

Anexo 2. Código implementado en RStudio

```
#####  
###CODIGO - ESTIMACIÓN Y AJUSTE DE LAS DISTRIBUCIONES CANDIDATAS###  
#####  
###LEER LAS SGTES LIBRERIAS###  
library(maxLik); library(rootSolve); library(NADA); library(dplyr); library(survival); library(MASS)  
library(fitdistrplus); library(LindleyR); library(rmutl); library(zipfR)  
  
#####LEER ARCHIVO DE DATOS#####  
#Un archivo de datos de dos columnas. En la primera columna los tiempos de supervivencia se identifican con "0" si hay censura y "1" si no hay censura. En  
la segunda columna se reportan los tiempos correspondientes#  
BDcancer= read.table(file = "clipboard", header=TRUE)  
#Otra forma 1  
#Un archivo de datos de dos columnas, respectivamente, llamadas "left" y "right", que describen cada valor observado como un intervalo. La columna derecha  
contiene NA para las observaciones censuradas a la derecha. Si no hay censura, cada observación aparecerá en los dos lados de intervalo#  
BDcancer1= read.table(file = "clipboard", header=TRUE)  
#Otrs forma 2  
#Un archivo de datos de dos columnas. En la primera columna están tiempos de supervivencia. En la segunda columna, se coloca "TRUE" cuando es una  
observación censurada, en caso contrario, "FALSE".  
BDcancer2= read.table(file = "clipboard", header=TRUE)  
#Datos sin censura  
BDcancer3= read.table(file = "clipboard", header=TRUE)  
#Datos con censura  
BDcancer4= read.table(file = "clipboard", header=TRUE)  
  
#####HISTOGRAMAS Y BOXPLOT###  
par(mfrow=c(1,2))  
h=hist(BDcancer[,2], xlab="Tiempo (meses)", breaks=22, ylab="Frecuencia", main="", col="lemonchiffon3", freq=TRUE, mar = c(5.1, 4.1, 4.1, 4), mgp = c(2.8,  
0.8, 0), cex = 1.3, cex.lab=1.5, cex.main = 2.5, cex.axis=1.5, las= 0)  
  
#####INFERENCIA CLÁSICA#####  
#inicialización# j=1  
#####DISTRIBUCIONES COMÚNES#####  
###Distribucion Exponencial  
#Estimación directa del parametro "beta"  
Qexp=sum(BDcancer[,2^j-1])/sum(BDcancer[,2^j])  
#Error estándar del estimador de "beta"  
Q1exp=sqrt(((sum(BDcancer[,2^j-1]))/(sum(BDcancer[,2^j]))^2))  
#Otra forma  
ajusExp <- fitdistscens(BDcancer1,"exp")  
  
#####Distribucion Gamma  
log.gam <- function(param) {  
  alpha <- param[1]  
  beta <- param[2]  
  sum(BDcancer[,2^j-1]*(log(dgamma(BDcancer[,2^j]), shape = alpha, scale = 1/beta)))+sum((1-BDcancer[,2^j-1])*log(1-pgamma(BDcancer[,2^j]), shape =  
alpha, scale = 1/beta, lower.tail = TRUE, log.p = FALSE)))  
}  
Q.ga<- maxLik(log.gam, start=c(0.03,1.2), method = "NM")  
#Estimaciones "alpha" y "beta"  
Qga<-Q.ga$estimate[1]  
Qga1<-Q.ga$estimate[2]  
#Errores estándar  
Qga2<- solve(Q.ga$hessian)[1]  
Qga3<- solve(Q.ga$hessian)[4]  
#Otra forma  
ajusGa <- fitdistscens(BDcancer1,"gamma", optim.method="Nelder-Mead")  
  
#####Distribucion Weibull  
log.we <- function(param) {  
  alpha<-param[1]  
  beta <- param[2]  
  sum((BDcancer[,2^j-1])*log(dweibull(BDcancer[,2^j], shape = alpha, scale = 1/beta, log = FALSE)))+sum(((1-BDcancer[,2^j-1])*log(1-pweibull(BDcancer[,2^j],  
shape = alpha, scale = 1/beta, lower.tail = TRUE, log.p = FALSE)))  
}  
Q.we<- maxLik(log.we, start=c(0.8, 0.2), method = "NM")  
#Estimaciones "alpha" y "beta"  
Qwe<-Q.we$estimate[1]  
Qwe1<-Q.we$estimate[2]  
#Errores estándar  
Qwe2<- solve(Q.we$hessian)[1]  
Qwe3<- solve(Q.we$hessian)[4]  
#Otra forma  
ajusWe <- fitdistscens(BDcancer1,"weibull", optim.method="Nelder-Mead", start=list(a=1,b=1))
```

```

#####Distribucion LogNormal
log.lg <- function(param) {
  mu <- param[1]
  sigma <-param[2]
  sum((BDcancer[,2]*-1)*log(dlnorm(BDcancer[,2*], meanlog = mu, sdlog = sigma, log = FALSE)))+sum((1-BDcancer[,2*]-1)*log(1-plnorm(BDcancer[,2*],
meanlog = mu, sdlog = sigma, lower.tail = TRUE, log.p = FALSE)))
}
Q.lg <- maxLik(log.lg, start=c(1,1), method = "NM")
#Estimaciones "mu" y "sigma"
Qlg<-Q.lg$estimate[1]
Qlg1<-Q.lg$estimate[2]
#Errores estándar
Qlg2<- solve(Q.lg$hessian)[1]
Qlg3<- solve(Q.lg$hessian)[4]

#Otra forma
ajusLn <- fitdistcens(BDcancer1,"lnorm", optim.method="Nelder-Mead")

#####DISTRIBUCIONES ALTERNATIVAS#####
#####Distribucion Lindley
log.lind <- function(param) {
  beta <- param
  sum((BDcancer[,2*]-1)*log(dlindley(BDcancer[,2*], theta = beta, log = FALSE)))+ sum((1-BDcancer[,2*]-1)*log(1-plindley(BDcancer[,2*], theta = beta, lower.
