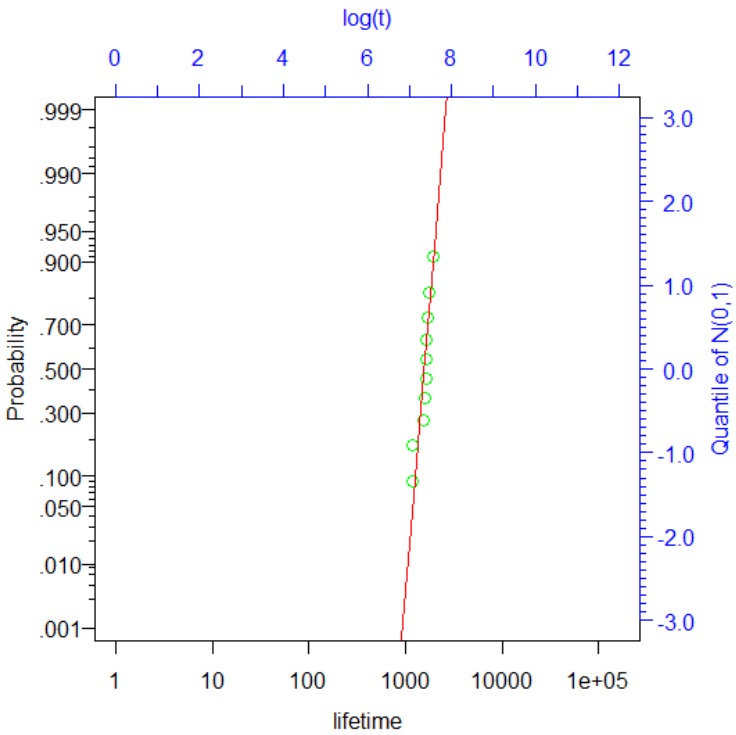
```
#--------------------------------
# MLE for log-normal distribution
#--------------------------------
# 240F Complete Data
#--------------------------------
data.x <- c(1175, 1175, 1521, 1567, 1617, 1665, 1665, 1713, 1761, 1953)

# log-likelihood function
lognorm.likelihood <- function(para) {
     mu <- para[1]
     sigma <- abs(para[2]) # should be positive
     n <- length(data.x)
     like.temp <- rep(NA, n)
     for(i in 1:n) {
          like.temp[i] <- -log(2*pi)/2 - log(sigma*data.x[i]) - (log(data.x[i])-mu)^2/(2*sigma^2)
     }
     -sum(like.temp)
}

# initial value
initial.value <- c(0, 1)
lognorm.likelihood(initial.value)
[1] 353.278
# optimization
(est.result <- optim(initial.value, lognorm.likelihood))
$par
[1] 7.741329 1.358772
$value
[1] 86.26968
$counts
function gradient
      41       NA
$convergence
[1] 10
$message
NULL

(est.result <- optim(est.result$par, lognorm.likelihood))
$par
[1] 7.3542952 0.1564733
$value
```

```
[1] 69.18586
$counts
function gradient
      67       NA
$convergence
[1] 0
$message
NULL

library(flexsurv)
flexsurvreg(formula = Surv(data.x) ~ 1, dist = "lnorm")

#--------------------------------
# 190F Censored Data
#--------------------------------
data.x <- c(7228, 7228, 7228, 8448, 9167, 9167, 9167, 9167, 10511, 10511)
status <- c(1, 1, 1, 1, 1, 1, 1, 1, 0, 0) # 1 = dead, 0 = censored
# log-likelihood function
lognorm.likelihood.censored <- function(para) {
     mu <- para[1]
     sigma <- abs(para[2]) # positive
     n <- length(data.x)
     like.temp <- rep(NA, n)
     for(i in 1:n) {
          like.temp[i] <- status[i] * (-log(2*pi)/2 - log(sigma*data.x[i]) - (log(data.x[i])-mu)^2/(2*sigma^2)) +
                                (1 - status[i]) * (log(1 - pnorm((log(data.x[i])-mu)/sigma)))
     }
     -sum(like.temp)
}
# initial value
initial.value <- c(0, 1)
lognorm.likelihood.censored(initial.value)
# optimization
(est.result <- optim(initial.value, lognorm.likelihood.censored))
(est.result <- optim(est.result$par, lognorm.likelihood.censored))

library(flexsurv)
flexsurvreg(formula = Surv(data.x, status) ~ 1, dist = "lnorm")
```

```
prob.plot.logn <- function(data.v){
    x.range = c(0, 12); y.range = c(-3,3)
    mgp = c(2.5, .5, 0)
    index2.p.v = c(.001, .01, .05, .1, .3, .5, .7, .9, .95, .99, .999)
    index2.lower.p.v = c(seq(.001, .01, by = .002), seq(.01, .1, by = .01),
                         seq(.1, .9, by = .1), seq(.9, .99, by = .01), seq(.99, .999, by = .002))
    index3.x.v = c(0, 2, 4, 6, 8, 10, 12)
    index4.q.v = c(-3, -2, -1, 0, 1, 2 ,3)
    index4.lower.q.v = seq(-3, 3, by = 0.1)
    cex1 = 1
    # sort the data
    data.v <- log(data.v)
    data.v = sort(data.v)
    n = length(data.v)
    # Control margin of the plot
    oldpar = par(mar = c(4,4,4,4), cex = cex1, las = 1)
    # empirical probability
    p.data.v = (1:n)/(n+1)
