STAD80: Homework #1

Yulun Wu

Due: 2023-02-02

Contents

Question 1 (30 Points) Conceptual Challenges 1
Question 2 (20 Points) Maximum Likelihood Estimator (MLE) and Asymptotic Normality 1
Question 3 (20 Points) Law of Large Numbers and Central Limit Theorem 4
Question 4 (20 Points) Basic R Programming for Big Data 10

Question 1 (30 Points) Conceptual Challenges

(1) Let $\{X_i\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p_\theta(x)$ be n random samples (Let X be their population variable). We denote their realizations (or outcomes) to be $\{x_i\}_{i=1}^n$. Select all the WRONG statements.
(2) Unbiasedness and Consistency. Select all the wrong statement:
(3) Law of Large Numbers (LLN) and Central Limit Theorem (CLT). Select all the WRONG statements:
(4) Linear Regression and Ordinary Least Squares (OLS). Select all the WRONG statements:
(5) Assuming the true distribution of the data follows a linear model $Y = \beta_0 + \beta_1 X + \epsilon$. We fit an ordinary least squares regression using this true model on the data, as the number of data points goes to infinity, your estimator will have
(6) Select all the WRONG statements:

Answer:

(1) B,C,E
(2) A,B,C,E
(3) C
(4) A
(5) A,F
(6) A,E

Question 2 (20 Points) Maximum Likelihood Estimator (MLE) and Asymptotic Normality

Answer:

Part 1:

Part 2: (a)

$$\prod_{i=1}^{n}(\theta-1)x_i^{-\theta}1(x_i \geq 1) = n\log(\theta-1) - \theta\sum_{i=1}^{n}\log(x_i) + c$$

We can ignore c part because it will disappear after taking derivative anyway.

$$\begin{aligned}
\frac{\partial}{\partial\theta}\prod_{i=1}^{n}(\theta-1)x_i^{-\theta}1(x_i \geq 1) &= \frac{n}{\theta-1} - \sum_{i=1}^{n}\log(x_i) = 0 \\
&\Leftrightarrow \frac{n}{\sum_{i=1}^{n}\log(x_i)} = \theta - 1 \\
&\Leftrightarrow \widehat{\theta}_n = \frac{n}{\sum_{i=1}^{n}\log(x_i)} + 1
\end{aligned} \tag{1}$$

The MLE $\widehat{\theta}_n$ is $\frac{n}{\sum_{i=1}^{n}\log(x_i)} + 1$.

(b)

$$\log(p_\theta(X)) = \log(\theta-1) - \theta\log(X)$$

$$\frac{\partial^2}{\partial\theta^2}\log(p_\theta(X)) = \frac{\partial}{\partial\theta}\frac{1}{\theta-1} - \log(X) = -\frac{1}{(\theta-1)^2}$$

$$\begin{aligned}
I(\theta) &= \mathbb{E}_\theta(-\frac{\partial^2}{\partial\theta^2}\log(p_\theta(X))) \\
&= \int_1^\infty \frac{1}{(\theta-1)^2}(\theta-1)x^{-\theta}dx \\
&= \int_1^\infty \frac{x^{-\theta}}{\theta-1}dx \\
&= \frac{1}{\theta-1}\int_1^\infty x^{-\theta}dx \\
&= \frac{1}{\theta-1}\frac{x^{1-\theta}}{1-\theta}|_1^\infty \\
&= -\frac{x^{1-\theta}}{(\theta-1)^2}|_1^\infty \\
&= -0 + \frac{1}{(\theta-1)^2}, given \theta > 1 \\
&= \frac{1}{(\theta-1)^2}
\end{aligned} \tag{2}$$

$$\begin{aligned}
Var(\sqrt{n}(\widehat{\theta}_n - \theta)) &= \frac{1}{I(\theta)} \\
&= (\frac{1}{(\theta-1)^2})^{-1} \\
&= (\theta-1)^2
\end{aligned} \tag{3}$$

(c)

$$\begin{aligned}
I(\widehat{\theta}_n) &= \frac{1}{(\widehat{\theta}_n - 1)^2} \\
&= \frac{1}{(\frac{n}{\sum_{i=1}^{n}\log(x_i)} + 1 - 1)^2} \\
&= \frac{(\sum_{i=1}^{n}\log(x_i))^2}{n^2}
\end{aligned} \tag{4}$$

```
alpha_q2 = 0.05
z_q2 = qnorm(1-alpha_q2/2,0,1)
z_q2
```

```
## [1] 1.959964
```