tail = TRUE, log.p = FALSE)))
}
Q.lind<- maxLik(log.lind, start=0.1, method = "NR")
#Estimación "beta"
Qlind<-Q.lind$estimate
#Error estándar
Qlind1<- solve(Q.lind$hessian)
#Otra forma
ajusLind <- fitdistcens(BDcancer1,'lindley', start = list(theta=0.2))

#####Distribucion Power Lindley Inversa
log.pw=function(param){
  mean<- param[1]
  beta <- param[2]
  sum((BDcancer[,2*]-1)*log(dgenilindley(BDcancer[,2*], theta = beta, alpha = mean, log = FALSE)))+sum((1-BDcancer[,2*]-1)*log(1-pgenilindley(BDcancer[,2*],
theta = beta, alpha = mean, lower.tail = TRUE, log.p = FALSE)))
}
Q.pw <- maxLik(log.pw, start=c(mean=0.5, beta=1), method = "NM")
#Estimaciones "mu" y "beta"
Qpw<-Q.pw$estimate[1]
Qpw1<-Q.pw$estimate[2]
#Errores estándar
Qpw2<- solve(Q.pw$hessian)[1]
Qpw3<- solve(Q.pw$hessian)[4]
#Otra forma
ajusPw <- fitdistcens(BDcancer1,"genilindley", start = list(theta = 1, alpha = 0.3), optim.method="Nelder-Mead")

#####Distribucion Exponencial Power Lindley
log.lge <- function(param) {
  alpha<-param[1]
  beta <- param[2]
  lambda <- param[3]
  sum((BDcancer[,2*]-1)*log(dexplindley(BDcancer[,2*], beta, alpha, theta, log = FALSE)))+sum((1-BDcancer[,2*]-1)*log(1-pexplindley(BDcancer[,2*],
beta, alpha, theta, lower.tail = TRUE, log.p = FALSE))))
}
Q.lge <- maxLik(log.lge, start=c(0.5,0.3), method = "NM")
#Estimaciones "alpha" y "beta", fijo "lambda=4.12"
Qlge<-Q.lge$estimate[1]
Qlge1<-Q.lge$estimate[2]
#Errores estándar
Qlge2<- solve(Q.lge$hessian)[1]
Qlge3<- solve(Q.lge$hessian)[4]

#Otra forma
ajusLge <- fitdistcens(BDcancer1,"explindley", start = list(theta =0.3, alpha =0.5))

#####Distribucion Lévy estándar
log.lev <- function(param) {
  beta <- param
  sum((BDcancer[,2*]-1)*log(dlevy(BDcancer[,2*], m=0, s=beta)))+sum((1-BDcancer[,2*]-1)*log(1-plevy(BDcancer[,2*], m=0, s=beta)))
}
Q.lev<- maxLik(log.lev, start=0.01, method = "NR")
#Estimación "beta"
Qlev<-Q.lev$estimate[1]

```

```

#Error estándar
Qlev1<- solve(Q.lev$hessian)
#Otra forma
dlevy<-function(x,s,m) ((s/2*pi)^(1/2))*(1/(x^(3/2)))*(exp(-(s/(2*x))))
plevy<-function(q,s,m) 1-(2*pnorm(sqrt(s/q))-1)
qlevy<-function (p,s,m) s/qnorm(1 - p/2)^2
ajusLev <- fitdistscens(BDcancer1, "levy", start = list(s=0.06))

#####BONDAD DE AJUSTE#####
#####CRITERIOS DE INFORMACIÓN
n=88 #Tamaño de la muestra
AICcExp=-2*ajusExp$loglik+2*(1)*(n/(n-1)); AICcGa=-2*Q.ga$maximum+2*(2)*(n/(n-2-1))
AICcWe=-2*Q.we$maximum+2*(2)*(n/(n-2-1)); AICcLn=-2*Q.lg$maximum+2*(2)*(n/(n-2-1))
AICcLind=-2*Q.lind$maximum+2*(1)*(n/(n-1-1)); AICcPw=-2*Q.pw$maximum+2*(2)*(n/(n-2-1))
AICcLge=-2*Q.lge$maximum+2*(3)*(n/(n-3-1)); AICcLev=-2*Q.lev$maximum+2*(1)*(n/(n-1-1))
#Resumen (LogL, AIC, BIC, AICc)
mB=data.frame("LogL"=c(-2*ajusExp$loglik,-2*Q.ga$maximum,-2*Q.we$maximum,-2*Q.lg$maximum,-2*Q.lind$maximum,
