STAD80: Assignment 1

Vladislav Trukhin

Due: Jan 27

Contents

Question 1

1.1

B, C

1.2

A, B, C

1.3

B

1.4

A, C

1.5

A, F

1.6

A

Question 2

2.1

As $\sqrt{n}(\theta - \hat{\theta}_n) \xrightarrow{D} N(0, \frac{1}{I(\theta)})$ and $\sqrt{I(\hat{\theta}_n)} \xrightarrow{P} \sqrt{I(\theta)}$, by Slutsky's Theorem:

$\sqrt{I(\hat{\theta}_n)}\sqrt{n}(\theta - \hat{\theta}_n) = \sqrt{nI(\hat{\theta}_n)}(\theta - \hat{\theta}_n) \xrightarrow{D} \sqrt{I(\theta)}N(0, \frac{1}{I(\theta)}) = N(0, 1)$

Using this result:

$lim_{n\to\infty}P(\theta \in C_n)$

$= lim_{n\to\infty} P(\hat{\theta}_n - \frac{z_{\alpha/2}}{\sqrt{nI(\hat{\theta}_n)}} \leq \theta \leq \hat{\theta}_n + \frac{z_{\alpha/2}}{\sqrt{nI(\hat{\theta}_n)}})$

$= lim_{n\to\infty} P(-z_{\alpha/2} \leq \sqrt{nI(\hat{\theta}_n)}(\theta - \hat{\theta}_n) \leq z_{\alpha/2})$

$= P(-z_{\alpha/2} \leq Y \leq z_{\alpha/2})$ where $Y \sim N(0,1)$

$= P(Y \leq z_{\alpha/2}) - P(Y \leq -z_{\alpha/2})$

$= 1 - P(Y \leq -z_{\alpha/2}) - P(Y \leq -z_{\alpha/2}))$

$= 1 - 2P(Y \leq -z_{\alpha/2})$

$= 1 - 2\alpha/2$

$= 1 - \alpha$

2.2a

$\ell(\theta, X_1, ..., X_n) = ln(\Pi(\theta-1)x_i^{-\theta}1(x_i \geq 1))$

$= \Sigma(\ln((\theta-1)x_i^{-\theta}1(x_i \geq 1)))$

$= \Sigma(\ln(\theta-1) + \ln(x_i^{-\theta}) + \ln(1(x_i \geq 1)))$

$= \Sigma(\ln(\theta-1) - \theta\ln(x_i) + \ln(1(x_i \geq 1)))$

$= n\ln(\theta-1) - \theta\Sigma(\ln(x_i)) + \Sigma(\ln(1(x_i \geq 1))))$

$\frac{\partial}{\partial\theta}\ell(\theta, X_1, ..., X_n) = \frac{\partial}{\partial\theta}(n\ln(\theta-1) - \theta\Sigma(\ln(x_i)) + \Sigma(\ln(1(x_i \geq 1))))$

$= \frac{\partial}{\partial\theta}n\ln(\theta-1) - \frac{\partial}{\partial\theta}\theta\Sigma(\ln(x_i)) + \frac{\partial}{\partial\theta}\Sigma(\ln(1(x_i \geq 1)))$

$= \frac{n}{\theta-1} - \Sigma(\ln(x_i)) = 0$

$\implies \frac{n}{\theta-1} = \Sigma(\ln(x_i))$

$\implies n = (\theta-1)\Sigma(\ln(x_i))$

$\implies n = \theta\Sigma(\ln(x_i)) - \Sigma(\ln(x_i))$

$\implies n + \Sigma(\ln(x_i)) = \theta\Sigma(\ln(x_i))$

$\implies \frac{n+\Sigma(\ln(x_i))}{\Sigma(\ln(x_i))} = \theta = \hat{\theta}_n$

2.2b

$I(\theta) = E(-\frac{\partial^2}{\partial\theta^2}\ln p_\theta(X))$

$= E(-\frac{\partial^2}{\partial\theta^2}(\ln((\theta-1)X^{-\theta}1(X \geq 1))))$

$= E(-\frac{\partial^2}{\partial\theta^2}(\ln(\theta-1) + \ln(X^{-\theta}) + \ln(1(X \geq 1))))$

$= E(-\frac{\partial^2}{\partial\theta^2}(\ln(\theta-1) - \theta\ln(X) + \ln(1(X \geq 1))))$

$= E(-\frac{\partial^2}{\partial\theta^2}\ln(\theta-1) + \frac{\partial^2}{\partial\theta^2}\theta\ln(X) - \frac{\partial^2}{\partial\theta^2}\ln(1(X \geq 1)))$

$= E(-\frac{\partial}{\partial\theta}\frac{1}{\theta-1})$

$= E(\frac{1}{(\theta-1)^2})$

$= \int_{-\infty}^{\infty}\frac{1}{(\theta-1)^2}(\theta-1)X^{-\theta}1(X \geq 1)$

