Analyse de survie Master ISN, TP1

Génia Babykina

Dans ce TP on utilisera un exemple suivant. Nous avons les données sur les durées de deux échantillons, A et B (par exemple, patients traités ou non traités) ainsi que les données simulées. Nous utiliserons les packages {*survival*}, {*survminer*}, {*fitdistrplus*}.

Estimation non-paramétrique de la fonction de survie.

1) Lecture de données

```
exo_cours = read.table("exo_cours.txt")
```

2) Calculer la courbe de survie avec l'intervalle de confiance pour l'échantillon A.

```
S.t=cumprod(c(1,1,1,(1-1/10), (1-0/10), (1-1/9), 1-0/9, 1-0/8, 1-1/7, 1-0/5,
              1-2/5, 1-0/3, 1-0/3, 1-0/2, 1-0/2, 1-1/2, 1-0/1, 1-0/1, 1-0/1,
              1-0/1, 1-0/1, 1-0/1, 1-0/1, 1-0/1, 1-0/1))

#autrement
Di.a = c(0, 0, 0, 1, 0, 1, 0, 0, 1,0,2,0,0,0,0,1,0,0,0, 0,0,0,0,0,0)
Ri.a = c(10, 10, 10, 10,  9, 9, 8,8, 7, 5, 5, 3, 3, 2, 2,2, 1,1,1,1,1,1,1,1,1 )
S.t.bis= cumprod(1-Di.a/Ri.a)
se.S.t_a = (S.t*sqrt(cumsum(Di.a /((Ri.a-Di.a)*Ri.a))))
fit=survfit(Surv(temps, event)~1, data=subset(exo_cours, ech=="A"), conf.type="plain")
summary(fit)
ll=S.t-se.S.t_a*qnorm(1-0.05/2)
ul=S.t+se.S.t_a*qnorm(1-0.05/2)
```

3) Représenter graphiquement la courbe de survie pour l'échantillon A.

```
plot(0:24, S.t, type="s" )
plot(fit, lwd=2, conf.type = "log")
lines(0:24, S.t , type="s", col="red")
lines(0:24, ll , type="s", col="red", lty=2)
lines(0:24, ul, type="s", col="red", lty=2)
```

4) Calculer la courbe pour l'échantillon B, superposer les deux courbes sur le graphique, commenter.

5) Tester les differents types d'intervalle de confiance :

- "plain" (obtenu par le delta-method)
- "log-log", "log" : par transformation de $S(t)$ en log (double log)

6) Représenter graphiquement le hasard cumulé $H(t)$ pour l'échantillon A. Rappel :

$$S(t) = \exp\left(-H(t)\right) = \exp\left(-\int_0^t h(u)du\right)$$

```r
plot(survfit(Surv(temps, event)~1,
             data=subset(exo_cours, ech=="A")),
             fun="cumhaz", main="Hasard cumulé")
```

7) Effectuer le test de log-rank à l'aide du fonction *survdiff*.

Sous l'hypothèse d'indépendance $d_{0i} \sim$ distribution hypergéométrique $\mathcal{H}\left(N = n_i, n = d_i, p = \frac{n_{0i}}{n_i}\right)$

$$p(d_{0i}|n_{0i}, n_{1i}, d_i) = \frac{\binom{n_{0i}}{d_{0i}}\binom{n_{1i}}{d_{1i}}}{\binom{n_i}{d_i}}$$

L'espérance de d_{0i} est donnée par

$$e_{0i} = E(d_{0i}) = \frac{n_{0i}d_i}{n_i}$$

et sa variance par :

$$v_{0i} = \text{var}(d_{0i}) = \frac{n_{0i}n_{1i}d_i(n_i - d_i)}{n_i^2(n_i - 1)}$$

