The Rayleigh distribution is used often to help model wind speed (Celik, *Energy Conversion and Management*, 2004, p. 1735-1747). Let Xi be a random variable denoting the average wind speed during a February day in Lincoln, where Xi has a Rayleigh distribution:



where β > 0 and x > 0. Note that a Rayleigh distribution is a special case of a Weibull distribution with a = 2 and b =  using the R parameterization.

The data file wind\_speed.csv contains the observed xi daily average wind speed values for 2000 – 2004 in Lincoln during each February. Assume that each observation is independent (the autocorrelations are all nonsignificant, except for the first autocorrelation in 2002).

The MLE of β is . We would like to obtain an estimate for T’s sampling distribution using a parametric bootstrap! Note that the asymptotic distribution for T is

 where 

Therefore, we could approximate the distribution of T with a N(β, β2/(4n)). Because β is unknown, we would use the maximum likelihood estimate for it.

1. Is the Rayleigh distribution appropriate for the data?

> wind<-read.table(file = "C:\\chris\\UNL\\STAT950\\  
 projects\\spring\_2010\\wind\_speed.csv", header=TRUE,

sep = ",")

> head(wind)

Year Day x

1 2004 1 9.4

2 2004 2 12.7

3 2004 3 3.9

4 2004 4 9.8

5 2004 5 9.5

6 2004 6 15.0

> x<-wind$x

> n<-length(x)

> beta.hat.mle<-sqrt(sum(x^2)/(2\*n))

> beta.hat.mle

[1] 7.871982

> plot.beta<-beta.hat.mle

> #EDF

> par(pty = "s", mfrow=c(1,2), lend = "square")

> plot.ecdf(x = x, verticals = TRUE, do.p = FALSE, main =

"EDF for wind speeds", lwd = 2, panel.first = grid(nx =

NULL, ny = NULL, col="gray", lty="dotted"), ylab =

expression(hat(F)), xlab = "Wind speed")

> curve(expr = pweibull(q = x, shape = 2, scale =

beta.hat.mle\*sqrt(2)), col = "red", add = TRUE)

> rayleigh.quant<-qweibull(p = seq(from = 1/(n+1), to = 1-

1/(n+1), by = 1/(n+1)), shape = 2, scale =

beta.hat.mle\*sqrt(2))

> plot(y = sort(x), x = rayleigh.quant, main = "QQ-Plot for

wind speeds", ylab = "Wind speed", xlab = "Rayleigh

quantiles", panel.first = grid(col="gray",

lty="dotted"))

> abline(a = 0, b = 1, col = "red")



> hist(x = x, main = "Histogram of wind speeds",

freq=FALSE, xlab = "Wind speed", xlim = c(0, max(x)+5))

> curve(expr = dweibull(x = x, shape = 2, scale =

beta.hat.mle\*sqrt(2)), col = 2, add = TRUE)

> #Try another set of histogram bars

> hist(x = x, main = "Histogram of wind speeds",

freq=FALSE, xlab = "Wind speed", xlim = c(0, max(x)+5),

breaks = 5)

> curve(expr = dweibull(x = x, shape = 2, scale =

beta.hat.mle\*sqrt(2)), col = 2, add = TRUE)



1. Take the resamples and calculate 

> R<-1999

> set.seed(9823)

> x.star<-rweibull(n = n\*R, shape = 2, scale =

beta.hat.mle\*sqrt(2))

> x.star.mat<-matrix(data = x.star, nrow = R, ncol = n,

byrow = TRUE)

> x.star.mat[1,1:5] #First 5 observations of first resample

[1] 5.065131 7.781296 11.389837 11.005791 10.169304

> #Function for the MLE

> find.mle<-function(x) {

n<-length(x)

sqrt(sum(x^2)/(2\*n))

}

> find.mle(x)

[1] 7.871982

> #Find the t\* values

> t.star<-apply(X = x.star.mat, MARGIN = 1, FUN = find.mle)

> head(t.star)

[1] 7.640871 7.846854 7.649977 8.253639 8.297641 7.742672

3) Estimate for the distribution of T\*

> par(pty = "s", mfrow=c(1,2), lend = "square")

> hist(x = t.star, main = substitute(paste("Histogram for

t\*, R=", Rnumb), list(Rnumb = 1999)), freq=FALSE, xlab

= "t\*")

> curve(expr = dnorm(x = x, mean = beta.hat.mle, sd =

beta.hat.mle/2/sqrt(n)), col = 2, add = TRUE)

> plot.ecdf(x = t.star, verticals = TRUE, do.p = FALSE,

main = substitute(paste("EDF for t\*, R=", Rnumb),

list(Rnumb = 1999)), lwd = 2, panel.first = grid(nx =

NULL, ny = NULL, col="gray", lty="dotted"), ylab =

expression(hat(G)), xlab = "t\*")

> curve(expr = pnorm(q = x, mean = beta.hat.mle, sd =

beta.hat.mle/2/sqrt(n)), col = "red", add = TRUE)



How would one find the exact distribution of T\*?

1. Estimated bias and variance for T

Remember that MLE’s are asymptotically unbiased (consistent estimators). Also, the asymptotic variance here is .

> data.frame(var.est = var(t.star), bias.est = mean(t.star)

- beta.hat.mle)

var.est bias.est

1 0.1088509 0.001428827

> var.asym<-beta.hat.mle^2/(4\*n) #asymptotic standard

deviation

> var.asym

[1] 0.1090988