$= \int_1^{\infty}\frac{1}{(\theta-1)}X^{-\theta}$

$= \frac{1}{(\theta-1)}\int_1^{\infty}X^{-\theta}$

$= \frac{1}{(\theta-1)} \frac{X^{-\theta+1}}{-\theta+1}|_1^\infty$

$= -\frac{1}{(\theta-1)^2} X^{-\theta+1}|_1^\infty$

$= -\frac{1}{(\theta-1)^2} * 0 + \frac{1}{(\theta-1)^2} * 1$

$= \frac{1}{(\theta-1)^2}$

$\implies \frac{1}{I(\theta)} = (\theta-1)^2$

2.2c

$C_n = [\hat{\theta}_n - \frac{z_{\alpha/2}}{\sqrt{nI(\hat{\theta}_n)}}, \hat{\theta}_n + \frac{z_{\alpha/2}}{\sqrt{nI(\hat{\theta}_n)}}]$

$= [\hat{\theta}_n - \frac{z_{\alpha/2}}{\sqrt{\frac{n}{(\hat{\theta}_n-1)^2}}}, \hat{\theta}_n + \frac{z_{\alpha/2}}{\sqrt{\frac{n}{(\hat{\theta}_n-1)^2}}}]$

$= [\hat{\theta}_n - \frac{z_{\alpha/2}}{\frac{\sqrt{n}}{(\hat{\theta}_n-1)}}, \hat{\theta}_n + \frac{z_{\alpha/2}}{\frac{\sqrt{n}}{(\hat{\theta}_n-1)}}]$

$= [\hat{\theta}_n - z_{\alpha/2}\frac{(\hat{\theta}_n-1)}{\sqrt{n}}, \hat{\theta}_n + z_{\alpha/2}\frac{(\hat{\theta}_n-1)}{\sqrt{n}}]$

$= [\hat{\theta}_n - 1.96\frac{(\hat{\theta}_n-1)}{\sqrt{n}}, \hat{\theta}_n + 1.96\frac{(\hat{\theta}_n-1)}{\sqrt{n}}]$

2.2d

```r
invcdf <- function(y, theta) {
  return ((1-y)^(1/(-theta+1)))
}

N=10000
n=100
theta=2
count=0
for (i in 1:N) {
  Y <- runif(n, 0, 1)
  X <- invcdf(Y, 2)
  theta_hat <- (n + sum(log(X))) / sum(log(X))
  c_l <- theta_hat - 1.96*(theta_hat-1)/sqrt(n)
  c_u <- theta_hat + 1.96*(theta_hat-1)/sqrt(n)
  if (c_l <= theta & theta <= c_u) {
    count = count + 1
  }
}
count/N
```

```
## [1] 0.9537
```

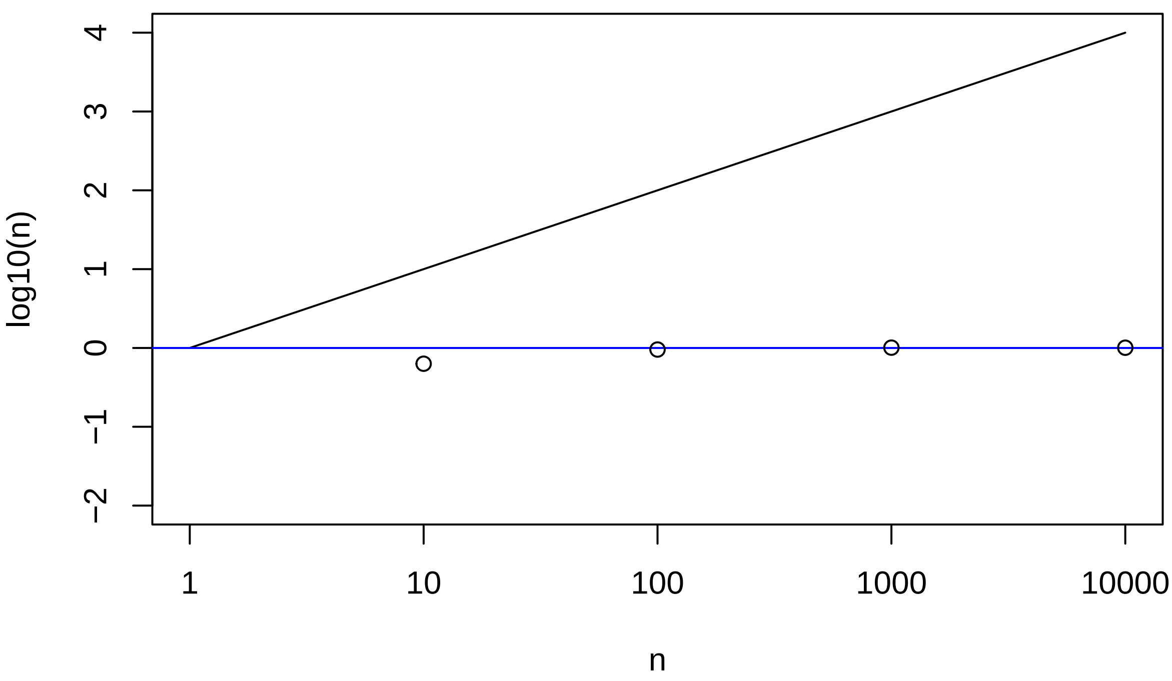
Therefore the 95% CI is effective.