-2*Q.pw$maximum,-2*Q.lge$maximum,-2*Q.lev$maximum), "AIC"=c(ajusExp$aic, ajusGa$aic, ajusWe$aic, ajusLn$aic,
ajusLind$aic,ajusPw$aic, ajusLge$aic,ajusLev$aic),"BIC"=c(ajusExp$bic, ajusGa$bic, ajusWe$bic, ajusLn$bic, ajusLind$bic,
ajusPw$bic, ajusLge$bic, ajusLev$bic), "AICc"=c(AICcExp, AICcGa, AICcWe, AICcLn, AICcLind, AICcPw, AICcLge, AICcLev)); Mb

#####ANALISIS GRÁFICO#####
#####AJUSTE DE LAS DISTRIBUCIONES VS EL HISTOGRAMA DE LOS DATOS#####
h1=hist(BDcancer[,1], xlab="Tiempo (meses)", breaks=22, ylab="Densidad", main="", col="lemonchiffon3", freq=FALSE, ylim=c(0, 0.15), mar = c (5.1, 4.1, 4.1, 4), mgp = c (2.8, 0.8, 0), cex = 1.3, cex.lab=1.8, cex.main = 2.5, cex.axis=1.5, las= 0)
curve(dexp(x, rate=Qexp), col=2, lty=1, lwd=4, add=T)
curve(dgamma(x, shape=Qga, rate = Qga1), col=3, lty=2, lwd=4, add=T)
curve(dweibull(x, shape=0.8455418, scale=1/Qwe1), col=4, lty=3, lwd=4, add=T)
curve(dlnorm(x, meanlog = Qlg, sdlog = Qlg1), col=5, lty=4, lwd=4, add=T)
curve(dlindley(x, theta=Qlind), col=6, lty=5, lwd=4, add=T)
curve(dgenilindley(x, theta=Qpw1, alpha=Qpw), col=7, lty=6, lwd=4, add=T)
curve(dexpplindley(x, theta=Qlge1, alpha=Qlge, beta=Qlge2), col="peru", lty=7, lwd=4, add=T)
curve(dlevy(x, m=0, s=Qlev), col="red4", lty=6, lwd=4, add=T)
legend(110, 0.13, bty="n", c("Exp", "Gam", "Weib", "LogN", "Lind", "PwL", "EPwL", "Lév"), col = c(2,3,4,5,6,7,"peru", "red4", cex = 1.6), lty=c(1,2,3,4,5,6,7,8), lwd = 4, cex=1.6)

#####DISTRIBUCIÓN EMPIRICA VS TEÓRICA#####
mycenfit=cefit(BDcancer2[,1], BDcancer2[,2])
plot(mycenfit, conf.int=TRUE, main="", ylab="Probabilidad acumulada", xlab="Tiempo (meses)", mar = c (5.1, 4.1, 4.1, 4), mgp = c (2.6, 0.8, 0), cex = 1.3, cex.lab=1.8, cex.main = 2.5, cex.axis=1.5, las= 0)
#Exponencial
supExp=exp(-Qexp*sort(BDcancer1[,2]))
fdaExp=1-supExp
par(new=TRUE)
lines(sort(BDcancer1[,2]), fdaExp, type = "l", lwd=4, col="2", lty=1, ylab = "", xlab = "Tiempo", main="", cex.lab=1.1, cex=1.5, cex.main=1.5)
#Gamma
supGa=(1-(lgamma(Qga, sort(BDcancer[,2])^Qga1, lower=TRUE)/gamma(Qga)))
fdaGa=1-supGa
par(new=TRUE)
lines(sort(BDcancer[,2]), fdaGa, type = "l", lwd=4, col="3", lty=2, ylab = "", xlab = "Tiempo", main="", cex.lab=1.1, cex=1.5, cex.main=1.5)
#Weibull
supWe=exp(-(Qwe1*sort(BDcancer[,2]))^(Qwe))
fdaWe=1-supWe
par(new=TRUE)
lines(sort(BDcancer[,2]), fdaWe, type = "l", lwd=4, col="4", lty=3, ylab = "", xlab = "Tiempo", main="", cex.lab=1.1, cex=1.5, cex.main=1.5)
#LogN
supLn=(1-pnorm((log(sort(BDcancer[,2]))-Qlg)/Qlg1))
fdaLn=1-supLn
par(new=TRUE)
lines(sort(BDcancer[,2]), fdaLn, type = "l", lwd=4, col="5", lty=4, ylab = "", xlab = "Tiempo", main="", cex.lab=1.1, cex=1.5, cex.main=1.5)
#Lindley
