

## Module 4 Homework

Chengbo Gu

### Problem 1. (20 points)

$X_1, \dots, X_5$  are independent random samples from a distribution with mean 5 and standard deviation 3. Complete the following:

(a) For the sample mean  $\bar{X} = \frac{1}{5} \sum_{i=1}^5 X_i$ , find its mean  $E(\bar{X})$  and standard deviation  $sd(\bar{X})$ .

$$E(\bar{X}) = \frac{1}{5} \sum_{i=1}^5 E(X_i) = \frac{1}{5} \times 25 = 5$$

$$Var(\bar{X}) = \left(\frac{1}{5}\right)^2 \sum_{i=1}^5 Var(X_i) = \frac{1}{25} \times 9 \times 5 = \frac{9}{5}$$

$$sd(\bar{X}) = \sqrt{Var(\bar{X})} = \sqrt{\frac{9}{5}} = \frac{3}{\sqrt{5}} \approx 1.3416$$

(b) Can you find the  $P(2 < \bar{X} < 5.1)$  approximately using CLT? If yes, what is your estimate for  $P(2 < \bar{X} < 5.1)$ ? If no, why not?

No, we can't use CLT to find  $P(2 < \bar{X} < 5.1)$  here because the  $n$  here is too small.

Generally, it is OK to use CLT when  $n \geq 30$ .

### Problem 2. (20 points)

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

Let  $X$  denote the number of purine in the 20 microRNAs.

Then we have  $X \sim \text{Bionom}(20, 0.7)$ .

Thus,

$$E(X) = n \times p = 20 \times 0.7 = 14$$

$$Var(X) = n \times p \times (1 - p) = 20 \times 0.7 \times 0.3 = 4.2$$

According to CLT,  $Y \sim N(\text{Mean} = 14, \text{Var} = 4.2 / 100 = 0.042)$

$$P(Y > 15) = 1 - P(Y \leq 15) = 1 - \text{pnorm}(15, \text{mean} = 14, \text{sd} = \text{sqrt}(0.042)) = 5.317746 \times 10^{-7}$$

**R code:**

```
meanX <- 20*0.7
VarX <- 20*0.7*0.3
print(1-pnorm(15, mean=meanX, sd=sqrt(VarX)/10))
[1] 5.317746e-07
```

### Problem 3. (20 points)

Two genes' expression values follow a bivariate normal distribution. Let  $X$  and  $Y$  denote their expression values respectively. Also assume that  $X$  has mean 9 and variance 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 genes are collected, and denoted as  $(X_1, Y_1), \dots, (X_{50}, Y_{50})$ . We wish to find the probability  $P(\bar{X} + 0.5 < \bar{Y})$ , that is, the probability that the sample mean for the second gene 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.

#### One loop solution:

```
require(mvtnorm)
sim <- 100000
blah <- rep(NA, sim)
for (i in 1:sim){
  matrix <- rmvnorm(50, mean=c(9,10), sigma=matrix(c(3,2,2,5), nrow=2))
  meanX <- mean(matrix[,1])
  meanY <- mean(matrix[,2])
  blah[i] <- meanX + 0.5 < meanY
}
p <- mean(blah)

print(p)
[1] 0.96149

print( p + c(-1, 1)*1.96*sqrt(var(blah)/sim))
[1] 0.9602973 0.9626827
```

#### Two loops solution:

```
sim <- 1000
res <- rep(NA, sim)
n <- 100
zeroOnes <- rep(NA, n)
for (i in 1:sim){
  for (j in 1:n){
    matrix <- rmvnorm(50, mean=c(9,10), sigma=matrix(c(3,2,2,5), nrow=2))
    meanX <- mean(matrix[,1])
    meanY <- mean(matrix[,2])
    zeroOnes[j] <- meanX + 0.5 < meanY
  }
  p <- mean(zeroOnes)
  res[i] <- p
}

print( mean(res) )
[1] 0.96146

print( mean(res) + c(-1, 1)*1.96*sqrt(var(res)/sim))
[1] 0.9602833 0.9626367
```

#### Summary:

Even though the answers from one and two loops solutions are similar, I still stick on my idea that two loops solution is statistically meaningful. To get confidence interval, we need lots of observed value of  $P(\bar{X} + 0.5 < \bar{Y})$ . One loop solution treats 0s and 1s as one single  $P(\bar{X} + 0.5 < \bar{Y})$ , which is wrong from my point of view, while there are lots of observed value around 0.96 in the res array in two loops solution.

**Theoretical Calculation:**

$$(X, Y) \sim N(\mu_x = 9, \mu_y = 10, \sigma_x^2 = 3, \sigma_y^2 = 5, \text{cov}(X, Y) = 2)$$

$$\begin{aligned} \text{cov}(\bar{X}, \bar{Y}) &= \text{cov}\left(\frac{1}{n} \sum_{i=1}^n X_i, \frac{1}{n} \sum_{j=1}^n Y_j\right) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \text{cov}(X_i, Y_j) \\ &= \frac{1}{n^2} \sum_{i=1}^n \text{cov}(X_i, Y_i) = \frac{1}{n} \text{cov}(X, Y) \end{aligned}$$

Then we have

$$(\bar{X}, \bar{Y}) \sim N(\mu_x = 9, \mu_y = 10, \sigma_x^2 = 3/50, \sigma_y^2 = 5/50, \text{cov}(\bar{X}, \bar{Y}) = 2/50)$$

⇓

$$(\bar{X}, \bar{Y}) \sim N(\mu_x = 9, \mu_y = 10, \sigma_x^2 = 0.06, \sigma_y^2 = 0.1, \text{cov}(\bar{X}, \bar{Y}) = 0.04)$$

