STATISTICS ASSIGNMENT 3 **-AMAN SINGH (BS-2308)**

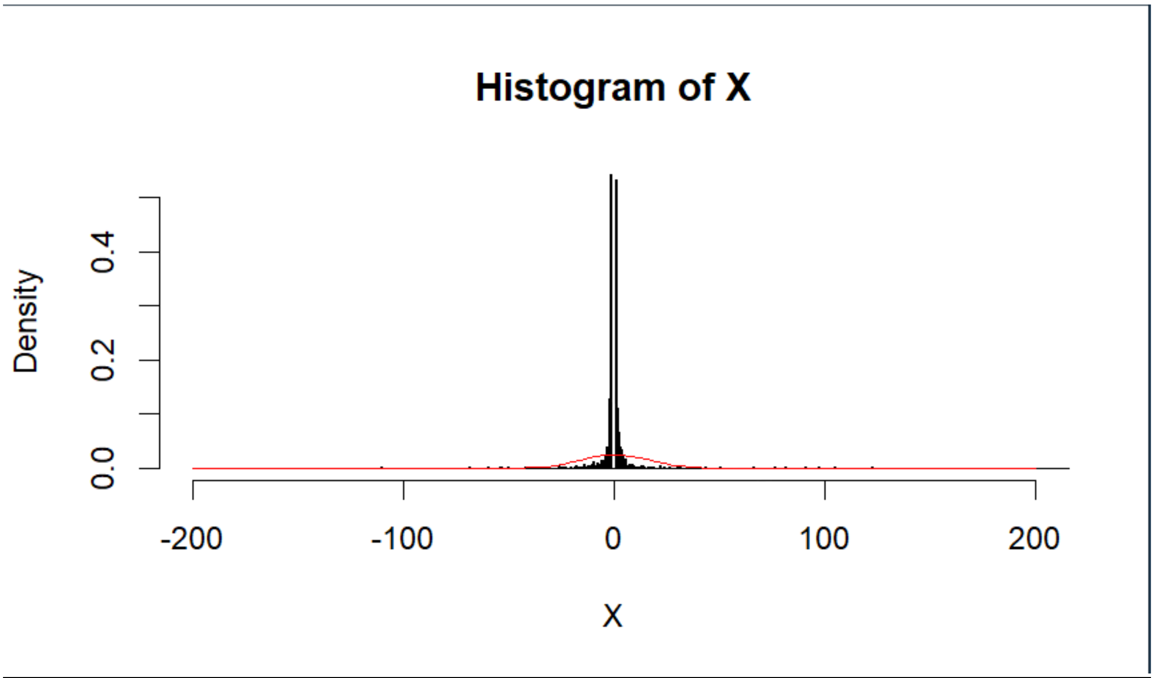
QUESTION 1

```
#We have a random sample of size 101 from a N(μ, 1) population, where μ ∈ R is unknown.
#We want to estimate μ. There are two contending estimators: the sample mean and the
sample median.
#We want to approximate the standard errors of these estimators. Do this using simulation
for μ = 10 and μ = 20.

#for mean = 10
y = numeric(1000) # generating a vector containing 1000 zeros for updating mean
z = numeric(1000) # generating a vector containing 1000 zeros for updating median
for (i in 1:1000){x = rnorm(101,10,1) #generating 101 normal distribution random numbers
y[i]=mean(x) # updating mean of 101 random deviates in y for the current iteration
z[i]=median(x) # updating median of 101 random deviates in y for the current iteration
}
seme1 = sd(y) #average standard error of mean
semd1 = sd(z) #average standard error of median


#for mean = 20
y = numeric(1000) # generating a vector containing 1000 zeros for updating mean
z = numeric(1000)# generating a vector containing 1000 zeros for updating median
for (i in 1:1000){x=rnorm(101,20,1) #generating 101 normal distribution random numbers
y[i]=mean(x) # updating mean of 101 random deviates in y for the current iteration
z[i]=median(x) # updating median of 101 random deviates in y for the current iteration
}
seme2 = sd(y) #average standard error of mean
semd2 = sd(z) #average standard error of median

# COMMENT
# when we compare the standard errors of mean and median for mean = 10 and 20
# we found that usually the error increase as we increase the mean, also standard error of
mean is less than that of median.
```