supLind=(1+((Qlind*sort(BDcancer[,2]))/(1+Qlind)))*exp(-Qlind*sort(BDcancer[,2]))
fdaLind=1-supLind
par(new=TRUE)
lines(sort(BDcancer[,2]), fdaLind, type = "l", lwd=4, col="6", lty=5, ylab = "Función de supervivencia, S(x)", xlab = "Tiempo", main="", cex.lab=1.1, cex=1.5, cex.main=1.5)
#PwL
supPw=(1-(exp(-(Qpw1/sort(BDcancer[,2])^Qpw)))+(exp(-(Qpw1/sort(BDcancer[,2])^Qpw))*((Qpw1)/(sort(BDcancer[,2])^Qpw*(1+Qpw1))))))
fdaPw=1-supPw
par(new=TRUE)
lines(sort(BDcancer[,2]), fdaPw, type = "l", lwd=4, col="7", lty=1, ylab = "", xlab = "Tiempo", main="", cex.lab=1.1, cex=1.5, cex.main=1.5)
#EPwL
supLge=(1-pexpplindley(sort(BDcancer[,2]), Qlge1, Qlge, 4.12, lower.tail = TRUE, log.p = FALSE))
fdaLge=1-supLge
par(new=TRUE)
lines(sort(BDcancer[,2]), fdaLge, type = "l", lwd=4, col="peru", lty=1, ylab = "", xlab = "Tiempo", main="", cex.lab=1.1, cex=1.5, cex.main=1.5)
#Lév

```

```

supLev=erf(sqrt(Qlev/(2*sort(BDcancer[,2])))
fdaLev=1-supLev
lines(sort(BDcancer[,2]), fdaLev, type = "l", lwd=4, col="magenta", lty=8, ylab = "", xlab = "Tiempo", main="", cex.lab=1.1, cex=1.5, cex.main=1.5)
legend(50, 0.83, c("Exp", "Gam", "Weib", "LogN", "Lind", "PwLl", "EPwL", "Lév"), col = c(2,3,4,5,6,7,"darkmagenta", "red4"), lty=1, bty="n", lwd = 4, cex = 1.6)
legend(50, 0.47, c("Exp", "Gam", "Weib", "LogN"), col = c(2,3,4,5), lty=1, lwd = 4, cex = 1.6, bty="n")
legend(50, 0.47, c("Lind", "PwLl", "EPwL", "Lév"), col = c(6,7,"peru", "red4"), lty=1, lwd = 4, cex = 1.6, bty="n")
#####FUNCIONES DE RIESGO#####
R.exp=plot(0, type = "l", xlim=c(0, max(BDcancer[,2])), main="", las = 0, ylab = "Función de riesgo", xlab="Tiempo (meses)", ylim=c(0,0.2), mar = c (5.1, 4.1, 4.1, 4), mgp = c (2.8, 0.8, 0), cex.lab=1.8, cex.main = 1.8, cex.axis=1.5, las= 0)
abline(h=Qexp, col=2, lty=1, lwd=5)
R.ga=((Qga1^Qga)*(sort(BDcancer[,2])^(Qga-1))*exp(-Qga1*sort(BDcancer[,2])))/((gamma(Qga)*(1-(lgamma(Qga, sort(BDcancer[,2])*Qga1, lower=TRUE)/gamma(Qga))))))
par(new=TRUE)
lines(sort(BDcancer[,2]), R.ga, type = "l", ylab = "", xlab = "", col=3, lty=1, lwd = 3, main="", cex.lab=1.1, cex.main=1.5, cex=1.5)
R.we=((Qwe*Qwe1)*((Qwe1*sort(BDcancer[,2])^(Qwe1-1)))
par(new=T)
lines(sort(BDcancer[,2]), R.we, type = "l", ylab = "", xlab = "", col=4, lty=1, lwd = 3, main="", cex.lab=1.1, cex.main=1.5, cex=1.5)
R.lg=((1/(sqrt(2*pi)*Qlg1*sort(BDcancer[,2])))*exp(-1/2*((log(sort(BDcancer[,2]))-Qlg)/Qlg1)^2))/(1-pnorm((log(sort(BDcancer[,2]))-Qlg)/Qlg1))
par(new=T)
lines(sort(BDcancer[,2]), R.lg, type = "l", ylab = "", xlab = "", col=5, lty=1, lwd = 4, main="", cex.lab=1.1, cex.main=1.5, cex=1.5)
R.lind=((Qlind^2)*(1+sort(BDcancer[,2]))/(1+Qlind+Qlind*sort(BDcancer[,2])))
par(new=T)
