HUDM6026 Homework_02

Chenguang Pan

Feb 03, 2023

Question 01 SCR 3.3

MY SOLUTION:
The inverse transformation of the `Pareto(a,b)`'s cdf function is as followed.

$$F^{-1}(u) = \frac{b}{(1-u)^{\frac{1}{a}}}$$

```
> # define the quantile function of Pareto(a,b) distribution
> quantile_Pareto <-function(prob, a, b){
+   x <- b * (1-prob)^(-1/a)
+   return(x)
+ }
> # define the simulated sample size
> n <- 100
> u <- runif(n)
> # based on the uniformly generated vector to get the random sample
> X <- quantile_Pareto(u, 2, 2)
> range(X)
[1]  2.018862 12.215290
```

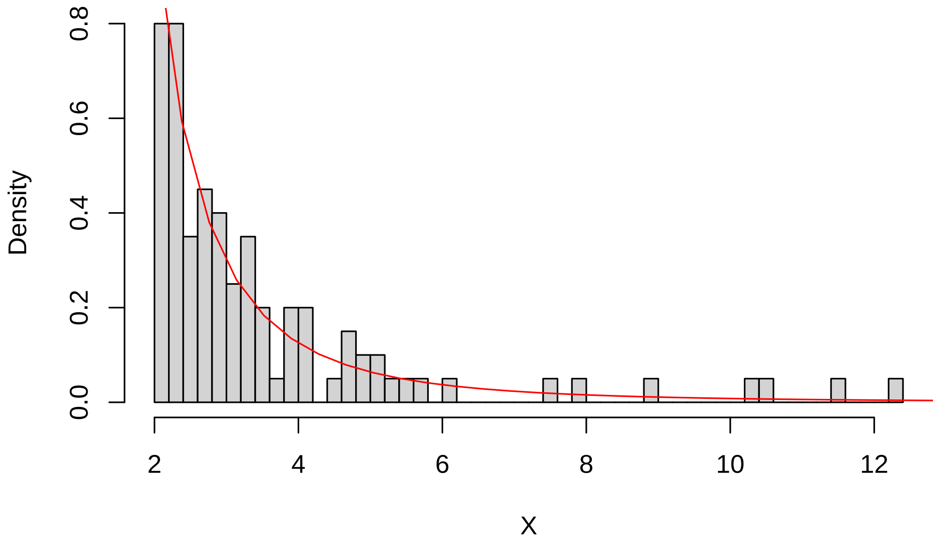
This inverse function runs well. Before comparing the simulated density and the original density, I derivate the CDF to get the pdf function of Pareto(a,b), that is:

$$f(x) = \frac{ab^a}{x^{a+1}}$$

```
> # draw the density histogram of the simulated data
> hist(X, prob = T,
+      breaks = 50,
+      main = expression(f(x)==ab^a/x^(a+1)))
> # prepare the Pareto(2,2) distribution
> x <- seq(2,40,.38)
> y <- 2*(2^2)/(x^(2+1))
> # superimpose the lines on the simulated density
> lines(x, y, col="red")
> mtext("Figure 1. Comparing the simulated data with Pareto(a,b)",
+       side = 3,
+       line = -1,
+       outer = T)
```

Figure 1. Comparing the simulated data with Pareto(a,b)

$$f(x) = ab^a / x^{(a+1)}$$

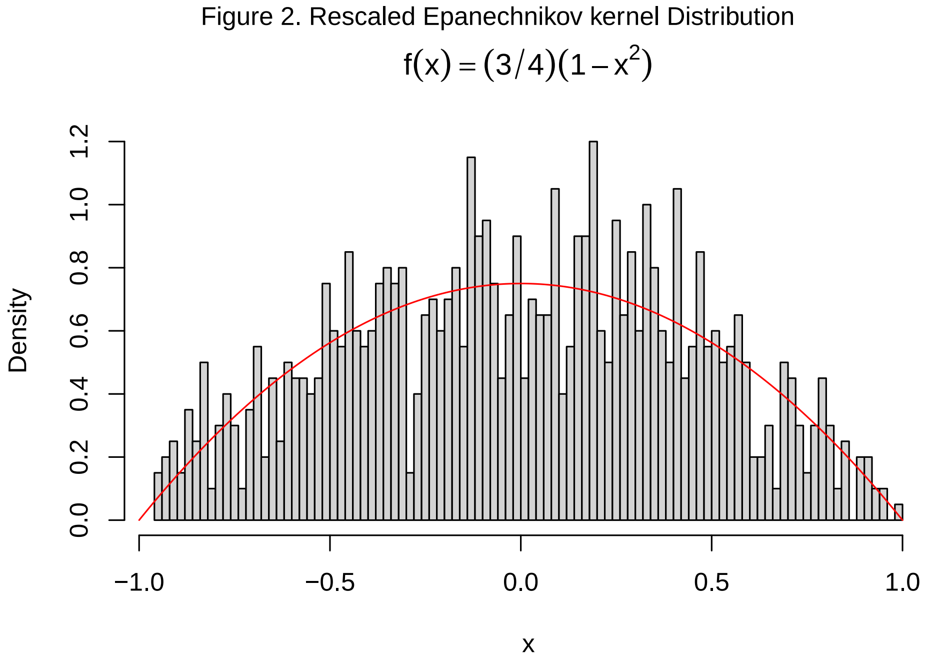


Question 02 SCR 3.9

MY SOLUTION:

