STP598-Assignment 3

Hao Wang
hwang306

October 25, 2017

1 Question 1

A monte carlo test for four bootstrap confidence intervals. This is a test of normal population. The α level I use here is 0.1.

```
x <- as.data.frame(rnorm(100, mean = 0, sd = 2))
colnames(x) <- c("x")

library(boot)
# bootstrapping with 1000 replications
fc <- function(data, indicies){
    d <- data[indicies,]
    return(mean(d))
}

results <- boot(data = x, fc, R = 1000)
# get 95% confidence intervals
a <- boot.ci(results, conf = 0.9, type = c("norm","basic", "perc", "bca"))
a
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, conf = 0.9, type = c("norm", "basic",
##     "perc", "bca"))
##
## Intervals :
## Level      Normal              Basic
## 90%   (-0.1654,  0.4194 )   (-0.1641,  0.4246 )
##
## Level     Percentile            BCa
## 90%   (-0.1732,  0.4155 )   (-0.1663,  0.4237 )
## Calculations and Intervals on Original Scale
```

```r
#monte carlo test
n <- 100 #sample size
m <- 10000 # number of replications
mean <- numeric(m)
for (i in 1:m) {
  y <- rnorm(n, 0, 2)
  mean[i] <- mean(y)
}

# The normal CI
normal.low <- a$normal[1, 2]
mean(mean < normal.low)
```

```
## [1] 0.1936
```

```r
normal.up <- a$normal[1, 3]
mean(mean > normal.up)
```

```
## [1] 0.0205
```

```r
# The Basic CI
bc.low <- a$basic[1, 4]
mean(mean < bc.low)
```

```
## [1] 0.1946
```

```r
bc.up <- a$basic[1, 5]
mean(mean > bc.up)
```

```
## [1] 0.0188
```

```r
# Percentile CI
pc.low <- a$percent[1, 4]
mean(mean < pc.low)
```

```
## [1] 0.1846
```

```r
pc.up <- a$percent[1, 5]
mean(mean > pc.up)
```

```
## [1] 0.0212
```

```r
# BCa CI

bca.low <- a$bca[1, 4]
mean(mean < bca.low)
```

```
## [1] 0.1923
```

```
bca.up <- a$bca[1, 5]
mean(mean > bca.up)
```

```
## [1] 0.0192
```

2 Question 2

For this question I checked t-test performance of three different distributions. $\alpha = 0.9$. The sample size I pick is 100.

2.1 chi-sq

```
# chi-sq distribution with 1 degrees of freedom
n <- 100 # sample size
alpha <- 0.1
mu0 <- 1
m <- 10000 # replication
p <- numeric(m)

for (i in 1:m) {
  x <- rchisq(n, 1)
  ttest <- t.test(x, alternative = "two.sided", mu = mu0)
  p[i] <- ttest$p.value
}
p.hat <- mean(p < alpha)
p.hat
```

```
## [1] 0.1094
```

2.2 uniform distribution

```
# uniform distribution
n <- 100 # sample size
alpha <- 0.1
mu0 <- 1
m <- 10000 # replication
p <- numeric(m)

for (i in 1:m) {
  x <- runif(n, min = 0, max = 2)
```

```r
  ttest <- t.test(x, alternative = "two.sided", mu = mu0)
  p[i] <- ttest$p.value
}
p.hat <- mean(p < alpha)
p.hat
```

```
## [1] 0.0992
```

2.3 Exponential distribution

```r
#Exponetial distribution
n <- 100 # sample size
alpha <- 0.1
mu0 <- 1
m <- 10000 # replication
p <- numeric(m)

for (i in 1:m) {
  x <- rexp(n, 1)
  ttest <- t.test(x, alternative = "two.sided", mu = mu0)
  p[i] <- ttest$p.value
}
p.hat <- mean(p < alpha)
p.hat
```

```
## [1] 0.1054
```

In all three cases, the monte carlo test error is very close to the nominal value of α.

3 Question 3

1. in bivariate normal, the Spearman correlation is less powerful than Pearson correlation
2. find an alternative where Spearman has better empirical power than Pearson correlation.

