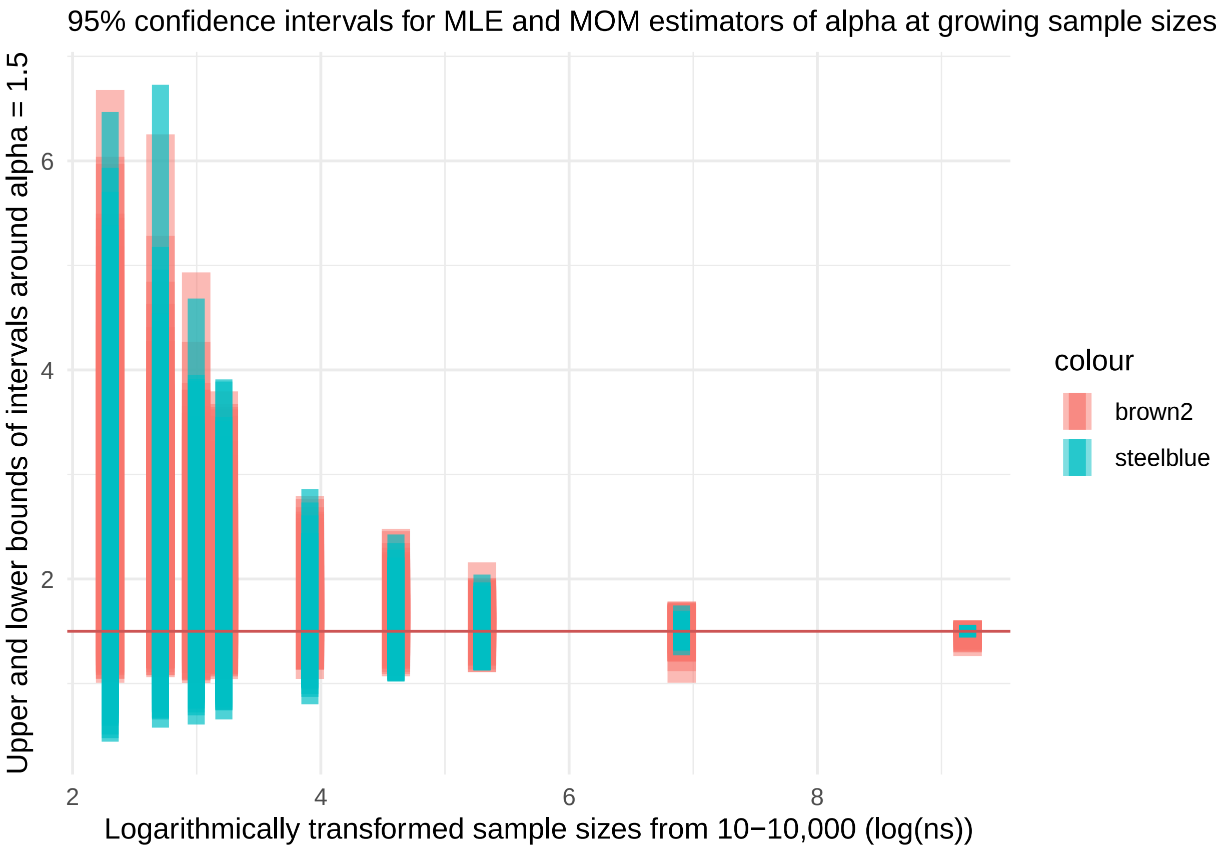
```r
#setup
library(ggplot2)


y_m <- 5e4
alphas <- c(1.5) #the only value of alpha we're interested in
num_alphas <- length(alphas)
ns <- c(10, 15, 20, 25, 50, 100, 200, 1000, 10000) # expanded numbers of sample
#sizes so we can get a better idea of the change in confidence interval spread.
num_ns <- length(ns)
numIter <- 100 # number of simulations (1000 in the chart)
B <- 100 # number of bootstrap iterations per simulation (1000 in the chart)


#arrays to store the estimates from our simulation - here we're only using
#boot.lower and boot.upper for both the MLE and MOM.