This question has already given the clues to generate random variable for the rescaled Epanechnikov kernel

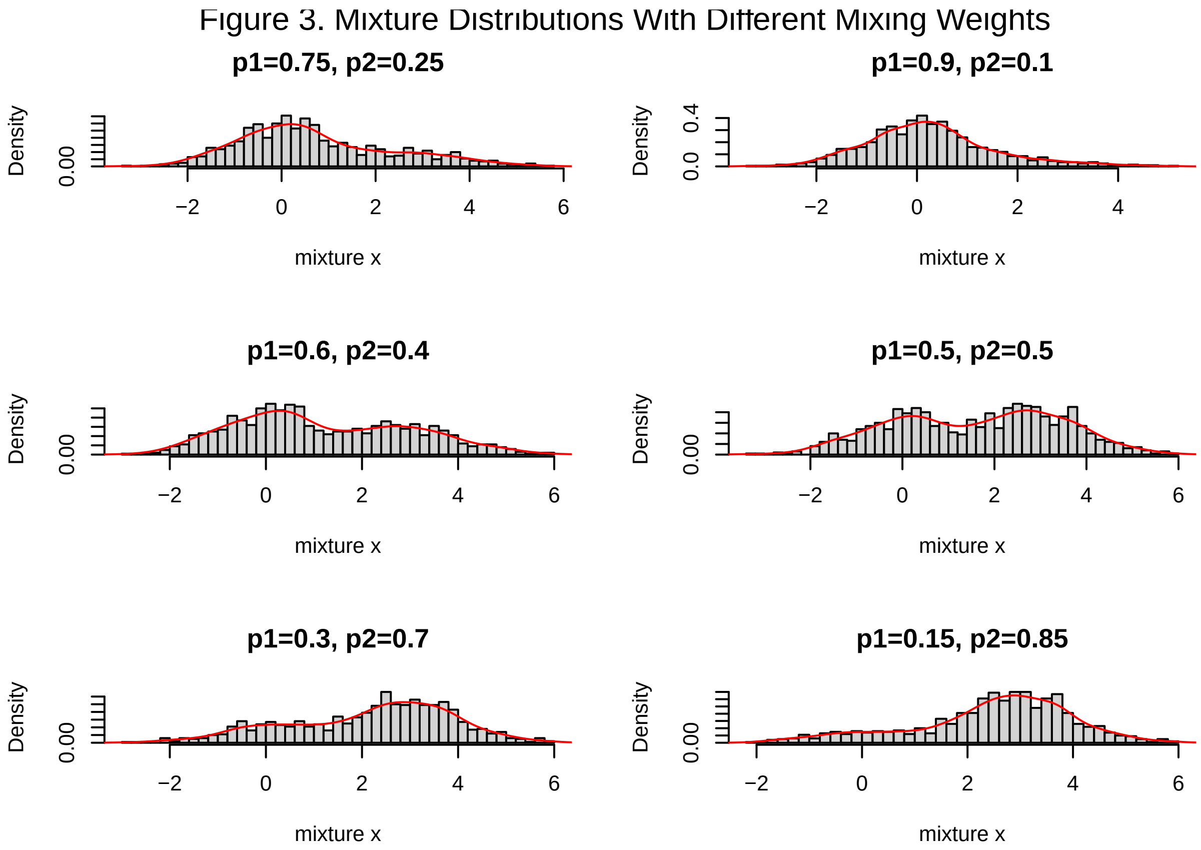
```
> # write a function based on text's information
> gen_var <- function(n){ # n is the sample size
+   U_1 <- runif(n, -1, 1)
+   U_2 <- runif(n, -1, 1)
+   U_3 <- runif(n, -1, 1)
+   U_output <- c()
+   for (i in c(1:n)) {
+     if (abs(U_3[i]) > abs(U_2[i]) &
+         abs(U_3[i]) > abs(U_1[i]))
+       {U_output[i] <- U_2[i]}
+     else
+       {U_output[i] <- U_3[i]}
+   }
+   return(U_output)
+ }
>
> # generate 1000 data
> U_output <- gen_var(1000)
> hist(U_output, prob = T,
+      breaks = 100,
+      xlab = "x",
+      main = expression(f(x)==(3/4)*(1-x^2)))
> x_vec <- seq(-1,1,0.001)
> f_x <- 0.75*(1-x_vec^2)
> lines(x_vec, f_x, col="red")
> mtext("Figure 2. Rescaled Epanechnikov kernel Distribution",
+       side = 3,
+       line = -1,
+       outer = T)
```



