# 3858 Tutorial 8

*Yifan Li, yli2763@uwo.ca*

*2017-03-21*

## Parametric vs Non-parametric Bootstrap for Gamma distribution

In class we discuss the non-parametric bootstrap in the estimation of expectation of $X$ (using sample mean), where ${T_n}^*$ will have normal limiting distribution (by CLT).

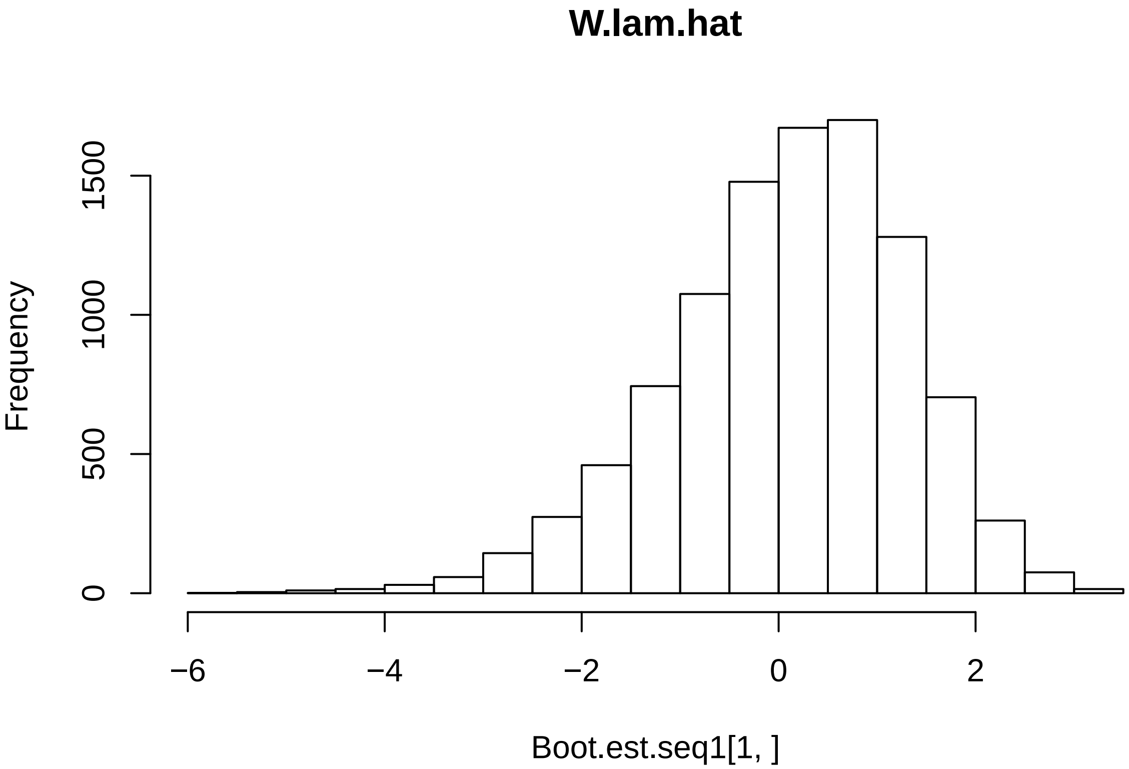
Here we try the non-parametric bootstrap for the ${T_n}^*$ that may not converge to normal distribution. We assume we construct the expressions of variance without knowing the distribution. Let's see what will happen.

```r
#rain.dat <- scan("http://www.stats.uwo.ca/faculty/kulperger/SS3858/Data/illinois60.txt")
set.seed(31415926)
rain.dat <- rgamma(100, shape = 2.5, rate = 1/3)
```

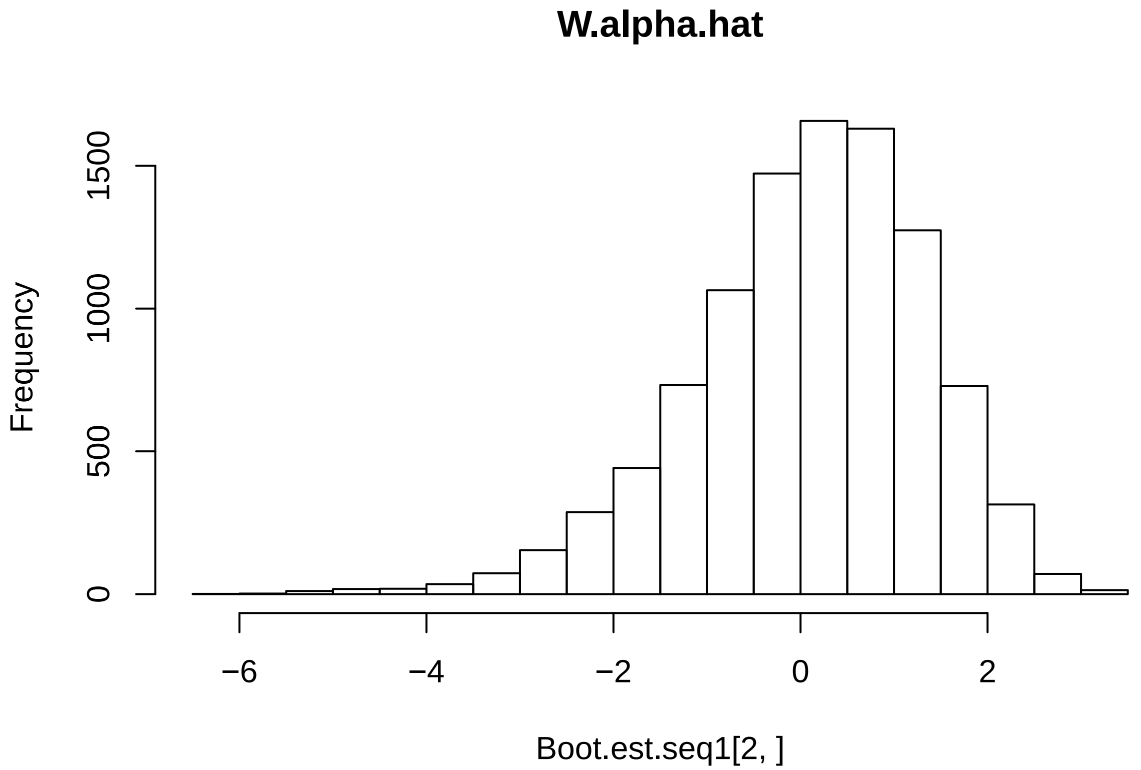
```r
gamma.est <- function(x){
  mu.1 <- mean(x)
  mu.2 <- mean(x^2)
  var.x <- mu.2-mu.1^2
  lam.hat <- mu.1/(mu.2-mu.1^2)
  alpha.hat <- mu.1*lam.hat
  c(lam.hat, alpha.hat)
}
```

```r
est <- gamma.est(rain.dat)
lam <- est[1] #lam and alpha are estimated values
alpha <- est[2]
#try to get better estimate at first (it needs more data)

#bootstrap
M <- 1e4
n <- length(rain.dat)

#using the Fisher information matrix in class

I.0.mat = function(theta){
 A = matrix(0, 2,2)
 lambda = theta[1]
 alpha = theta[2]
 A[1,1] = alpha/lambda^2
 A[2,1] = - 1/lambda
 A[1,2] = A[2,1]
 A[2,2] = trigamma(alpha)
 A
}

V.est <- solve(I.0.mat(est)) #inverse of the Fisher Information matrix
var.lam <- V.est[1,1]
var.alpha <- V.est[2,2]

Boot.est <- function(x){
  theta <- gamma.est(x)
  V.hat <- solve(I.0.mat(theta))
  W.lam <- sqrt(n)*(theta[1]-lam)/sqrt(V.hat[1,1])
  W.alpha <- sqrt(n)*(theta[2]-alpha)/sqrt(V.hat[2,2])
  c(W.lam, W.alpha)
}
```