En sommant sur les N instants auxquels se produisent des événement

$$U_0 = \sum_{i=1}^{N}(d_{0i} - e_{0i}) = \sum d_{0i} - \sum e_{0i}$$

$$\text{var}(U_0) = \sum v_{0i} = V_0$$

Ainsi, on peut construire une statistique de test qui a une distribution normale

$$\frac{U_0}{\sqrt{V_0}} \sim \mathcal{N}(0, 1)$$

où de manière équivalente

$$\frac{U_0^2}{V_0} \sim \chi_1^2$$

```r
# Exemple de calcul "à la main"
Patient = 1:6
Survtime = c(6,7,10,15,19,25)
Censor = c(1,0,1,1,0,1)
Group = c("C","C","T","C","T","T")
data = cbind.data.frame(Patient = Patient, Survtime = Survtime ,
                        Censor = Censor,
                        Group = Group)
temps = unique(Survtime[Censor == 1])
U0 = 0
V0 = 0
ti = 6
lignes = c()
for (ti in temps){
  X = subset(data,Survtime >= ti)
  ni = nrow(X)
  di = sum(X$Survtime == ti)
  d0i = sum((X$Group == "C")&(X$Survtime == ti)&(X$Censor == 1))
```

```r
  d1i = sum((X$Group == "T")&(X$Survtime == ti)&(X$Censor == 1))
  n0i = sum(X$Group == "C")
  n1i = sum(X$Group == "T")
  M = matrix(c(d0i,d1i,n0i - d0i,n1i - d1i),2,2,byrow = T)
  dimnames(M) = list(c("Failure","Non-failure"),c("Control","Treatement"))
  print(paste("Tableau pour ti =",ti))
  print(M)
  e0i = n0i*di/ni
  v0i = n0i*n1i*di*(ni-di)/(ni^2*(ni - 1))
  if (ni == 1) v0i = 0
  lignes = rbind.data.frame(lignes,c(ti,ni,di,n0i,d0i,n1i,d1i,e0i,v0i))
}
colnames(lignes) = c("ti","ni","di","n0i","d0i","n1i","d1i","e0i","v0i")
lignes = round(lignes,4)
lignes
sum(lignes$d0i)
sum(lignes$e0i)
U0 = sum(lignes$d0i) - sum(lignes$e0i)
U0
V0 = sum(lignes$v0i)
V0
X2 = U0^2/V0
X2
pchisq(X2,df = 1,lower.tail = F) # p-value du test statistique
```

On note l'équivalence suivante :

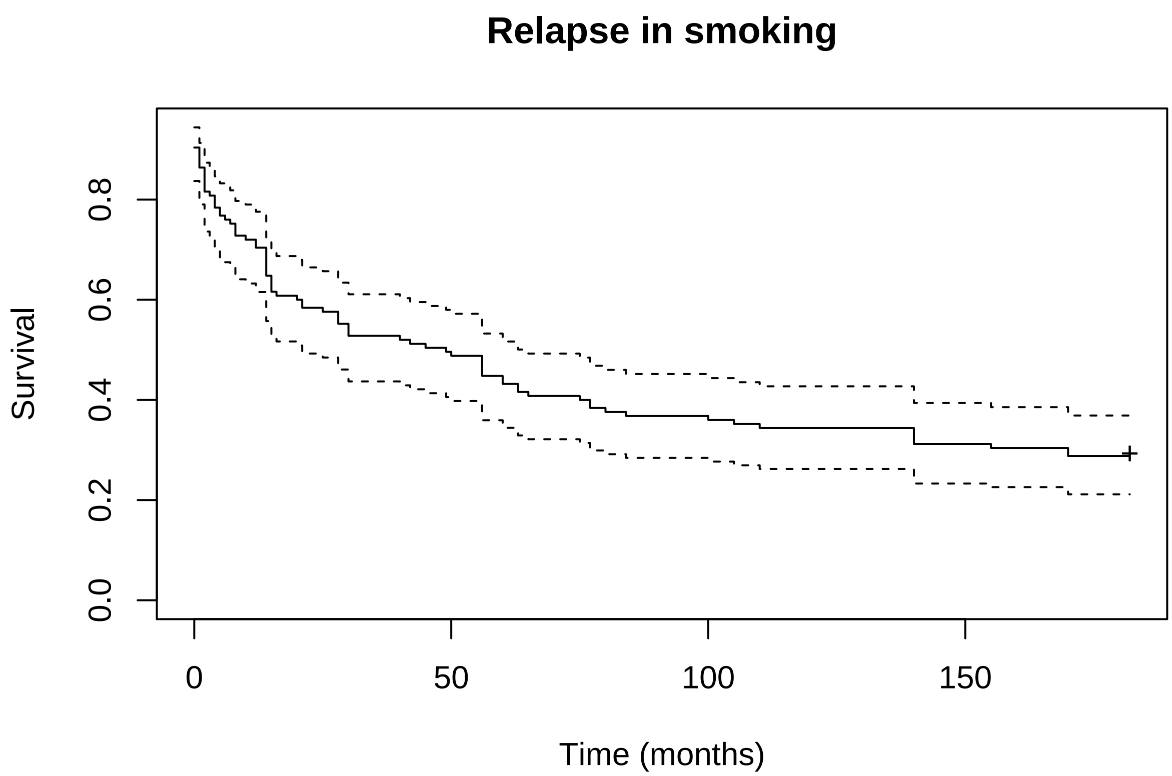
$$u = \frac{u_0}{\sqrt{Var(u_0)}} \sim \mathcal{N}(0,1) \Leftrightarrow \frac{u_0^2}{Var(u_0)} \sim \chi_1^2$$

```r
log.rank.test=survdiff(Surv(temps, event)~ech, data=exo_cours)
```