$$
\begin{aligned}
&\widehat{\theta}_n \pm \frac{z_{\alpha/2}}{\sqrt{nI(\widehat{\theta}_n)}} \\
&\quad \Leftrightarrow \frac{n}{\sum_{i=1}^{n}\log(x_i)} + 1 \pm \frac{1.959964}{\sqrt{\frac{(\sum_{i=1}^{n}\log(x_i))^2}{n}}} \\
&\quad \Leftrightarrow \frac{n}{\sum_{i=1}^{n}\log(x_i)} + 1 \pm \frac{1.959964}{\frac{\sum_{i=1}^{n}\log(x_i)}{\sqrt{n}}} \\
&\quad \Leftrightarrow \frac{n}{\sum_{i=1}^{n}\log(x_i)} + 1 \pm \frac{1.959964\sqrt{n}}{\sum_{i=1}^{n}\log(x_i)} \\
&C_n = [\frac{n - 1.959964\sqrt{n}}{\sum_{i=1}^{n}\log(x_i)} + 1, \frac{n + 1.959964\sqrt{n}}{\sum_{i=1}^{n}\log(x_i)} + 1]
\end{aligned} \tag{5}
$$

(d)

$$
\begin{aligned}
\int_1^x (\theta-1)t^{-\theta}dt &= (\theta-1)\int_1^x t^{-\theta}dt \\
&= (\theta-1)\frac{t^{1-\theta}}{1-\theta}|_1^x \\
&= -t^{1-\theta}|_1^x \\
&= -x^{1-\theta} + 1 \\
&= (1+x)^{1-\theta}
\end{aligned} \tag{6}
$$

CDF is

$$(1+x)^{1-\theta}$$

```
N_q2 = 10000
n_q2=100
theta_q2=2
count_contain_theta = 0
CIcalc_q2 = function(x,z,n){
  CI_low = (n-z*sqrt(n))/(sum(log(x)))+1
  CI_up = (n+z*sqrt(n))/(sum(log(x)))+1
  CI = cbind(CI_low,CI_up)
  #print(x)
  return(CI)
}
q2cdf = function(x,theta){
  return((1-x)^(1/(1-theta))) # Is this correct
}
for (i in 1:N_q2){
  U_q2 = runif(n_q2,min=0,max=1)
```

```
  X_q2 = q2cdf(U_q2,theta_q2)
  CI_q2 = CIcalc_q2(X_q2,z_q2,n_q2)
  if ((CI_q2[1]<=theta_q2)&(CI_q2[2]>=theta_q2)){
    count_contain_theta = count_contain_theta+1
  }
}
count_contain_theta/N_q2
```

```
## [1] 0.9522
```

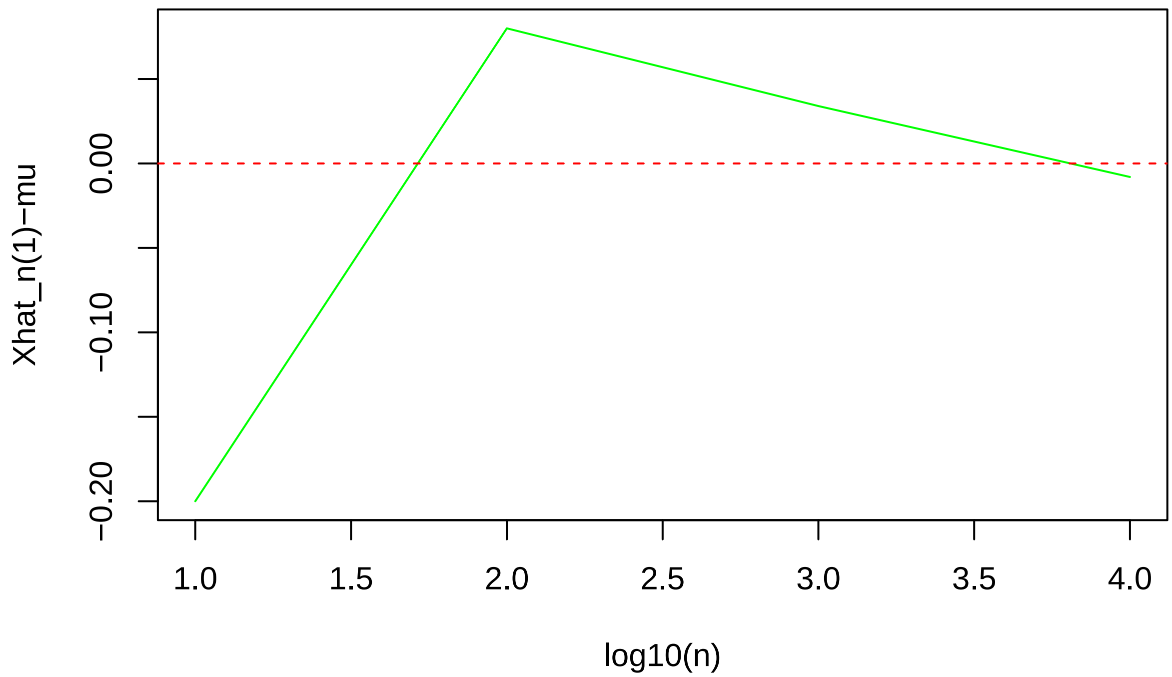
Hence, CI will cover the true θ with probability around 95%.