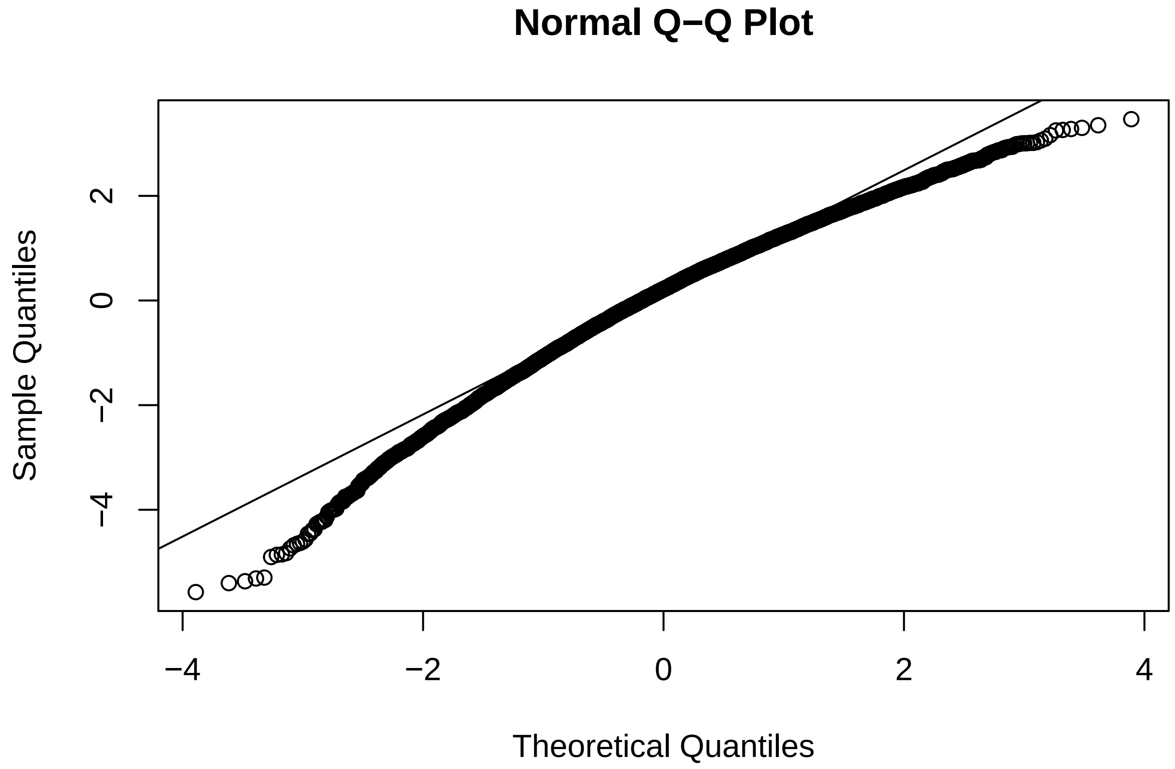
```
#parametric
Boot.est.seq1 <- replicate(M, Boot.est(rgamma(n, shape = alpha, rate = lam)))

hist(Boot.est.seq1[1,], main = "W.lam.hat")
```



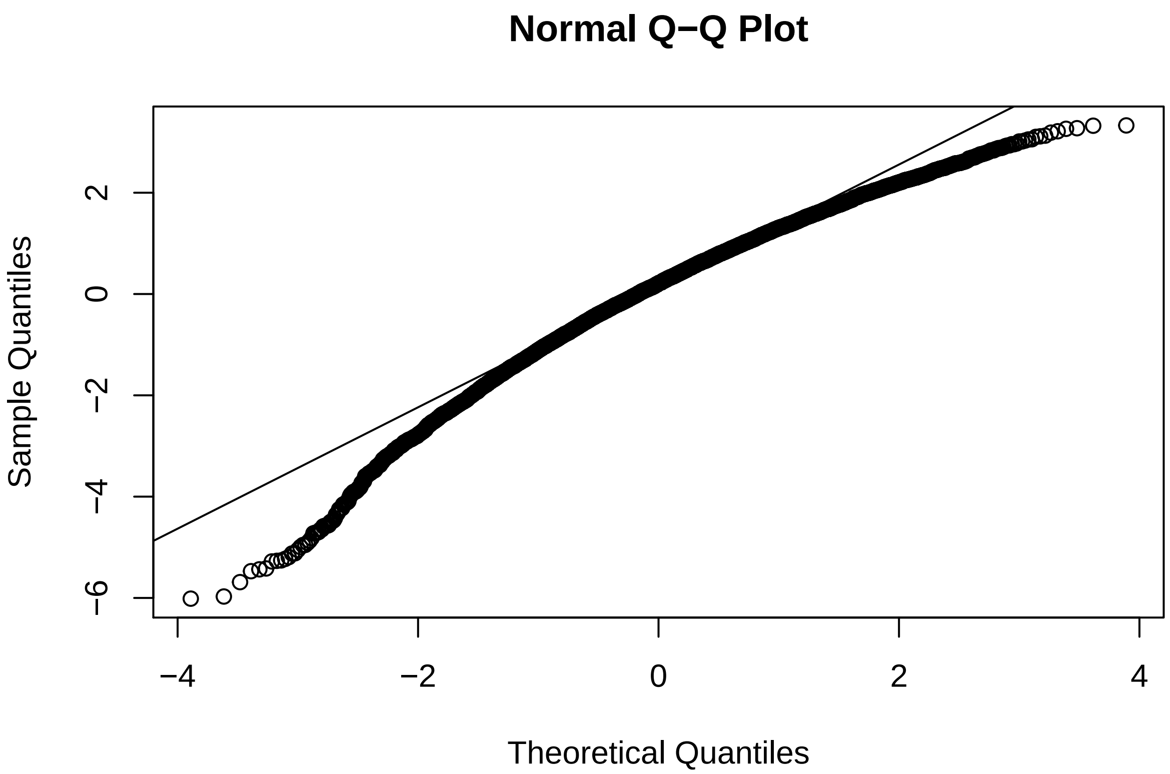
```
hist(Boot.est.seq1[2,], main = "W.alpha.hat")
```



