exp(estmedl1-1.96*estm
exp(estmedl1+1.96*estmedl1.ep))
names(IC.mediana_log) ← c("mediana Logn","ICI","ICS")
IC.mediana_log
```

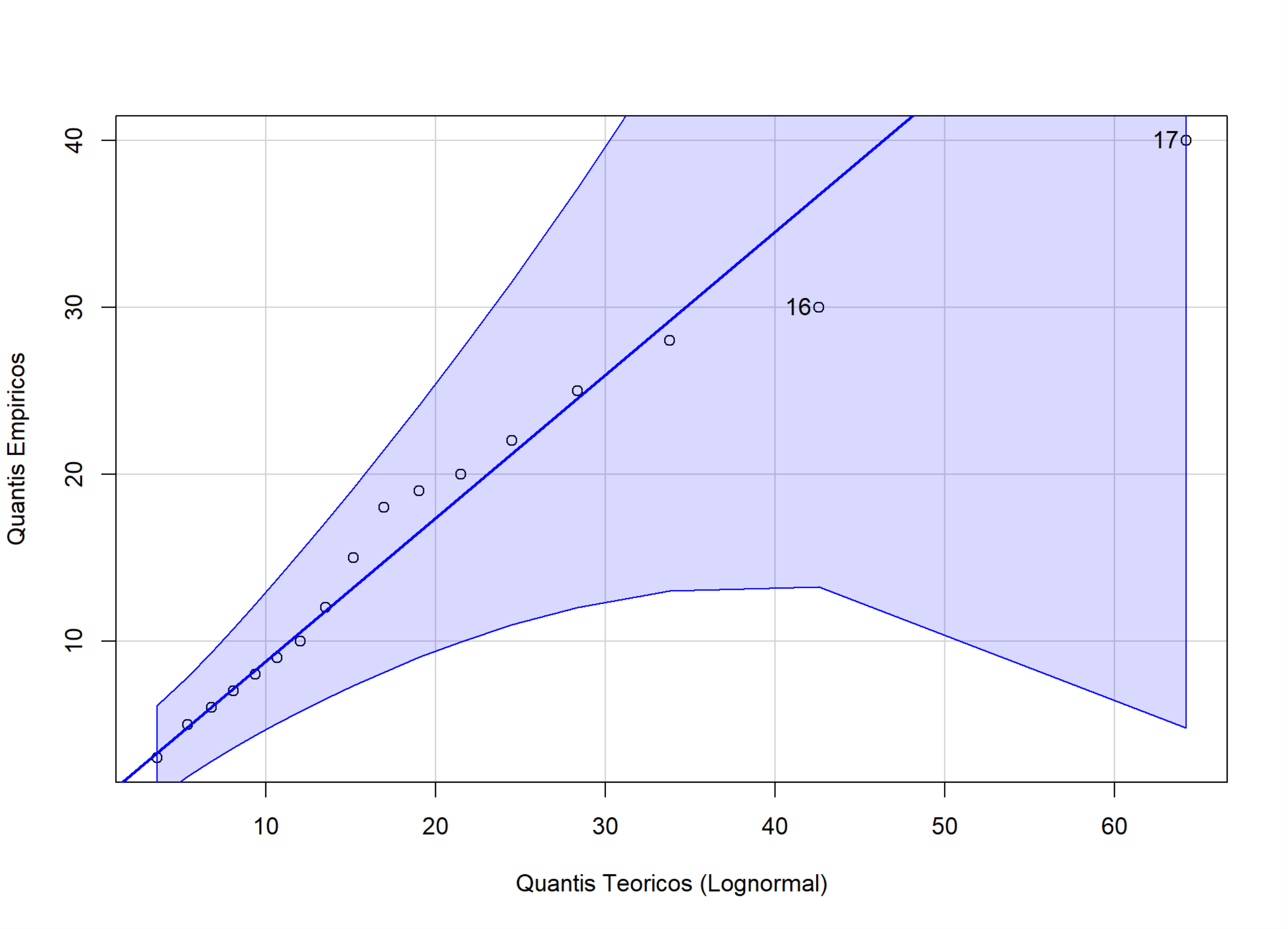
```
[1] 15.13751

mediana Logn          ICI          ICS
    15.13751     10.71601     21.38334
```

Modelos Paramétricos - Lognormal

```r
n.censurado <- dados$tempos[dados$status == 1]
```

Modelos Paramétricos - Lognormal

```
n.censurado ← dados$tempos[dados$status == 1]
qqPlot(n.censurado,
dist = "lnorm",
meanlog = mulog,
sdlog = sigmalog,
xlab = "Quantis Teoricos (Lognormal)",
ylab = "Quantis Empiricos")
```



```
[1] 17 16
```

Modelos Paramétricos - Log-Logistico

```
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
```

Modelos Paramétricos - Log-Logistico

```
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLLog
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "loglogistic")

Coefficients:
(Intercept)
   2.737415

Scale= 0.4496653

Loglik(model)= -66   Loglik(intercept only)= -66
n= 20
```

Modelos Paramétricos - Log-Logistico

```
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLLog
betall← ajustLLog$icoef[1]
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "loglogistic")

Coefficients:
(Intercept)
   2.737415

Scale= 0.4496653

Loglik(model)= -66   Loglik(intercept only)= -66
n= 20
```

Modelos Paramétricos - Log-Logistico

```r
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLLog
betall← ajustLLog$icoef[1]
betall
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "loglogistic")

Coefficients:
(Intercept)
   2.737415

Scale= 0.4496653

Loglik(model)= -66   Loglik(intercept only)= -66
n= 20

(Intercept)
   2.737415
```

Modelos Paramétricos - Log-Logistico

```r
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLLog
betall← ajustLLog$icoef[1]
betall
alfall ← ajustLLog$scale
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "loglogistic")

Coefficients:
(Intercept)
   2.737415

Scale= 0.4496653

Loglik(model)= -66   Loglik(intercept only)= -66
n= 20

(Intercept)
   2.737415
```

Modelos Paramétricos - Log-Logistico

```r
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLLog
betall← ajustLLog$icoef[1]
betall
alfall ← ajustLLog$scale
alfall
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "loglogistic")

Coefficients:
(Intercept)
   2.737415

Scale= 0.4496653

Loglik(model)= -66   Loglik(intercept only)= -66
n= 20

(Intercept)
   2.737415

[1] 0.4496653
```

Modelos Paramétricos - Log-Logistico

```
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLLog
betall← ajustLLog$icoef[1]
betall
alfall ← ajustLLog$scale
alfall
mullog ← betall
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "loglogistic")

Coefficients:
(Intercept)
   2.737415

Scale= 0.4496653

Loglik(model)= -66   Loglik(intercept only)= -66
n= 20

(Intercept)
   2.737415

[1] 0.4496653
```

Modelos Paramétricos - Log-Logistico

```r
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
ajustLLog
betall← ajustLLog$icoef[1]
betall
alfall ← ajustLLog$scale
alfall
mullog ← betall
sigmallog ← alfall
```

```
Call:
survreg(formula = Surv(tempos, status) ~ 1, data = dados, dist = "loglogistic")

Coefficients:
(Intercept)
   2.737415

Scale= 0.4496653

Loglik(model)= -66   Loglik(intercept only)= -66
n= 20

(Intercept)
   2.737415

[1] 0.4496653
```

Modelos Paramétricos - Log-Logistico

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
```

Modelos Paramétricos - Log-Logistico

```r
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum <- summary(ajusteKM)
```

Modelos Paramétricos - Log-Logistico

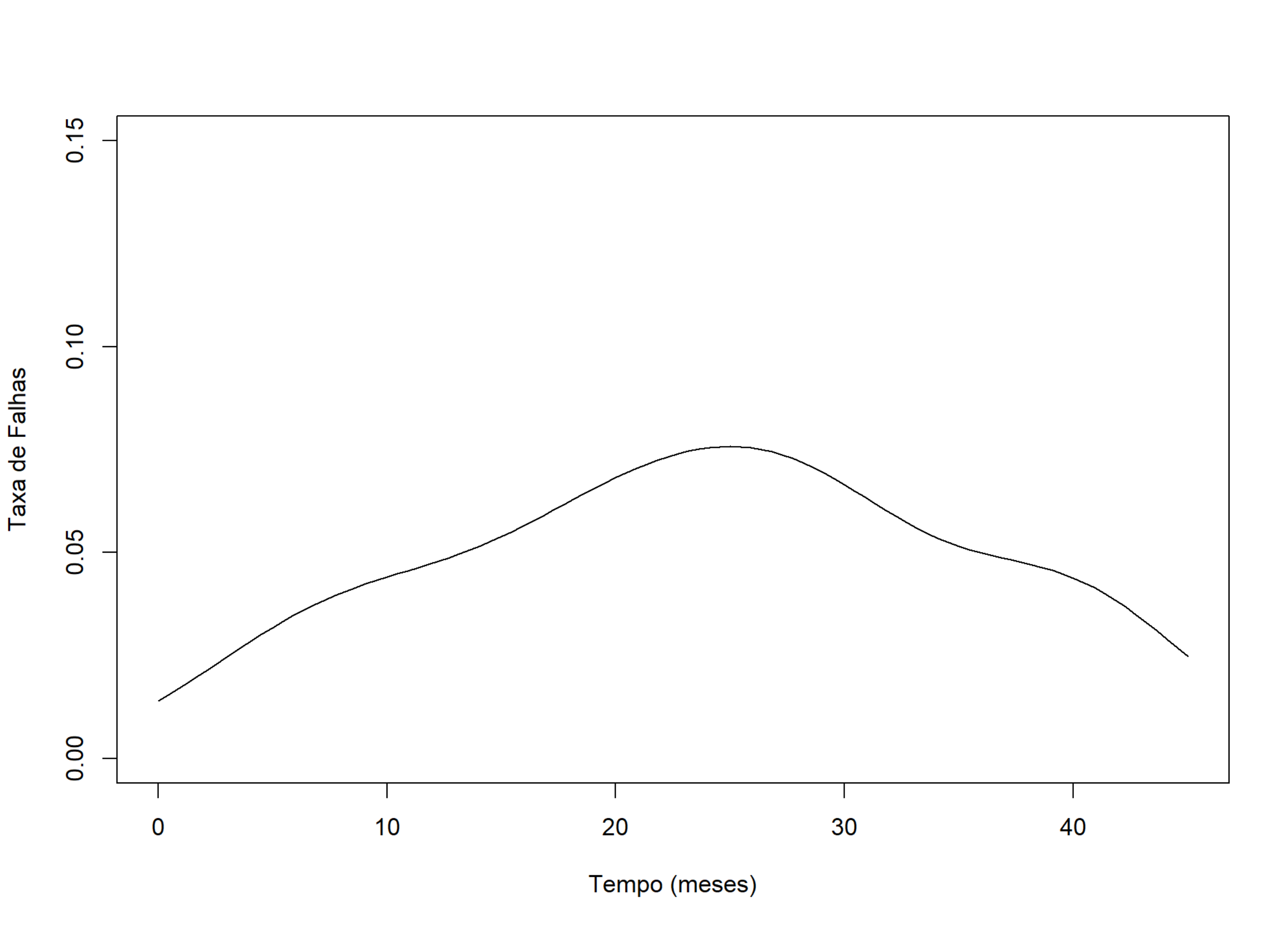
```r
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
```

