Transmuted New Weibull-Pareto Distribution

Ayodeji Ayodele Theophilus

4/16/2022

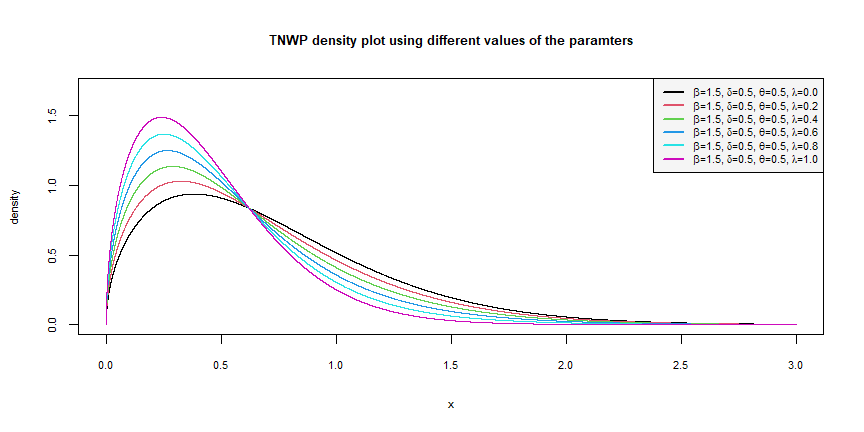
Table of Contents

# Libraries

library(maxLik)   
library(rnoaa)  
library(dplyr)  
library(ggplot2)  
library(lubridate)  
options(scipen = 999, digits = 3)

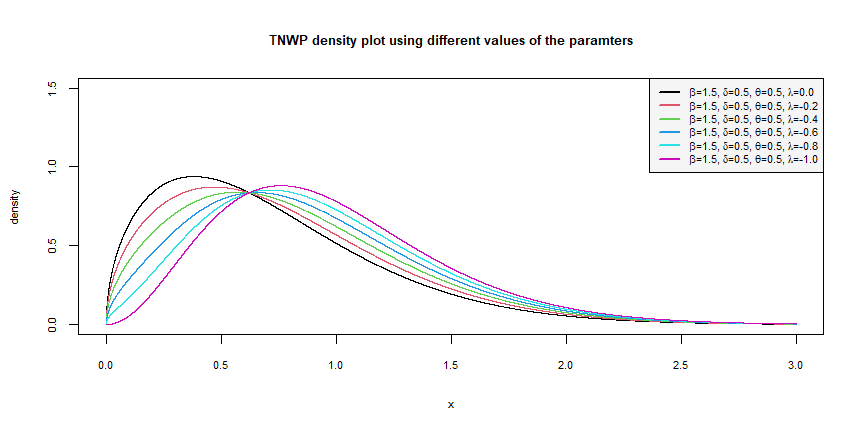
# Density Plot

## PDF #=============================================  
  
x <- seq(0, 3, by = 0.0001)  
beta <- 1.5  
delta <- 0.5  
theta <- 0.5  
lambda <- 0.0  
  
tnwp.pdf <- function(x, beta=1.5, delta=0.5,   
 theta=0.5, lambda=0.0){  
 (beta\*delta/theta) \* ((x/theta)^(beta-1))\*  
 exp(-(delta\*(x/theta)^beta)) \*(1-lambda + 2\*lambda\*exp(-(delta\*(x/theta)^ beta)))  
}  
  
  
 # par(mfrow = c(2,2), cex.axis = 0.5, cex.lab = 0.5,  
 # mar = c(3.5, 3.5, 3.5, 3.5) - 2.5)  
  
plot(x, tnwp.pdf(x,1.5,0.5,0.5,0.0), type = "l", xlab = "x",   
 ylab = "density", ylim = c(0, 1.7),   
 main = "TNWP density plot using different values of the paramters",  
 cex.axis = 0.7,   
 cex.lab = 0.7,  
 cex.main = 0.8,  
 ann = TRUE)  
  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,0.2), col = 2)  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,0.4), col = 3)  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,0.6), col = 4)  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,0.8), col = 5)  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,1.0), col = 6)  
legend("topright", cex=0.7,   
 c(expression(paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=0.0"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=0.2"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=0.4"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=0.6"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=0.8"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=1.0"))),  
 col = c(1,2,3,4,5,6),  
 lty = c(1,1,1,1,1,1),   
 lwd = c(2,2,2,2,2,2),   
 horiz = FALSE,  
 bg = "grey96")