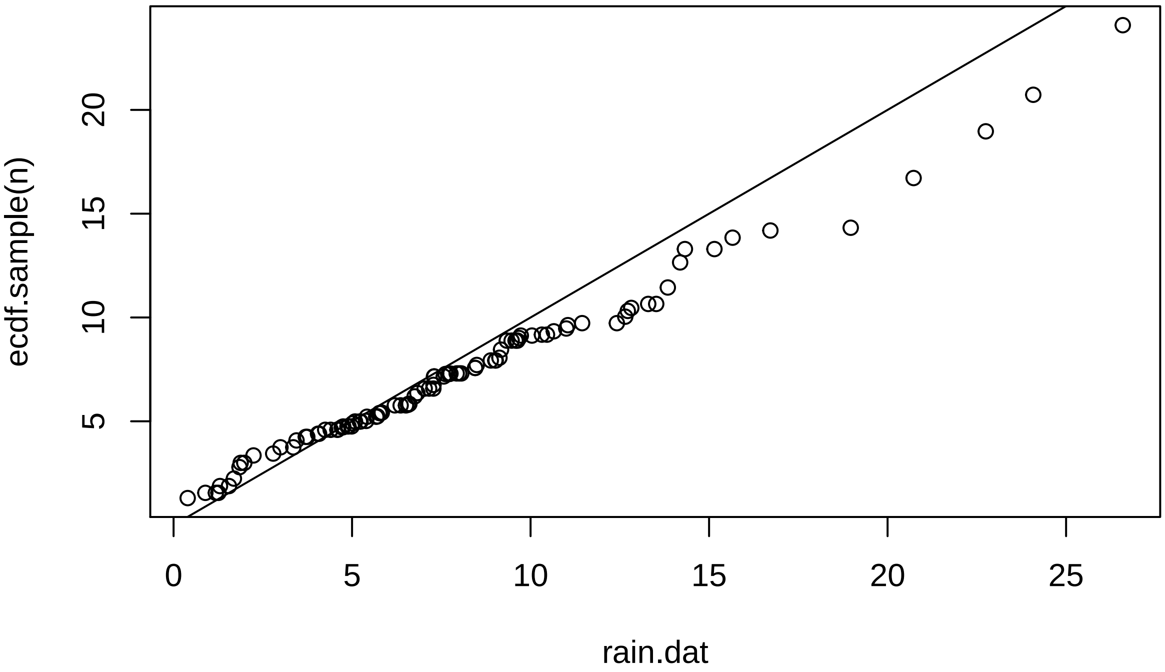
```
qqnorm(Boot.est.seq1[1,]); qqline(Boot.est.seq1[1,])
```



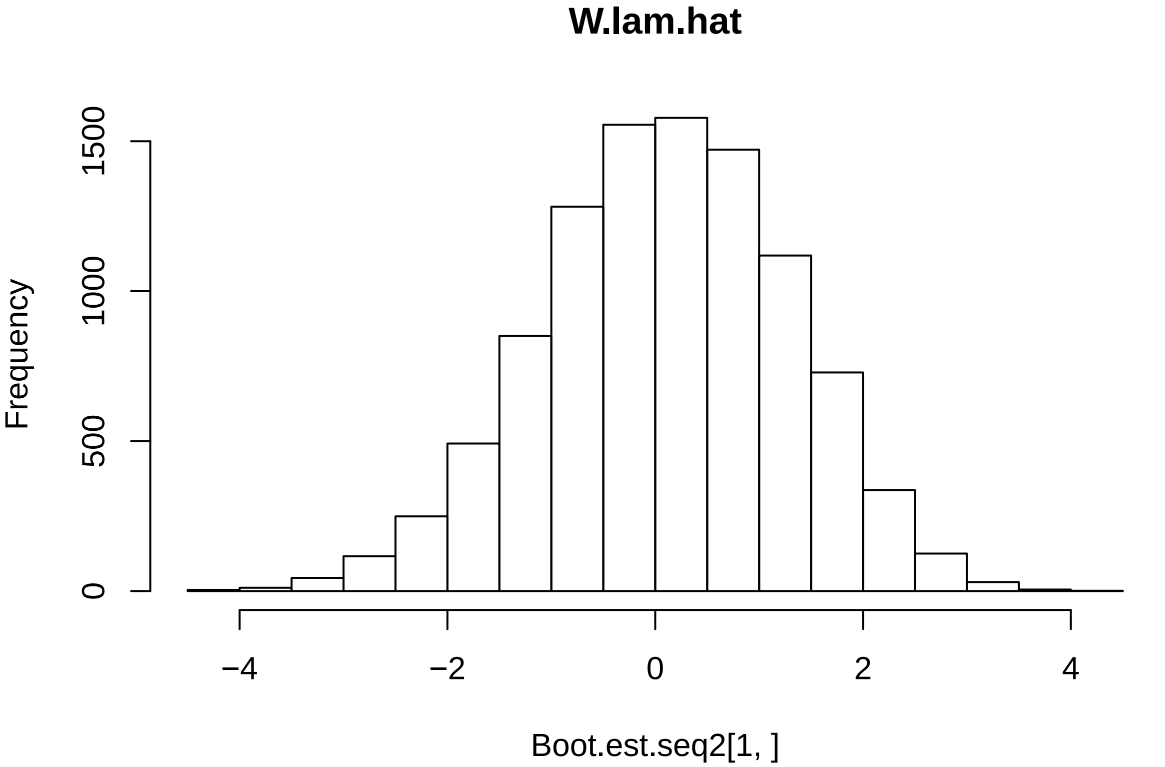
```
qqnorm(Boot.est.seq1[2,]); qqline(Boot.est.seq1[2,])
```



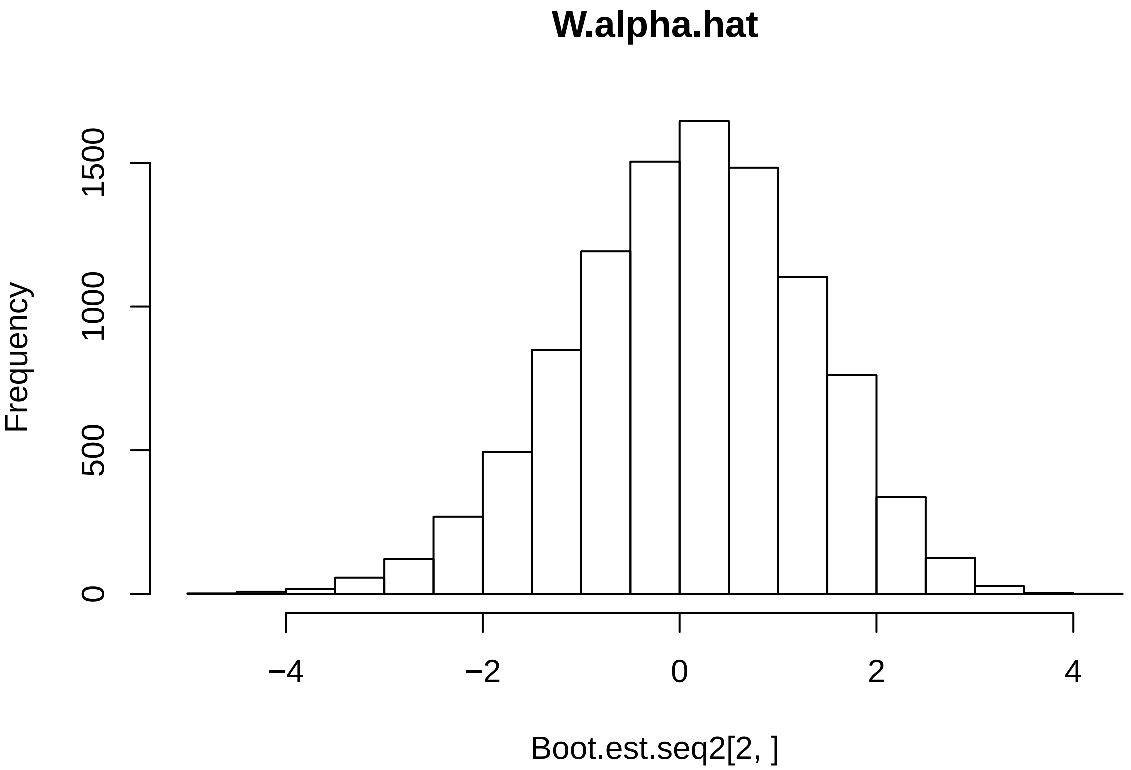
```
#non parametric

ecdf.sample <- function(num=n, data.sample=rain.dat){
  #n=length(rain.dat)
  x <- data.sample
  i.sample = sample(1:n , size = num , replace = T)
  x[i.sample]
}

#compare data with X.star
qqplot(rain.dat, ecdf.sample(n)) #n is so small
abline(0,1)
```