Question 3

3a

```r
generate <- function(n){
  N <- 10000
  Xbar_n <- vector(mode = "list", length = N)
```

```
  for (j in 1:N) {
    X = runif(n, 0, 1)
    for (i in 1:n) {
      if (X[i] < 0.5) {X[i] = -1}
      else {X[i] = 1}
    }
  Xbar_n[j] <- mean(X)
  }
return(Xbar_n)
}

Xbar_10 <- as.numeric(generate(10))
Xbar_100 <- as.numeric(generate(100))
Xbar_1000 <- as.numeric(generate(1000))
Xbar_10000 <- as.numeric(generate(10000))

curve(log10(x), from=1, to=10000, ylim=c(-2,4), log="x", xlab = "n", ylab = "log10(n)")
abline(h = 0, col="blue")
points(10, Xbar_10[1] - 0)
points(100, Xbar_100[1] - 0)
points(1000, Xbar_1000[1] - 0)
points(10000, Xbar_10000[1] - 0)
```

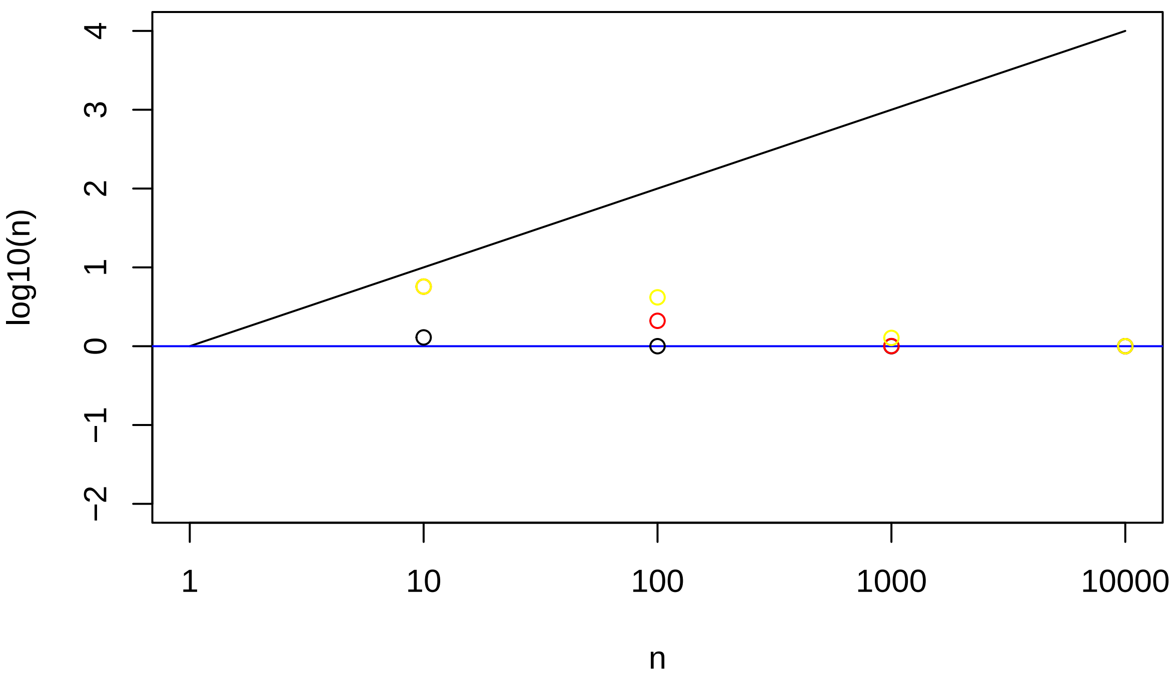


The plot shows as $n \to \infty$, $(\bar{X}_n^{(1)} - \mu) \to 0$ or $\bar{X}_n^{(1)} \to \mu$.

3b

```
lln <- function(X, e) {
  N <- 10000
  for (i in 1:N) {
    if (abs(X[i] - 0) > e) {X[i] = 1}
    else {X[i] = 0}
```

