–Build a confidence interval estimator for the mean based on the bootstrap (use 10000 =nboot)

–Build a simulator that draws n samples form a lognormal distribution (rlnorm) and builds both the central limit theorem based confidence interval, and compares it to the coverage rate for the bootstrap (confidence interval based on the the bootstrap program). (1000 simulation runs minimum)

- Now build the bootstrap estimator into a simulator which compares how well the Pivotal bootstrap confidence interval covers the true value vs a normal theory confidence interval.
- Run it for lognormal (nsim=1000) for n=3,10,30,100, alpha= .1,.05 That's 8 runs **(Fix the program for small n!?!)**
- Compare the coverage of the 2 confidence intervals with nominal coverage
- Write up the results with a table

	alpha=0.1		alpha=0.05	
n	$boot.coverage	$norm.coverage	$boot.coverage	$norm.coverage
3	0.731	0.764	0.722	0.83
10	0.868	0.806	0.909	0.853
30	0.875	0.834	0.924	0.882
100	0.883	0.876	0.946	0.924

Observations:
The bootstrap confidence intervals coverage always suffer at small n because the bootstrap function still relies on the representative data and its is not an immediate solution to small n especially at n=3, which is why at n=3 and both a=0.1 and a=0.05 the bootstrap CI coverage is less than the normal theory coverage.

At all other values of n tested however, the bootstrap confidence interval coverage outperformed the normal theory CI coverage. This is because without the use of bootstrap, we have to assume that the data follows the normal distribution or some other distribution. So for the normal distribution, the central limit theorem will let you bypass this assumption for sample sizes that are larger than ~30. But even at sample sizes larger than 30 the boot strap performs better because it does not require any assumptions to be made and simply resamples the data and we use whatever sampling distribution emerges.

R script (with comments):

```
simfunc<- function(mu.val=3,n=30,nsim=1000)
{
   #initializes the coverage vectors for the bootstrap confidence interval and the normal confidence
intervals
```

```
    cvec.boot<-NULL
    cvec.norm<-NULL
    # calculates mean for lognormal as exp(μ+1 /2)
    mulnorm<-(exp(mu.val+1/2))
    #runs the following lines of simulation code nsim times
    for(i in 1:nsim){
        if((i/10)==floor(i/10)){
            print(i)
            # prints the simulation number by 10s
        }
        #sample vector is created useing the rlnorm function
        vec.sample<-rlnorm(n,mu.val)
        #the function my.bootstrapci is run on the sample vector and then output is stored
        boot.list<-my.bootstrapci(vec.sample)
        #the boot strap and normal confidence intervals are sourced from the boot.list and stored
        boot.conf<-boot.list$bootstrap.confidence.interval
        norm.conf<-boot.list$normal.confidence.interval
        #the bootstrap CI coverage vector is appended with the calculated coverage
        cvec.boot<-c(cvec.boot,(boot.conf[1]<mulnorm)*(boot.conf[2]>mulnorm))
        #the normal CI coverage vector is appended with the calculated coverage
        cvec.norm<-c(cvec.norm,(norm.conf[1]<mulnorm)*(norm.conf[2]>mulnorm))
    }
    #output of the average bootstrap coverage and the average normal coverage out of all the simulations run
    list(boot.coverage=(sum(cvec.boot)/nsim),norm.coverage=(sum(cvec.norm)/nsim))
}

my.bootstrapci<- function(vec0,nboot=10000,alpha=0.1)
{
    #stores the length, mean and standard deviation for the sample vector
    n0<-length(vec0)
    mean0<-mean(vec0)
    sd0<-sqrt(var(vec0))
    #bootstrap vector is initialized
    bootvec<-NULL
    #the following bootstrap resampling loop runs nboot times
    for( i in 1:nboot){
        #vector is resampled with replacement
        vecb<-sample(vec0,replace=T)
        #standard deviation is calulated for the resampled vector
        sdb<-sqrt(var(vecb))
        # the following while loop insures that for small n if the resampled vector has a sd of 0 it is resampled until the resampled vector no longer has a std of 0 wherein it exits the while loop
        while(sdb==0){
            #vector is resampled with replacement
            vecb<-sample(vec0,replace=T)
            #standard deviation is recalulated for the resampled vector
            sdb<-sqrt(var(vecb))
        }
```

```
    #the mean for the resampled vector (with sd not equal to 0 ) is calculated and stored
    meanb<-mean(vecb)
    #the boot strap vector is appended with resampled vector distribution
    bootvec<-c(bootvec,(meanb-mean0)/(sdb/sqrt(n0)))

}
#the lower and upper quantiles are found for the bootstrapped distribution
lq<-quantile(bootvec,alpha/2)
uq<-quantile(bootvec,1-alpha/2)
#the pivotal bootstrap confidence interval bounds are calculated
LB<-mean0-(sd0/sqrt(n0))*uq
UB<-mean0-(sd0/sqrt(n0))*lq
#the normal theory confidence interval bounds are calculated
NLB<-mean0-(sd0/sqrt(n0))*qt(1-alpha/2,n0-1)
NUB<-mean0+(sd0/sqrt(n0))*qt(1-alpha/2,n0-1)
#outputs both confidence intervals
list(bootstrap.confidence.interval=c(LB,UB),normal.confidence.interval=c(NLB,NUB))}
```