3.1 bivariate normal

In this case, we want to show

$$H_0 : corr = 0$$

$$H_a : corr \neq 0$$

codes for comparing powers of tests are attached

```
n <- 30 #sampling size
m <- 10000 #number of replications
alpha <- 0.1
test.p <- test.s <- numeric(m) # storage of test statistics


#generate bivariate normal distribution, assume mu = 0,
# covaraiance matrix is given by
Sigma <- matrix(c(1, -0.5, -0.5, 1),
                nrow = 2, byrow = TRUE) # covariance matrix
mu <- c(0,0) #mean
library(MASS)


# run the loop
for (i in 1:m) {
bvn1 <- mvrnorm(n, mu = mu, Sigma = Sigma ) # from MASS package
bvn1 <- as.data.frame(bvn1)
colnames(bvn1) <- c("x","y")
test.p[i] <- as.integer(cor.test(bvn1$x, bvn1$y, method = "pearson")$p.value <= alpha)
test.s[i] <- as.integer(cor.test(bvn1$x, bvn1$y, method = "spearman")$p.value <= alpha)
}

mean(test.p)
```

```
## [1] 0.8974
```

```
mean(test.s)
```

```
## [1] 0.8557
```

In this case the power of Pearson is higher than Spearman.

3.2 Dependent cases

I generated bivariate expoenetial distribution. In this case The variables y1, y2 have exponential distribution with rates lambda1, lambda2 and they are positively correlated. And I foudn spearman test has better power statistics.

```r
n <- 30 #sampling size
m <- 10000 #number of replications
alpha <- 0.1
test.p <- test.s <- numeric(m) # storage of test statistics

#generate bivariate distribution
# In this case The variables y1, y2 have exponential distribution with rates lambda1, la
lambda1 <- 2
lambda2 <- 3
common <- 1


# run the loop
for (i in 1:m) {
x1 <- rexp(n, rate = lambda1 - common)
x2 <- rexp(n, rate = lambda2 - common)
z <- rexp(n, rate = common)
y1 <- pmin(x1, z)
y2 <- pmin(x2, z)
test.p[i] <- as.integer(cor.test(y1, y2, method = "pearson")$p.value <= alpha)
test.s[i] <- as.integer(cor.test(y1, y2, method = "spearman")$p.value <= alpha)
}

mean(test.p)
```

```
## [1] 0.4112
```

```r
mean(test.s)
```

```
## [1] 0.5595
```

4 Question 4

Monte Carlo integration

$$\theta = \int_0^{0.5} e^{-x} dx$$

let $f(x) = \frac{1}{0.5-0}$ then

$$\theta = (0.5-0) \int_0^{0.5} e^{-x} \frac{1}{0.5-0} dx = 0.5E[e^{-x}]$$

then

$$\hat{\theta} = 0.5Eg(X) = 0.5E(e^{-x})$$

The sample mean variance is

$$Var(\hat{\theta}) = 0.25/m * Var(g(x)) = \frac{0.25}{m} Var(e^{-x})$$

```
# generate unif(0, 0.5)
m <- 10000
theta <- numeric(m)
x <- runif(m, min = 0, max = 0.5)
theta.hat <- mean(exp(-x))*0.5
theta.hat
```

```
## [1] 0.3940847
```