Question 3 (20 Points) Law of Large Numbers and Central Limit Theorem

Answer:

```
generate_xhat_q3 = function(n){
  x = runif(n)
  x=replace(x,x<0.5,-1)
  x=replace(x,x>=0.5,1)
  xhat = mean(x)
  varx = var(x)
  return(list(xhat,varx))
}
N_q3 = 10000
n_q3 = c(10,100,1000,10000)
# row 1 is mean, row 2 is variance
meanVar_10 <- replicate(N_q3,generate_xhat_q3(n_q3[1]))
meanVar_100 <- replicate(N_q3,generate_xhat_q3(n_q3[2]))
meanVar_1000 <- replicate(N_q3,generate_xhat_q3(n_q3[3]))
meanVar_10000 <- replicate(N_q3,generate_xhat_q3(n_q3[4]))
```

```
mu_q3 = 0.5*(1-1)
sigma_q3 = ((1-mu_q3)^2)*0.5+((-1-mu_q3)^2)*0.5
y_q3 = c(meanVar_10[1,1],meanVar_100[1,1],meanVar_1000[1,1],meanVar_10000[1,1])
y_q3 = unlist(y_q3)-mu_q3
#curve(log10(x),from=1,to=10000,log="x",ylab="log10(n)",xlab="n",col="green",ylim=c(-3,5))
plot(log10(n_q3),y_q3,ylab="Xhat_n(1)-mu",xlab="log10(n)",col="green",type="l")
abline(h = 0, lty = 2,col="red")
```




```
#points(n_q3,y_q3)
```

This plot shows that as

$$n \to \infty$$

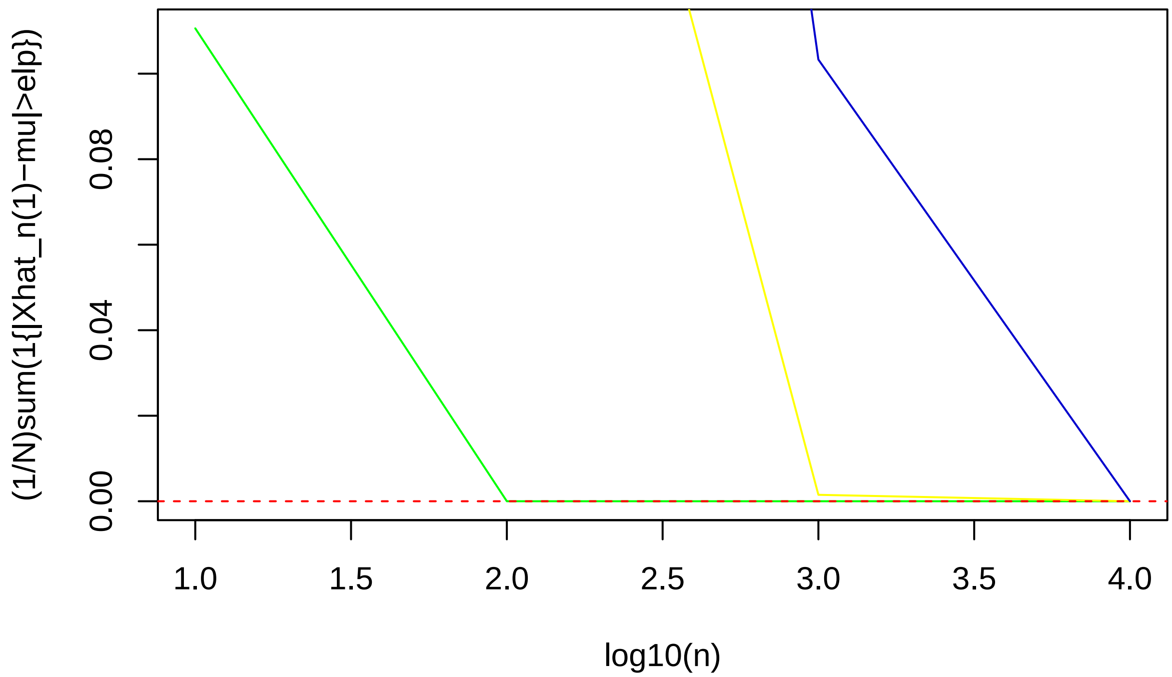
,

$$\overline{X}_n^{(1)} - \mu \to 0$$

.