Description de données pharmacoSmoking

8) Estimation de Kaplan-Meier

```r
data=read.csv2("/Users/babykinaevgenia/Documents/recherche/christophe/cours_isn/TP/smoking.csv")
result.km = survfit(Surv(ttr, relapse)~1, conf.type="log-log", data=data)
plot(result.km, conf.int=TRUE, mark="+", xlab="Time (months)", ylab ="Survival")
title("Relapse in smoking")
```




```
## Les quantiles
quantile(result.km)
```

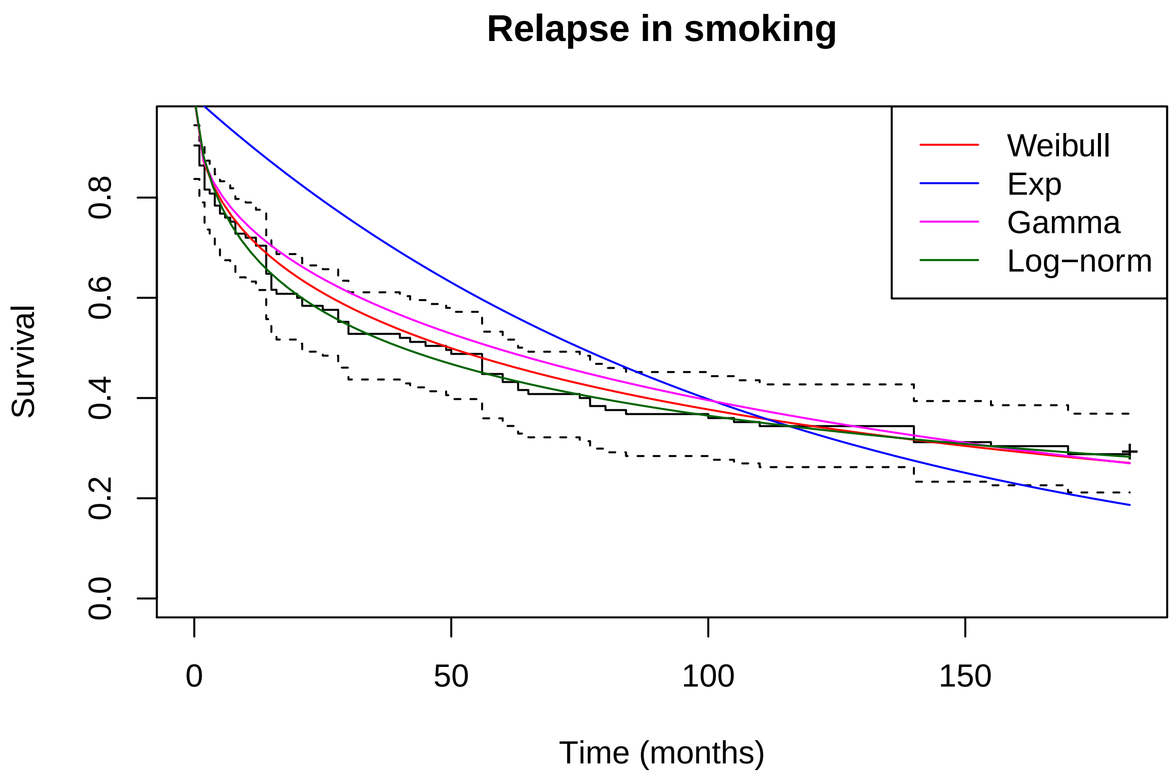
9) Estimation des paramètres pour loi de Weibull, Gamma et Exponentielle. Ajouter les fonctions de survie estimées sur l'estimation non-paramétrique de Kaplan-Meier

```
library(fitdistrplus)
plot(result.km, conf.int=TRUE, mark="+", xlab="Time (months)", ylab ="Survival")
title("Relapse in smoking")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
left=data[,"ttr"]
left[left== 0 ] = 0.5
right=ifelse(data[,"relapse"]==1, left, NA)
datacens=cbind.data.frame(left=left, right=right)
par_weib=fitdistcens(datacens, "weibull")
curve(pweibull(x, shape=par_weib$estimate["shape"],
```

```
                scale=par_weib$estimate["scale"], lower.tail=FALSE), add=TRUE, col="red")

par_exp=fitdistcens(datacens, "exp")
curve(pexp(x,
                rate=par_exp$estimate["rate"], lower.tail=FALSE), add=TRUE, col="blue")
par_gamma=fitdistcens(datacens, "gamma")
curve(pgamma(x, shape=par_gamma$estimate["shape"],
                rate=par_gamma$estimate["rate"], lower.tail=FALSE), add=TRUE, col="magenta")
par_lnorm=fitdistcens(datacens, "lnorm")
curve(plnorm(x, meanlog=par_lnorm$estimate["meanlog"],
                sdlog=par_lnorm$estimate["sdlog"], lower.tail=FALSE), add=TRUE, col="darkgreen")
legend("topright",
       legend=c("Weibull", "Exp", "Gamma", "Log-norm"),
       col=c("red", "blue", "magenta", "darkgreen"),
       lty=rep(1,4))
```