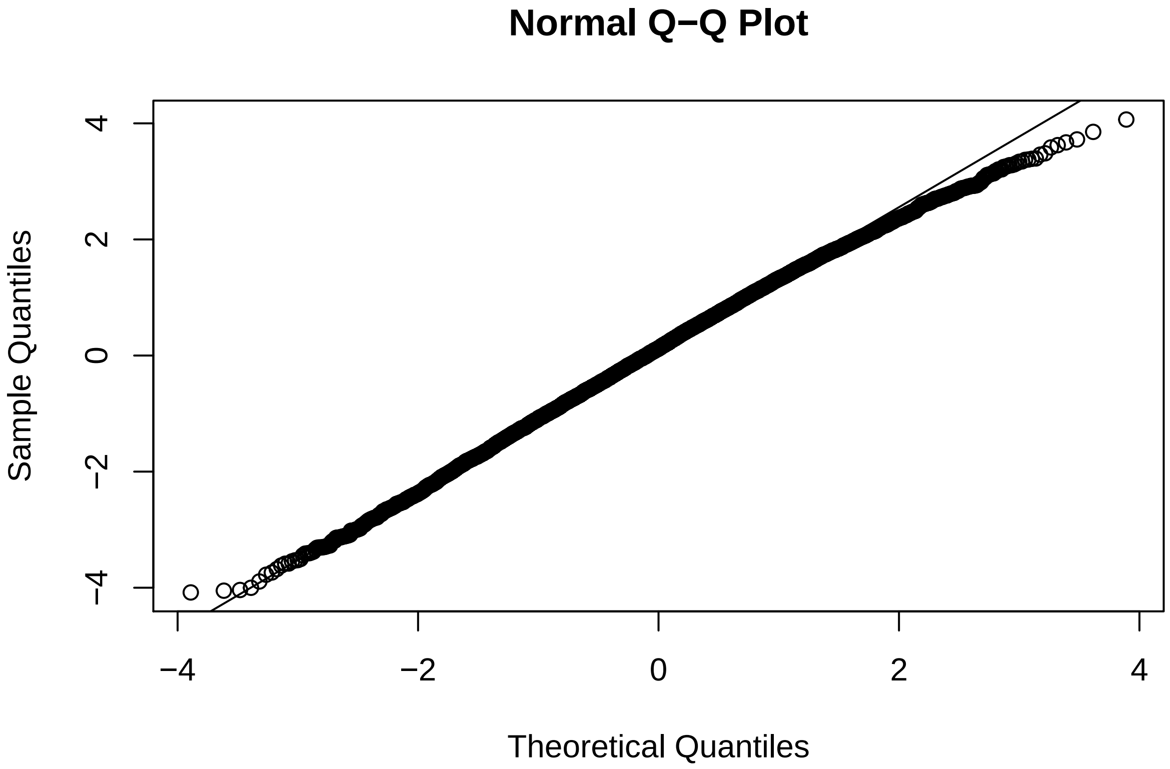
```r
Boot.est.seq2 <- replicate(M, Boot.est(ecdf.sample(n)))

hist(Boot.est.seq2[1,], main = "W.lam.hat") #non-par makes it like normal
```



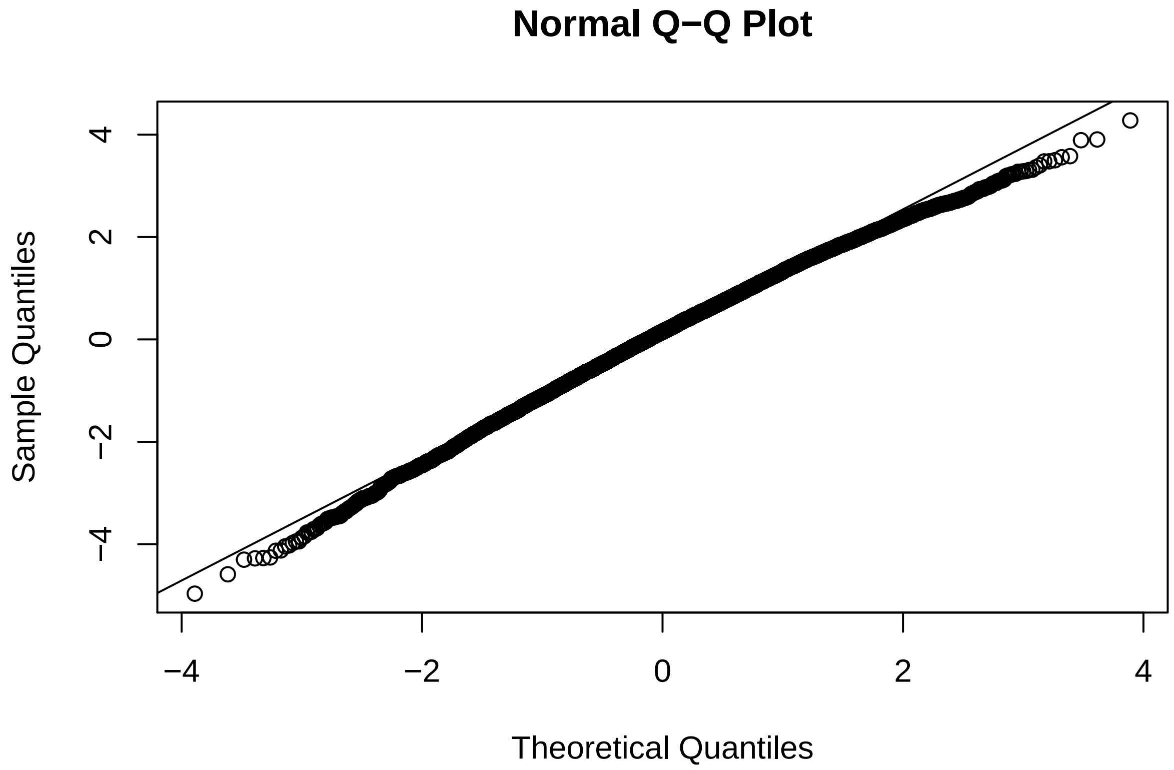
```r
hist(Boot.est.seq2[2,], main = "W.alpha.hat") #non-par make it like normal
```