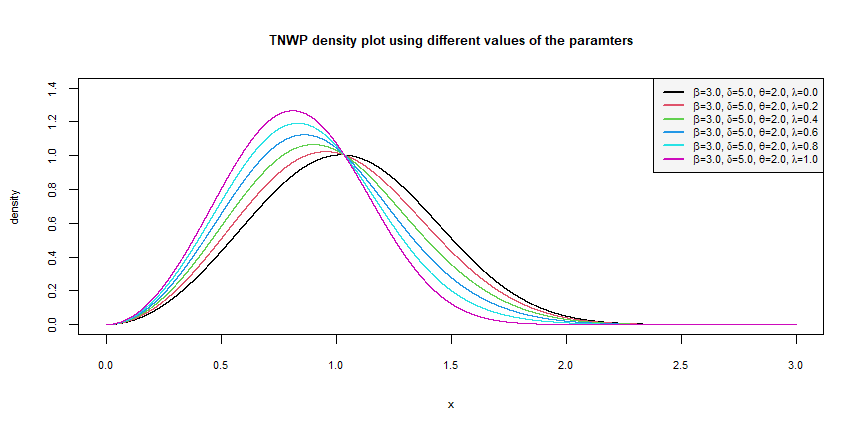
TNWP density plot using different values of the paramters

####========== 2nd plot  
  
  
plot(x, tnwp.pdf(x,1.5,0.5,0.5,0.0), type = "l", xlab = "x",   
 ylab = "density", ylim = c(0, 1.5),  
 main = "TNWP density plot using different values of the paramters",  
 cex.axis = 0.7,   
 cex.lab = 0.7,  
 cex.main = 0.8,  
 ann = TRUE)  
  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,-0.2), col = 2)  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,-0.4), col = 3)  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,-0.6), col = 4)  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,-0.8), col = 5)  
lines(x, tnwp.pdf(x,1.5,0.5,0.5,-1.0), col = 6)  
legend("topright", cex=0.7,  
 c(expression(paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=0.0"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=-0.2"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=-0.4"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=-0.6"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=-0.8"),  
 paste(beta, "=1.5, ",  
 delta, "=0.5, ",  
 theta, "=0.5, ",  
 lambda, "=-1.0"))),  
 col = c(1,2,3,4,5,6),  
 lty = c(1,1,1,1,1,1),   
 lwd = c(2,2,2,2,2,2),   
 horiz = F,  
 bg = "grey96")



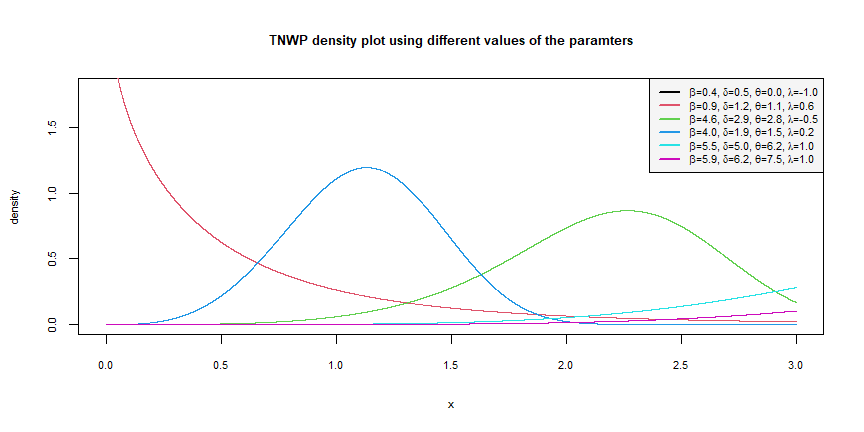
TNWP density plot using different values of the paramters