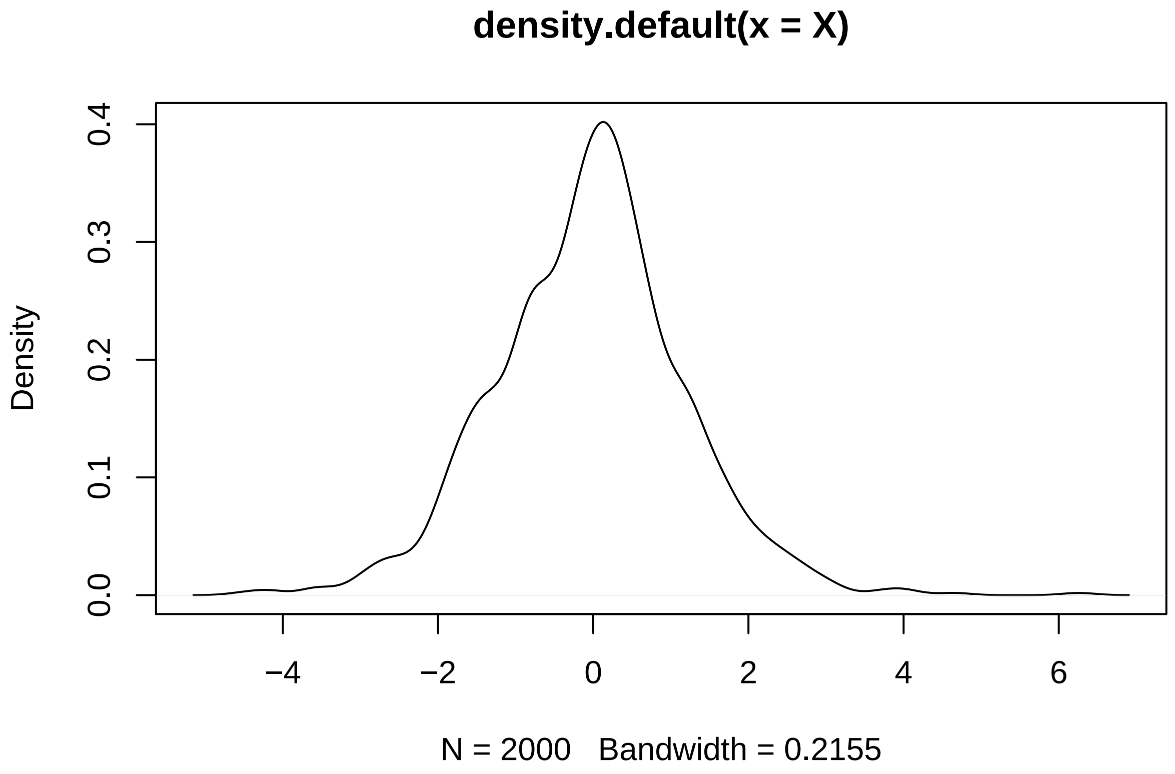
```
# variance of theta.hat
for (i in 1:m) {
  theta[i] <- exp(-x[i])
}
variance <- 0.25/m * var(theta)
variance
```

```
## [1] 3.176632e-07
```