Modelos Paramétricos - Log-Logistico

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
             kernel = "gaussian", n = 100, from = 0, to
```

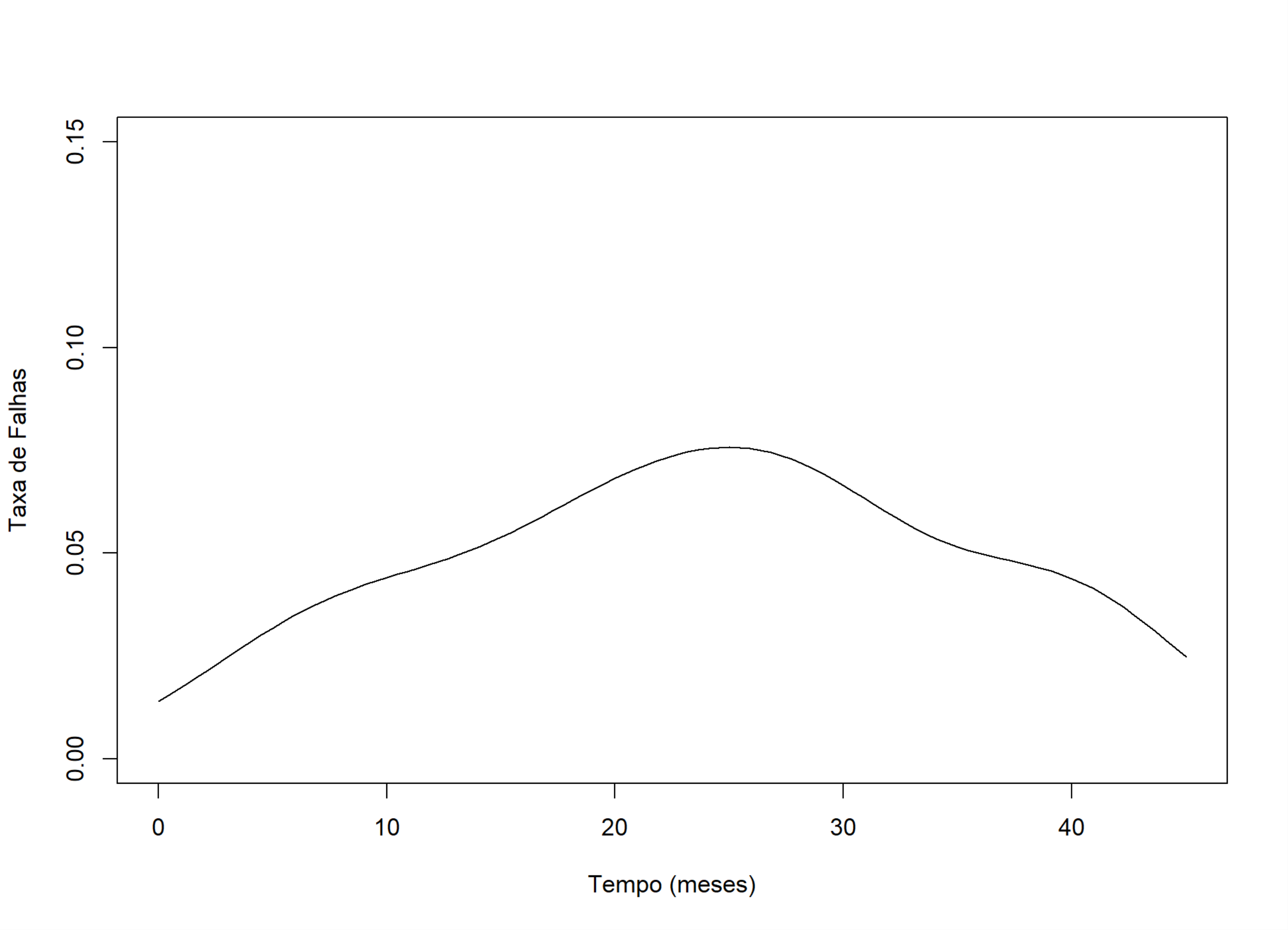
Modelos Paramétricos - Log-Logistico

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
                kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
      ylim=c(0,0.15))
```



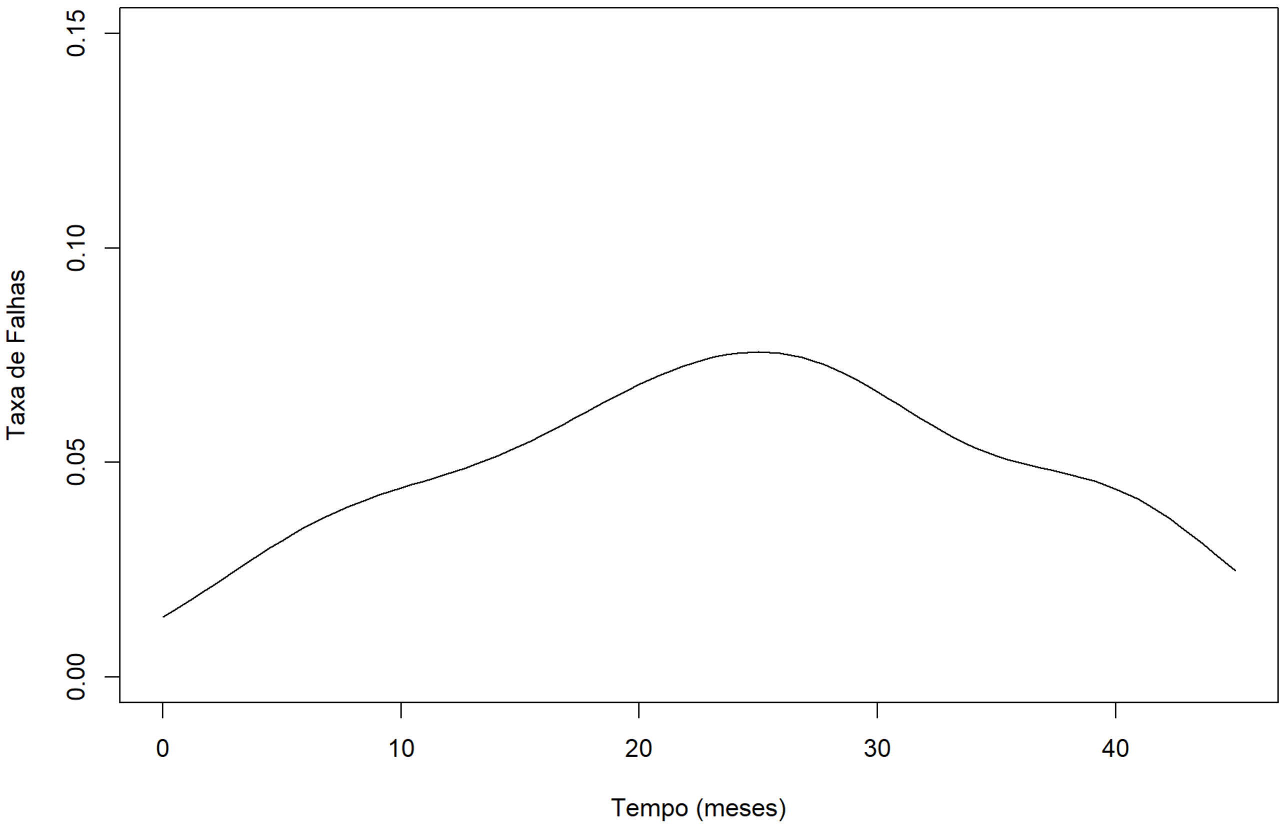
Modelos Paramétricos - Log-Logistico

```
ajusteKM <- survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum <- summary(ajusteKM)
pesos <- ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h <- density(ajusteKM_sum$time, weights = pesos,
                kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.15))
library(flexsurv)
```



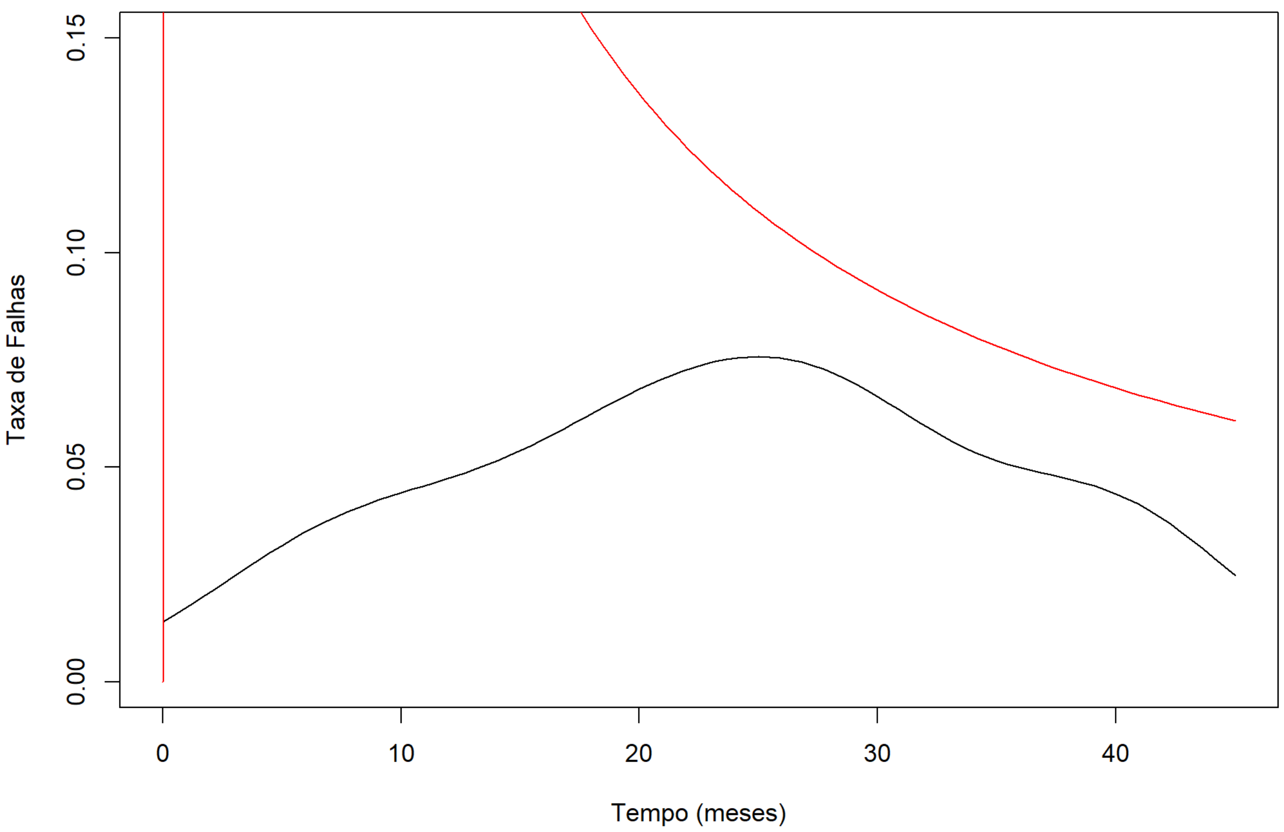
Modelos Paramétricos - Log-Logistico

```
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.15))
library(flexsurv)
llogisHaz ← function(x, alfa, beta) {
  haz ← hllogis(x, shape = beta, scale = alfa, log = FA
}
```



Modelos Paramétricos - Log-Logistico

```r
ajusteKM ← survfit(Surv(tempos, status) ~ 1, data=dados
ajusteKM_sum ← summary(ajusteKM)
pesos ← ajusteKM_sum$n.event/ajusteKM_sum$n.risk
h ← density(ajusteKM_sum$time, weights = pesos,
            kernel = "gaussian", n = 100, from = 0, to
plot(h$x, h$y, type = "l", xlab = "Tempo (meses)", ylab
     ylim=c(0,0.15))
library(flexsurv)
llogisHaz ← function(x, alfa, beta) {
  haz ← hllogis(x, shape = beta, scale = alfa, log = FA
}
curve(llogisHaz(x, alfa=alfall, beta=betall), from=0, to
      ylab="h(t)", xlab="Tempo", col="red", add=TRUE)
```



Modelos Paramétricos - Log-Logistico

```
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
```

Modelos Paramétricos - Log-Logistico

```
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
estmedll ← predict(ajustLLog,type="uquantile",p=0.5,se.
```

Modelos Paramétricos - Log-Logistico

```
ajustLLog <- survreg(Surv(tempos, status)~1, data=dados,
estmedll <- predict(ajustLLog,type="uquantile",p=0.5,se.
estmedll1 <- estmedll$fit[1]
```