10) Estimation des paramètres de la loi de Weibull "à la main". On utilisera le paramétrage suivant:

$$S(t) = \exp\left(-\left(\frac{t}{\theta}^{\delta}\right)\right), h(t) = \left(\frac{1}{\theta}\right)^{\delta} \times \delta \times t^{\delta-1}$$

.

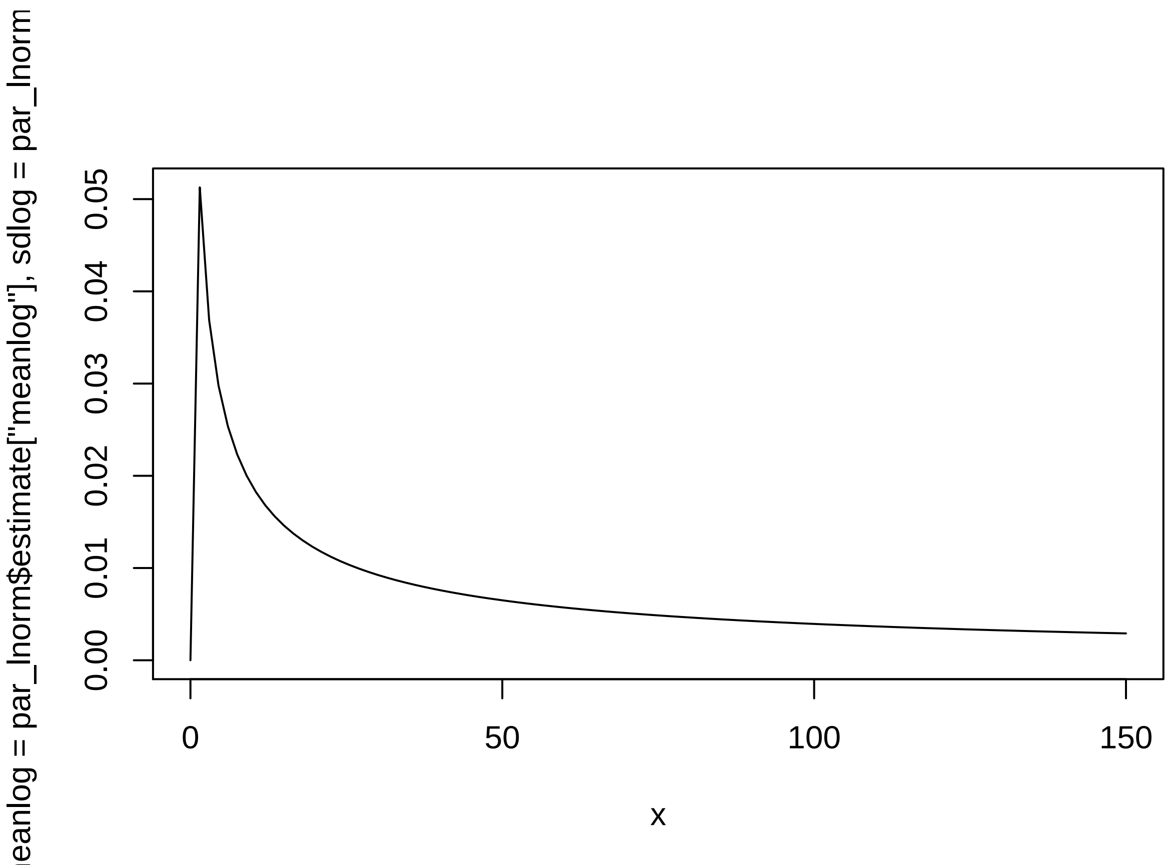
```
ti=data$ttr
#ti[which(ti==0)]=0.5
ci=data$relapse
  LnLweib_opt = function(x){
    sc=x[1]
    sh=x[2]

    sum(log(
```

```
    (((1/sc)^sh)*sh*(ti^(sh-1)))^ci*exp(- ((1/sc)^(sh))*(ti^sh))
   )
   )
  }
  res_opt=maxNR(LnLweib_opt, start=c(100, 1))
```