####========== 3rd plot  
  
  
plot(x, tnwp.pdf(x,3.0,5.0,2.0,0.0), type = "l", xlab = "x",   
 ylab = "density",   
 ylim = c(0, 1.4),  
 main = "TNWP density plot using different values of the paramters",  
 cex.axis = 0.7,   
 cex.lab = 0.7,  
 cex.main = 0.8,  
 ann = TRUE)  
  
lines(x, tnwp.pdf(x,3.0,5.0,2.0,0.2), col = 2)  
lines(x, tnwp.pdf(x,3.0,5.0,2.0,0.4), col = 3)  
lines(x, tnwp.pdf(x,3.0,5.0,2.0,0.6), col = 4)  
lines(x, tnwp.pdf(x,3.0,5.0,2.0,0.8), col = 5)  
lines(x, tnwp.pdf(x,3.0,5.0,2.0,1.0), col = 6)  
legend("topright",cex=0.7,   
 c(expression(paste(beta, "=3.0, ",  
 delta, "=5.0, ",  
 theta, "=2.0, ",  
 lambda, "=0.0"),  
 paste(beta, "=3.0, ",  
 delta, "=5.0, ",  
 theta, "=2.0, ",  
 lambda, "=0.2"),  
 paste(beta, "=3.0, ",  
 delta, "=5.0, ",  
 theta, "=2.0, ",  
 lambda, "=0.4"),  
 paste(beta, "=3.0, ",  
 delta, "=5.0, ",  
 theta, "=2.0, ",  
 lambda, "=0.6"),  
 paste(beta, "=3.0, ",  
 delta, "=5.0, ",  
 theta, "=2.0, ",  
 lambda, "=0.8"),  
 paste(beta, "=3.0, ",  
 delta, "=5.0, ",  
 theta, "=2.0, ",  
 lambda, "=1.0"))),  
 col = c(1,2,3,4,5,6),  
 lty = c(1,1,1,1,1,1),   
 lwd = c(2,2,2,2,2,2),   
 horiz = F,  
 bg = "grey96")



TNWP density plot using different values of the paramters

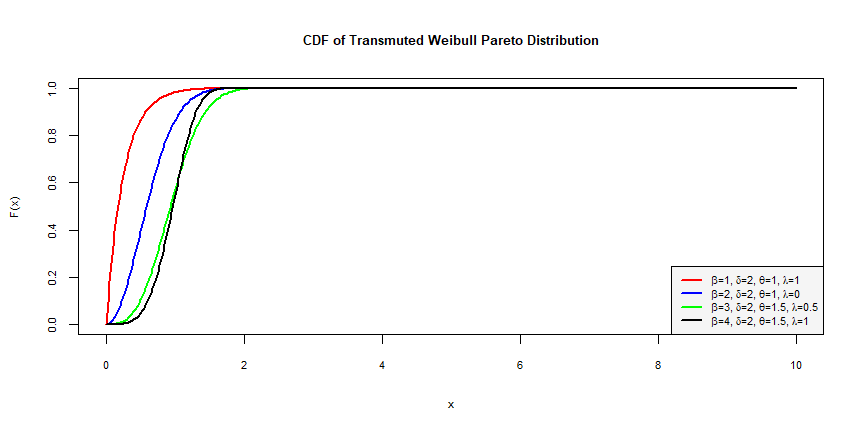
####========== 4th plot  
  
  
plot(x, tnwp.pdf(x,0.4,0.5,0.0,-1.0), type = "l",   
 xlab = "x",   
 ylab = "density",   
 ylim = c(0, 1.8),  
 main = "TNWP density plot using different values of the paramters",  
 cex.axis = 0.7,   
 cex.lab = 0.7,  
 cex.main = 0.8,  
 ann = TRUE)  
  