Modelos Paramétricos - Log-Logistico

```
ajustLLog <- survreg(Surv(tempos, status)~1, data=dados,
estmedll <- predict(ajustLLog,type="uquantile",p=0.5,se.
estmedll1 <- estmedll$fit[1]
estmedll1.ep <- estmedll$se.fit[1]
```

Modelos Paramétricos - Log-Logistico

```
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
estmedll ← predict(ajustLLog,type="uquantile",p=0.5,se.
estmedll1 ← estmedll$fit[1]
estmedll1.ep ← estmedll$se.fit[1]
exp(estmedll1)
```

```
[1] 15.447
```

Modelos Paramétricos - Log-Logistico

```r
ajustLLog ← survreg(Surv(tempos, status)~1, data=dados,
estmedll ← predict(ajustLLog,type="uquantile",p=0.5,se.
estmedll1 ← estmedll$fit[1]
estmedll1.ep ← estmedll$se.fit[1]
exp(estmedll1)
IC.mediana_llog ← c(exp(estmedll1),exp(estmedll1-1.96*e
exp(estmedll1+1.96*estmedll1.ep))
```

```
[1] 15.447
```

Modelos Paramétricos - Log-Logistico

```r
ajustLLog <- survreg(Surv(tempos, status)~1, data=dados,
estmedll <- predict(ajustLLog,type="uquantile",p=0.5,se.
estmedll1 <- estmedll$fit[1]
estmedll1.ep <- estmedll$se.fit[1]
exp(estmedll1)
IC.mediana_llog <- c(exp(estmedll1),exp(estmedll1-1.96*e
exp(estmedll1+1.96*estmedll1.ep))
names(IC.mediana_llog) <- c("mediana LogL","ICI","ICS")
```

```
[1] 15.447
```

Modelos Paramétricos - Log-Logistico

```r
ajustLLog <- survreg(Surv(tempos, status)~1, data=dados,
estmedll <- predict(ajustLLog,type="uquantile",p=0.5,se.
estmedll1 <- estmedll$fit[1]
estmedll1.ep <- estmedll$se.fit[1]
exp(estmedll1)
IC.mediana_llog <- c(exp(estmedll1),exp(estmedll1-1.96*e
exp(estmedll1+1.96*estmedll1.ep))
names(IC.mediana_llog) <- c("mediana LogL","ICI","ICS")
IC.mediana_llog
```

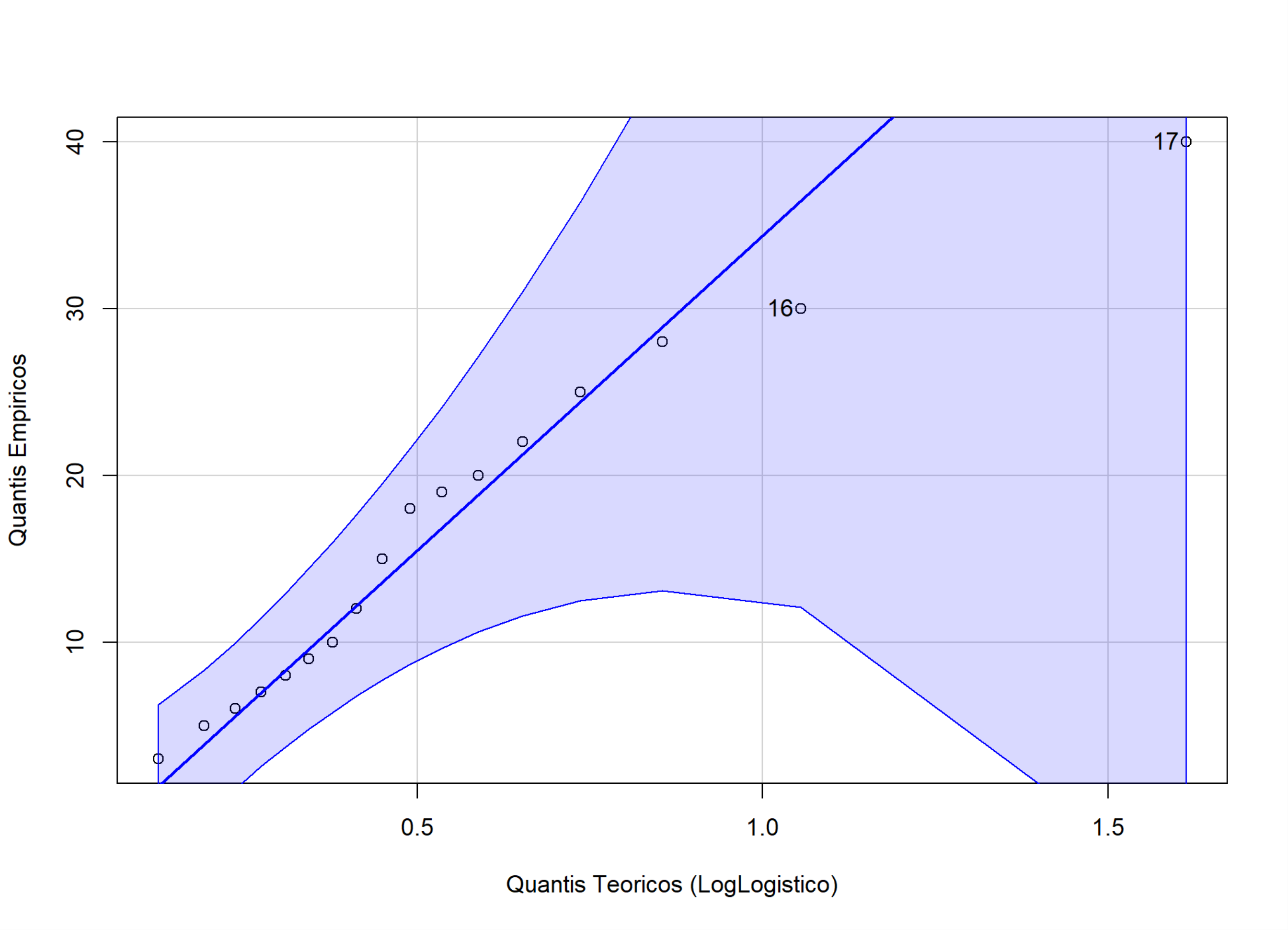
```
[1] 15.447

mediana LogL          ICI          ICS
    15.44700     10.85399     21.98361
```

Modelos Paramétricos - Log-Logistico

```r
n.censurado <- dados$tempos[dados$status == 1]
```

Modelos Paramétricos - Log-Logistico

```r
n.censurado <- dados$tempos[dados$status == 1]
qqPlot(n.censurado,
       dist = "llogis",
       shape = betall,
       scale = alfall,
       xlab = "Quantis Teoricos (LogLogistico)",
       ylab = "Quantis Empiricos")
```



```
[1] 17 16
```

1a Comparação dos Modelos

```
tempo ← ekm$time
```

1a Comparação dos Modelos

```
tempo ← ekm$time
st ← ekm$surv
```

1a Comparação dos Modelos

```
tempo ← ekm$time
st ← ekm$surv
ste ← exp(-tempo/alfa)
```

1a Comparação dos Modelos

```r
tempo <- ekm$time
st <- ekm$surv
ste <- exp(-tempo/alfa)
stw <- exp(-(tempo/alfaw)^betaw)
```

1a Comparação dos Modelos

```r
tempo <- ekm$time
st <- ekm$surv
ste <- exp(-tempo/alfa)
stw <- exp(-(tempo/alfaw)^betaw)
stln <- pnorm((-log(tempo) + mulog)/sigmalog)
```

1a Comparação dos Modelos

```
tempo ← ekm$time
st ← ekm$surv
ste ← exp(-tempo/alfa)
stw ← exp(-(tempo/alfaw)^betaw)
stln ← pnorm((-log(tempo) + mulog)/sigmalog)
stll ← plogis((-log(tempo) + mullog)/sigmallog)
```

1a Comparação dos Modelos

```
tempo ← ekm$time
st ← ekm$surv
ste ← exp(-tempo/alfa)
stw ← exp(-(tempo/alfaw)^betaw)
stln ← pnorm((-log(tempo) + mulog)/sigmalog)
stll ← plogis((-log(tempo) + mullog)/sigmallog)
cbind(tempo,st,ste,stw,stln, stll)
```

```
      tempo         st       ste        stw       stln      stll
 [1,]     3 0.95000000 0.8633164 0.95274148 0.98283934 0.9745310
 [2,]     5 0.90000000 0.7827384 0.89897484 0.92624322 0.9247343
 [3,]     6 0.85000000 0.7453152 0.86839357 0.88685752 0.8911964
 [4,]     7 0.80000000 0.7096812 0.83609525 0.84337638 0.8532358
 [5,]     8 0.75000000 0.6757509 0.80253272 0.79781416 0.8120295
 [6,]     9 0.70000000 0.6434428 0.76809812 0.75169629 0.7687589
 [7,]    10 0.65000000 0.6126794 0.73313414 0.70611769 0.7245227
 [8,]    12 0.59583333 0.5554947 0.66278292 0.61931883 0.6368104
 [9,]    15 0.54166667 0.4795676 0.55966698 0.50475984 0.5163200
[10,]    18 0.48148148 0.4140186 0.46346069 0.41042396 0.4157713
[11,]    19 0.42129630 0.3942241 0.43345774 0.38317692 0.3868919
[12,]    20 0.36111111 0.3753760 0.40461041 0.35784922 0.3602069
[13,]    22 0.30092593 0.3403401 0.35056108 0.31248031 0.3129370
[14,]    25 0.24074074 0.2938212 0.27891623 0.25592025 0.2552681
[15,]    28 0.18055556 0.2536607 0.21851514 0.21065456 0.2103630
[16,]    30 0.12037037 0.2299851 0.18419003 0.18556452 0.1860066
[17,]    40 0.06018519 0.1409071 0.07154720 0.10195229 0.1075572
[18,]    45 0.06018519 0.1102934 0.04229094 0.07714984 0.0848755
```

2a Comparação dos Modelos

```
med.km
```

```
mediana KM        ICI        ICS
        18         10         28
```

2a Comparação dos Modelos

```
med.km
IC.mediana_exp
```

```
mediana KM        ICI        ICS
        18         10         28

mediana Exp         ICI         ICS
  14.148357    8.795399   22.759175
```

2a Comparação dos Modelos

```
med.km
IC.mediana_exp
IC.mediana_w
```

```
mediana KM        ICI        ICS
        18         10         28

mediana Exp         ICI         ICS
  14.148357    8.795399   22.759175

mediana Wei         ICI         ICS
   16.82823    11.95670    23.68456
```

2a Comparação dos Modelos

```
med.km
IC.mediana_exp
IC.mediana_w
IC.mediana_log
```

```
mediana KM        ICI        ICS
        18         10         28

mediana Exp         ICI         ICS
  14.148357    8.795399   22.759175

mediana Wei         ICI         ICS
   16.82823    11.95670    23.68456

mediana Logn          ICI          ICS
    15.13751     10.71601     21.38334
```

2a Comparação dos Modelos

```
med.km
IC.mediana_exp
IC.mediana_w
IC.mediana_log
IC.mediana_llog
```