lines(sort(BDcancer[,2]), R.lind, type = "l", ylab = "", xlab = "", col=6, lty=1, lwd = 4, main="", cex.lab=1.1, cex.main=1.5, cex=1.5)
R.pw=((Qpw*Qpw1^2)*(1+sort(BDcancer[,2])^(-Qpw)))/(sort(BDcancer[,2])^(-Qpw1+sort(BDcancer[,2])^(Qpw)*(1+Qpw1)*(exp(Qpw1/
sort(BDcancer[,2])^Qpw-1)))
par(new=T)
lines(sort(BDcancer[,2]), R.pw, type = "l", ylab = "", xlab = "", col=7, lty=1, lwd = 4, main="", cex.lab=1.1, cex.main=1.5, cex=1.5)

R.lge= hexppbindley(sort(BDcancer[,2]), Qlge1, Qlge, Qlge2, log = FALSE)
par(new=T)
lines(sort(BDcancer[,2]), R.lge, type = "l", ylab = "", xlab = "", col="peru", lty=1, lwd = 4, main="", cex.lab=1.1, cex.main=1.5, cex=1.5)

D.lev=((Qlev/2*pi)^(0.5))*1/(sort(BDcancer[,2])^(1.5))*exp(-(Qlev/(2*(sort(BDcancer[,2])))))
S.lev=erf(sqrt(Qlev/(2*(sort(BDcancer[,2])))))
R.lev=D.lev/S.lev
lines(sort(BDcancer[,2]), R.lev, type = "l", ylab = "", xlab = "", col="red4", lty=1, lwd = 4, main="", cex.lab=1.1, cex.main=1.5, cex=1.5)

legend("topright", c("Exp", "Gam", "Weib", "LogN"), col = c(2,3,4,5), lty=1, bty="n", lwd = 3)
legend("topright", c("Lind", "PwLl", "EPwL", "Lév"), col = c(6,7,"peru", "red4"), lty=1, bty="n", lwd = 3)
legend("topright", c("Exp", "Gam", "Weib", "LogN", "Lind", "PwLl", "EPwL", "Lév"), col = c(2,3,4,5,6,7,"peru", "red4"), lty=1, bty="n", lwd = 3)

#####
#Bootstrap - bandas de confianza de la función de supervivencia S(x)
#####
N=88 ## Tamaño de muestra
B=3000 ## Numero de realizaciones Bootstrap

#Bootstrap con la base de datos individual (No censurados)
listaB1=list()
BoosM1=matrix(0, N-(N-sum(BDcancer[,1])), 2*B)
for(j in 1:B){
  listaB1[[j]]=sample_n(BDcancer3, N-(N-sum(BDcancer[,1])), replace = TRUE)
  BoosM1[,2*j-1]<-listaB1[[j]][[1]]
  BoosM1[,2*j]<-listaB1[[j]][[2]]
}
#Bootstrap con la base de datos individual (censurados)
listaB2=list()
BoosM2=matrix(0, N-sum(BDcancer[,1]), 2*B)
for(j in 1:B){
  listaB2[[j]]=sample_n(BDcancer4, N-sum(BDcancer[,1]), replace = TRUE)
  BoosM2[,2*j-1]<-listaB2[[j]][[1]]
  BoosM2[,2*j]<-listaB2[[j]][[2]]
}
#Muestra Bootstrap completa
listaB3=list()
BoosM3=matrix(0, N, 2*B)
for(j in 1:B){
  listaB3[[j]]=rbind(BoosM1[, (2*j-1):(2*j)], BoosM2[, (2*j-1):(2*j)])
  BoosM3[,2*j-1]<-listaB3[[j]][,1]
  BoosM3[,2*j]<-listaB3[[j]][,2]
}

#####
##Estimación de parámetros vía Bootstrap
#####
QexpB=0; Qexp1B=0; QgaB=0; Qga1B=0; Qga2B=0; Qga3B=0; Qwe3B=0; QweB=0; Qwe1B=0; Qwe2B=0; Qwe3B=0;

```

QlgB=0; Qlg1B=0; Qlg2B=0; Qlg3B=0; QlindB=0; Qlind1B=0; QpwB=0; Qpw1B=0; Qpw2B=0; Qpw3B=0;
 QlgeB=0; Qlge1B=0; Qlge2B=0; Qlge3B=0; QlevB=0; Qlev1B=0; Qlev2B=0; Qlev3B=0;

```

for(j in 1:B){
#####Exponencial
QexpB[j]=sum(BoosM3[,2^j-1])/sum(BoosM3[,2^j]);Qexp1B[j]=sqrt(((sum(BoosM3[,2^j-1])/sum(BoosM3[,2^j]))^2))
}

#####Gamma
log.gam <- function(param) {