lines(x, tnwp.pdf(x,0.9,1.2,1.1,0.6), col = 2)  
lines(x, tnwp.pdf(x,4.6,2.9,2.8,-0.5), col = 3)  
lines(x, tnwp.pdf(x,4.0,1.9,1.5,0.2), col = 4)  
lines(x, tnwp.pdf(x,5.5,5.0,6.2,1.0), col = 5)  
lines(x, tnwp.pdf(x,5.9,6.2,7.5,1.0), col = 6)  
legend("topright",cex=0.7,   
 c(expression(paste(beta, "=0.4, ",  
 delta, "=0.5, ",  
 theta, "=0.0, ",  
 lambda, "=-1.0"),  
 paste(beta, "=0.9, ",  
 delta, "=1.2, ",  
 theta, "=1.1, ",  
 lambda, "=0.6"),  
 paste(beta, "=4.6, ",  
 delta, "=2.9, ",  
 theta, "=2.8, ",  
 lambda, "=-0.5"),  
 paste(beta, "=4.0, ",  
 delta, "=1.9, ",  
 theta, "=1.5, ",  
 lambda, "=0.2"),  
 paste(beta, "=5.5, ",  
 delta, "=5.0, ",  
 theta, "=6.2, ",  
 lambda, "=1.0"),  
 paste(beta, "=5.9, ",  
 delta, "=6.2, ",  
 theta, "=7.5, ",  
 lambda, "=1.0"))),  
 col = c(1,2,3,4,5,6),  
 lty = c(1,1,1,1,1,1),   
 lwd = c(2,2,2,2,2,2),   
 horiz = F,  
 bg = "grey96")



TNWP density plot using different values of the paramters

# CDF Plot

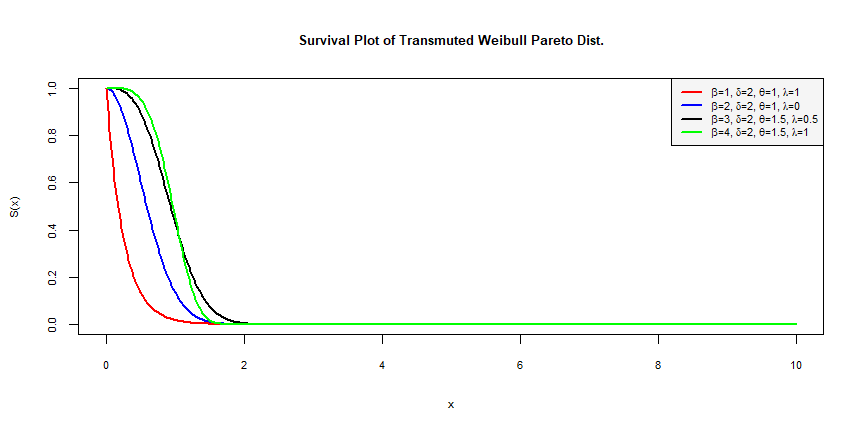
## ==== R-code for the CDF plot =========================  
  
x <- seq(0,10,0.025)  
  
cd <- function(x,beta,delta,theta,lambda){  
   
 (1-exp(-(delta\*(x/theta)^beta)))\*  
 (1+lambda\*(exp(-(delta\*(x/theta)^beta))))  
   
 }  
  
  
c1 <- cd(x,beta=1,delta=2,theta=1,lambda=1)   
  
plot(x,c1, main="CDF of Transmuted Weibull Pareto Distribution",  
 type="n", xlab = "x",ylab = "F(x)",   
 cex.main = 0.8,   
 cex.lab = 0.7,  
 cex.axis = 0.7)   
  
lines(x, cd(x,beta=1,delta=2,theta=1,lambda=1),   
 col="red",lwd=2,lty=1)   
  
lines(x, cd(x,beta=2,delta=2,theta=1,lambda=0),  
 col="blue",lwd=2,lty=1)   
  
lines(x, cd(x,beta=3,delta=2,theta=1.5,lambda=0.5),  
 col="green",lwd=2,lty=1)   
  
lines(x, cd(x,beta=4,delta=2,theta=1.5,lambda=1),  
 col="black",lwd=2,lty=1)   
  
  
legend("bottomright",cex=0.7,   
 c(expression(paste(beta, "=1, ",  
 delta, "=2, ",  
 theta, "=1, ",  
 lambda, "=1"),  
 paste(beta, "=2, ",  
 delta, "=2, ",  
 theta, "=1, ",  
 lambda, "=0"),  
 paste(beta, "=3, ",  
 delta, "=2, ",  
 theta, "=1.5, ",  
 lambda, "=0.5"),  
 paste(beta, "=4, ",  
 delta, "=2, ",  
 theta, "=1.5, ",  
 lambda, "=1"))),  
 horiz=F,   
 lty=c(1,1,1,1),  
 lwd=c(2,2,2,2),   
 col=c("red","blue","green","black"),   
 bg="grey96")