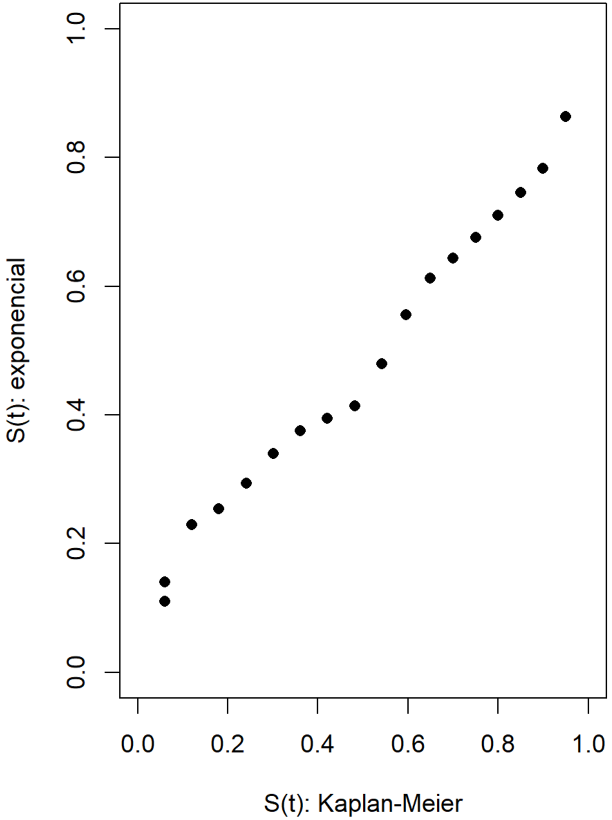
```
mediana KM        ICI        ICS
        18         10         28

mediana Exp         ICI         ICS
  14.148357    8.795399   22.759175

mediana Wei         ICI         ICS
   16.82823    11.95670    23.68456

mediana Logn          ICI          ICS
    15.13751     10.71601     21.38334

mediana LogL          ICI          ICS
    15.44700     10.85399     21.98361
```

3a Comparação dos Modelos

```r
par(mfrow=c(1,2))
```

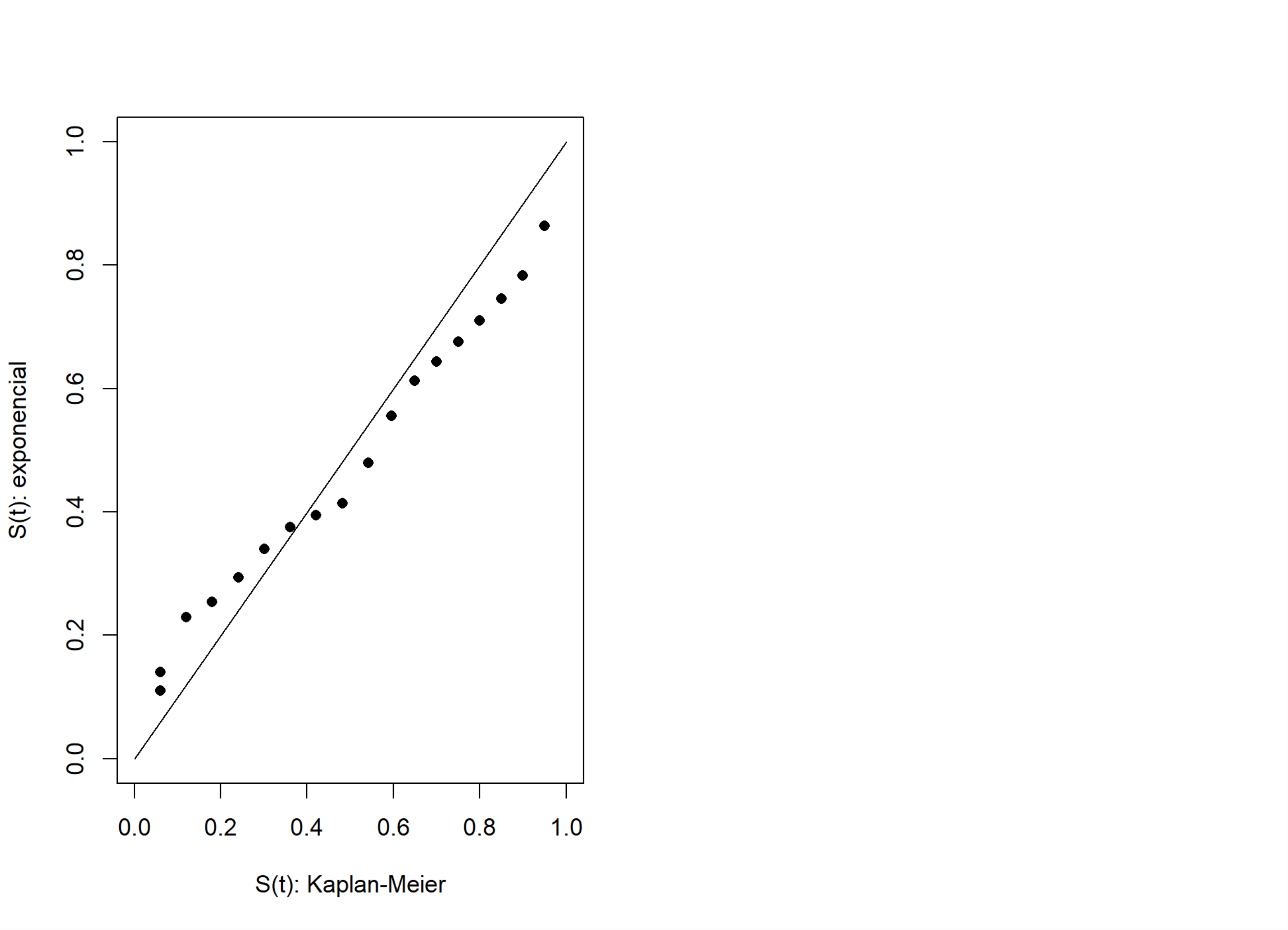
3a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(st,ste,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0,
```



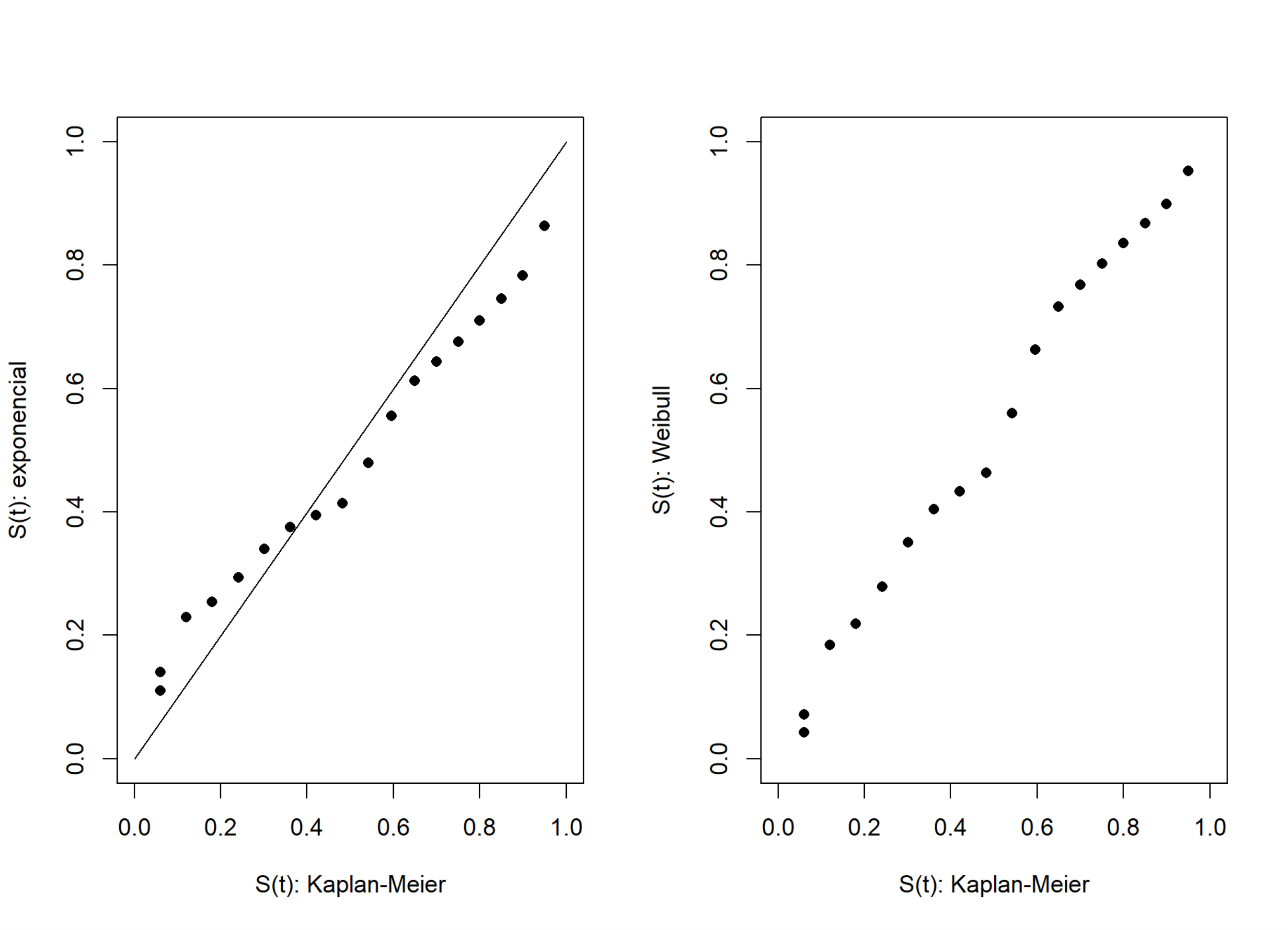
3a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(st,ste,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0,
lines(c(0,1), c(0,1), type="l", lty=1)
```



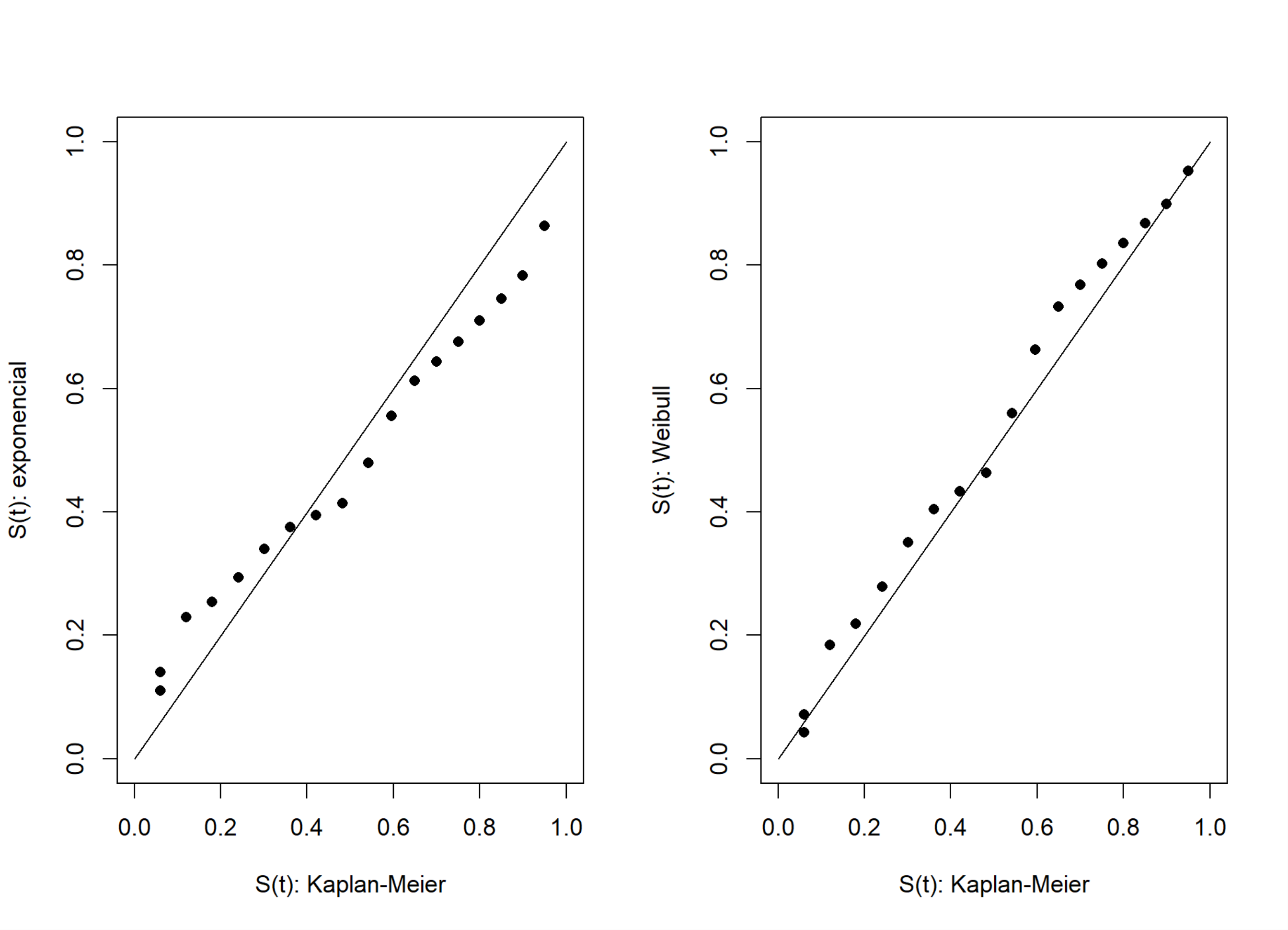
3a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(st,ste,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0,
lines(c(0,1), c(0,1), type="l", lty=1)
plot(st,stw,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0,
     ylab="S(t): Weibull")
```



3a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(st,ste,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0,
lines(c(0,1), c(0,1), type="l", lty=1)
plot(st,stw,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0,
     ylab="S(t): Weibull")
lines(c(0,1), c(0,1), type="l", lty=1)
```