(b)

```
elp_q3 = c(0.5,0.1,0.05)
sum_ind1 = c((1/N_q3)*sum(abs(unlist(meanVar_10[1,])-mu_q3)>elp_q3[1]),(1/N_q3)*sum(abs(unlist(meanVar_
sum_ind2 = c((1/N_q3)*sum(abs(unlist(meanVar_10[1,])-mu_q3)>elp_q3[2]),(1/N_q3)*sum(abs(unlist(meanVar_
sum_ind3 = c((1/N_q3)*sum(abs(unlist(meanVar_10[1,])-mu_q3)>elp_q3[3]),(1/N_q3)*sum(abs(unlist(meanVar_
#curve(log10(x),from=1,to=10000,log="x",ylab="log10(n)",xlab="n",col="green",ylim=c(-3,5))
plot(log10(n_q3),sum_ind1,ylab="(1/N)sum(1{|Xhat_n(1)-mu|>elp})",xlab="log10(n)",col="green",type="l")
lines(sum_ind2,col="yellow")
lines(sum_ind3,col="blue3")
abline(h = 0, lty = 2,col="red")
```




```
#points(n_q3,sum_ind1)
#points(n_q3,sum_ind2,col="yellow")
#points(n_q3,sum_ind3,col="blue3")
```

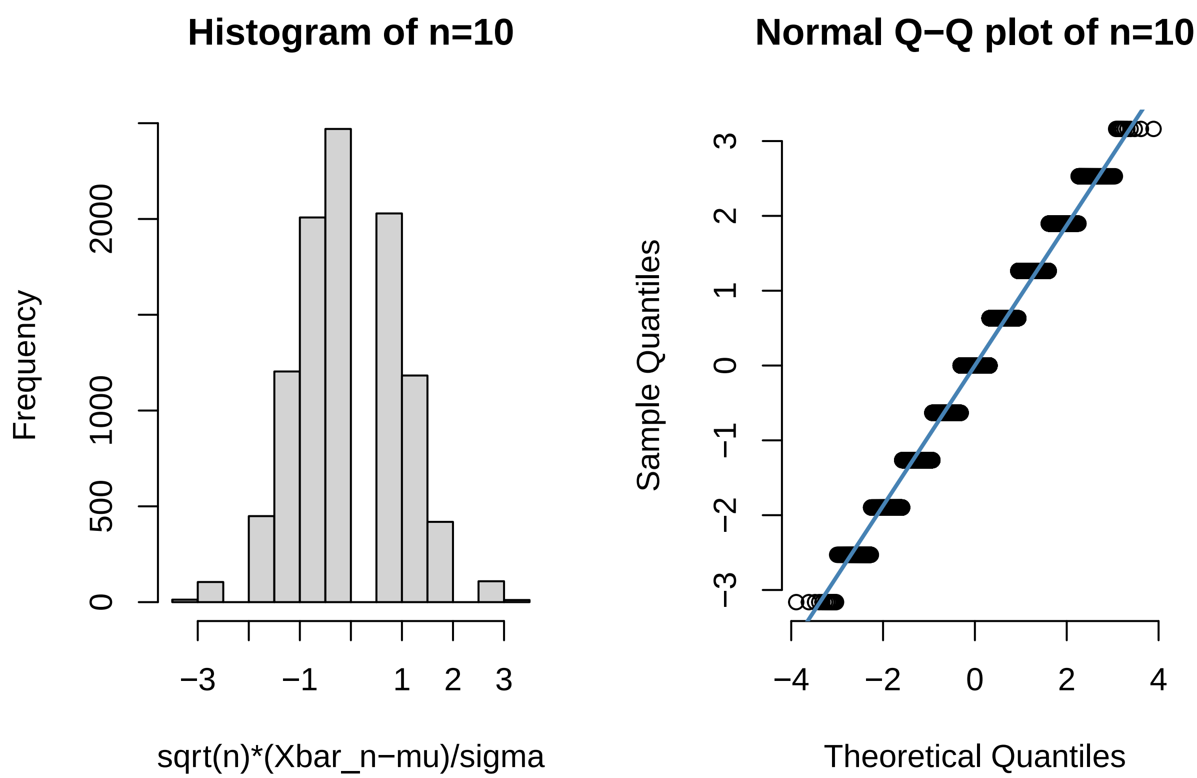
This plot shows that

$$\forall \epsilon, \forall i \in \{1, ..., N\} \lim_{n\to\infty} P(|\overline{X}_n^{(i)} - \mu| > \epsilon) = 0$$