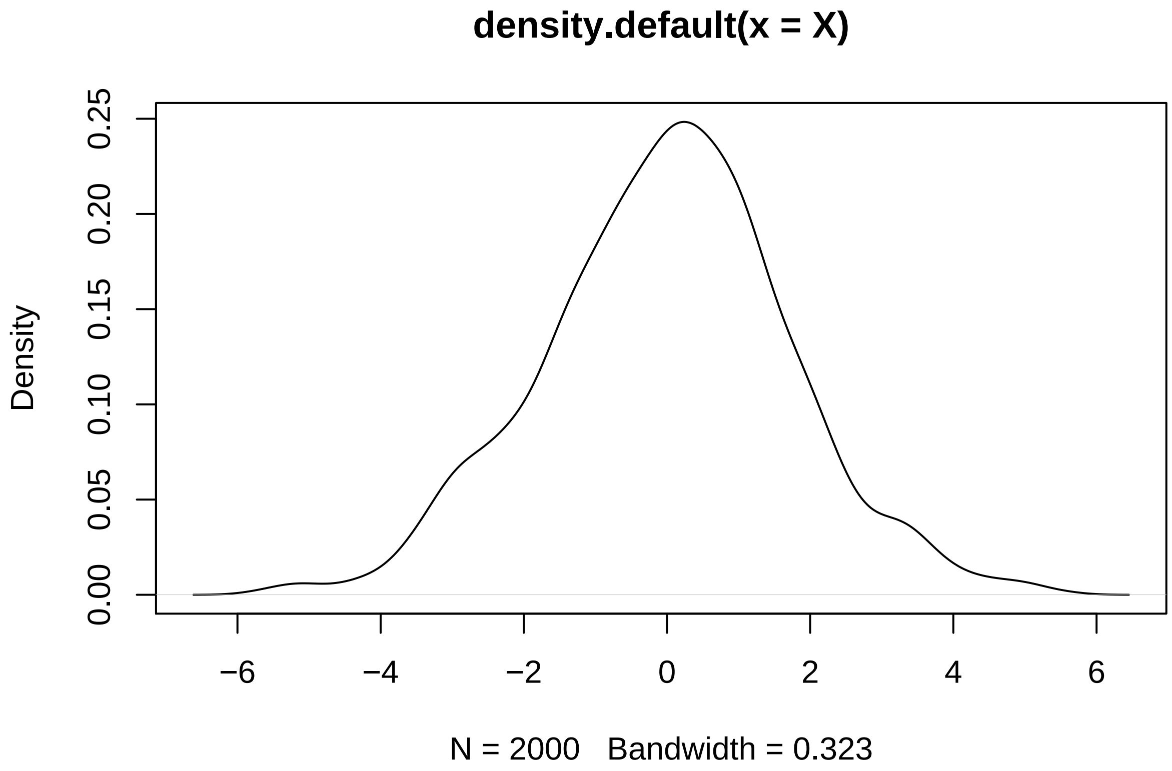
5 Question 5

mixture of normal distribution

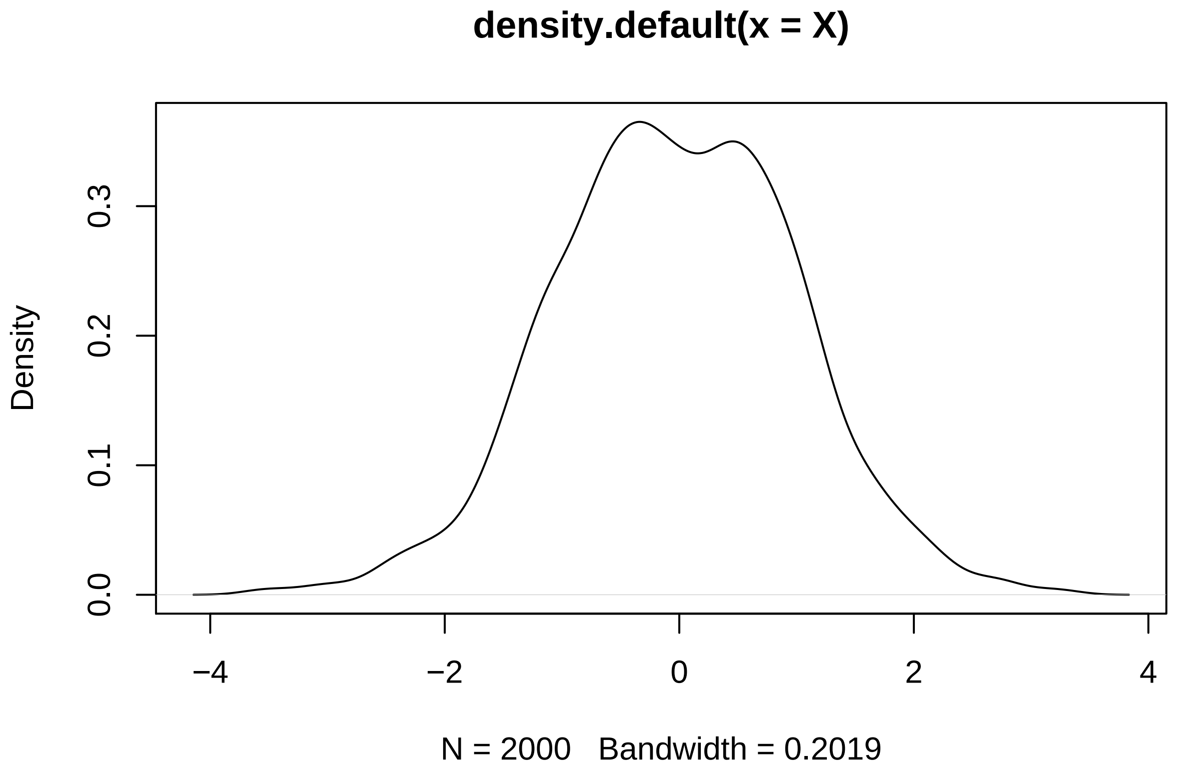
```
library(MASS)
# write function
loc.mix.0 <- function(n, p, mu1, mu2, Sigma1, Sigma2) {
#generate sample from BVN location mixture
X <- matrix(0, n, 2)
for (i in 1:n) {
k <- rbinom(1, size = 1, prob = p)
if (k)
X[i,] <- mvrnorm(1, mu = mu1, Sigma1) else
X[i,] <- mvrnorm(1, mu = mu2, Sigma2)
}
return(X)
}

X <- loc.mix.0(1000, 0.75, 0, 0 ,1, 3)
plot(density(X))
```



```r
X <- loc.mix.0(1000, 0.1, 0, 0 ,1, 3)
plot(density(X))
```



```r
X <- loc.mix.0(1000, 0.9, 0, 0 ,1, 3)
plot(density(X))
```