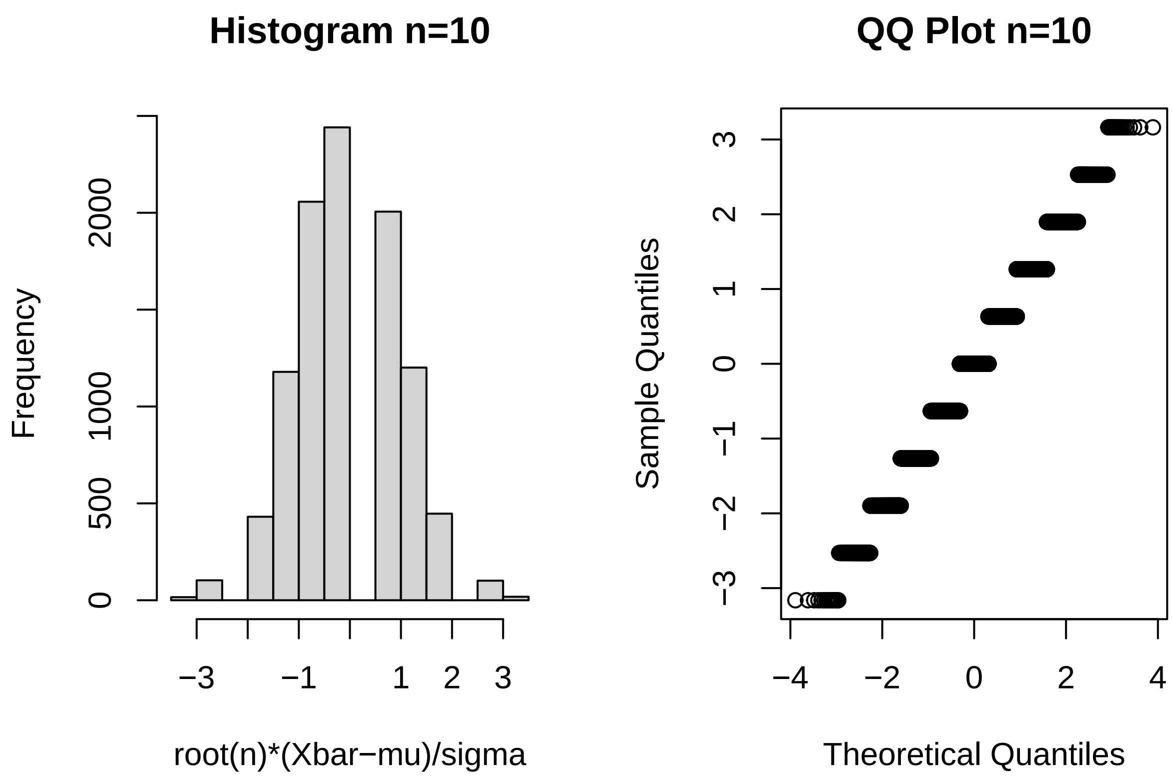
```
  }
  return(mean(X))
}

curve(log10(x), from=1, to=10000, ylim=c(-2,4), log="x", xlab = "n", ylab = "log10(n)")
abline(h = 0, col="blue")
points(10, lln(Xbar_10, 0.5) - 0)
points(100, lln(Xbar_100, 0.5) - 0)
points(1000, lln(Xbar_1000, 0.5) - 0)
points(10000, lln(Xbar_10000, 0.5) - 0)

points(10, lln(Xbar_10, 0.1) - 0, col = "red")
points(100, lln(Xbar_100, 0.1) - 0, col = "red")
points(1000, lln(Xbar_1000, 0.1) - 0, col = "red")
points(10000, lln(Xbar_10000, 0.1) - 0, col = "red")

points(10, lln(Xbar_10, 0.05) - 0, col = "yellow")
points(100, lln(Xbar_100, 0.05) - 0, col = "yellow")
points(1000, lln(Xbar_1000, 0.05) - 0, col = "yellow")
points(10000, lln(Xbar_10000, 0.05) - 0, col = "yellow")
```



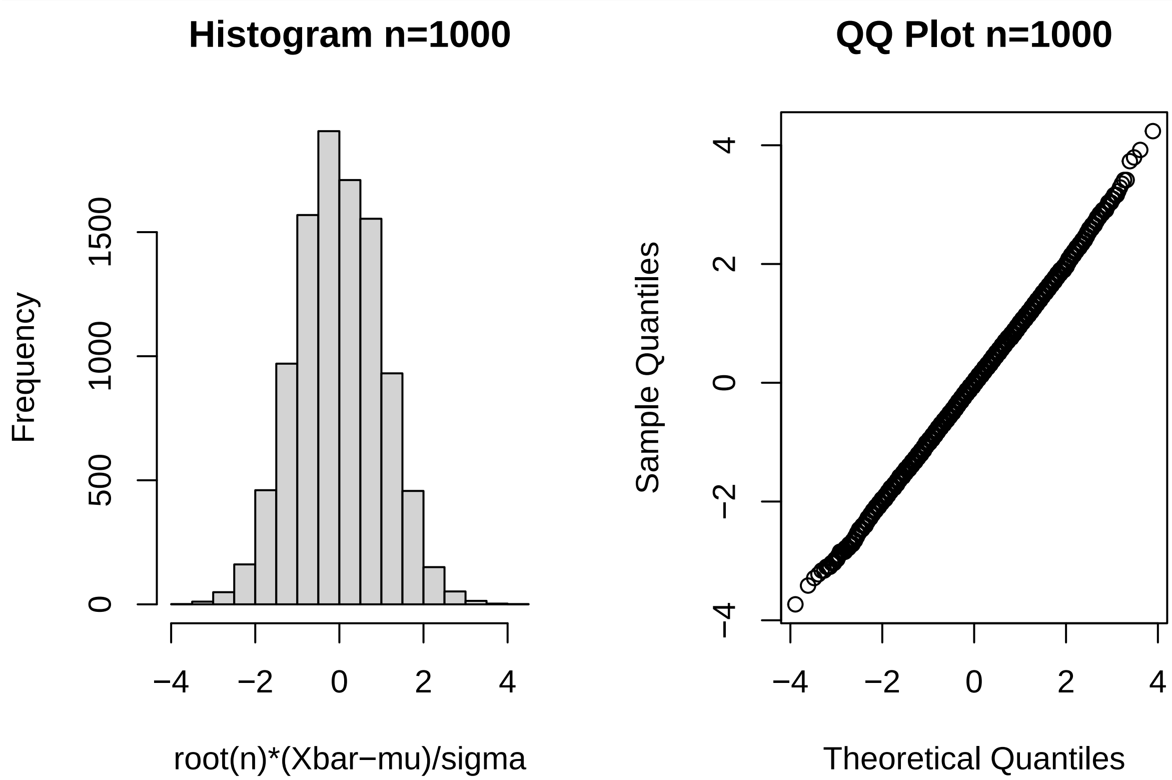
The plot shows that $\underset{n\to\infty}{lim} P(|\bar{X}_n^{(i)} - \mu| > \epsilon) = 0 \ \forall \epsilon \ \forall i$, which illustrates the Law of Large Numbers, or $\bar{X}_n^{(i)} \xrightarrow{P} \mu$.

