**Module 4 Home Work**

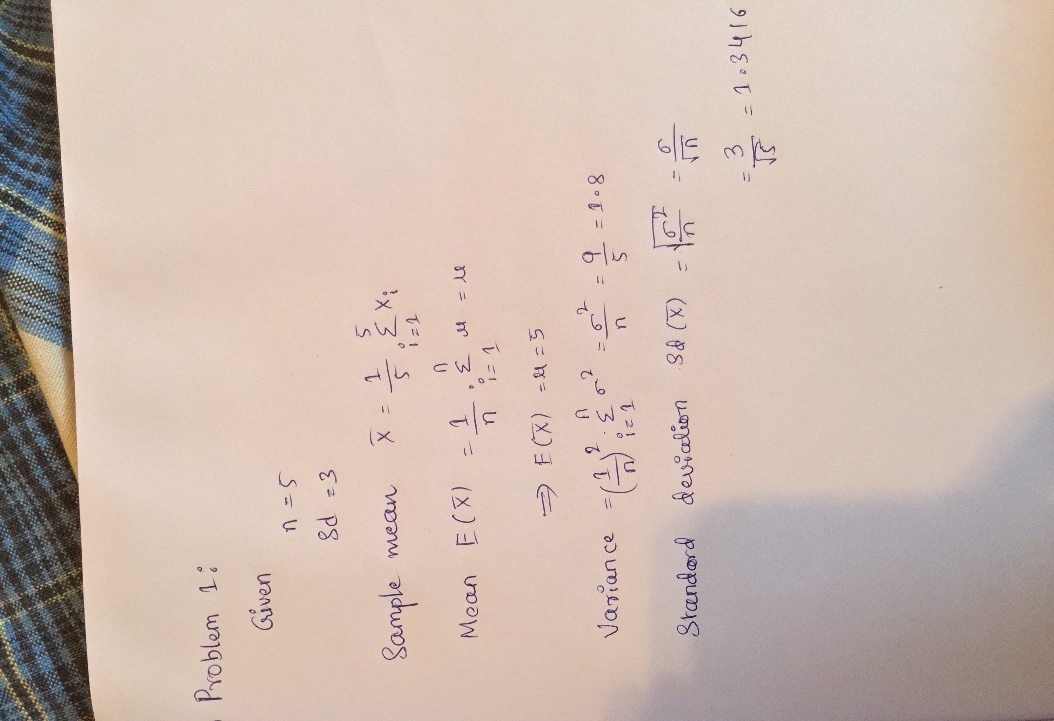
**Problem 1:**

X1………….X5 are independent random samples from a distribution with mean 5 and

standard deviation 3. Complete the following:

a) For the sample mean {X} = frac{1}{5} sum\_{i=1...5}X{i} find its mean E(X) and

standard deviation sd(X).



b) Can you find the approximate value of P(2<{X}<5.1)? If yes, what is your estimate

for P(2<{X}<5.1)?

No

We can not find the approximate value of P(2<{X}<5.1) as the sample size n=5 is not

enough which is very less to converge distribution into the normal distribution.

**Problem 2:**

Suppose that for certain microRNA of size 20 the probability of a purine is binomiallydistributed with probability 0.7. Say there are 100 such microRNAs, each independentof the other. Let Y denote the average number of purine in these microRNAs. Find theprobability that Y is great than 15. Please do a theoretical calculation; do NOT useMonte Carlo simulation to approximate. Show all the steps and formulas in your calculation.

Answer:

We can now calculate E(X) and Var(X) using the formula for the binomial

distribution

E(X) =np

Mean = n\*p = 20\*0.7 = 14

Var(X) = np(1-p).

Variance = np(1-p) = 20\*0.7\*0.3 = 4.2

Therefore, standard deviation is sd(X) = sqrt{Var(X)} = sqrt{4.2} = 2.04939

Each microRNA is then treated as an independent random variable Xi, such that we

get X1..........X100.

Y is the average number of purines in the microRNAs, we can write this formula as

follows: Y = frac{1}{100}sum\_{i=1...100}Xi

X1,.....,X100 are independent random samples drawn from a distribution with mean =

14 and standard deviation = 2.04939. We can use CLT to calculate probability of

events about the sample mean Y, regardless of the exact distribution of X1,.......,X100.

P(Y>15) = 1 – P(Y15)

**R script:**

1-pnorm(15,mean=14,sd=sqrt(4.2)/sqrt(100)) = 5.317746e-07

Therefore P(Y>15) is approximately zero because the value seems to be almost

negligible.