I changed multiple p values, this does not look like a bimodal distribution. I think it is because the mu1 = mu2 = 0. The parts with highest probability are overlapping, thus the mixture looks like a normal distribution.

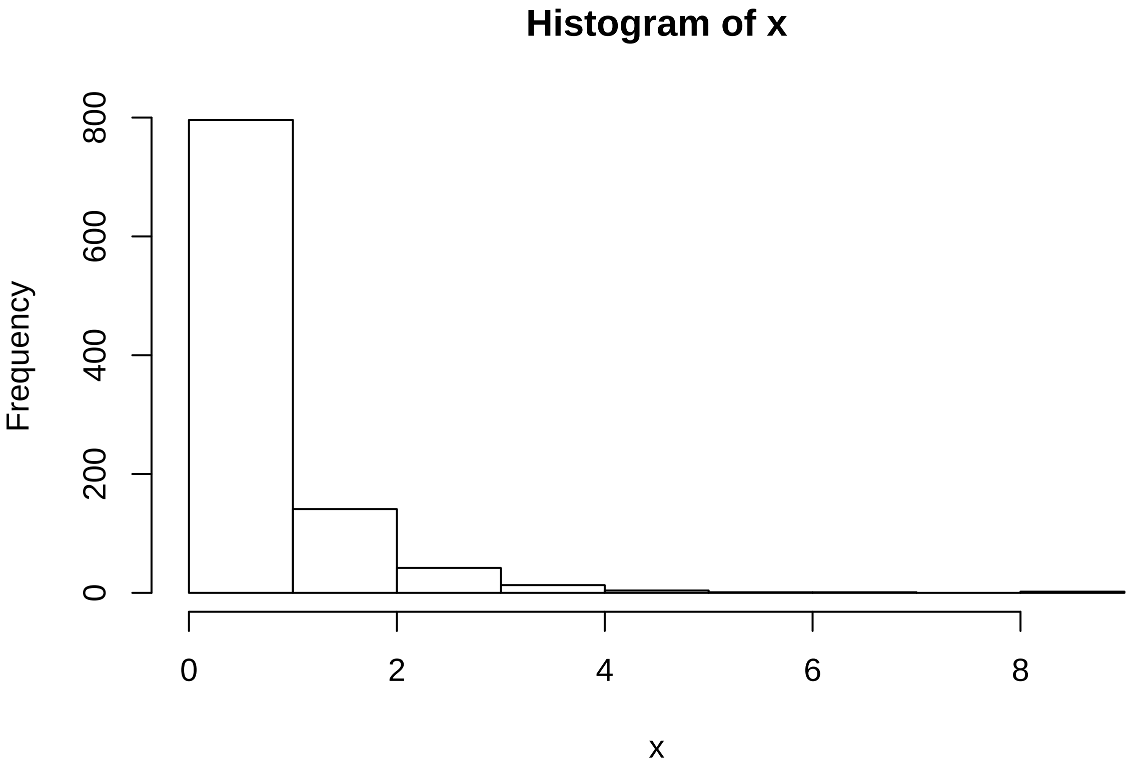
6 Question 6

Gamma-Exponetial mixture.