11) Ecrire la vraisemblance pour la loi exponentielle de durée, maximiser la vraisemblance analytiquement, comparer les résultats obtenus aux résultats numériques.

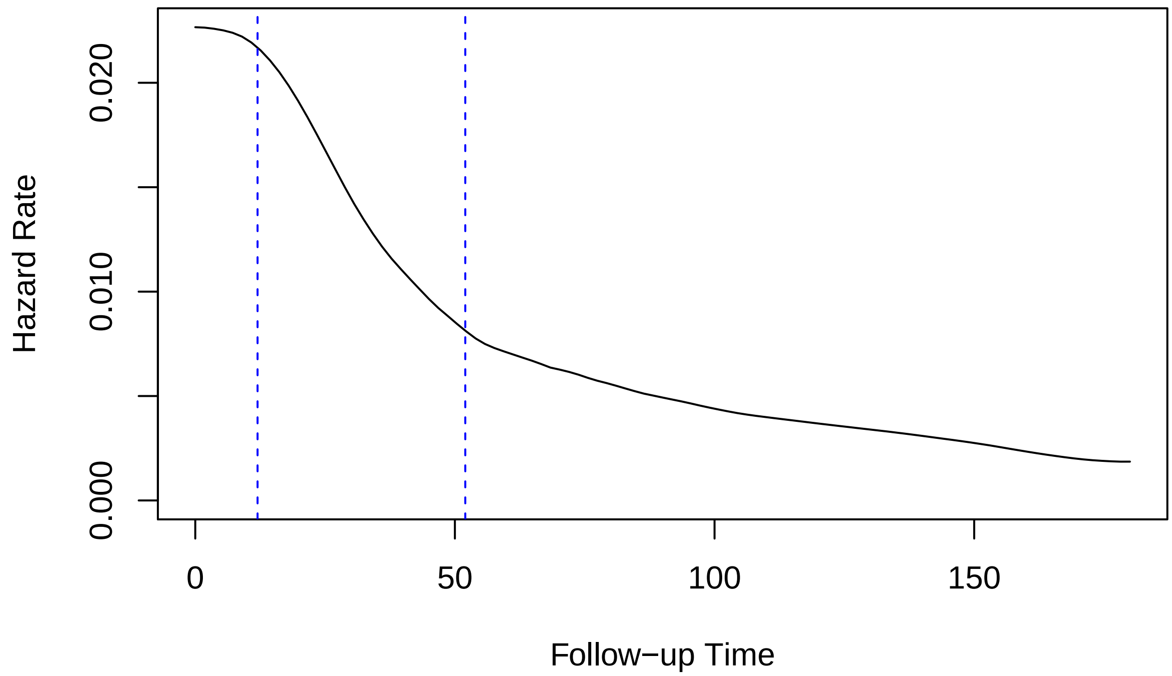
12) Tracer la fonction de hasard aussociée à la distribution log-normale estimée :

```
lognormHaz<-{function(x, meanlog, sdlog) dlnorm(x, meanlog=meanlog,
                                               sdlog=sdlog)/
  plnorm(x, meanlog=meanlog, sdlog=sdlog, lower.tail=FALSE)}
curve(lognormHaz(x, meanlog=par_lnorm$estimate["meanlog"],
                   sdlog=par_lnorm$estimate["sdlog"]),
      xlim=c(0, 150))
```



13) Estimation non-paramétrique (noyau) de la fonction de hasard :

```
library(muhaz)
ti=data[,"ttr"]
ci=data[,"relapse"]
fit=muhaz(ti, ci, min.time=0, max.time=180)
plot(fit)
abline(v=c(12,52), lty=2, col="blue")
```



##

Estimation paramétrique de la fonction survie.