CDF of Transmuted Weibull Pareto Distribution

# Survival Plot

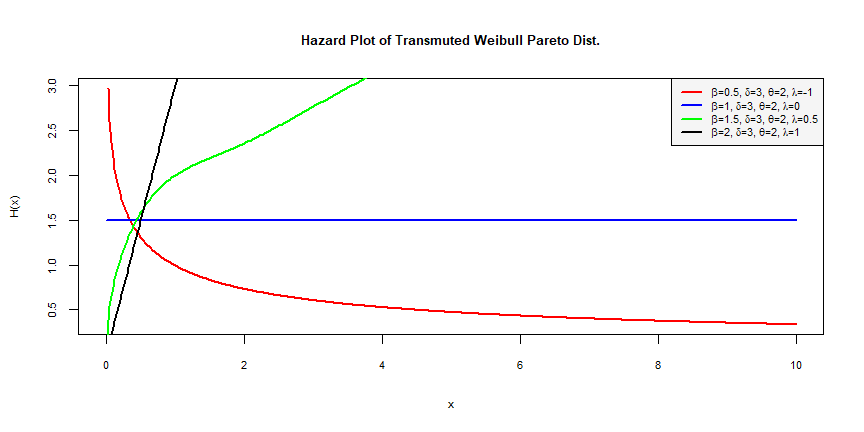
## ====R-code for the Survival Plot ======  
  
x <- seq(0,10,0.025)   
  
h <- function(x, beta, delta, theta, lambda){  
   
 (1-((1-exp(-(delta\*(x/theta)^beta)))\*  
 (1+(lambda\*(exp(-(delta\*(x/theta)^beta)))))))  
   
}   
  
x1 <- h(x,beta=1,delta=2,theta=1,lambda=1)   
  
x2 <- h(x,beta=2,delta=2,theta=1,lambda=0)   
  
x3 <- h(x,beta=3,delta=2,theta=1.5,lambda=0.5)   
  
x4 <- h(x,beta=4,delta=2,theta=1.5,lambda=1)  
  
plot(x, x1,   
 main = "Survival Plot of Transmuted Weibull Pareto Dist.",  
 ylim = c(0,1),type = "n", xlab = "x",ylab = "S(x)",  
 cex.main = 0.8, cex.lab = 0.7,  
 cex.axis = 0.7)  
  
lines(x,x1,col="red",lwd=2,lty=1)   
  
lines(x,x2,col="blue",lwd=2,lty=1)   
  
lines(x,x3,col="black",lwd=2,lty=1)   
  
lines(x,x4,col="green",lwd=2,lty=1)   
  
  
legend("topright",cex=0.7,   
 c(expression(paste(beta, "=1, ",  
 delta, "=2, ",  
 theta, "=1, ",  
 lambda, "=1"),  
 paste(beta, "=2, ",  
 delta, "=2, ",  
 theta, "=1, ",  
 lambda, "=0"),  
 paste(beta, "=3, ",  
 delta, "=2, ",  
 theta, "=1.5, ",  
 lambda, "=0.5"),  
 paste(beta, "=4, ",  
 delta, "=2, ",  
 theta, "=1.5, ",  
 lambda, "=1"))),  
 horiz=F,   
 lty=c(1,1,1,1),  
 lwd=c(2,2,2,2),   
 col=c("red","blue","black","green"),   
 bg="grey96")



Survival Plot of Transmuted Weibull Pareto Distribution

# Hazard Plot

## ==== R-code for the Hazard Plot   
  
x <- seq(0,10,0.025)   
  
ha <- function(x,beta,delta,theta,lambda){  
   
 ((beta\*delta/theta)\*((x/theta)^(beta-1))\*  
 (1-lambda+2\*lambda\*exp(-(delta\*(x/theta)^beta))))/  
 (lambda\*exp(-(delta\*(x/theta)^beta))+1-lambda)  
   