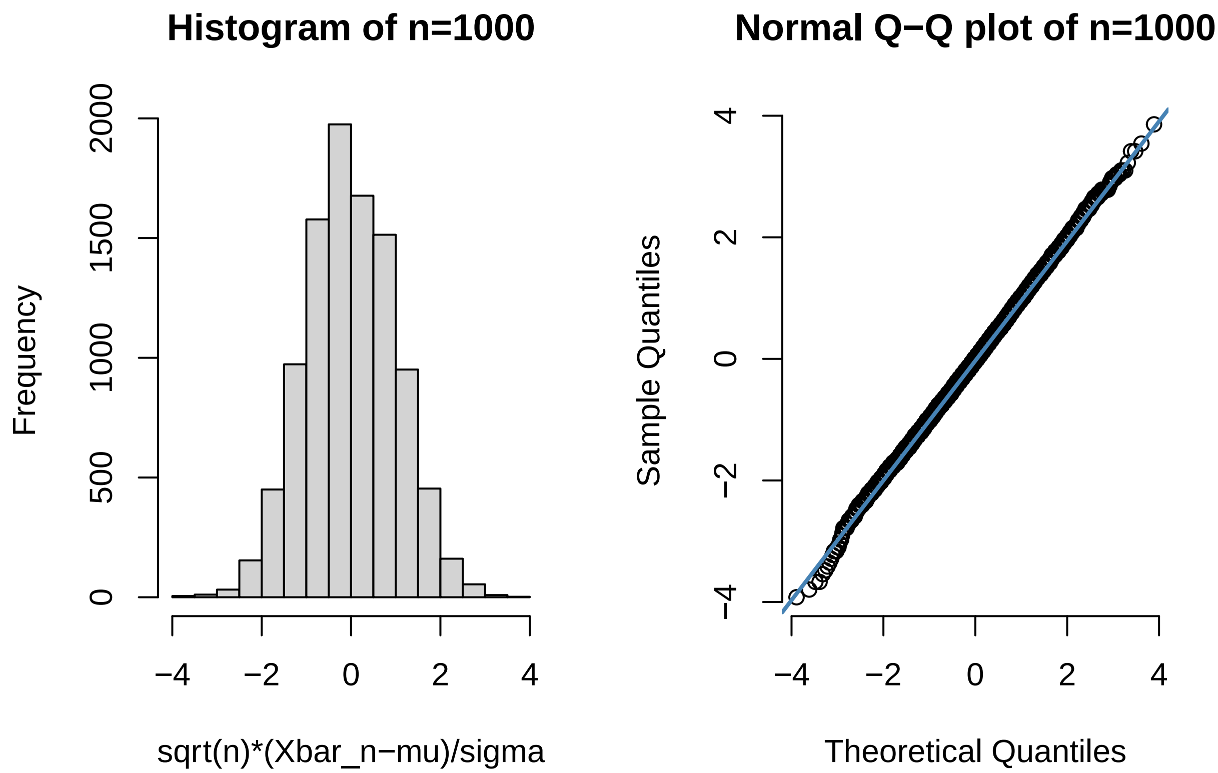
which illustrates the law of large numbers.

(c)