Thus,

$$\bar{Y} - \bar{X} \sim N(\mu_y - \mu_x, \text{Var} = \sigma_y^2 - 2\text{cov}(\bar{X}, \bar{Y}) + \sigma_x^2) = N(\text{mean} = 1, \text{sd} = \text{sqrt}(0.08))$$

$$P(\bar{Y} - \bar{X} > 0.5) = 1 - \text{pnorm}(0.5, \text{mean} = 1, \text{sd} = \text{sqrt}(0.08)) = 0.9614501$$

**R code:**

```
1-pnorm(0.5, 1, sqrt(0.08))
[1] 0.9614501
```

**Problem 4. (20 points)**

Assume there are three independent random variables  $X_1 \sim \text{chisq}(df = 10)$ ,  $X_2 \sim \text{Gamma}(\alpha = 1, \beta = 2)$ ,  $X_3 \sim t\text{-distribution}$  with  $m=3$  degrees of freedom.

Define a new random variable Y as  $Y = \sqrt{X_1}X_2 + 4(X_3)^2$ .

Use 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.

```
x1 <- rchisq(100000, df = 10)
x2 <- rgamma(100000, shape = 1, scale = 2)
x3 <- rt(100000, df = 3)
y <- sqrt(x1)*x2+4*(x3^2)
print(mean(y))
[1] 18.26831
```

**Summary:**

Theoretically,

$$E(Y) = E(\sqrt{X_1}) \times E(X_2) + 4 \times E(X_3^2)$$

$$E(\sqrt{X_1}) = \text{integrate}(\text{function}(x) \text{sqrt}(x) * \text{dchisq}(x, \text{df} = 10), \text{lower} = 0, \text{upper} = \text{Inf})\$value$$

$$E(X_2) = \alpha\beta$$

$$E(X_3^2) = \text{integrate}(\text{function}(x) x^2 * \text{dt}(x, \text{df} = 3), \text{lower} = -\text{Inf}, \text{upper} = \text{Inf})\$value$$

Thus,

```
Esqrtx1 <- integrate(function(x) sqrt(x) * dchisq(x, df = 10), lower = 0, upper = Inf)$value
Ex2 <- 1*2
Esquarex3 <- integrate(function(x) x^2 * dt(x, df = 3), lower = -Inf, upper = Inf)$value
Ey <- Esqrtx1*Ex2 + 4*Esquarex3
print(Ey)
```

```
[1] 18.16866
```

So the theoretical expectation of Y is 18.6866.

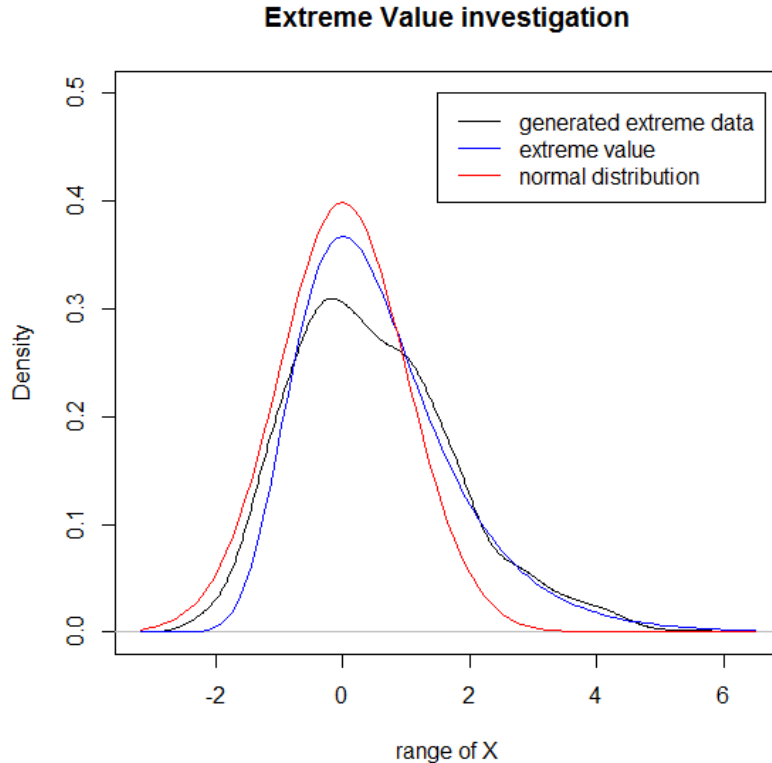
```
print( abs(mean(y)-Ey)/Ey * 100 )  
[1] 0.5484747
```

And the percentage error of this exact Monte Carlo simulation is only 0.55%.

### Problem 5. (20 points)

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.

```
n <- 1000  
a <- function(n) {sqrt(2*log(n)) - 0.5*(log(log(n)) + log(4*pi)) * (2*log(n))^(-1/2)}  
b <- function(n) {(2*log(n))^(-1/2)}  
an <- a(n)  
bn <- b(n)  
res <- rep(NA, n)  
for (i in 1:n){  
  res[i] = (max(rnorm(n, 0, 1))-an)/bn  
}  
plot(density(res), ylim=c(0,0.5), xlab="range of X", main="Extreme Value investigation")  
f <- function(x) {exp(-x)*exp(-exp(-x))}  
curve(f, range(density(res)$x), add=TRUE, col = "blue")  
curve(dnorm, add=TRUE, col = "red")  
legend(1.5,0.5, c("generated extreme data", "extreme value", "normal distribution"), lty =  
"solid", col = c("black","blue","red"))
```



#### Summary:

The extreme value (blue line) fits to the density of generated extreme data (black line) much better than normal distribution (red line).