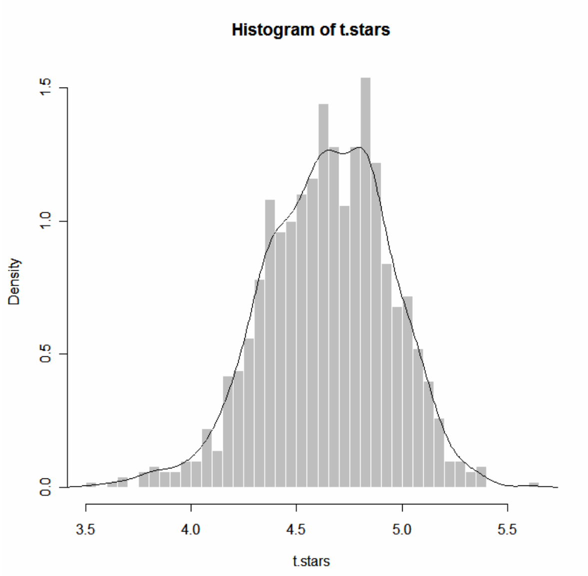
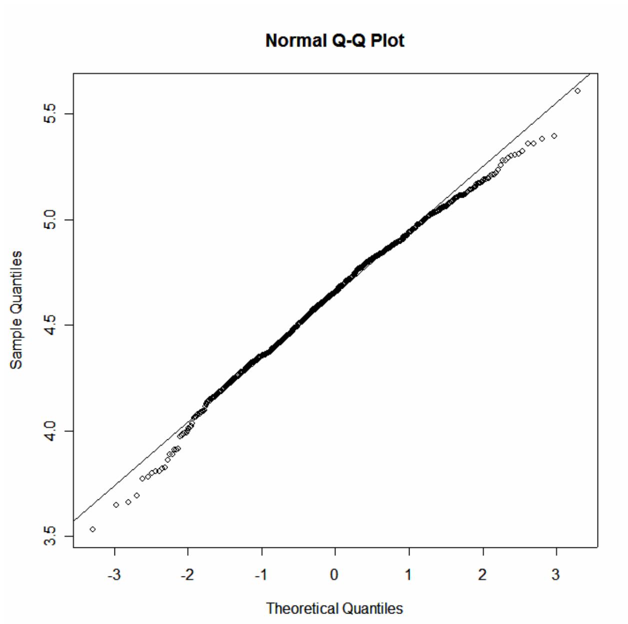
Stat 6601 Midterm Solution

We are interested in estimating θ=log(μ), the log mean inter-failure time.

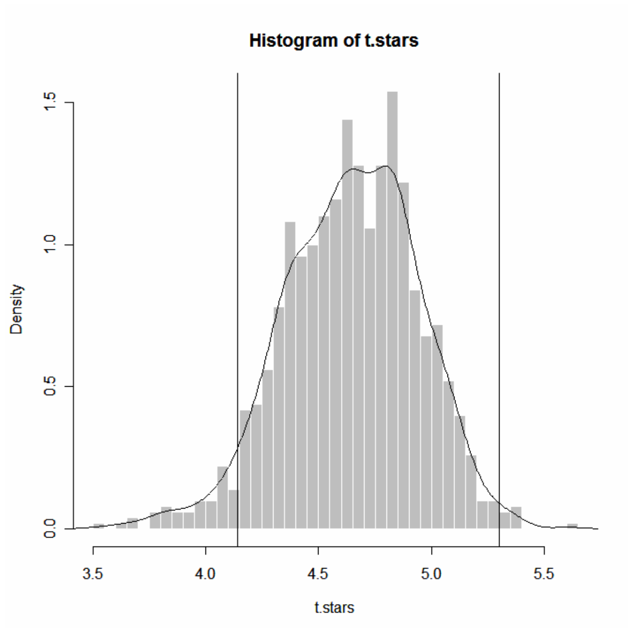
```
                                        #
                                        # Generate data
                                        #
y <- c(3, 5, 7, 18, 43, 85, 91, 98, 100, 130, 230, 487)
n <- length(y)

                                        # 1)
log(mean(y))
4.682903

                                        # 2)
y.stars <- rexp(n, 1/mean(y))
log(mean(y.stars))

                                        # 3)
R <- 1000
t.stars <- numeric(R)
set.seed(101)
for(r in 1:R){
  y.stars <- rexp(n, 1/mean(y))
  t.stars[r] <- log(mean(y.stars))
}

                                        # 4)
hist(t.stars, nclass=50, prob=TRUE,col='gray',border='white')
lines(density(t.stars))
```