14) Représenter graphiquement et commenter les distribution des durées suivant différentes lois :

$$T \sim Weibull\,(3.5, 2.2)\,, T \sim Exp\,(0.2)\,, T \sim LogN\,(2.5, 1.2)$$

.

Remarque : le paramétrage des loi de probabilité peut être différent. Par exemple, pour la loi de Weibull :

	Paramétrage 1 (cours)	Paramétrage 2 (R)
$H(t)$	$H(t) = \alpha t^{\gamma} = \left(\frac{1}{1/(\alpha^{1/\gamma})}t\right)^{\gamma}$	$H(t) = \left(\frac{x}{b}\right)^{a}$

```
curve(dweibull(x,shape=2.2, scale=3.5), col="red", from=0, to=30)
#curve(2.2*(1/3.5)^2.2*x^(2.2-1)*exp(-(1/3.5)^2.2*x^2.2),
      add=TRUE, col="black")
curve(dexp(x,rate=0.2), col="blue", add=TRUE)
#curve(0.2*exp(-0.2*x), col="red", add=TRUE)
curve(dlnorm(x, meanlog = 2.5, sdlog = 1.2, log = FALSE),
                                col="black", add=TRUE)
```

15) Générer un temps d'événement suivant la loi de Weibull de paramètres shape 1.2 et scale=0.2, représenter graphiquement la variable et les fonctions associées (la survie, le hasard, la fonction de densité). Changer les paramètres de la loi de Weibull, commenter. Remarque : la médiane de la loi de Weibull avec scale λ et shape γ est égale à $\lambda \times \log(2)^{1/\gamma}$

```
T_weib = rweibull(n=100, shape=1.2, scale=0.2)
hist(T_weib, probability=TRUE, main="f(t)")
abline(v=log(2)^(1/1.2)*0.2, col="red") # médiane
#median(T_weib)
curve(dweibull(x, shape=1.2, scale=0.2), add=TRUE, col="red")

curve(pweibull(x, shape=1.2, scale=0.2, lower.tail=FALSE),
              from=0, to=0.8, ylim=c(0,1),
              xlab="Temps", main="S(t)", ylab="")
```

```r
curve(dweibull(x, shape=1.2, scale=0.2)/
        pweibull(x, shape=1.2, scale=0.2,lower.tail=F),
        xlab="Temps", main="h(t)", ylab="")
```

16) Répeter la question précédente pour le temps suivant la loi exponentielle de paramètre $\lambda = 1/5$. La médiane de la loi de Weibull est égale à $\log(2)/\lambda$

17) Estimer les paramètres de la distriution de Weibull et Exponentielle pour les variables aléatoires générées dans les questions prcédentes. Attention au paramètrage.

```r
ff_weib=fitdist(T_weib, distr="weibull")
ff_exp=fitdist(T_exp, distr="exp")
rate=ff_exp$estimate
shape=ff_weib$estimate[1]
scale=ff_weib$estimate[2]
# Moyennes observéee
mean(T_weib) ; mean(T_exp)
# Médianes observées
median(T_weib) ; median(T_exp)
# Espéerance de la loi exponentielle
mean_exp=1/rate
# Espéerance de Weibull
mean_weib=scale*gamma(1+shape)
# Médiane de loi exponentielle
med_exp=log(2)/rate
# Médiane de Weibull
med_weib=scale*log(2)^(1/shape)
```

Visualisation améliorée

```r
library(survminer)
library(ggplot2)
fit1 = survdiff(Surv(ttr, relapse)~grp, data=data)
fit2 = survfit(Surv(ttr, relapse)~grp, data=data)
plot(fit2, xlab="Time (days)", ylab= "Relapse probability",
     col=c("blue", "red"))
legend("topright", legend=c("combination", "patch only"), col=c("blue", "red") , lty=c(1,1))

# ou bien
ggsurvplot(fit2, data=data)

ggsurvplot(fit2, risk.table = TRUE, pval=TRUE, conf.int=TRUE,
           ggtheme=theme_minimal(),
           risk.table.y.text.col=TRUE,
           risk.table.y.text=FALSE, data=data)
```