Question 03 SCR 3.11

MY SOLUTION:
How to better understand the mixing weights (i.e., the mixing probabilities)? The mixing weights is about **how much** each individual distribution contributes to the mixture distribution(Stephanie Glen. StatisticsHowTo.com). Therefore, when constructing the mixture function, one should not directly use the probability of each parent distribution as a coefficient!!

```
> set.seed(1000)
> n <- 1000
> # generate two vectors from normal distribution
> x1 <- rnorm(n,0,1)
> x2 <- rnorm(n,3,1)
>
> # use a for-loop to draw graphs at different mixing weights
> par(mfrow=c(3,2))
> for (p1 in c(0.75, 0.90, 0.60, 0.5, 0.30, 0.15)){
+   # define the mixing prob
+   p2 <- 1 - p1
+   # use n data from uniform distribution to construct
+   # the proportion of each parent distribution.
+   u <- runif(n)
+   k <- as.integer(u > p2)
+   x <- k * x1 + (1-k) * x2
+   hist(x, prob = T,
+         breaks = 50,
+         xlab = "mixture x",
+         main = sprintf("p1=%s, p2=%s", p1, p2))
+   lines(density(x), col= "red")
+ }
> mtext("Figure 3. Mixture Distributions With Different Mixing Weights",
+       side = 3,
+       line = -1,
+       outer = T)
```



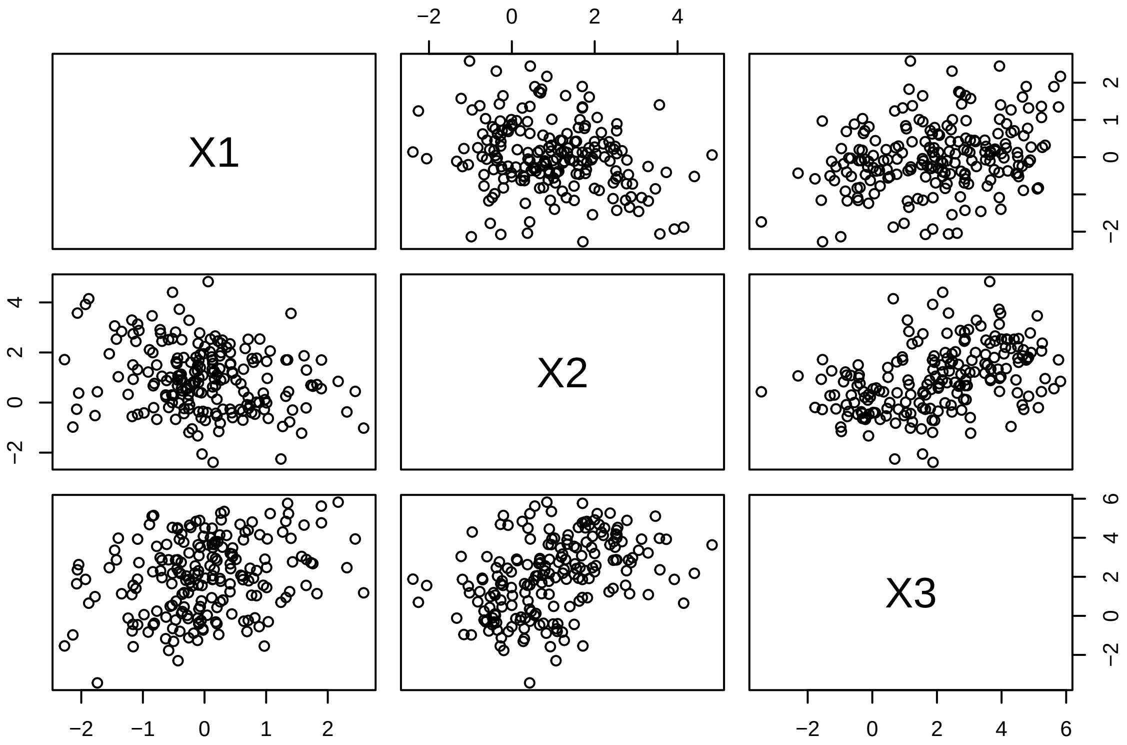
From the graphs, one can find that when the mixing weights are .5 and .5, the mixture distribution is apparently a bimodal distribution. At this circumstance the two samples contribute equally to the final mixture. But this bimodal distribution might not be symmetrical because the two parent distribution's shapes are different in variance. With increasing in P1, the bimodal distribution will be more positively skewed.