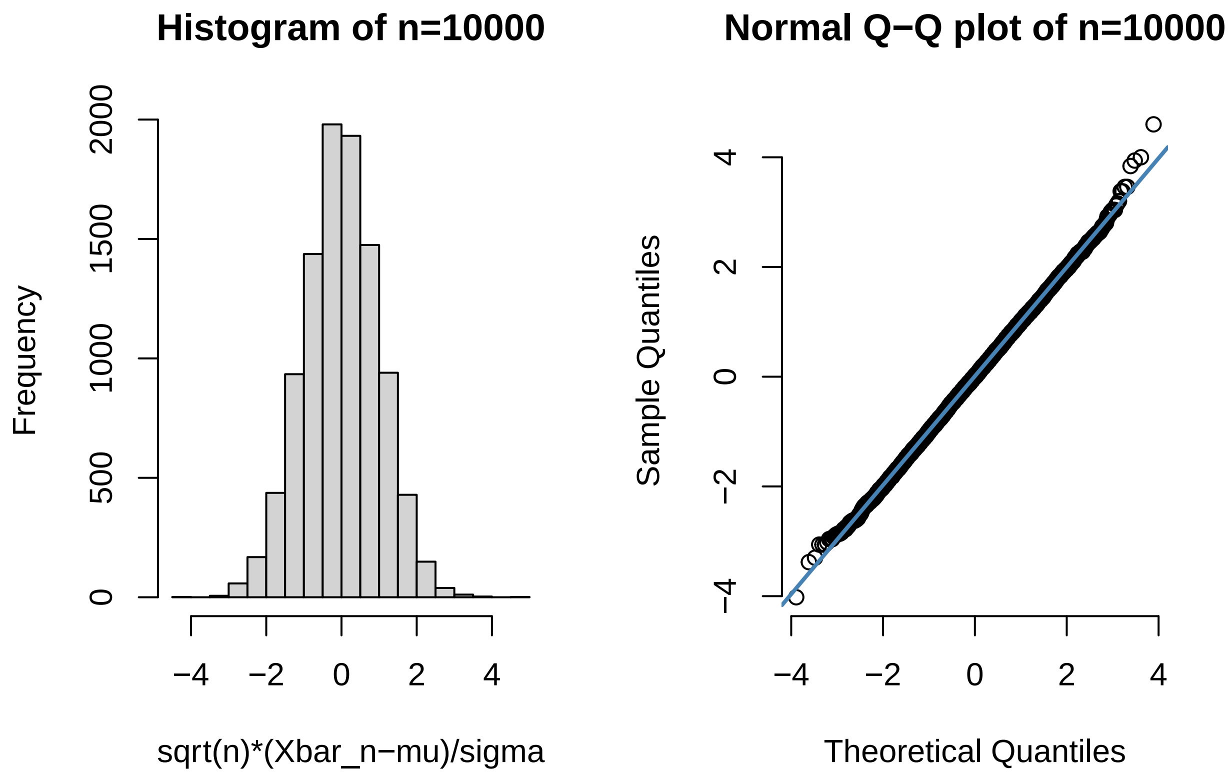
```
par(mfrow=c(1,2))
CLT1_q3 = sqrt(n_q3[1])*(unlist(meanVar_10[1,])-mu_q3)/sqrt(sigma_q3)
hist(CLT1_q3,main="Histogram of n=10",xlab="sqrt(n)*(Xbar_n-mu)/sigma")
qqnorm(CLT1_q3, pch = 1, frame = FALSE,main="Normal Q-Q plot of n=10")
qqline(CLT1_q3, col = "steelblue", lwd = 2)
```




```
par(mfrow=c(1,2))
CLT2_q3 = sqrt(n_q3[3])*(unlist(meanVar_1000[1,])-mu_q3)/sqrt(sigma_q3)
hist(CLT2_q3,main="Histogram of n=1000",xlab="sqrt(n)*(Xbar_n-mu)/sigma")
qqnorm(CLT2_q3, pch = 1, frame = FALSE,main="Normal Q-Q plot of n=1000")
qqline(CLT2_q3, col = "steelblue", lwd = 2)
```




```
par(mfrow=c(1,2))
CLT3_q3 = sqrt(n_q3[4])*(unlist(meanVar_10000[1,])-mu_q3)/sqrt(sigma_q3)
hist(CLT3_q3,main="Histogram of n=10000",xlab="sqrt(n)*(Xbar_n-mu)/sigma")
qqnorm(CLT3_q3, pch = 1, frame = FALSE,main="Normal Q-Q plot of n=10000")
qqline(CLT3_q3, col = "steelblue", lwd = 2)
```



These plots shows that as

$$n \to \infty$$

, the histograms of

$$\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma$$

are more look like the histograms of standard Normal Distribution which has $\mu = 0$ and $\sigma^2 = 1$ and the Normal Q-Q plots are more close to the line of $y = x$, which means as