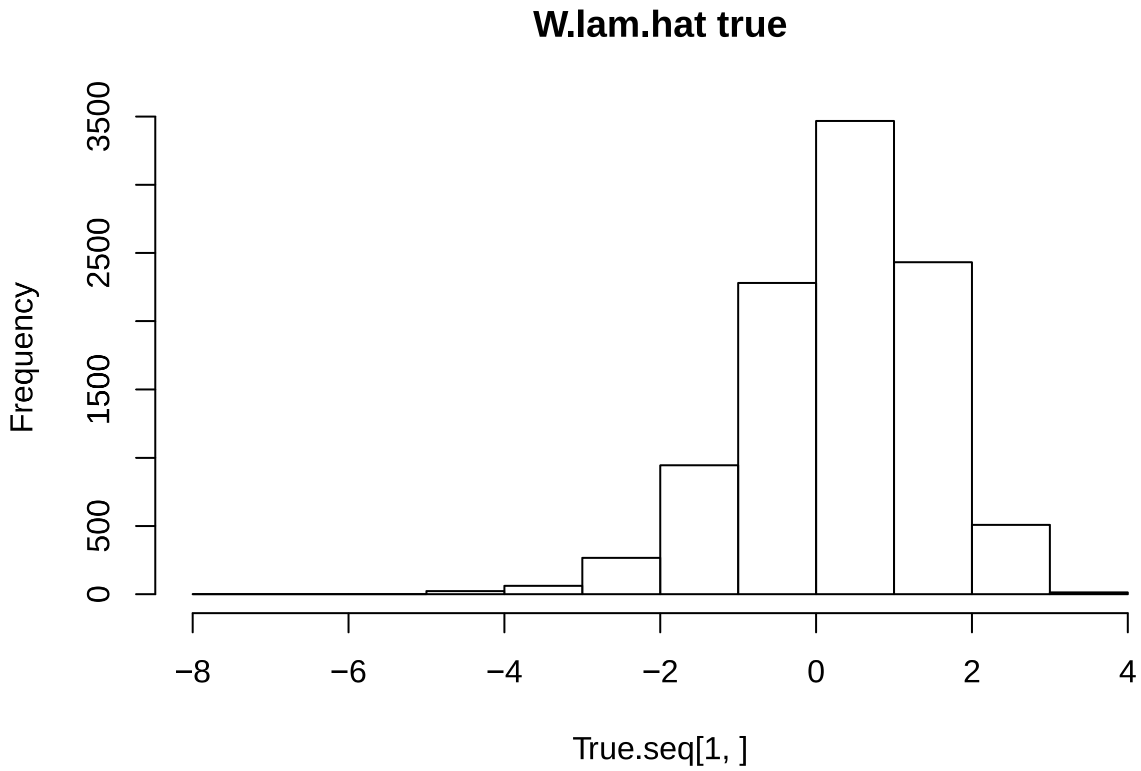
```
#non-par vs normal
qqnorm(Boot.est.seq2[1,]); qqline(Boot.est.seq2[1,])
```



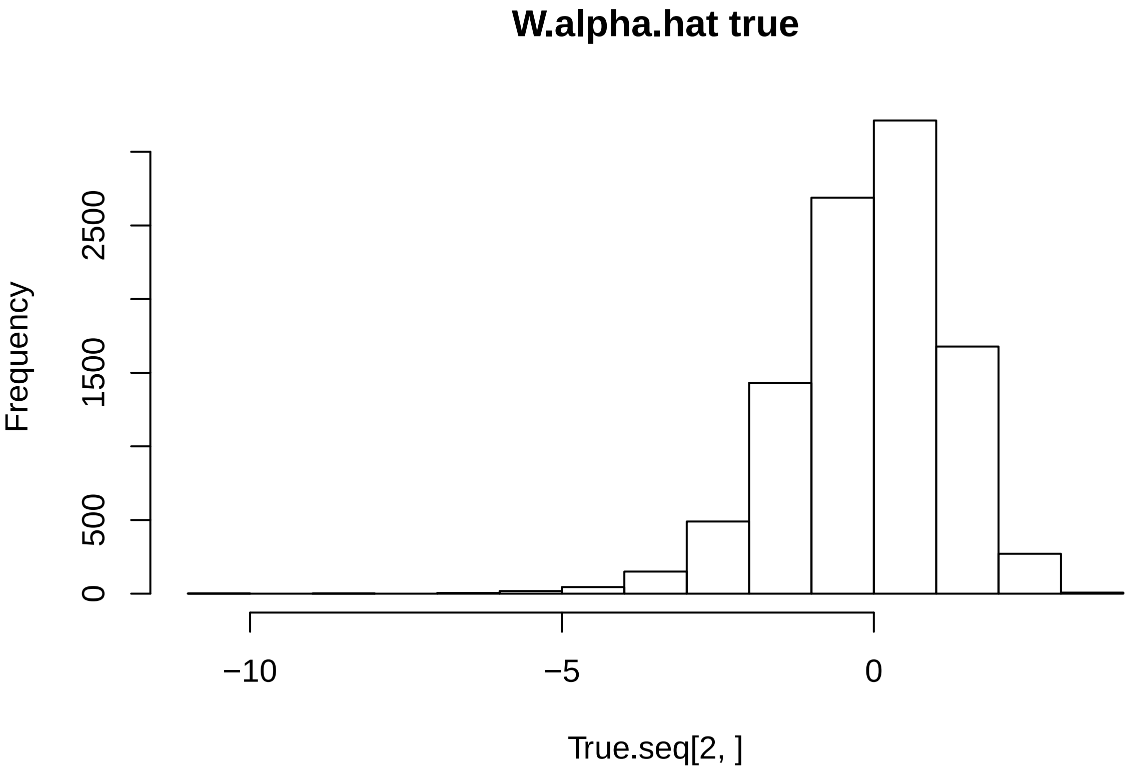
```
qqnorm(Boot.est.seq2[2,]); qqline(Boot.est.seq2[2,])
```



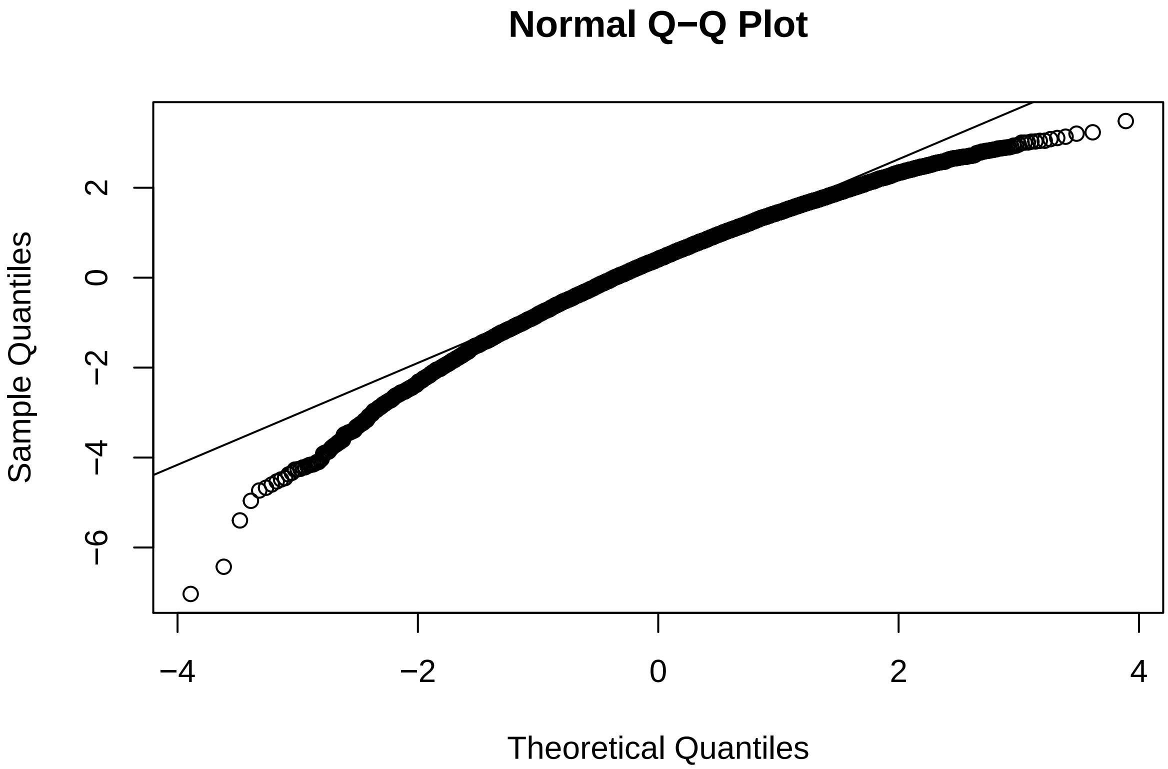
```
#but the true distribution are not normal
True.seq <- replicate(M, Boot.est(rgamma(n, shape = 2.5, rate = 1/3))) #with true parameter
hist(True.seq[1,], main = "W.lam.hat true")
```