  alpha <- param[1]; beta <- param[2]
  sum(BoosM3[,2^j-1]*(log(dgamma(BoosM3[,2^j], shape = alpha, scale = 1/beta))))+sum((1-BoosM3[,2^j-1])*(log(1-pgamma(BoosM3[,2^j], shape = alpha,
scale = 1/beta, lower.tail = TRUE, log.p = FALSE))))
Q.ga<- maxLik(log.gam, start=c(0.8, 0.01), method = "NM"); QgaB[j]<-Q.ga$estimate[1]; Qga1B[j]<-Q.ga$estimate[2]; Qga2B[j]<- solve(Q.ga$hessian)[1];
Qga3B[j]<- solve(Q.ga$hessian)[4]
}
#####Weibull
log.we <- function(param) {
  alpha<-param[1]; beta <- param[2]
  sum((BoosM3[,2^j-1]*log(dweibull(BoosM3[,2^j], shape = alpha, scale = 1/beta, log = FALSE)))+sum((1-BoosM3[,2^j-1])*log(1-pweibull(BoosM3[,2^j], shape =
alpha, scale = 1/beta, lower.tail = TRUE, log.p = FALSE))))
Q.we<- maxLik(log.we, start=c(1, 0.01), method = "NM"); QweB[j]<-Q.we$estimate[1]; Qwe1B[j]<-Q.we$estimate[2]
Qwe2B[j]<- solve(Q.we$hessian)[1]; Qwe3B[j]<- solve(Q.we$hessian)[4]
}
#####LogNormal
log.lg <- function(param) {
  mu <- param[1]; sigma <-param[2]
  sum((BoosM3[,2^j-1]*log(dlnorm(BoosM3[,2^j], meanlog = mu, sdlog = sigma, log = FALSE)))+sum((1-BoosM3[,2^j-1])*log(1-plnorm(BoosM3[,2^j], meanlog
= mu, sdlog = sigma, lower.tail = TRUE, log.p = FALSE))))
Q.lg <- maxLik(log.lg, start=c(1, 1), method = "NM")
QlgB[j]<-Q.lg$estimate[1]; Qlg1B[j]<-Q.lg$estimate[2]; Qlg2B[j]<- solve(Q.lg$hessian)[1]; Qlg3B[j]<- solve(Q.lg$hessian)[4]
}
#####Lindley
log.lind <- function(param) {
  beta <- param
  sum((BoosM3[,2^j-1]*log(dlindley(BoosM3[,2^j], theta = beta, log = FALSE)))+ sum((1-BoosM3[,2^j-1])*log(1-plindley(BoosM3[,2^j], theta = beta, lower.tail =
TRUE, log.p = FALSE))))
Q.lind<- maxLik(log.lind, start=0.15); QlindB[j]<-Q.lind$estimate; Qlind1B[j]<- solve(Q.lind$hessian)
}
#####PwLl
log.pw=function(param){
  mean<- param[1]; beta <- param[2]
  sum((BoosM3[,2^j-1]*log(dgenilindley(BoosM3[,2^j], theta = beta, alpha = mean, log = FALSE)))+sum((1-BoosM3[,2^j-1])*log(1-pgenilindley(BoosM3[,2^j],
theta = beta, alpha = mean, lower.tail = TRUE, log.p = FALSE))))
Q.pw <- maxLik(log.pw, start=c(mean=1, beta=2), method = "NM")
QpwB[j]<-Q.pw$estimate[1]; Qpw1B[j]<-Q.pw$estimate[2]; Qpw2B[j]<- solve(Q.pw$hessian)[1]; Qpw3B[j]<- solve(Q.pw$hessian)[4]
}
#####EPwL
log.lge <- function(param) {
  alpha<-param[1]; beta <- param[2]; lambda <- param[3]
  sum((BoosM3[,2^j-1]*log(dexpplindley(BoosM3[,2^j], beta, alpha, theta, log = FALSE)))+sum((1-BoosM3[,2^j-1])*log((1-pexpplindley(BoosM3[,2^j], beta,
alpha, theta, lower.tail = TRUE, log.p = FALSE))))
Q.lge <- maxLik(log.lge, start=c(0.6, 1, 1), method = "NM")
QlgeB[j]<-Q.lge$estimate[1]; Qlge1B[j]<-Q.lge$estimate[2]; Qlge2B[j]<- solve(Q.lge$hessian)[1]; Qlge3B[j]<- solve(Q.lge$hessian)[4]
}

#####Lév
log.lev <- function(param) {
  beta <- param