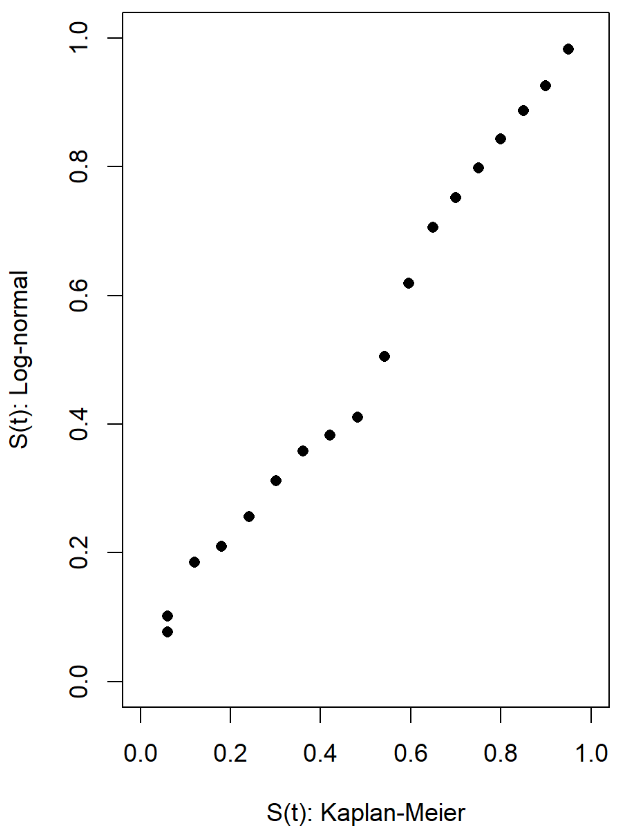


3a Comparação dos Modelos

```
par(mfrow=c(1,2))
```

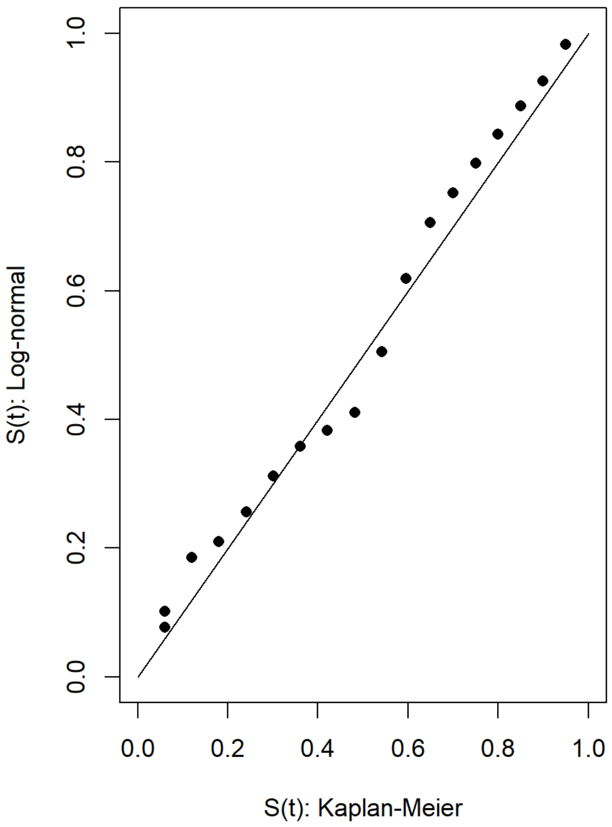
3a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(st,stln,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0
     ylab="S(t): Log-normal")
```



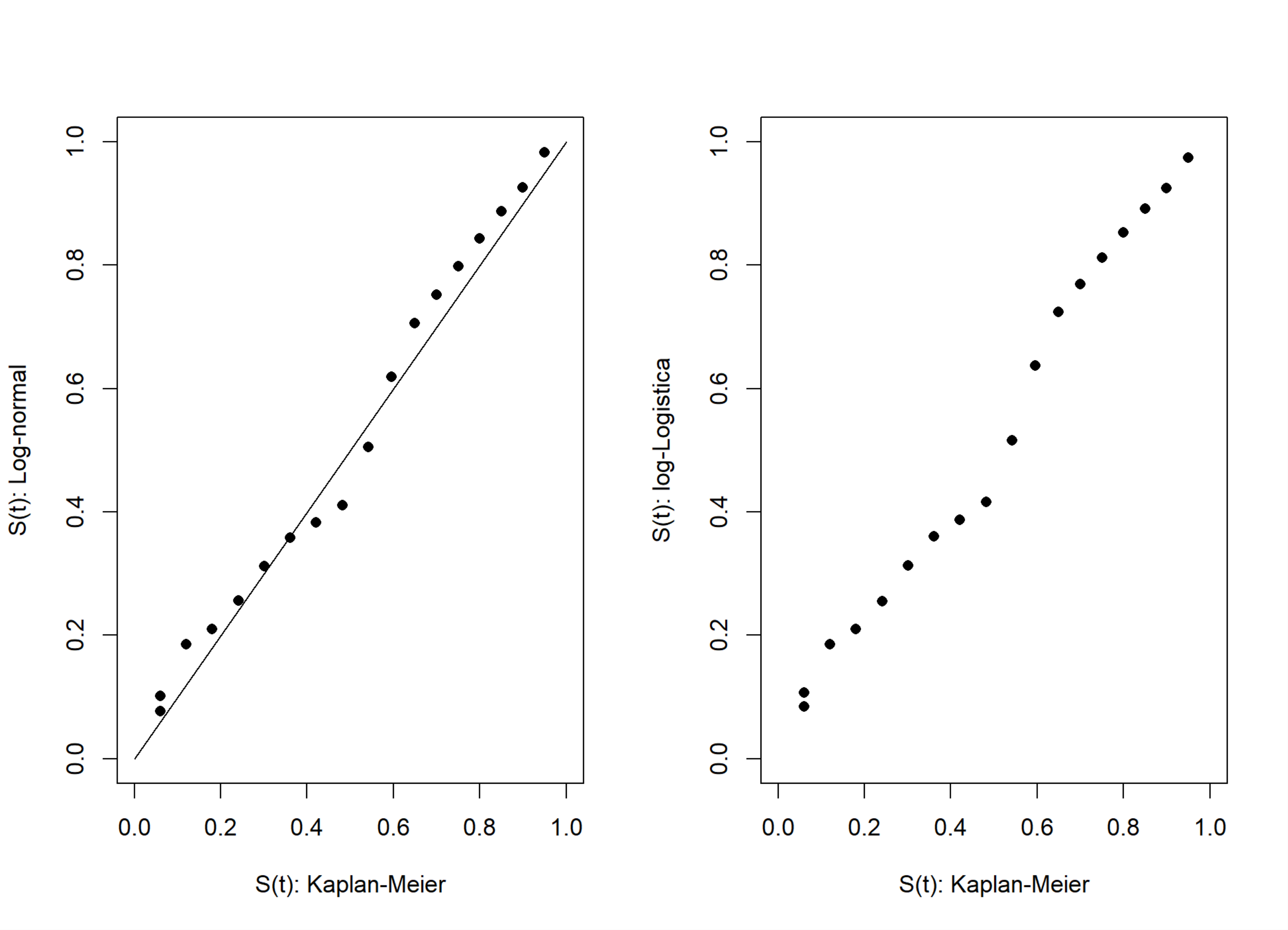
3a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(st,stln,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0
     ylab="S(t): Log-normal")
lines(c(0,1), c(0,1), type="l", lty=1)
```



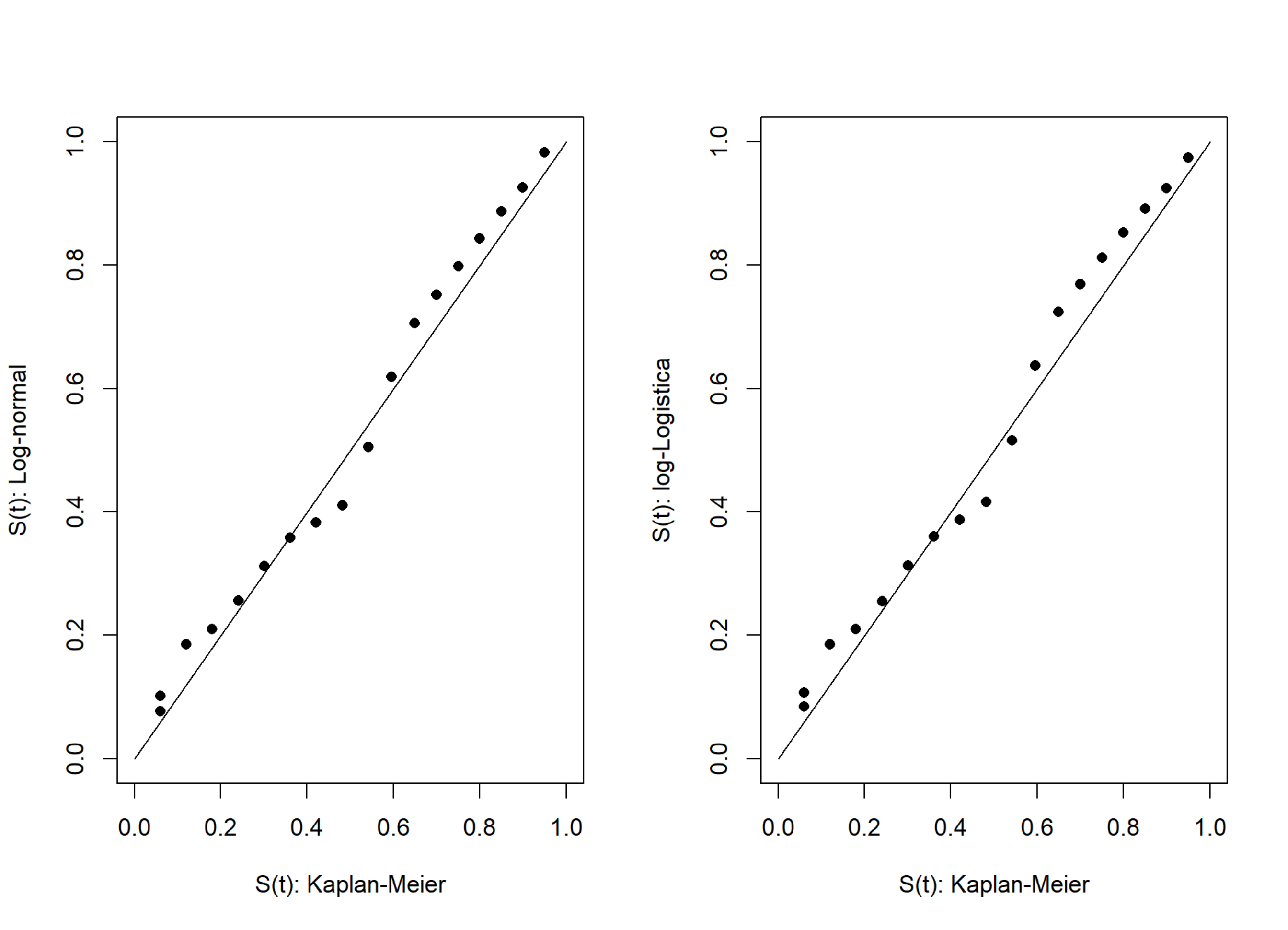
3a Comparação dos Modelos

```r
par(mfrow=c(1,2))
plot(st,stln,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0
     ylab="S(t): Log-normal")
lines(c(0,1), c(0,1), type="l", lty=1)
plot(st,stll,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0
     ylab="S(t): log-Logistica")
```