alpha_hat_MLE <- array(dim = c(num_alphas, num_ns, numIter))
alpha_hat_MOM <- array(dim = c(num_alphas, num_ns, numIter))

exact.lower<- array(dim = c(num_alphas, num_ns, numIter))
exact.upper <- array(dim = c(num_alphas, num_ns, numIter))

CLT.lower<- array(dim = c(num_alphas, num_ns, numIter))
CLT.upper <- array(dim = c(num_alphas, num_ns, numIter))

boot.lowerMLE<- array(dim = c(num_alphas, num_ns, numIter))
boot.upperMLE <- array(dim = c(num_alphas, num_ns, numIter))

boot.lowerMOM<- array(dim = c(num_alphas, num_ns, numIter))
boot.upperMOM <- array(dim = c(num_alphas, num_ns, numIter))


rmse_MLE <- array(dim = c(num_alphas, num_ns))

#Bootstrapping loops for each sample size and simulation number.
for(i in 1:num_alphas){ #i indexes the alpha value
    for(j in 1:num_ns){ #j indexes the sample size
        for(k in 1:numIter){ #k indexes the simulation number for a fixed n and alpha
            n<-ns[j]
            X <- rexp(n, rate = alphas[i])
            Y <- y_m * exp(X) # Y ~ Pareto(y_m, alpha)

            Ybar = mean(Y)
            alpha_hat_MLE[i,j,k] <- 1/mean(log(Y/y_m)) #MLE Formula

            alpha_hat_MOM[i,j,k] <- Ybar/(Ybar-y_m) #MOM Formula

            # confidence interval calculations
            exact.lower[i, j, k] <- alpha_hat_MLE[i,j,k]*qgamma(0.025, n, n)
            exact.upper[i, j, k] <- alpha_hat_MLE[i,j,k]*qgamma(0.975, n, n)

            CLT.lower[i, j, k] <-alpha_hat_MLE[i,j,k]- (sd(log(Y))/(sqrt(n)*qt(0.975, n-1)))
            CLT.upper[i, j, k] <-alpha_hat_MLE[i,j,k]+ (sd(log(Y))/(sqrt(n)*qt(0.975, n-1)))
            # bootstrapping loop.
            bootMLE <- numeric(B)
            bootMOM <- numeric(B)
            for(b in 1:B){
                bootY<-sample(Y, replace=TRUE) #sample from Y
                bootmean<-mean(bootY)
                bootMLE[b]<-1/mean(log(bootY/y_m)) #calculate a MLE estimator
                bootMOM[b]<-bootmean/(bootmean-y_m) #calculate a MOM estimator
            }

            #From the distribution of bootstrapped MLEs and MOMs,
            #calculate the spread via 95% confidence interval.
            boot.lowerMLE[i, j, k] <-quantile(bootMLE, 0.025)
            boot.upperMLE[i, j, k] <-quantile(bootMLE, 0.975)
            boot.lowerMOM[i, j, k] <-quantile(bootMOM, 0.025)
            boot.upperMOM[i, j, k] <-quantile(bootMOM, 0.975)
        }

    }
}


#Create a data frame with vectors for sample size, upper bound, and lower bound for each interval
upperMLE <- c(boot.upperMLE[1,1:num_ns, ])
lowerMLE <- c(boot.lowerMLE[1,1:num_ns, ])
upperMOM <- c(boot.upperMOM[1,1:num_ns, ])
lowerMOM <- c(boot.lowerMOM[1,1:num_ns, ])
condata <- data.frame(ns, upperMLE, lowerMLE, upperMOM, lowerMOM)

# plot the data with sample sizes on x, confidence intervals on y, and alpha indicated as a red line
ggplot(condata, aes(x=log(ns)))+ geom_point(y=1.5) +
  geom_linerange(aes(ymin=lowerMOM,ymax=upperMOM, color="brown2"), alpha=0.5, linewidth=5) +
  geom_linerange(aes(ymin=lowerMLE,ymax=upperMLE, color="steelblue"), alpha=0.7, linewidth=3)+
  geom_hline(yintercept = 1.5, color="indianred3") +
  labs(title= "95% confidence intervals for MLE and MOM estimators of alpha at growing sample sizes", x
  theme_minimal()+
  theme(plot.title = element_text(size=11))
```



95% confidence intervals for MLE and MOM estimators of alpha at growing sample sizes