3c

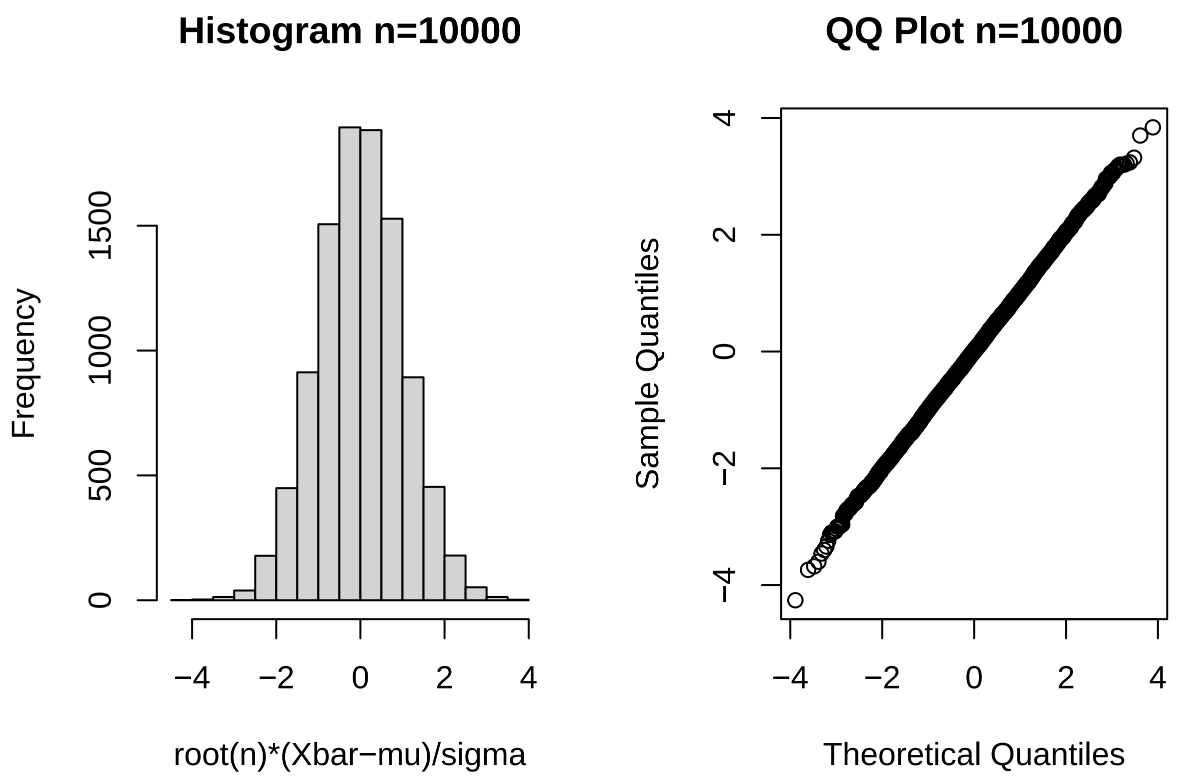
```
par(mfrow=c(1,2))
hist(sqrt(10)*(Xbar_10 - 0)/1, main = "Histogram n=10", xlab = "root(n)*(Xbar-mu)/sigma")
qqnorm(sqrt(10)*(Xbar_10 - 0)/1, main = "QQ Plot n=10")
```




```
hist(sqrt(1000)*(Xbar_1000 - 0)/1, main = "Histogram n=1000", xlab = "root(n)*(Xbar-mu)/sigma")
qqnorm(sqrt(1000)*(Xbar_1000 - 0)/1, main = "QQ Plot n=1000")
```