 }   
  
x1 <- ha(x,beta=0.5,delta=3,theta=2,lambda=-1)   
  
x2 <- ha(x,beta=1,delta=3,theta=2,lambda=0)   
  
x3 <- ha(x,beta=1.5,delta=3,theta=2,lambda=0.5)   
  
x4 <- ha(x,beta=2,delta=3,theta=2,lambda=1)   
  
plot(x,x1, main="Hazard Plot of Transmuted Weibull Pareto Dist.",type="n",xlab = "x",ylab = "H(x)",cex.main = 0.8,   
 cex.lab = 0.7,  
 cex.axis = 0.7)   
  
lines(x,x1,col="red",lwd=2,lty=1)  
  
lines(x,x2,col="blue",lwd=2,lty=1)  
  
lines(x,x3,col="green",lwd=2,lty=1)   
  
lines(x,x4,col="black",lwd=2,lty=1)   
  
legend("topright",cex=0.7,   
 c(expression(paste(beta, "=0.5, ",  
 delta, "=3, ",  
 theta, "=2, ",  
 lambda, "=-1"),  
 paste(beta, "=1, ",  
 delta, "=3, ",  
 theta, "=2, ",  
 lambda, "=0"),  
 paste(beta, "=1.5, ",  
 delta, "=3, ",  
 theta, "=2, ",  
 lambda, "=0.5"),  
 paste(beta, "=2, ",  
 delta, "=3, ",  
 theta, "=2, ",  
 lambda, "=1"))),  
 horiz=F,   
 lty=c(1,1,1,1),  
 lwd=c(2,2,2,2),   
 col=c("red","blue","green","black"),   
 bg="grey96")



# Maximum Likelihood Estimation

## Transmuted New Weibull-Pareto

The transmuted new weibull-pareto density function is given below as:

The log likelihood is given below as:

#### Data

* The dataset used for the analysis was randomly selected from the weibull distribution using the rweibull function in R.

## ===R-code for the Maximum likelihood estimation =====  
  
#x <- prcp\_df$value  
set.seed(123)  
x <- rweibull(120, 2)  
  
loglik <- function(a) {  
   
 beta <- a[1]  
 delta <- a[2]  
 theta <- a[3]  
 lambda <- a[4]  
   
log\_lik <- n\*log(a[1])+n\*log(a[2])-n\*log(a[3])-(a[2])\*  
 sum(x/a[3])^a[1]+(a[1]-1)\*sum(log(x/a[3]))+  
 sum(log(1 - a[4] + 2 \* a[4]\*exp(-(a[2]\*(x/a[3])^a[1]))))  
   
 }   
  
n <- length(x)   
  
mle\_estimate <- maxLik(loglik,   
 start = c(beta = 3.421,   
 delta = 8.6231,  
 theta = 2.25,  
 lambda = -0.4608),  
 control=list(printLevel=2))

----- Initial parameters: -----  
fcn value: -4246515   
 parameter initial gradient free  
beta 3.421 -16269975.0 1  
delta 8.623 -492436.8 1  
theta 2.250 6456312.2 1  
lambda -0.461 78.4 1  
Condition number of the (active) hessian: 106948438   
-----Iteration 1 -----  
-----Iteration 2 -----  
-----Iteration 3 -----  
-----Iteration 4 -----  
-----Iteration 5 -----  
-----Iteration 6 -----  
-----Iteration 7 -----  
-----Iteration 8 -----  
-----Iteration 9 -----  
-----Iteration 10 -----  
-----Iteration 11 -----  
-----Iteration 12 -----  
-----Iteration 13 -----  
-----Iteration 14 -----  
-----Iteration 15 -----  
-----Iteration 16 -----  
-----Iteration 17 -----  
-----Iteration 18 -----  
-----Iteration 19 -----  
-----Iteration 20 -----  
-----Iteration 21 -----  
-----Iteration 22 -----  
-----Iteration 23 -----  
-----Iteration 24 -----  
-----Iteration 25 -----  
-----Iteration 26 -----  
-----Iteration 27 -----  
-----Iteration 28 -----  
-----Iteration 29 -----  
-----Iteration 30 -----  
-----Iteration 31 -----  
-----Iteration 32 -----  
-----Iteration 33 -----  
-----Iteration 34 -----  
-----Iteration 35 -----  
-----Iteration 36 -----  
-----Iteration 37 -----  
-----Iteration 38 -----  
-----Iteration 39 -----  
-----Iteration 40 -----  
-----Iteration 41 -----  
-----Iteration 42 -----  
-----Iteration 43 -----  
--------------  
successive function values within relative tolerance limit (reltol)   
43 iterations  
estimate: 0.202 22.2 0.0245 -42099   
Function value: 1457

summary(mle\_estimate)