  sum((BoosM3[,2^j-1]*log(dlevy(BoosM3[,2^j], m=0, s=beta)))+sum((1-BoosM3[,2^j-1])*log(1-plevy(BoosM3[,2^j], m=0, s=beta))))
Q.lev<- maxLik(log.lev, start=0.8); QlevB[j]<-Q.lev$estimate; Qlev1B[j]<- solve(Q.lev$hessian)
}

#####Distribución Bootstrap de la supervivencia S(x)
t.int=sort(BDcancer1[,2])
p.sup=matrix(0, B, length(t.int))
for(j in 1:B){
  for(i in 1:length(t.int)){
    p.sup[i,j]=exp(-QexpB[j]*(t.int[i]))
    p.sup[i,j]=(1-(lgamma(QgaB[j], (t.int[i]*Qga1B[j]), lower=TRUE)/gamma(QgaB[j])))
    p.sup[i,j]=exp(-(Qwe1B[j]*t.int[i])^(QweB[j]))
    p.sup[i,j]=(1-pnorm((log(t.int[i])-QlgB[j])/Qlg1B[j]))
    p.sup[i,j]=(1+((QlindB[j]*t.int[i])/(1+QlindB[j])))^exp(-QlindB[j]*t.int[i]))
    p.sup[i,j]=(1-(exp(-(Qpw1B[j]/t.int[i])^(QpwB[j])))+(exp(-(Qpw1B[j]/t.int[i])^(QpwB[j]))*(Qpw1B[j])/(t.int[i]^QpwB[j]*(1+Qpw1B[j])))))
    p.sup[i,j]=(1-pexpplindley(t.int[i], Qlge1B[j], QlgeB[j], Qlge2B[j], lower.tail = TRUE, log.p = FALSE))
    p.sup[i,j]=erf(sqrt(QlevB[j]/(2*(t.int[i]))))
  }
}

```

```

}
quan1=0; quan2=0
for(j in 1:length(BDcancer1[,2])){
  quan1[j]=quantile(p.sup[j], 0.025)
  quan2[j]=quantile(p.sup[j], 0.975)
}
tiemQ=cbind(t.int, quan1, quan2)
dif=tiemQ[,3]-tiemQ[,2]
#Clasico#
supPar=exp(-Qexp*(sort(BDcancer[,2])))
supPar=(1-(gamma(Qga, sort(BDcancer[,2])*Qga1, lower=TRUE)/gamma(Qga)))
supPar=exp(-(Qwe1*sort(BDcancer[,2]))^(Qwe))
supPar=(1-pnorm((log(sort(BDcancer[,2]))-Qlq)/Qlg1))
supPar=(1+((Qlind*sort(BDcancer[,2]))/(1+Qlind)))^exp(-Qlind*sort(BDcancer[,2]))
supPar=(1-(exp(-(Qpw1/sort(BDcancer[,2]))^(Qpw)))+(exp(-(Qpw1/sort(BDcancer[,2]))^(Qpw)))/((sort(BDcancer[,2])^Qpw)^(1+Qpw1))))
supPar=(1-pexpplndley(sort(BDcancer[,2]), Qlge1, Qlge, 4.12, lower.tail = TRUE, log.p = FALSE))
supPar=erf(sqrt(Qlev/(2*(sort(BDcancer1[,2])))))

```

#####Curvas de supervencia KM y de cada modelo paramétrico que se desee estimar, ambos funciones con sus IC####

```

kmM<-survfit(Surv(BDcancer[,2], BDcancer[,1], type='right')~1, conf.int=TRUE, conf.type='log')
plot(kmM, mark.time=TRUE, mark=3, main="", col='black', lty=2, lwd=3, xlab = "Time (month)", ylab = "Estimated survival probability", ylim = c(0,1), xlim=c(0,
max(BDcancer1[,2])), mar = c(5.1, 4.1, 4.1, 4), mgp = c(2.7, 0.8, 0), cex = 1.3, cex.lab=1.8, cex.main = 2.5, cex.axis=1.5, las=0)
kmM<-survfit(Surv(BDcancer[,2], BDcancer[,1], type='right')~1, conf.int=TRUE, conf.type='log-log')
lines(kmM, mark.time=TRUE, mark=3, col='deeppink2', lty=2, lwd=3, xlab = "", ylab = "", ylim = c(0,1), xlim=c(0, max(BDcancer1[,2])), main="")
kmM<-survfit(Surv(BDcancer[,2], BDcancer[,1], type='right')~1, conf.int=TRUE, conf.type='plain')
lines(kmM, mark.time=TRUE, mark=3, col='orange', lty=2, lwd=3, xlab = "", ylab = "", ylim = c(0,1), xlim=c(0, max(BDcancer1[,2])), main="")