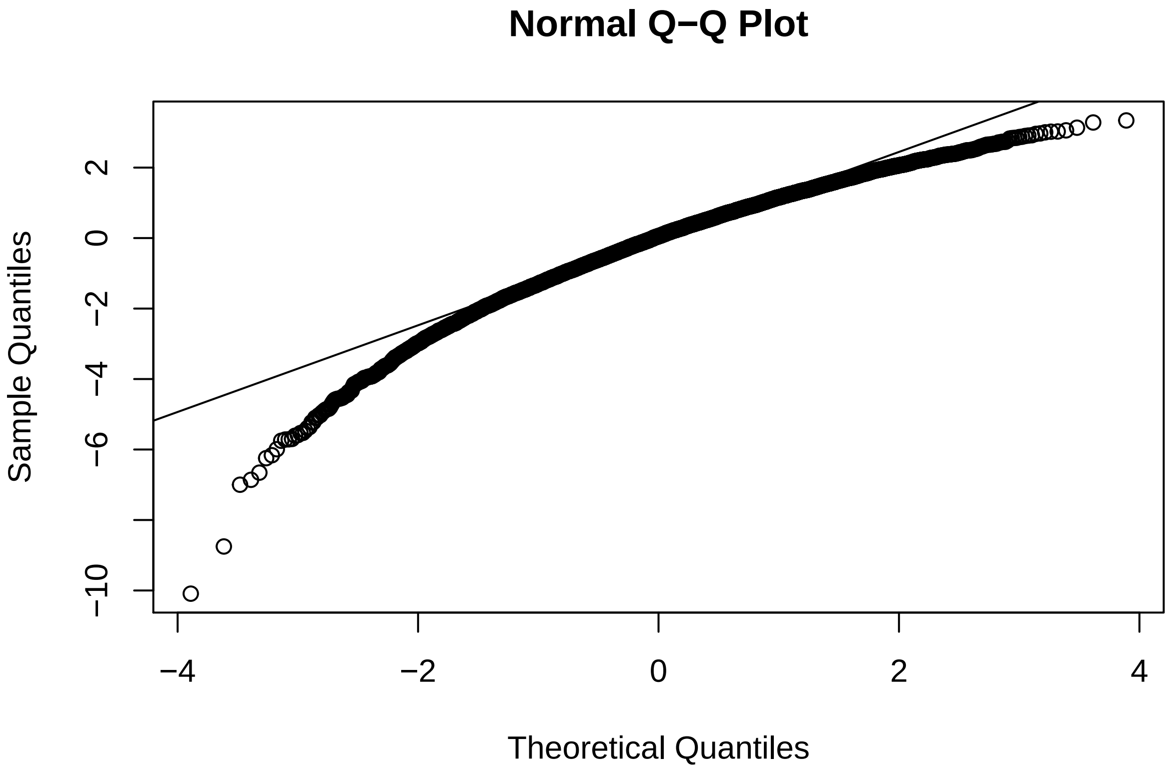
```r
hist(True.seq[2,], main = "W.alpha.hat true")
```



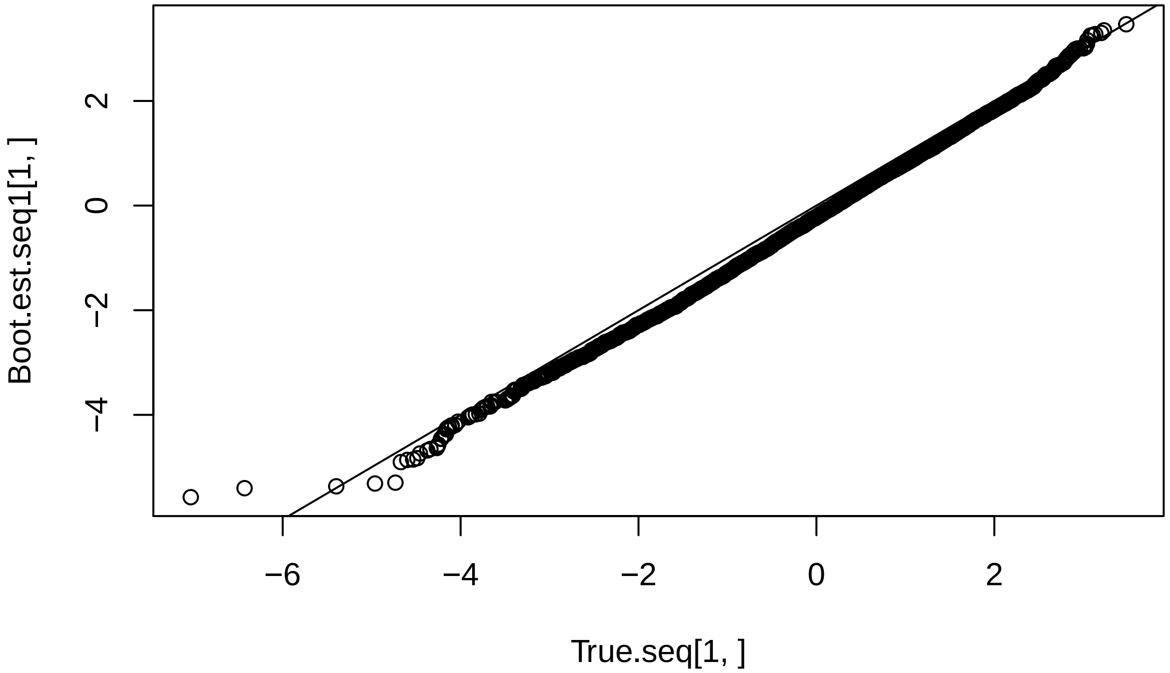
```r
#true vs normal
qqnorm(True.seq[1,]); qqline(True.seq[1,])
```