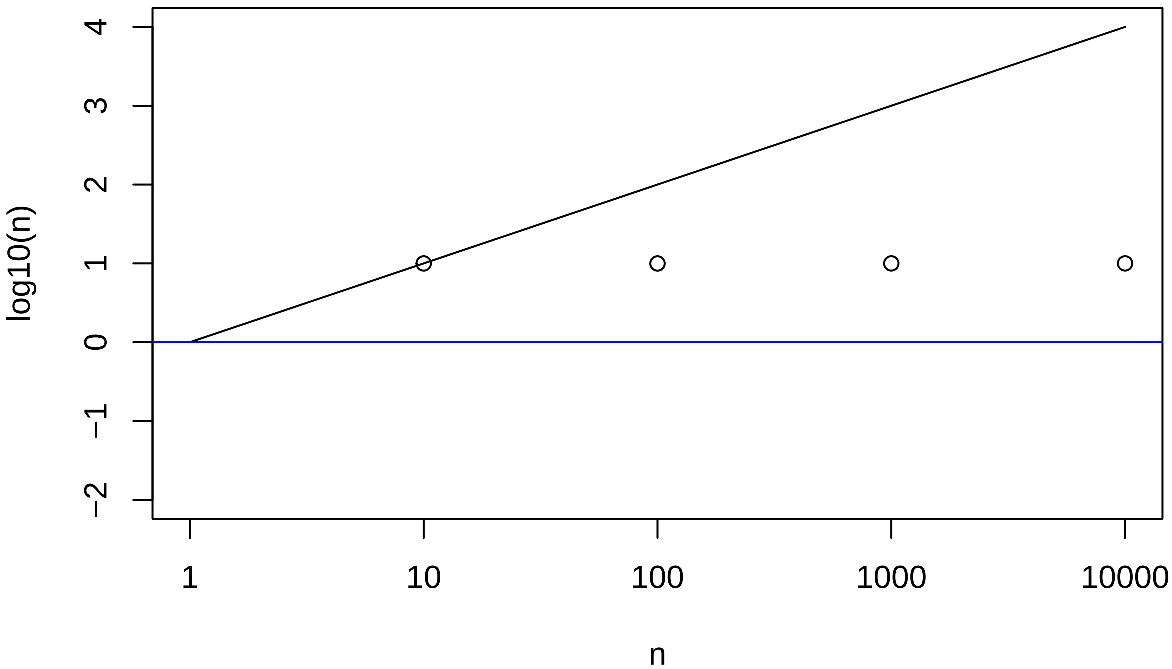

```
hist(sqrt(10000)*(Xbar_10000 - 0)/1, main = "Histogram n=10000", xlab = "root(n)*(Xbar-mu)/sigma")
qqnorm(sqrt(10000)*(Xbar_10000 - 0)/1, main = "QQ Plot n=10000")
```



As $n \to \infty$, the histogram of $\sqrt{n}(\bar{X}_n^{(i)} - \mu)/\sigma$ begins to more closely resemble a random sample generated by a standard Normal distribution. The QQ plot begins to more closely resemble the $x = y$ line, suggesting the data follows a standard Normal distribution. This illustrates the Central Limit Theorem, where $\sqrt{n}(\bar{X}_n^{(i)} - \mu)/\sigma \xrightarrow{D} N(0, 1)$.

3.d

```
conv_prob <- function(X, e) {
  for (i in 1:10000) {
    if(abs(X[i] - rnorm(1)) > e) {X[i] = 1}
    else {X[i] = 0}
  }
return(mean(X))
}

curve(log10(x), from=1, to=10000, ylim=c(-2,4), log="x", xlab = "n", ylab = "log10(n)")
abline(h = 0, col="blue")
points(10, conv_prob(sqrt(10)*(Xbar_10 - 0)/1, 0.001))
points(100, conv_prob(sqrt(100)*(Xbar_100 - 0)/1, 0.001))
points(1000, conv_prob(sqrt(1000)*(Xbar_1000 - 0)/1, 0.001))
points(10000, conv_prob(sqrt(10000)*(Xbar_10000 - 0)/1, 0.001))
```