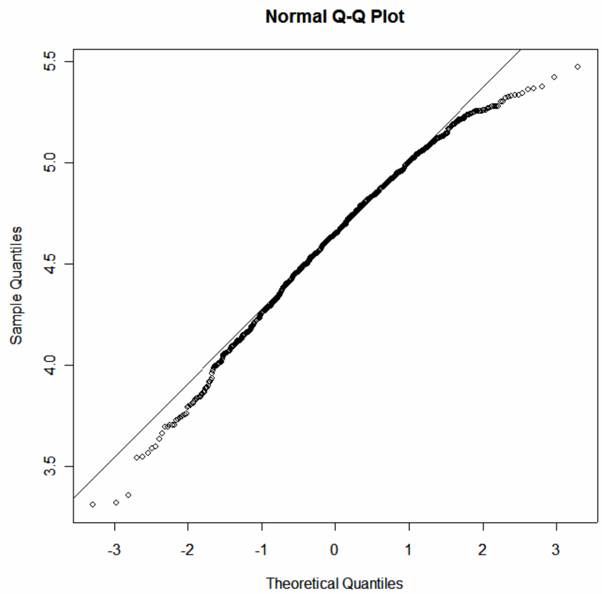

```
                                        # 5)
qqnorm(t.stars)
qqline(t.stars)
```

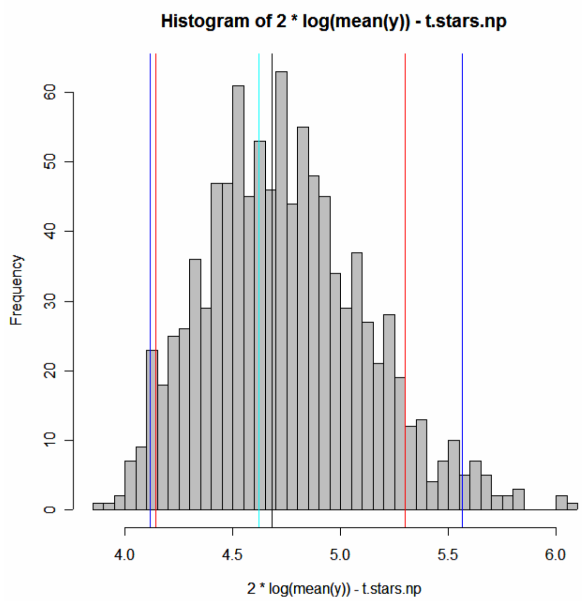



```
                                                        # 6)
B <- mean(t.stars)-log(mean(y)) # vs 4.68
V <- var(t.stars)

B
[1] -0.03787144
> V
[1] 0.08765792
>

                                                        # 7)
hist(t.stars, nclass=50, prob=TRUE,col='gray',border='white')
lines(density(t.stars))
p.ci <- c(log(mean(y))-B-qnorm(.975)*sqrt(V),
          log(mean(y))-B+qnorm(.975)*sqrt(V))
abline(v=p.ci)
```




```
                                        # 8
R <- 999
t.stars.np <- numeric(R)
set.seed(101)
for(r in 1:R){
  y.stars <- sample(y, replace=TRUE)
  t.stars.np[r] <- log(mean(y.stars))
}

                                        # 9, 10, 11
                                        # doesn't look very normal
alpha <- .025
par(mfrow=c(1,1))
np.ci <- c(log(mean(y))-(quantile(t.stars.np, 1-alpha) - log(mean(y))),
         log(mean(y))-(quantile(t.stars.np, alpha) - log(mean(y))))

qqnorm(t.stars.np); qqline(t.stars.np)
hist(t.stars.np, nclass=50, col='gray') #
hist(2*log(mean(y))- t.stars.np, nclass=50, col='gray') # better, but not a
must
abline(v=p.ci, col='red')
abline(v=np.ci, col='blue')
abline(v=log(mean(y)))
abline(v=mean(t.stars.np), col='cyan')
```