3a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(st,stln,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0
     ylab="S(t): Log-normal")
lines(c(0,1), c(0,1), type="l", lty=1)
plot(st,stll,pch=16,ylim=range(c(0.0,1)), xlim=range(c(0
     ylab="S(t): log-Logistica")
lines(c(0,1), c(0,1), type="l", lty=1)
```

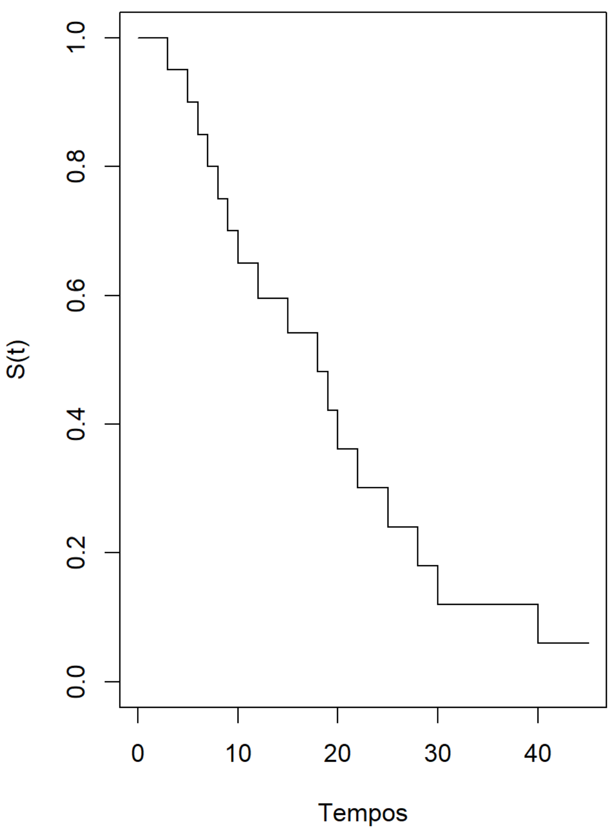


4a Comparação dos Modelos

```r
par(mfrow=c(1,2))
```

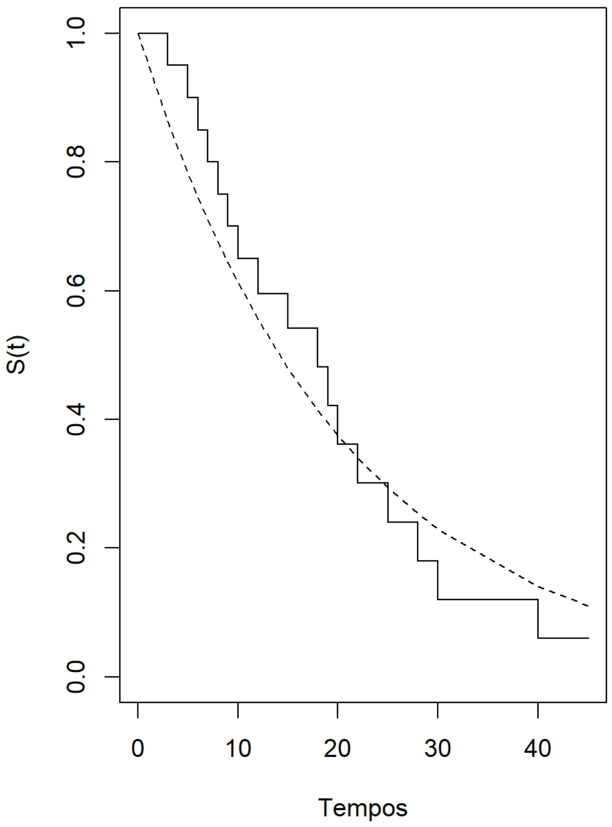
4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
```



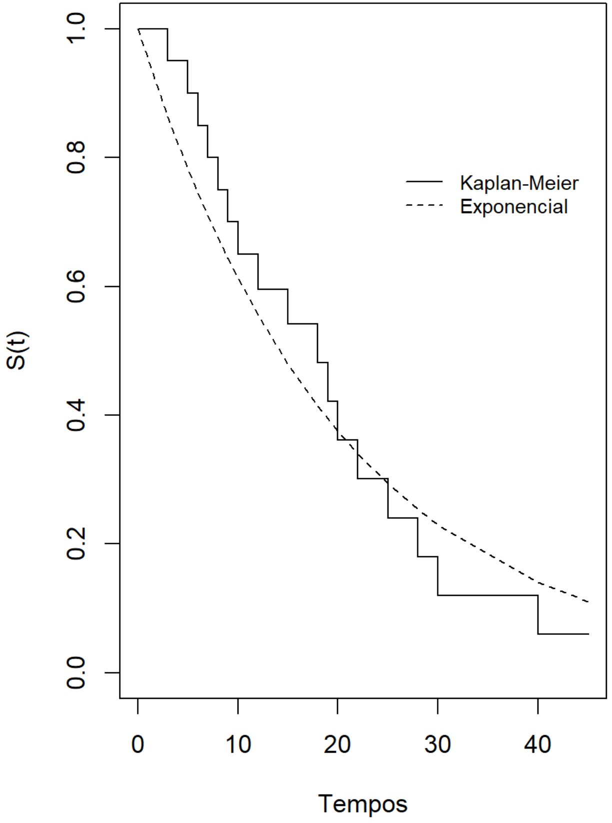
4a Comparação dos Modelos

```r
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,ste), lty=2)
```



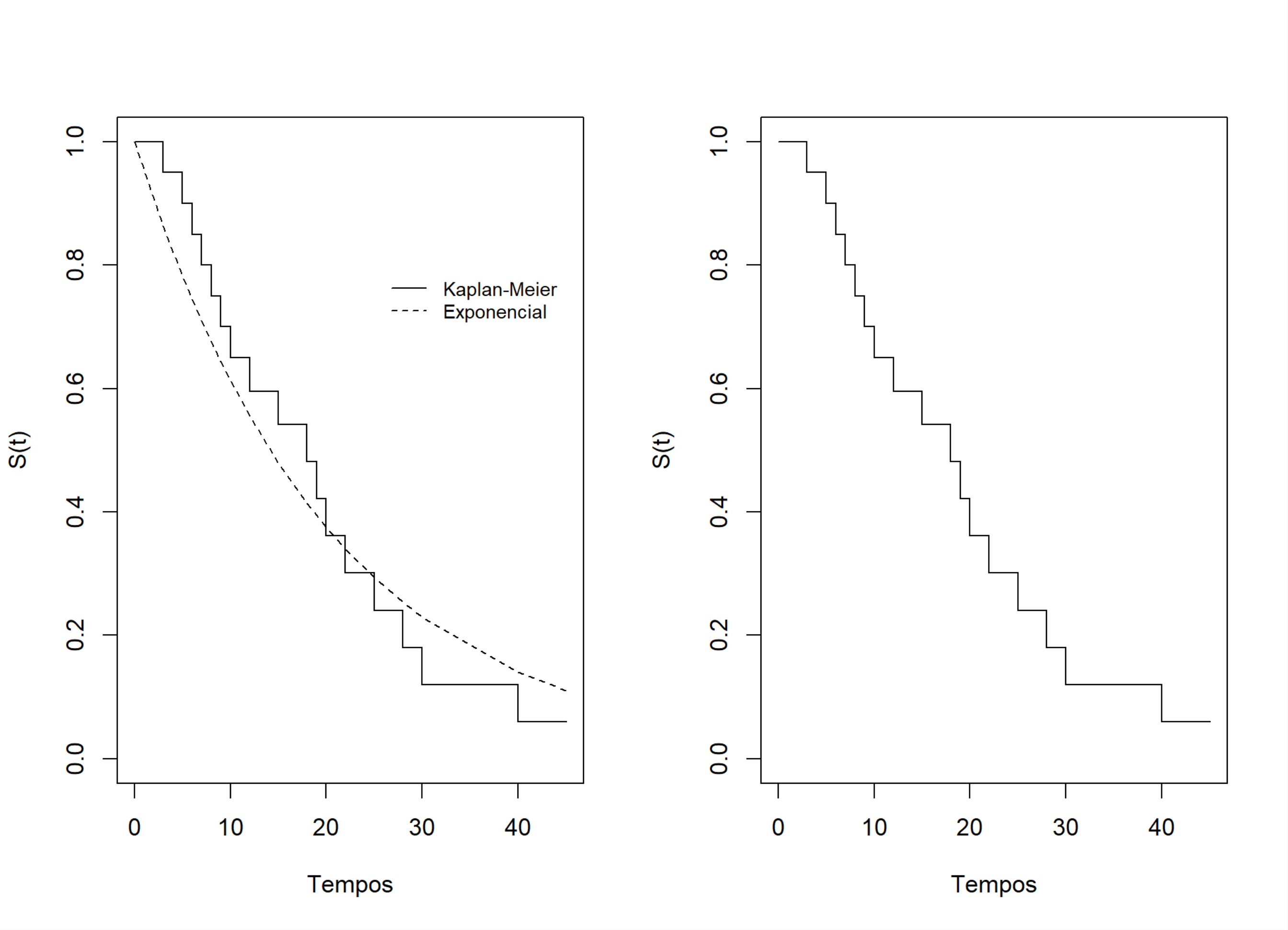
4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,ste), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Exponencial"
```



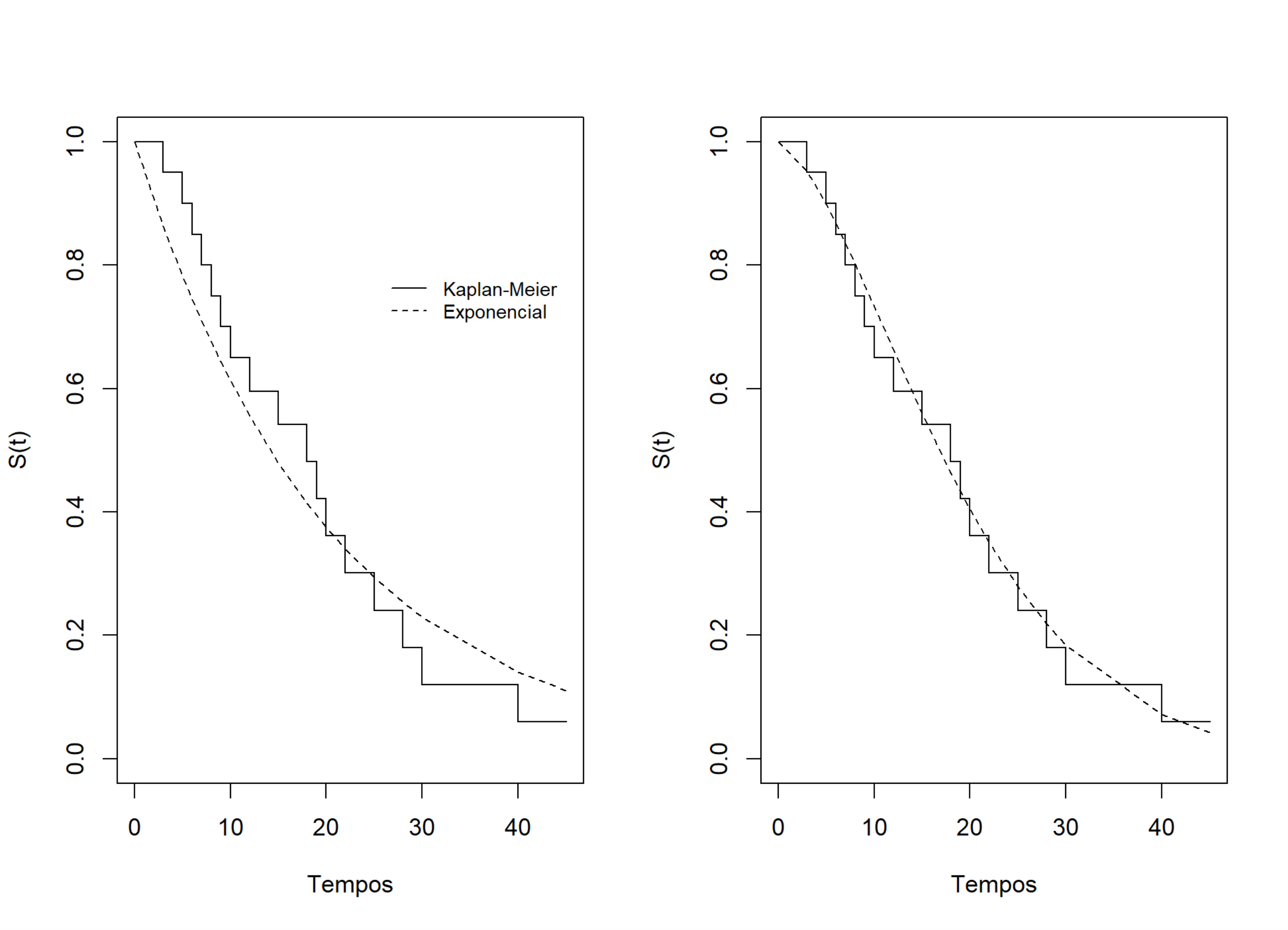
4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,ste), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Exponencial"
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
```