$$n \to \infty$$

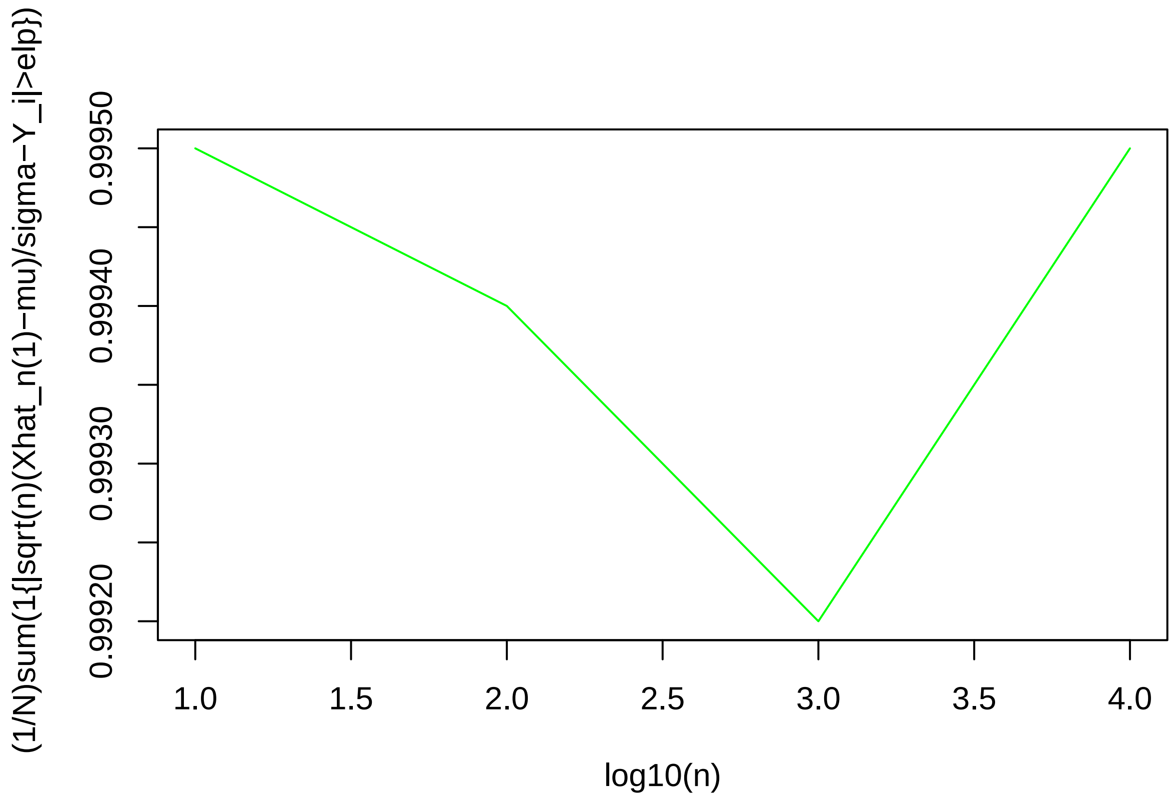
,

$$\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma \xrightarrow{D} N(0, 1)$$

. This illustrates the Central Limit Theorem.

(d)

```
Y_q3 = rnorm(N_q3,0,1)
point1_q3 = (1/N_q3)*sum(abs(sqrt(n_q3[1])*(unlist(meanVar_10[1,])-mu_q3)/sqrt(sigma_q3)-Y_q3)>0.001)
point2_q3 = (1/N_q3)*sum(abs(sqrt(n_q3[2])*(unlist(meanVar_100[1,])-mu_q3)/sqrt(sigma_q3)-Y_q3)>0.001)
point3_q3 = (1/N_q3)*sum(abs(sqrt(n_q3[3])*(unlist(meanVar_1000[1,])-mu_q3)/sqrt(sigma_q3)-Y_q3)>0.001)
point4_q3 = (1/N_q3)*sum(abs(sqrt(n_q3[4])*(unlist(meanVar_10000[1,])-mu_q3)/sqrt(sigma_q3)-Y_q3)>0.001
#curve(log10(x),from=1,to=10000,log="x",ylab="log10(n)",xlab="n",col="green",ylim=c(-2,4))
plot(log10(n_q3),c(point1_q3,point2_q3,point3_q3,point4_q3),ylab=" (1/N)sum(1{|sqrt(n)(Xhat_n(1)-mu)/si
abline(h = 0, lty = 2,col="red")
```




```
#points(n_q3,c(point1_q3,point2_q3,point3_q3,point4_q3),col="blue3")
```

This plot shows that for

$$\epsilon = 0.001, \forall i \in \{1, ..., N\} \lim_{n\to\infty} P(|\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma| > \epsilon) = 1 \Longrightarrow \sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma \nrightarrow Y_i$$

in probability.

Question 4 (20 Points) Basic R Programming for Big Data

Answer:

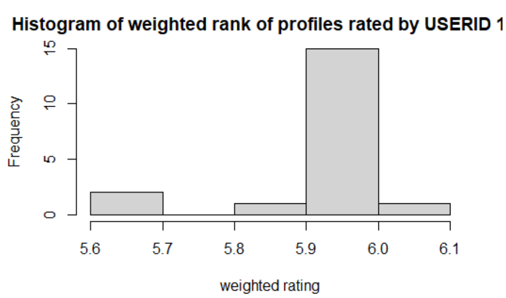
```
library(bigmemory)
X = read.big.matrix("ratings.dat", col.names = c("UserID", "ProfileID", "Rating"))
```

```
N=3000000       # number of rating records
Nu=135359       # maximum of UserID
Np=220970       # maximum of ProfileID
user.rat=rep(0,Nu)       # user.rat[i] denotes the sum of ratings given by user i
user.num=rep(0,Nu)       # user.num[i] denotes the number of ratings given by user i
profile.rat=rep(0,Np)       # profile.rat[i] denotes the sum of ratings given to profile i
profile.num=rep(0,Np)       # user.rat[i] denotes the number of ratings given to profile i
for (i in 1:N){ # In each iteration, we update the four arrays, i.e. user.rat, user.num, profile.rat, p
```

```
    user.rat[X[i,'UserID']]=user.rat[X[i,'UserID']]+X[i,'Rating'] # The matrix X here comes from the fi
    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
    if (i %% 100000==0) print(i/100000)
}
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'
X1[,'Rat']=X[,'Rating']
X1[,'UsrAveRat']=user.ave[X[,'UserID']]
X1[,'PrfAveRat']=profile.ave[X[,'ProfileID']]       # X1 is the new data matrix we will work with in re
```

```
head(X)
C = mean(X[,3])
m = 4182
unique_profile = unique(X[,2]) # unique profile id
```

```
head(X1)
weighted.rank = function(ProfileID){
  ProfileID_index = mwhich(X,2,ProfileID,'eq')
  v = length(ProfileID_index)
  R = X1[ProfileID_index[1],2]
  WR = (v/(v+4182))*R+(4182/(v+4182))*C
  return(WR)
}
index_profile = mwhich(X,1,100,'eq')
prof_id = X[index_profile,2]
WR_100 = lapply(prof_id, weighted.rank)
WR_100=unlist(WR_100)
WR_100
hist(WR_100,xlab="weighted rating",main="Histogram of weighted rank of profiles rated by USERID 100")
```



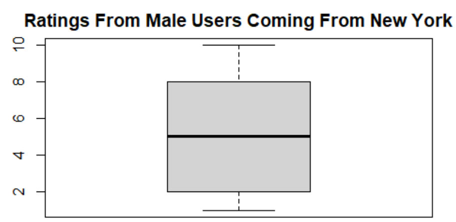
(b)

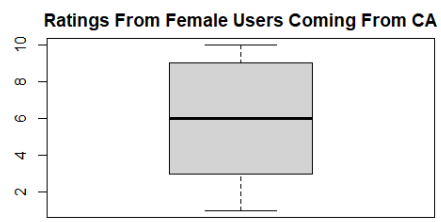
```
head(User)
```

```
# Male and New York
nyuser = grep('.*ny|.*york',User$State,perl = T,ignore.case = T)
mnyuser=nyuser[which(User$Gender[nyuser]=="M")]
mny_rat = c(X[which(X[,1]==mnyuser[1]),3])
```

```
for (i in 2:length(mnyuser)){
  mny_rat = append(mny_rat,X[which(X[,1]==mnyuser[i]),3])
}
boxplot(mny_rat,main="Ratings From Male Users Coming From New York")

# Female and CA
causer = grep('(^(?!.car).*ca)|(.*california)',User$State,perl = T,ignore.case = T)
fcauser=nyuser[which(User$Gender[causer]=="F")]
fca_rat = c(X[which(X[,1]==fcauser[1]),3])
for (i in 2:length(fcauser)){
  fca_rat = append(fca_rat,X[which(X[,1]==fcauser[i]),3])
}
boxplot(fca_rat,main="Ratings From Female Users Coming From CA")
```





(c)

```
library(biganalytics)
# fitting all data to biglm
model1 = biglm.big.matrix(Rat~UsrAveRat +PrfAveRat,X1)
coef(model1)
Ybar=mean(X1[,3])
SST=sum((X1[,3]-Ybar)^2)
Yhat=predict(model1,as.data.frame(X1[,1:3],colnames=c("UsrAveRat","PrfAveRat")))
SSR=sum((Yhat-Ybar)^2)
SSR/SST
```

Coefficients for fitting all data are

$$\{\theta_1, \theta_2, \theta_3\} = \{-2.1271, 0.4460, 0.9122\}$$

, R^2 is 0.6294795.

```
size_sub = 100000
n_sub = 10 # ie: I will train 10 model each with 100000 rows from X1
cof = c()
Rsquared = 0
for (i in 1:n_sub){
  subsample = X1[(i*size_sub):((i+1)*size_sub),]
  submodel = biglm.big.matrix(Rat~UsrAveRat +PrfAveRat,subsample)
  cof=rbind(cof,coef(submodel))
  Ybar=mean(subsample[,3])
  SST=sum((subsample[,3]-Ybar)^2)
```

```
  Yhat=predict(model1,as.data.frame(subsample[,1:3],colnames=c("UsrAveRat","PrfAveRat")))
  SSR=sum((Yhat-Ybar)^2)
  Rsquared = Rsquared+(SSR/SST)
}
cof
colMeans(cof)
Rsquared/n_sub
```

Coefficients for averaging 10 models each with 100000 data are

$$\{\theta_1, \theta_2, \theta_3\} = \{-2.1148911, 0.4442422, 0.9121603\}$$

, R^2 is 0.6295416.