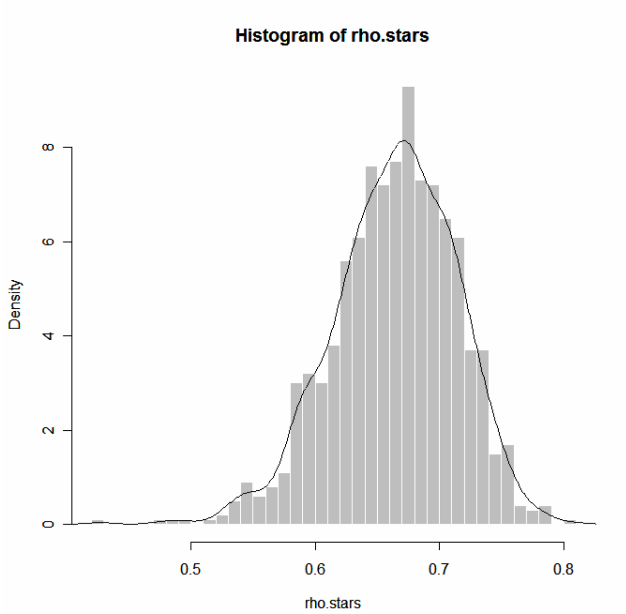





```
# 12, 13
# NP bootstrap has more accurate
# coverage probability
length(which(t.stars.np < np.ci[2] & t.stars.np > np.ci[1]))/R
length(which(t.stars.np < p.ci[2] & t.stars.np > p.ci[1]))/R
length(which(2*log(mean(y))-t.stars.np < np.ci[2] &
             2*log(mean(y))-t.stars.np > np.ci[1]))/R
0.94995
length(which(2*log(mean(y))-t.stars.np < p.ci[2] &
             2*log(mean(y))-t.stars.np > p.ci[1]))/R
0.8898899
```

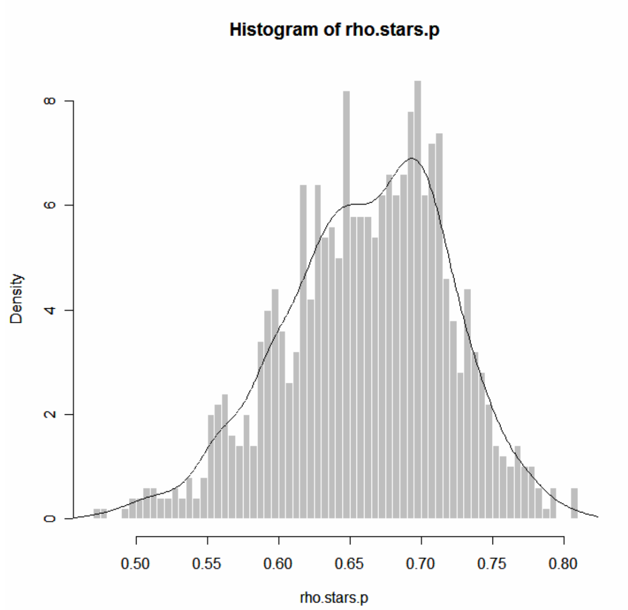
#2. (50 pt + 20 pt) Suppose n=100 independent observations of $Y_j = (X_j, Z_j)$ are observed. It's stored in "corr.dat" on the class webpage.

```
                                                # 1
data <- read.table("corr.dat")
n <- nrow(data)
cor(data[,1], data[,2])
cor(data$x, data$y)
0.6632465

                                                # 2
data.star <- data[sample(1:n, replace=TRUE),]
cor(data.star$x, data.star$y)
0.6395324


                                                # 3
R <- 1000
rho.stars <- numeric(R)
set.seed(101)
for(r in 1:1000){
  data.star <- data[sample(1:n, replace=TRUE),]
  rho.stars[r] <- cor(data.star[,1], data.star[,2])
}

                                                # 4
hist(rho.stars, nclass=50, prob=TRUE,col='gray',border='white')
lines(density(rho.stars))
```




```
> rho
[1] 0.6632465
> mean(rho.stars)
[1] 0.6639841
> rho.np.ci
    97.5%      2.5%
0.5757941 0.7708829
>

                                                # 6
```

```
library(MASS)
R <- 1000
rho.stars.p <- numeric(R)
set.seed(101)
n <- nrow(data)
for(r in 1:1000){
  data.star <- mvrnorm(n, apply(data, 2, mean),
                       cov(data))
  rho.stars.p[r] <- cor(data.star[,1], data.star[,2])
}

alpha <- .025
rho.p.ci <- c(2*rho-quantile(rho.stars.p, 1-alpha),
              2*rho-quantile(rho.stars.p, alpha))
mean(rho.stars.p)
[1] 0.661292
rho.p.ci
   97.5%      2.5%
0.560350 0.786871
```



* By the way, this is how I generated the data: (True rho=0.7)

```
library(MASS)
set.seed(101)
data <- mvrnorm(100, c(0,0), matrix(c(1, .7, .7, 1), 2,2))
dimnames(data) <- list(NULL, c('x','y'))
data <- round(data*100)
write.table(data,"corr.dat")
```