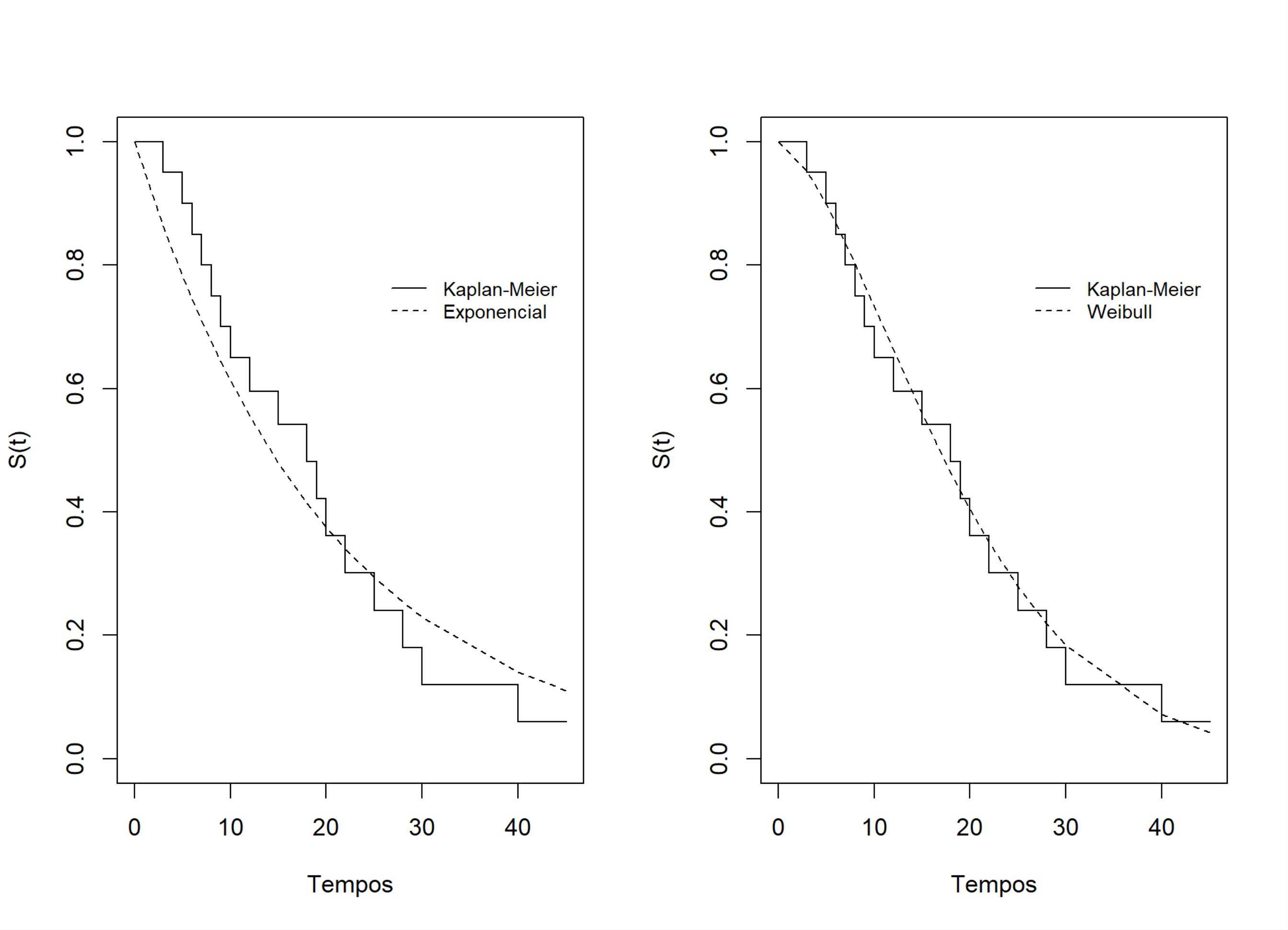
4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,ste), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Exponencial"
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stw), lty=2)
```



4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,ste), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Exponencial"
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stw), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Weibull"),bt
```

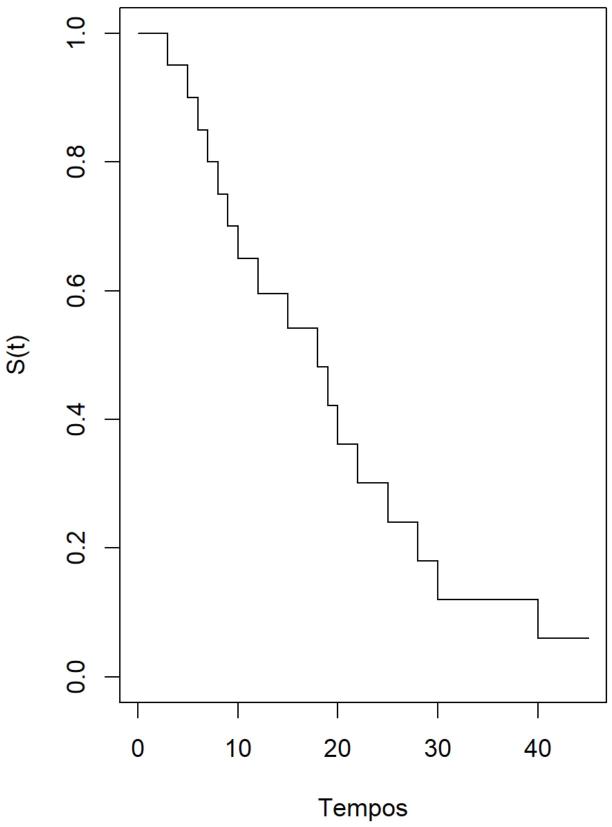


4a Comparação dos Modelos

```r
par(mfrow=c(1,2))
```

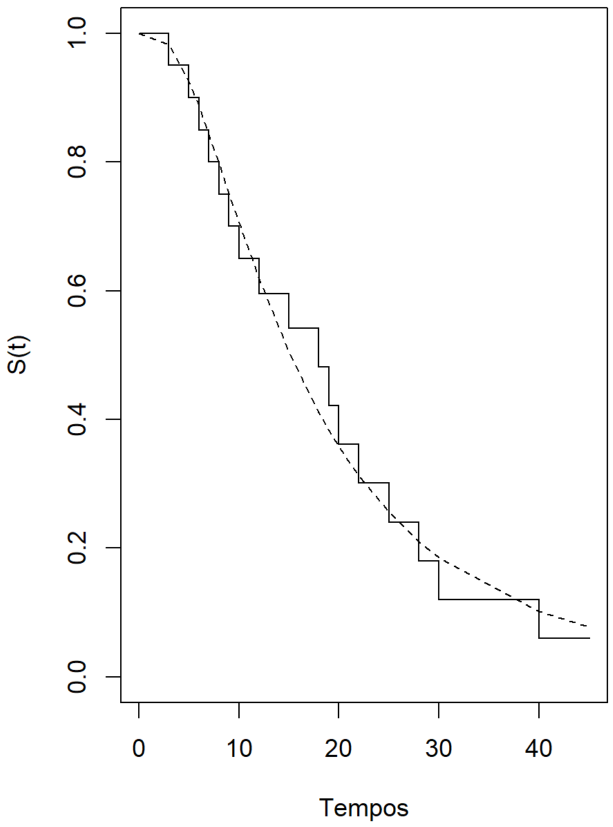
4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
```



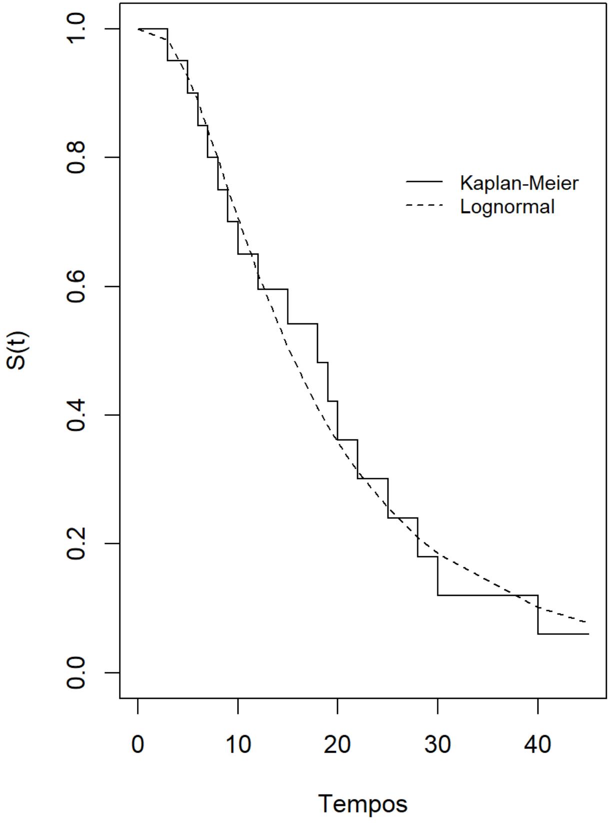
4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stln), lty=2)
```