kmM<-survfit(Surv(BDcancer[,2], BDcancer[,1], type='right')~1, conf.int=FALSE)
lines(kmM, mark.time=TRUE, mark=3, col='tomato4', lty=1, lwd=4, xlab = "", ylab = "", ylim = c(0,1), xlim=c(0, max(BDcancer1[,2])), main="")
par(new=TRUE)
lines(sort(BDcancer[,2]), supPar, type = "l", lwd=4, col="green", lty=1, ylab = "", xlab = "Tiempo", main="")
par(new=TRUE)
lines(tiemQ[,1], quan1, col="blue", lty=2, lwd=3)
par(new=TRUE)
lines(tiemQ[,1], quan2, col="blue", lty=2, lwd=3)

```

#Legendas para las gráficas según la distribución de probabilidad analizada.

```

legend("topright", c("Estimación KM", "KM.Lim (log)", "KM.Lim (log-log)", "KM.Lim (lineal)", "Estimación Sup.Exp", "LimBootstrap.Exp"), col = c("tomato4",
"black", "deeppink2", "orange", "green", "blue"), lty=c(1,2,2,2,1,2), bty="n", lwd = c(4,3,3,3,4,3))
legend("topright", c("Estimación KM", "KM.Lim (log)", "KM.Lim (log-log)", "KM.Lim (lineal)", "Estimación Sup.Gam", "LimBootstrap.Gam"), col = c("tomato4",
"black", "deeppink2", "orange", "green", "blue"), lty=c(1,2,2,2,1,2), bty="n", lwd = c(4,3,3,3,4,3))
legend("topright", c("Estimación KM", "KM.Lim (log)", "KM.Lim (log-log)", "KM.Lim (lineal)", "Estimación Sup.Weib", "LimBootstrap.Weib"), col = c("tomato4",
"black", "deeppink2", "orange", "green", "blue"), lty=c(1,2,2,2,1,2), bty="n", lwd = c(4,3,3,3,4,3))
legend("topright", c("Estimación KM", "KM.Lim (sin transf.)", "KM.Lim (log)", "KM.Lim (log-log)", "Estimación Sup.LogN", "LimBootstrap.LogN"), cex= 1.6, col
= c("tomato4", "orange", "black", "deeppink2", "green", "blue"), lty=c(1,2,2,2,1,2), bty="n", lwd = c(4,3,3,3,4,3))
legend("topright", c("Estimación KM", "KM.Lim (log)", "KM.Lim (log-log)", "KM.Lim (lineal)", "Estimación Sup.Lind", "LimBootstrap.Lind"), col = c("tomato4",
"black", "deeppink2", "orange", "green", "blue"), lty=c(1,2,2,2,1,2), bty="n", lwd = c(4,3,3,3,4,3))
legend("topright", c("Estimación KM", "KM.Lim (log)", "KM.Lim (log-log)", "KM.Lim (lineal)", "Estimación Sup.PwLI", "LimBootstrap.PwLI"), col = c("tomato4",
"black", "deeppink2", "orange", "green", "blue"), lty=c(1,2,2,2,1,2), bty="n", lwd = c(4,3,3,3,4,3))
legend("topright", c("Estimación KM", "KM.Lim (sin transf.)", "KM.Lim (log)", "KM.Lim (log-log)", "Estimación Sup.EPwL", "LimBootstrap.EPwL"), cex= 1.6, col
= c("tomato4", "orange", "deeppink2", "black", "green", "blue"), lty=c(1,2,2,2,1,2), bty="n", lwd = c(4,3,3,3,4,3))
legend("topright", c("Estimación KM", "KM.Lim (sin transf.)", "KM.Lim (log)", "KM.Lim (log-log)", "Estimación Sup.Lév", "LimBootstrap.Lév"), cex= 1.6, col
= c("tomato4", "orange", "deeppink2", "black", "green", "blue"), lty=c(1,2,2,2,1,2), bty="n", lwd = c(4,3,3,3,4,3))

```