QUESTION 2

```
#A point, P, is chosen at random on the circumference of the unit circle
#centered at the origin. All points are equally likely. Let (X, 0) be the point
#where the tangent hits the x-axis. Take X = 0 if the P is at (0, -1) or
#(0, 1). Use simulation to form an idea about the distribution of X. Is the
#distribution normal? Answer this question by overlaying the best normal
#PDF on the histogram, and then visually ascertaining the fit
```

```r
p = runif(1000,0,2*pi) #generating 1000 random values between 0 to 2*pi as angles formed
by joining a point and the origin of the unit circle

#tangent at (a,b) will have equation ax+by=1, so for y=0 we have x as 1/a where
(a,b)~(cos(p),sin(p)) so x-intercept of tangent at the point for a given angle is 1/cos(p)

t = numeric(1000) # generating a vector containing 1000 zeros for updating x-intercept
values

# updating x intercept values for each angle according to condition given in the question
for (i in 1:1000){
    if ((p[i] == pi/2) | (p[i] == 3*pi/2)){
        t[i] = 0}
    else{t[i] = 1/cos(p[i])
    }}

hist(t,1000,prob=T,xlim=c(-200,200),main = ‘Histogram of X’, xlab = ‘X’) # plotting the
histogram of x-intercepts
library(MASS) # importing MASS library for using fitdistr
a = fitdistr(t,"normal") # using fitdistr to find mean and standard deviation of the data
for normal distribution
b = a$estimate # extracting values of mean and sd
curve(dnorm(x,b[1],b[2]),add=T,col = 'red') # plotting the normal distribution curve for
the founded mean and sd, trying to fit it in the distribution of x-intercepts


# we can see from the plot of histogram and the curve that the x-intercepts do not fit in
normal distribution for the same mean and standard deviation.
# also our distribution of X have no value between -1 and 1 , because of the range of
1/cos(x) = sec(x) function,
# and it can't take value zero also because of the computer limitation that it our P can't
take value exactly pi/2 and 3*pi/2 for which it could have been zero
```



QUESTION 3

```
#The same set up as above. Find (using simulation) two numbers L and
#U such that X lies between them with 90% probability. The smaller is
#U −L, the happier I would be. Also find (using simulation) the probability
#that X exceeds 5.

k = numeric(1000) #generating 1000 random values for U
z = numeric(1000) #generating 1000 random values for L
for (j in 1:1000){
p = runif(1000,0,2*pi) #generating 1000 random values between 0 to 2*pi as angles formed
by joining a point and the origin of the unit circle

#tangent at (a,b) will have equation ax+by=1, so for y=0 we have x as 1/a where
(a,b)~(cos(p),sin(p)) so x-intercept of tangent at the point for a given angle is 1/cos(p)

t = numeric(1000) # generating a vector containing 1000 zeros for updating x-intercept
values
# updating x intercept values for each angle according to condition given in the question
for (i in 1:1000){
    if ((i==pi/2) | (i==3*pi/2)){
        t[i]=0}
    else{t[i]=1/cos(p[i])
    }}

m = numeric(100) # generating a vector containing 100 zeros for U-L values
for (i in 1:100){q1 = quantile(t,(900+i)/1000) # will give a number such that 900+i% of x-
intercepts are below it
q2 = quantile(t,i/1000) # will give a number such that i% of x-intercepts are below it
m[i]=abs(q2-q1)} # updating U-L
c = which.min(m) # finding index of minimum U-L
k[j] = quantile(t,(900+c)/1000) # will give the min U such that 900+c % of x-intercepts
are below it
z[j] = quantile(t,c/1000) # will give the min L such that c % of x-intercepts are below it
}

qa = mean(k) # mean of U
qb = mean(z) # mean of L
minc = qa-qb # minimum of U-L

temp=numeric(1000) # generating a vector containing 1000 zeroes for storing probability
that x-intercept exceeds 5 at a given iteration
for (i in 1:1000){r= runif(1000,0,2*pi)
l = 1/cos(r)
temp[i]=(sum(l>5))/1000 #probability that x-intercept exceeds 5 in this iteration
}
prob5 = mean(temp) # average probability that x-intercept exceeds 5


#thus we found the U(that is qa) and L(that is qb) such that X lies between them with 90%
probability and U-L(that is minc) being minimum , and it vary everytime we take different
observations
#also we found the probability that X exceeds 5 (prob5) which is approximately 0.06%
```