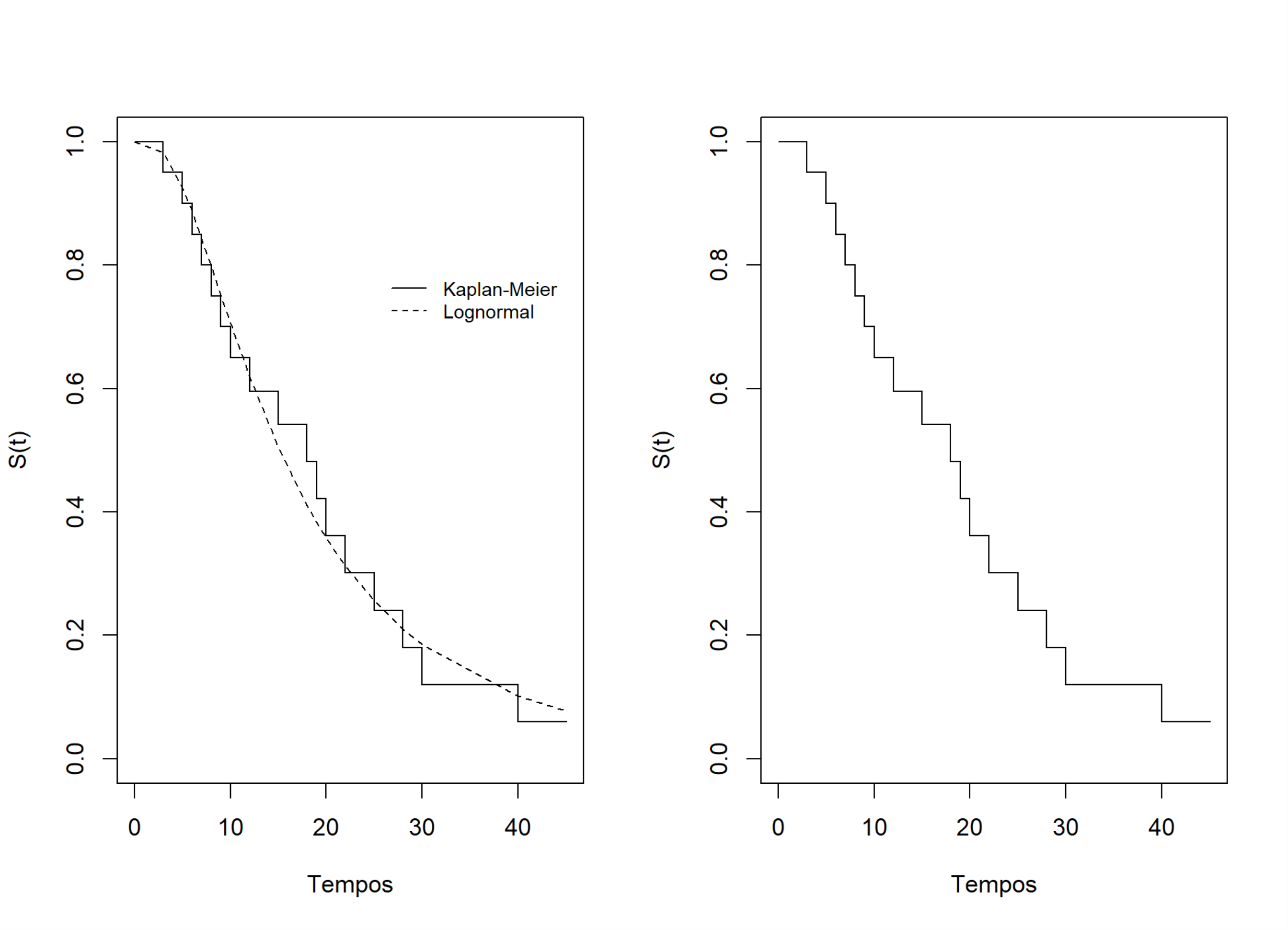
4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stln), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Lognormal"),
```



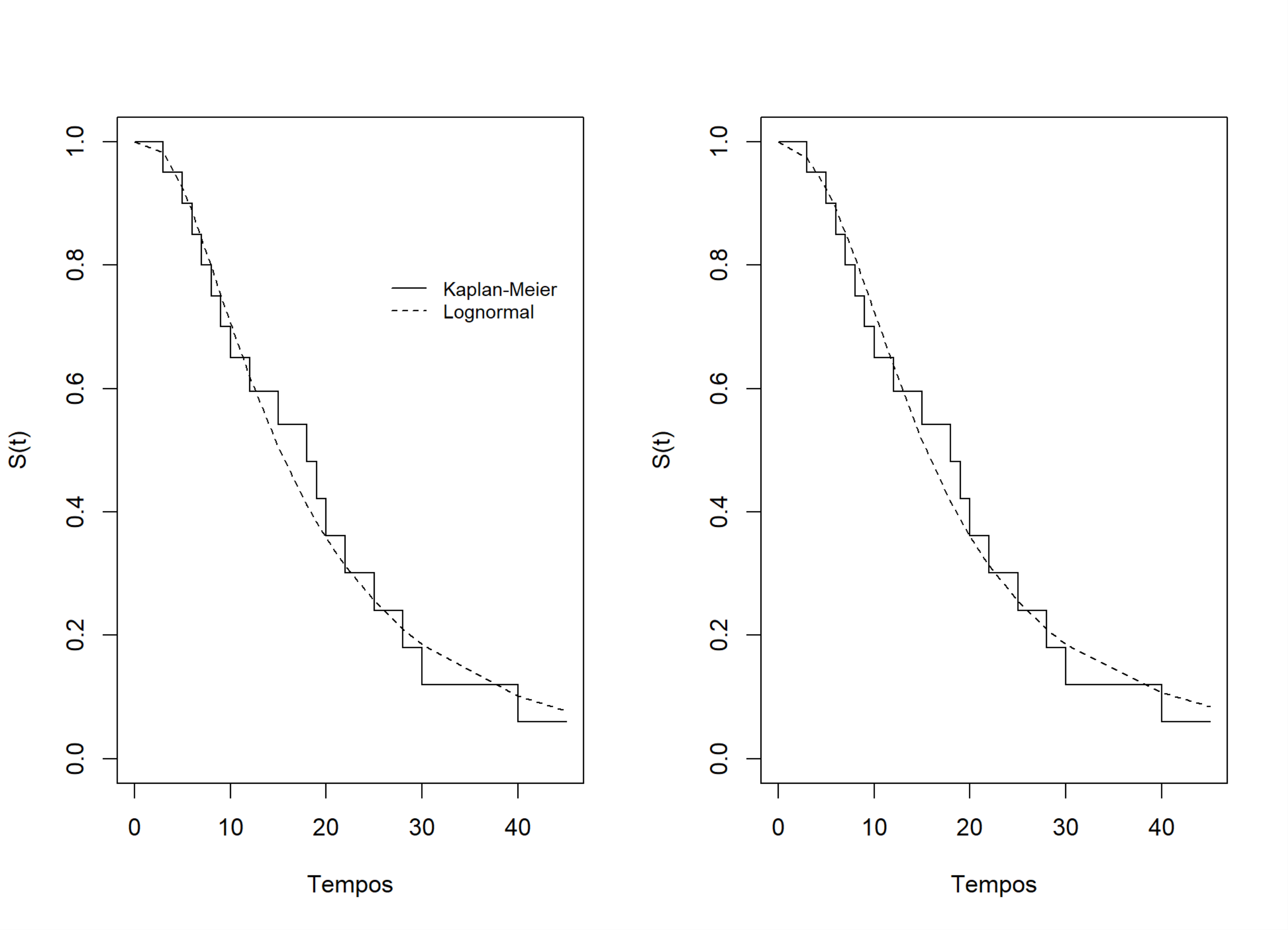
4a Comparação dos Modelos

```r
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stln), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Lognormal"),
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
```



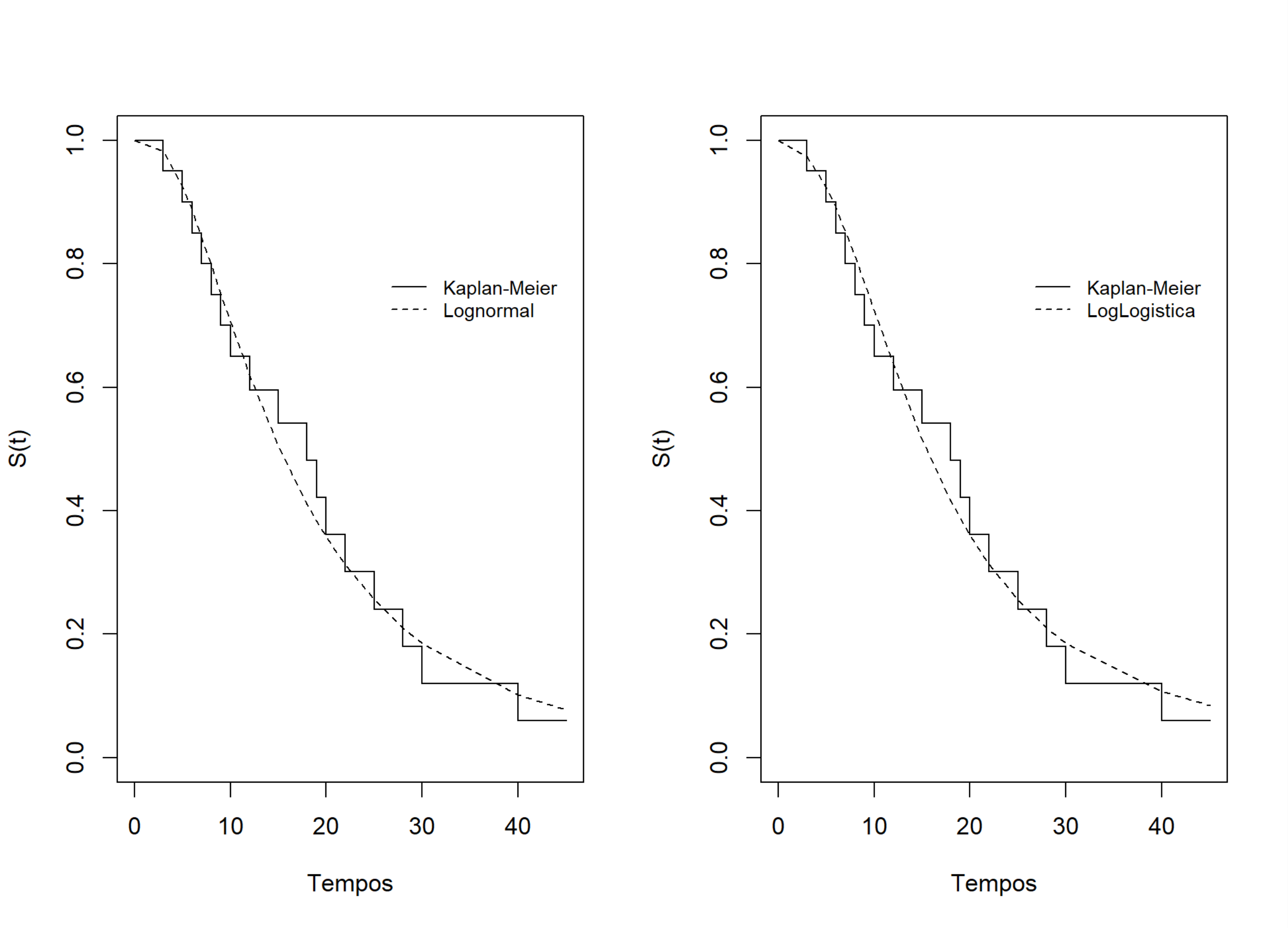
4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stln), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Lognormal"),
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stll), lty=2)
```



4a Comparação dos Modelos

```
par(mfrow=c(1,2))
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stln), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "Lognormal"),
plot(ekm, conf.int=F, xlab="Tempos", ylab="S(t)")
lines(c(0,tempo),c(1,stll), lty=2)
legend(25,0.8,lty=c(1,2),c("Kaplan-Meier", "LogLogistica
```


S(t)
Tempos
Kaplan-Meier
Lognormal
S(t)
Tempos
Kaplan-Meier
LogLogistica