--------------------------------------------  
Maximum Likelihood estimation  
Newton-Raphson maximisation, 43 iterations  
Return code 8: successive function values within relative tolerance limit (reltol)  
Log-Likelihood: 1457   
4 free parameters  
Estimates:  
 Estimate Std. error t value Pr(> t)   
beta 0.2021 0.0183 11 <0.0000000000000002 \*\*\*  
delta 22.2168 NaN NaN NaN   
theta 0.0245 NaN NaN NaN   
lambda -42099.4888 2.0972 -20074 <0.0000000000000002 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
--------------------------------------------

AIC(mle\_estimate)

[1] -2906  
attr(,"df")  
[1] 4

## Weibull 2-parameter distribution

Let

The loglikelihood function becomes:

weibull\_loglik <- function(a) {  
   
 beta <- a[1]  
 lambda <- a[2]  
   
log\_lik <- n\*log(a[1]) - n\*a[1]\*log(a[2]) - (sum(x/a[2])^a[1]) + (a[1] - 1)\* sum(log(x))   
   
 }   
  
n <- length(x)   
  
mle\_weibull <- maxLik(weibull\_loglik,   
 start = c(beta = 3.421,   
 lambda = 0.5),  
 control=list(printLevel=2))

----- Initial parameters: -----  
fcn value: -84528512   
 parameter initial gradient free  
beta 3.42 -451000117 1  
lambda 0.50 578345605 1  
Condition number of the (active) hessian: 36.7   
-----Iteration 1 -----  
-----Iteration 2 -----  
-----Iteration 3 -----  
-----Iteration 4 -----  
-----Iteration 5 -----  
-----Iteration 6 -----  
-----Iteration 7 -----  
-----Iteration 8 -----  
-----Iteration 9 -----  
-----Iteration 10 -----  
-----Iteration 11 -----  
-----Iteration 12 -----  
-----Iteration 13 -----  
-----Iteration 14 -----  
-----Iteration 15 -----  
-----Iteration 16 -----  
-----Iteration 17 -----  
-----Iteration 18 -----  
-----Iteration 19 -----  
-----Iteration 20 -----  
-----Iteration 21 -----  
-----Iteration 22 -----  
-----Iteration 23 -----  
-----Iteration 24 -----  
-----Iteration 25 -----  
-----Iteration 26 -----  
-----Iteration 27 -----  
-----Iteration 28 -----  
-----Iteration 29 -----  
--------------  
Last step could not find a value above the current.  
Boundary of parameter space?   
Consider switching to a more robust optimisation method temporarily.   
29 iterations  
estimate: 0.28 0.00000216   
Function value: 170

summary(mle\_weibull)

--------------------------------------------  
Maximum Likelihood estimation  
Newton-Raphson maximisation, 29 iterations  
Return code 3: Last step could not find a value above the current.  
Boundary of parameter space?   
Consider switching to a more robust optimisation method temporarily.  
Log-Likelihood: 170   
2 free parameters  
Estimates:  
 Estimate Std. error t value Pr(> t)  
beta 0.28030632 NaN NaN NaN  
lambda 0.00000216 NaN NaN NaN  
--------------------------------------------

AIC(mle\_weibull)

[1] -337  
attr(,"df")  
[1] 2

# Citation

* R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
* Arne Henningsen and Ott Toomet (2011). maxLik: A package for maximum likelihood estimation in R. Computational Statistics 26(3), 443-458. DOI 10.1007/s00180-010-0217-1.
* Scott Chamberlain (2021). rnoaa: ‘NOAA’ Weather Data from R. R package version 1.3.8. <https://CRAN.R-project.org/package=rnoaa>