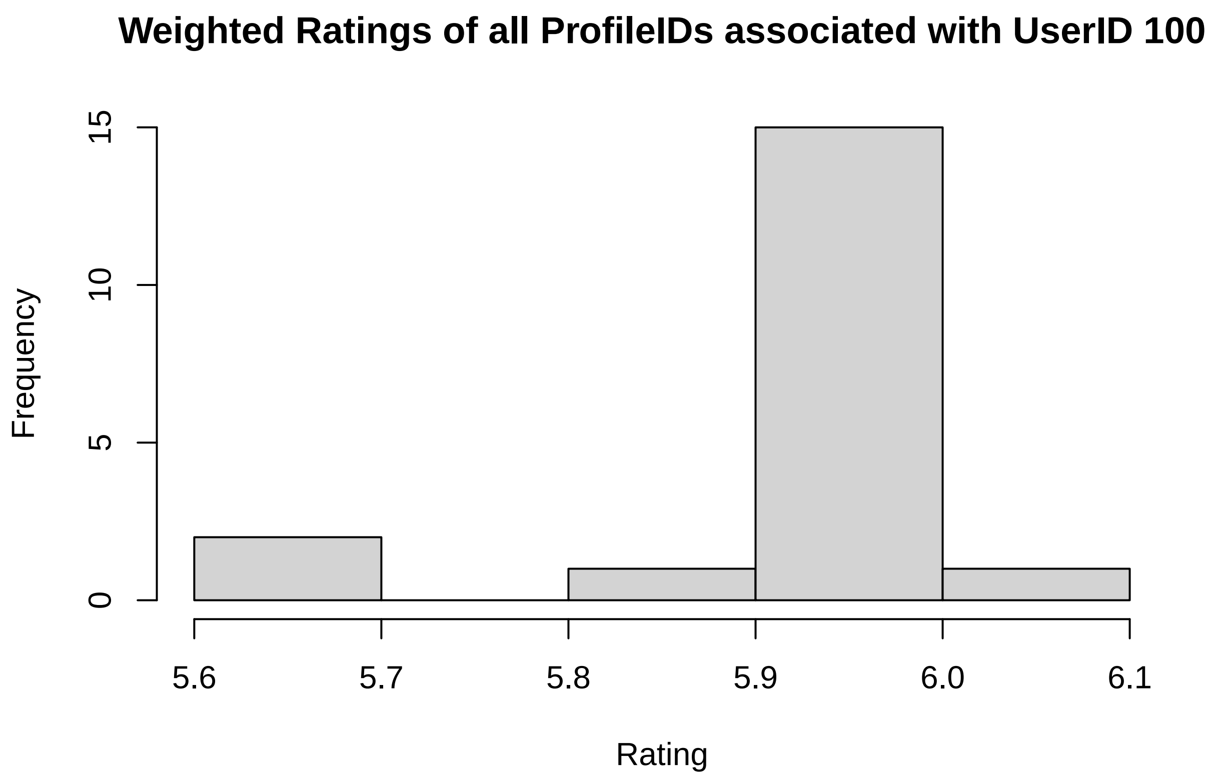


The plot shows that $\lim_{n\to\infty} P(|\sqrt{n}(\bar{X}_n^{(i)} - \mu)/\sigma - Y_i| > \epsilon) = 1$ for $\epsilon = 0.001\ \forall i$, which implies that $\sqrt{n}(\bar{X}_n^{(i)} - \mu)/\sigma$ does not converge in probability to Y_i.

Question 4

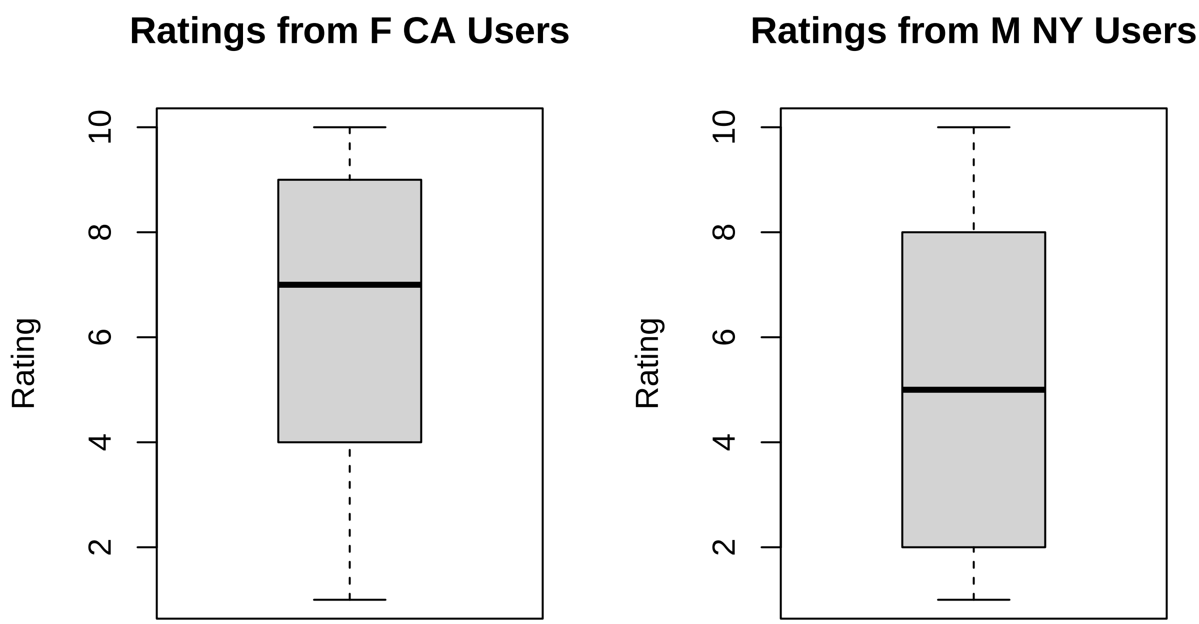
4a

```
X <- read.table("/Users/vladislavtrukhin/Downloads/datasets_all/ratings.dat",
                sep= ",")
names(X) <- c("UserID", "ProfileID", "Rating")

#Function to calculate weighted rank
weighted.rank <- function(ProfileID) {
  R <- mean(X[which(X$ProfileID == ProfileID), 'Rating'])
  v <- nrow(X[which(X$ProfileID == ProfileID), ])
  m <- 4182
  C <- mean(X[, 3])
  return ((v/(v+m))*R + (m/(v+m))*C)
}

#Histogram of weighted ranks of all ProfileIDs associated with UserID 100
results <- c()
for (i in X[which(X$UserID == 100), 'ProfileID']) {
  results <- c(results, weighted.rank(i))
}
hist(results, main = "Weighted Ratings of all ProfileIDs associated with UserID 100", xlab="Rating")
```