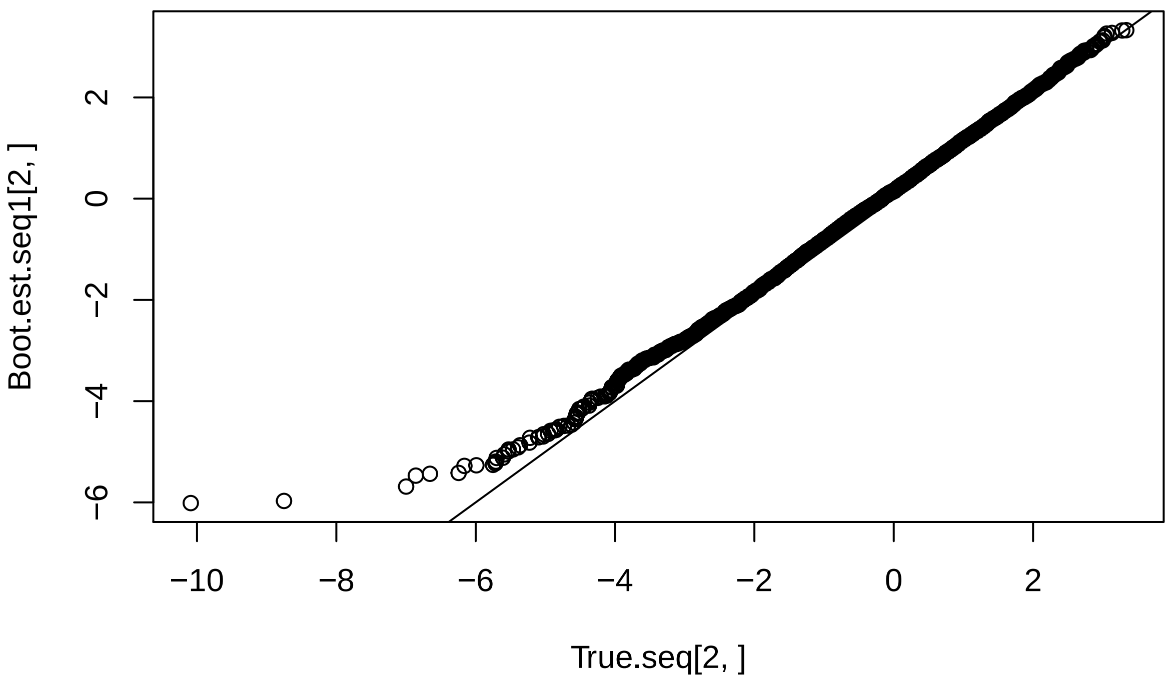
```
qqnorm(True.seq[2,]); qqline(True.seq[2,])
```



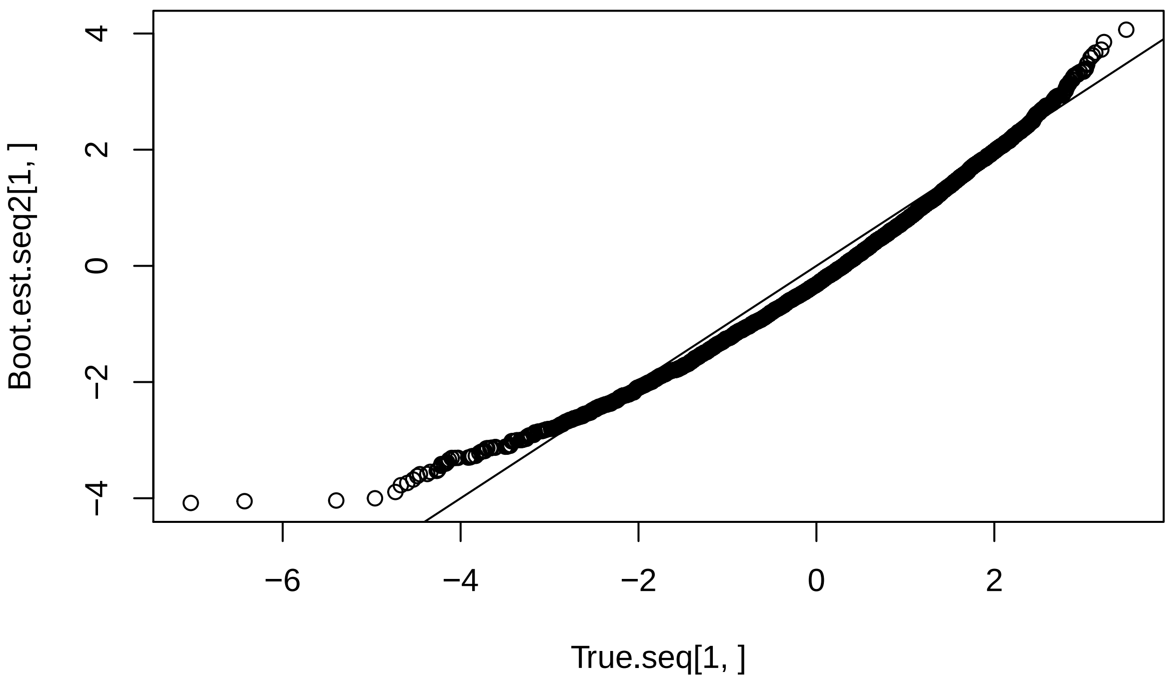
```
#true vs par
#maybe some translation happens, due to the low accuarcy of the estimates
qqplot(True.seq[1,], Boot.est.seq1[1,]); abline(0,1)
```



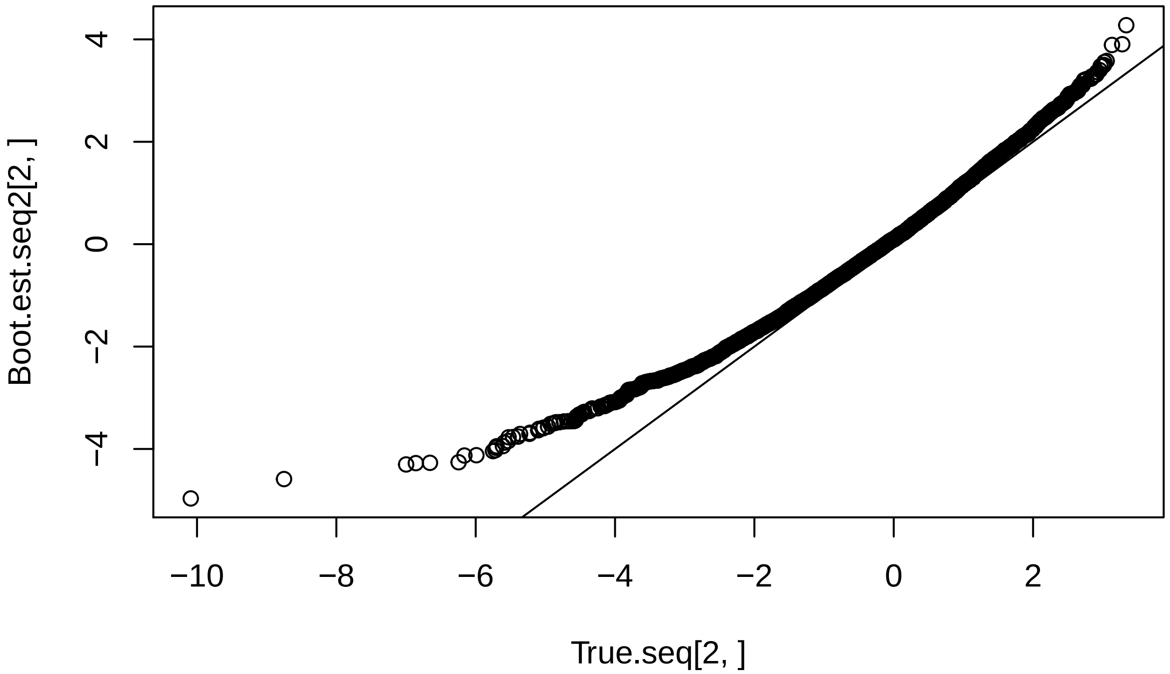
```
qqplot(True.seq[2,], Boot.est.seq1[2,]); abline(0,1)
```