$$\Lambda \sim \Gamma(r, \beta)$$
$$(Y|\Lambda = \lambda) \sim exp(\Lambda)$$

```
# generate distributions of Lambda
n <- 1000 #1000 random variables
r <- 4
beta <- 2
lambda <- rgamma(n, r, beta)

#now apply the sample of lambda as the exponetial
```

```
x <- rexp(n, lambda)
hist(x)
```



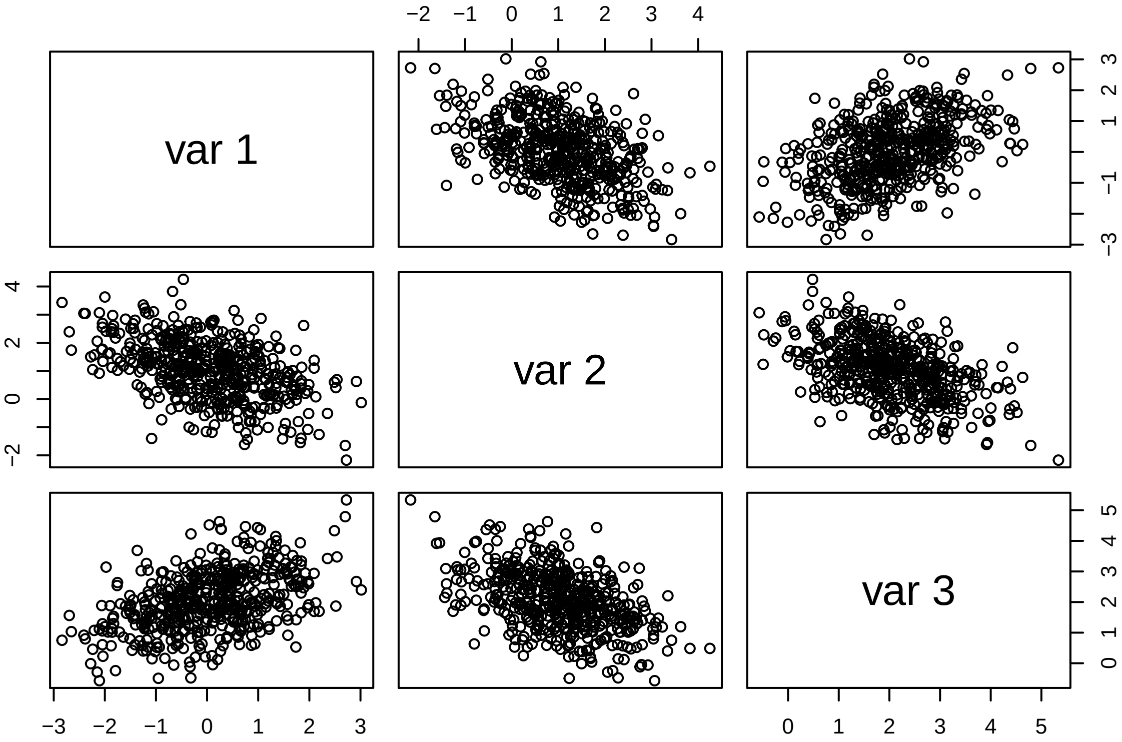
7 Question 7

requires to generate 500 random obsevations of the 3-dimensional MVN, using Choleski factorization method.

write a function of factorization method

```
rmvn.Choleski <-
function(n, mu, Sigma){
# generate n random vectors from MVN(mu, Sigma)
# dimension is inferred from mu and Sigma
d <- length(mu)
Q <- chol(Sigma) # Choleski factorization of Sigma
Z <- matrix(rnorm(n*d), nrow = n, ncol = d)
X <- Z %*% Q + matrix(mu, n, d, byrow = TRUE)
X
}
```

```r
# write input mean matrix and covriance matrix

mu <- c(0,1,2)
Sigma <- matrix(c(1, -0.5, 0.5, -0.5, 1, -0.5, 0.5, -0.5, 1),
                nrow = 3, byrow = TRUE)

X <- rmvn.Choleski(500, mu, Sigma)
pairs(X)
```



From the graph, it satisfies the theoretical expectation. The joint distribution of each pair of marginal distributions is theoretically bivariate normal. var1 and var2 are negatively correlated, and var1 and var 3 are postively correlated.