4b

```
par(mfrow=c(1,2))
load("/Users/vladislavtrukhin/Downloads/datasets_all/users.Rdata")

#Female CA Users
ca <- grep("^(?!.*CAR).*CA", User$State, perl=TRUE, ignore.case = TRUE)
ca_f <- ca[which(User$Gender[ca] == "F")]

#Male NY Users
ny <- grep(".*ny|.*york", User$State, perl=TRUE, ignore.case = TRUE)
ny_m <- ny[which(User$Gender[ny] == "M")]

#Box-plot of ratings given out by female CA users
ca_f_ratings <- c()
for (i in ca_f) {
  ca_f_ratings <- c(ca_f_ratings, X[which(X$UserID == i), 'Rating'])
}
boxplot(ca_f_ratings, main="Ratings from F CA Users", ylab="Rating")

#Box-plot of ratings given out by male NY users
ny_m_ratings <- c()
for (i in ny_m) {
  ny_m_ratings <- c(ny_m_ratings, X[which(X$UserID == i), 'Rating'])
}
boxplot(ny_m_ratings, main="Ratings from M NY Users", ylab="Rating")
```



4c

```
library(biganalytics)
```

```
## Loading required package: bigmemory

## Loading required package: foreach

## Loading required package: biglm

## Loading required package: DBI
```

```
#Given
N=3000000
Nu=135359
Np=220970
user.rat=rep(0,Nu)
user.num=rep(0,Nu)
profile.rat=rep(0,Np)
profile.num=rep(0,Np)
for (i in 1:N){
    user.rat[X[i,'UserID']]=user.rat[X[i,'UserID']]+X[i,'Rating']
    user.num[X[i,'UserID']]=user.num[X[i,'UserID']]+1
    profile.rat[X[i,'ProfileID']]=profile.rat[X[i,'ProfileID']]+X[i,'Rating']
    profile.num[X[i,'ProfileID']]=profile.num[X[i,'ProfileID']]+1
}
user.ave=user.rat/user.num
profile.ave=profile.rat/profile.num
X1=big.matrix(nrow=nrow(X), ncol=ncol(X), type= "double",
              dimnames=list(NULL, c('UsrAveRat','PrfAveRat','Rat')))
X1[,'Rat']=X[,'Rating']
X1[,'UsrAveRat']=user.ave[X[,'UserID']]
X1[,'PrfAveRat']=profile.ave[X[,'ProfileID']]

#Normal Method
fit <- lm(Rat ~ UsrAveRat + PrfAveRat, as.data.frame(as.matrix(X1)))
```

```r
#Coefficients and R2
summary(fit)$coefficients[1:3]
```

```
## [1] -2.1270532  0.4459886  0.9121571
```

```r
summary(fit)$r.squared
```

```
## [1] 0.6294795
```

```r
#Sub-sampling Method (Sample 100 times of 1000 sample size)
coeff <- c()
for (i in 1:100) {
  n1 <- as.integer(trunc(runif(1, 0, N-1000)))
  n2 <- n1 + 1000
  fit <- lm(Rat ~ UsrAveRat + PrfAveRat, as.data.frame(as.matrix(X1[n1:n2,])))
  coeff <- rbind(coeff, summary(fit)$coefficients[1:3])
}

#Coefficients and R2
colMeans(coeff)
```

```
## [1] -2.1556274  0.4454397  0.9170613
```

```r
preds <- as.matrix(cbind(1, X1[, 1:2]))%*%as.matrix(colMeans(coeff))
actual <- as.matrix(X1[,3])
rss <- sum((preds - actual) ^ 2)
tss <- sum((actual - mean(actual)) ^ 2)
rsq <- 1 - rss/tss
rsq
```

```
## [1] 0.629465
```

The sub-sampling method for big data linear regression obtained similar coefficients values as the normal method while being less computationally expensive.