# P(Y >15).

n <- 20

p <- 0.7

MeanY <- n\*p

MeanY

VarY <- n\*p\*(1-p)

VarY

1-pnorm(15, mean=14, sd=sqrt(4.2)/sqrt(100))

**Problem 3:**

Two genes’ expression values follow a bivariate normal distribution. Let X and Ydenote their expression values respectively. Also assume that X has mean 9 andvariance 3; Y has mean 10 and variance 5; and the covariance between X and Y is 2.In a trial, 50 independent measurements of the expression values of the two genesare collected, and denoted as (X1,Y1) ………(X50,Y50). We wish to find theprobability P (X+0.5<Y), that is, the probability that the sample mean for the secondgene exceeds the sample mean of the first gene by more than 0.5.Conduct a Monte Carlo simulation to approximate this probability, providing a 95%confidence interval for your estimation. Submit your R script for the Monte Carlosimulation, and a brief summary of the actual simulation results

Extra Bonus: Provide a theoretical calculation of this probability. While the formulahas not been given in the course lecture, it can be calculated from a bivariate normaldistribution. You do not have to do this theoretical calculation if have no idea. Youwill get extra bonus points for doing it correctly

**Answer:**

Using sigma (1)^2 = 3, sigma (2)^2 = 5, and sigma (12) = Cov(X,Y) = 2 we derive

the following matrix: [3 2 2 5]

The mean of this vector is equal to the probability that we were successful and thatthe sample mean of the second gene (denoted by Y) exceeded the sample mean of thefirst gene (denoted by X) by more than 0.5. We then run 100 computations of the probability P(X+ 0.5 < Y) and store our computed probabilities of success. We may then calculate a confidence interval.

install.packages ("mvtnorm")

require(mvtnorm)

test <- function(niter=100){

mean <- NULL

sigma <- matrix(c(3, 2, 2, 5), nrow=2)

meanx <- 9

meany <- 10

results <- NULL

for (i in 1:niter){

data <- rmvnorm(50, mean=c(9,10), sigma=matrix(c(3,2,2,5), nrow=2))

new\_mean<- apply(data,2,mean)

mean1 <- rbind(mean, new\_mean)

if (new\_mean[1]+0.5 < new\_mean[2])

results <- c(results, 1)

else

results <- c(results, 0)

}

mean(results)

}

print(paste("P(X+0.5<Y)",test(),sep="="))

trials <- NULL

num\_trials <- 100

for(i in (1:num\_trials)){

trials <- c(trials, test())

}

CI <- function(x, ntrials) mean(x) + c(-1.96,1.96) \* sqrt(var(x)/ntrials)

print("95% Confidence Interval =", mean(trails))

print(CI(trials, num\_trials))

[1] "P(X+0.5<Y)=0.94"

[1] 0.9606757 0.9679243

**Problem 4:**

Assume there are three independent random variables X1 ~ chisq(df=10), X2 ~Gamma(alpha =1, Beta = 2), X3 ~ t-distribution with m=3 degrees of freedom. Define a new random variable Y as Y=sqrt(X1)X2 + 4(X3)^2Use Monte Carlo simulation to find the mean of Y. Submit your R script for the Monte Carlo simulation, and a brief summary of the actual simulation results**.**

Answer:

R Script:

filters<- 10000

X1 <- rchisq(filters, df=10)

X2 <- rgamma(filters, shape=1, scale=2)

X3 <- rt(filters, df=3)

Y <- sqrt(X1)\*X2+4\*(X3^2)

meanY <- mean(Y)

print(paste("Results after",filters,"iterations.",sep=" "))

print(paste("mean(Y)",meanY,sep=" = "))

[1] "Results after 10000 iterations."

[1] "mean(Y) = 17.8668534278341"

>

Summary:

I simulated 10,000 Y’s. The mean and of these 10,000 simulated Y is

17.8668534278341. Hence the

mean of Y is estimated as 17.8668534278341

**Problem 5:**

Complete exercise 10 in Chapter 3 of Applied Statistics for Bioinformatics using R

(page 45 - 46). Submit the plot, and a brief explanation of your observation. The

problem refers to the density function of extreme value distribution in another book.

You do not have to look for the other book, the density function is

f(x) = (exp(-x) \* e ^ (-exp(-x)))

**Answer:**

size <- 1000

nsim <- 1000

mydata <- matrix(rnorm(size \* nsim, mean=0, sd=1), nrow=nsim)

mymaxima <- apply(mydata, 1, max)

n <- size

as <- sqrt(2\*log(n)) - 0.5\*(log(log(n))+log(4\*pi))\*(2\*log(n))^(-1/2)

bs <- (2\*log(n))^(-1/2)

plot\_data <- (mymaxima - as)/bs

hist(plot\_data, freq=F, main="n=1000", breaks=100, xlim=c(-5,5), ylim=c(0,0.5))

f <- function(x) exp(-x)\*exp(-exp(-x))

curve(f(x), col=2, add=T)

curve(dnorm(x,mean=0, sd=1),col=3,add=T)

legend(-4.5,0.5,c("maxima","f(x)","dnorm"),col=c(1,2,3),lty=c(1,1,1),bg="gray90")