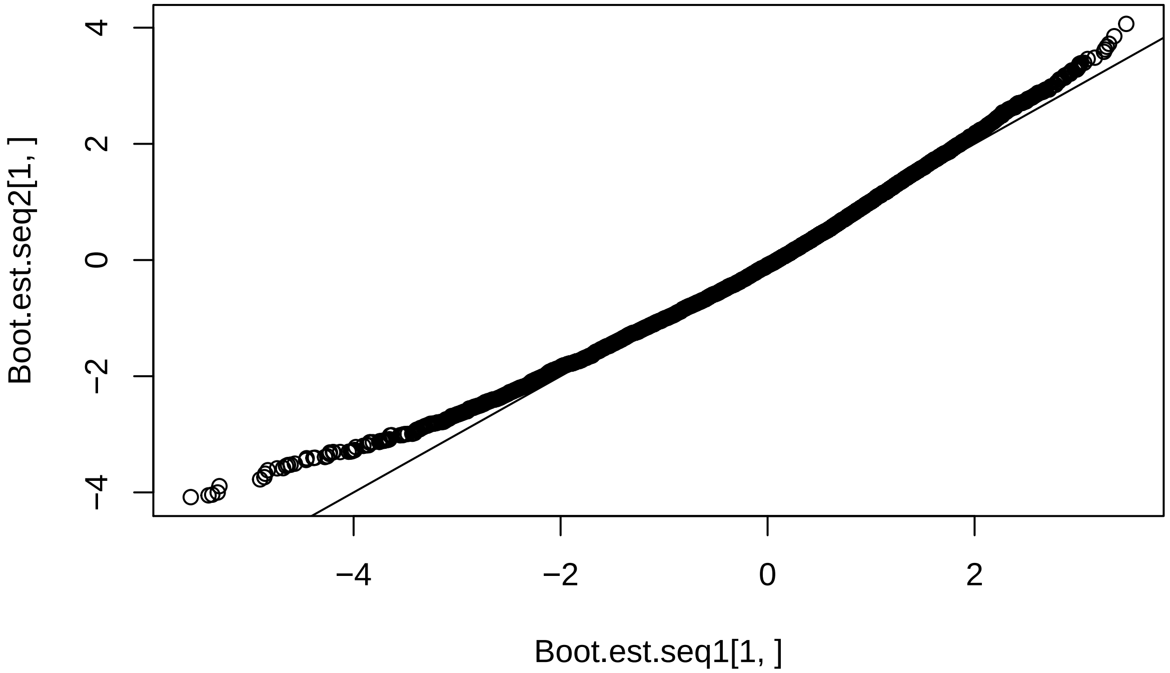
```r
#true vs non-par
qqplot(True.seq[1,], Boot.est.seq2[1,]); abline(0,1)
```



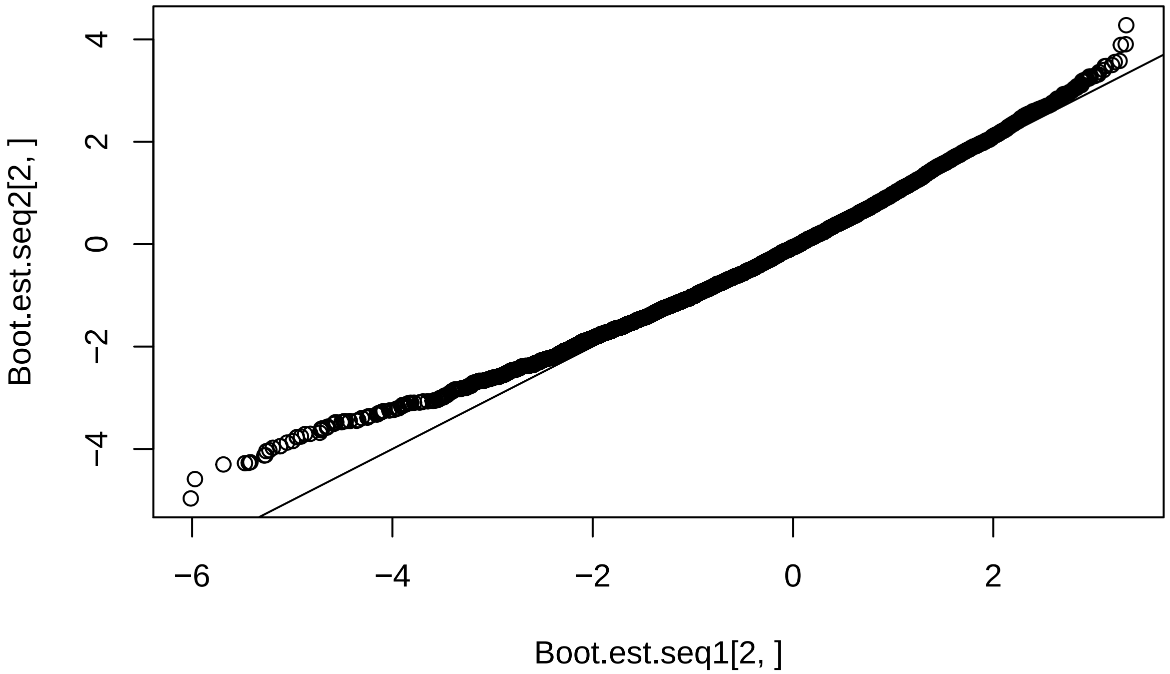
```r
qqplot(True.seq[2,], Boot.est.seq2[2,]); abline(0,1)
```



```
#compare par vs non-par
qqplot(Boot.est.seq1[1, ], Boot.est.seq2[1,]) #they are different
abline(0 , 1)
```



```
qqplot(Boot.est.seq1[2, ], Boot.est.seq2[2,]) #they are different
abline(0 , 1)
```



```
#par quantile
pr <- c(.025 , .975)
q.lam <- quantile(Boot.est.seq1[1,], pr)
q.lam
```

```
##      2.5%     97.5%
## -2.535069  2.137008
```

```
q.alpha <- quantile(Boot.est.seq1[2,], pr)
q.alpha
```

```
##      2.5%     97.5%
## -2.706911  2.167158
```

```
#confidence interval
unname(lam - q.lam[c(2,1)] * sqrt(var.lam)/sqrt(n))
```

```
## [1] 0.2211551 0.4429961
```

```
unname(alpha - q.alpha[c(2,1)] * sqrt(var.alpha)/sqrt(n))
```

```
## [1] 1.819385 3.480495
```

```
#non-par quantile
pr <- c(.025 , .975)
q.lam <- quantile(Boot.est.seq2[1,], pr)
q.lam
```

```
##      2.5%     97.5%
## -2.321283  2.325222
```

```
q.alpha <- quantile(Boot.est.seq2[2,], pr)
q.alpha
```

```
##     2.5%    97.5%
## -2.39227  2.30511
```

```
#confidence interval
unname(lam - q.lam[c(2,1)] * sqrt(var.lam)/sqrt(n))
```

```
## [1] 0.2122183 0.4328450
```

```
unname(alpha - q.alpha[c(2,1)] * sqrt(var.alpha)/sqrt(n))
```

```
## [1] 1.772370 3.373263
```