    # probability to quantile
    q.data.v = qnorm(p.data.v)
    if(is.null(y.range)) y.range = range(q.data.v)
    if(is.null(x.range)) x.range = range(data.v)
    plot(x.range, y.range, xlab = "lifetime", ylab = "Probability", type = "n", xaxt = "n", yaxt = "n", mgp = mgp)
    points(data.v, q.data.v, col = "green", cex = 1.2)
    #add the estimated line
     mu.hat = mean(data.v)
     sigma.hat = sd(data.v)
     x.v = seq(x.range[1], x.range[2], length = 50)
     p.v = pnorm(x.v, mean = mu.hat, sd = sigma.hat)
     q.v = qnorm(p.v)
     lines(x.v, q.v, type= "l", col = "red", lwd = 1.5)
    # add x axis
    index1.v <- c(0.1, 1, 1e1, 1e2, 1e3, 1e4, 1e5)
    axis(1, at = log(index1.v), labels = index1.v)
    # add probability axis
    if(is.null(index2.p.v)){
        index2.p.v = pnorm(seq(range(q.data.v)[1], range(q.data.v)[2], length = 7))
        temp.v = rowMeans(cbind(index2.p.v[-7], index2.p.v[-1]))
        temp.m = rbind(index2.p.v[-7], temp.v)
        index2.lower.p.v = c(as.vector(temp.m), index2.p.v[7])
    }
```

```
    index2.q.v = qnorm(index2.p.v)
    index2.p.label.v = formatC(index2.p.v, format = "f", digits = 3, width = 4)
    index2.p.label.v = lapply(index2.p.label.v, substr, 2, 5)
    axis(2, index2.q.v, labels = index2.p.label.v, cex = 0.2, mgp = mgp)
    index2.lower.q.v = qnorm(index2.lower.p.v)
    axis(2, index2.lower.q.v, labels = F, tck = -0.01)
    # add x axis
    if(is.null(index3.x.v)) index3.x.v = seq(ceiling(x.range[1]), floor(x.range[2]), length = 5)
    axis(3, index3.x.v, labels = as.character(round(index3.x.v,1)), cex = cex1, col = "blue", col.axis = "blue")
    index3.x.lower.v = seq(min(index3.x.v), max(index3.x.v), by = (index3.x.v[2] - index3.x.v[1]) / 2)
    axis(3, index3.x.lower.v, labels = F, tck = -0.02, col = "blue", col.axis = "blue")
    mtext(side = 3, line = mgp[1], "log(t)", cex = cex1, las = 0, col = "blue")
    # add quantile axis
    if(is.null(index4.q.v)){
        index4.q.v = seq(range(q.data.v)[1], range(q.data.v)[2], length = 7)
        index4.lower.q.v = seq(range(q.data.v)[1], range(q.data.v)[2], length = 13)
    }
    index4.q.label.v = formatC(index4.q.v, format = "f", digits = 1, width = 4)
    axis(4, index4.q.v, labels = index4.q.label.v, mgp = mgp + c(0,0.2,0), col = "blue", col.axis = "blue")
    axis(4, index4.lower.q.v, labels = F, tck = -0.01, col = "blue")
    mtext(side = 4, line = mgp[1], "Quantile of N(0,1)", cex = cex1, las = 0, col = "blue")
    par(oldpar)
}
data.x <- c(1175, 1175, 1521, 1567, 1617, 1665, 1665, 1713, 1761, 1953)
prob.plot.logn(data.x)
```



Package 'flexsurv'

"gengamma"	Generalized gamma (stable parameterisation)
"gengamma.orig"	Generalized gamma (original parameterisation)
"genf"	Generalized F (stable parameterisation)
"genf.orig"	Generalized F (original parameterisation)
"weibull"	Weibull
"gamma"	Gamma
"exp"	Exponential
"lnorm"	Log-normal
"gompertz"	Gompertz