Question 04 SCR 3.14

MY SOLUTION:
For solving this question, I refer to Prof.Keller's in-class demo codes and remove unnecessary if-else expressions.

```
> # create a function to gen data matrix from a multivariate normal distribution.
> mvn_gen <- function(n, mu, sigma){
+   # to determine the dimension
+   d <- length(mu)
+   # generate a n*d matrix from a standard normal distribution
+   Z <- matrix(rnorm(n*d), nrow = n, ncol = d)
+   # use Cholesky function to factorize the Cov-var matrix
+   Q <- chol(sigma)
+   # to transform the mu from dataframe to matrix
+   mu <- matrix (mu, nrow = d, ncol = 1)
+   # J is column vector of ones
+   J <- matrix(mu, nrow = n, ncol = 1)
+   # define the output X matrix
+   X <- Z %*% Q + J %*% t(mu)
+   return(data.frame(X))
+ }
>
> # input the given cov-var matrix
> sig <- matrix(c(1.0, -0.5, 0.5,
+                 -0.5, 1.0, -0.5,
+                 0.5, -0.5, 1.0),3,3)
> set.seed(100)
> X <- mvn_gen(n=200, mu = c(0,1,2), sigma = sig)
> pairs(X)
> mtext("Figure 4. Generate Data From Multivariate Normal Distribution",
+       side = 3,
+       line = -1,
+       outer = T)
```

Figure 4. Generate Data From Multivariate Normal Distribution



From the cov-var matrix, one can easily get the correlation coefficient between the x_1 and x_2 is -0.5. The middle-right graph demonstrated that these two variables are negatively correlated. By visually check, the joint mean point is at around`(0,1)`, which agrees with the given condition. The bottom-right graph also meets the given condition. However, the correlation between the x_2 and x_3 is obscure by visual check.

Question 05 [Bonous]

First show that the sample mean estimator is unbiased for the true population mean. Next, show that the mle estimator for the variance...

MY SOLUTION for Part 1:

Before proving the first statement, I want to refresh the understanding of several stats concepts. By definition, an **estimator** is a rule that is used to estimate an unknown parameter based on sample. The concept **Unbiased** means that the expectation of an estimator equals to the population parameter, i.e., $E(estimator) = PopulationParamter$.

The sample mean estimator is

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

, where $\overline{x}$ is the mean of a sample with n size. therefore,

$$E(estimator) = E(\overline{x}) = E(\frac{1}{n}\sum_{i=1}^{n} x_i) = \frac{1}{n}\sum_{i=1}^{n} E(x_i)$$

. Since x_i is draw from the population with mean μ, one can have $E(x_i) = \mu$. Put this equation to the above, I get

$$E(estimator) = E(\overline{x}) = \frac{1}{n}\sum_{i=1}^{n} \mu = \frac{1}{n}n\mu = \mu$$

. In short, here I have

$$E(\overline{x}) = \mu$$

, which proved the sample mean estimator is unbiased for the true population mean.

MY SOLUTION for Part 2:
By definition, the maximum likelihood estimator for the variance is

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2$$

. The variance of the population is

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$$

. Visually checking the two equations, it seems identical. Based on the definition of "unbiased", we need to further check whether the two above follow this rule $E(estimator) = PopulationParamter$.

Here, I have

$$E(estimator) = E(s^2) = E(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2) = \frac{1}{n}E(\sum_{i=1}^{n} {x_i}^2 - 2\sum_{i=1}^{n} x_i\overline{x} + \sum_{i=1}^{n} \overline{x}^2)$$

. Because $\sum_{i=1}^{n} x_i = n\overline{x}$, combining these two equation above one can have

$$E(estimator) = E(s^2) = \frac{1}{n}E(\sum_{i=1}^{n} {x_i}^2 - 2n\overline{x}\overline{x} + n\overline{x}^2) = E(\frac{1}{n}\sum_{i=1}^{n} {x_i}^2) - E(\overline{x}^2) = E(x^2) - E(\overline{x}^2)$$

.

Based on the definition of variance, we can have a equation of population variance, that is,

$$\sigma^2 = E\{[x - E(x)]^2\} = E[x^2 - 2xE(x) + E(x)^2] = E(x^2) - E(x)^2$$

. Based on this formula, we can also have

$$\sigma^2_{\overline{x}} = E(\overline{x}^2) - E(\overline{x})^2$$

. Since $E(x) = E(\overline{x}) = \mu$, based on all the equation above, one can have

$$E(estimator) = E(s^2) = \sigma^2 - \sigma^2_{\overline{x}}$$

. We need to go further to derive the $\sigma^2_{\overline{x}}$. Here, by the rule of variance, one can have

$$\sigma^2_{\overline{x}} = var(\overline{x}) = var(\frac{1}{n}\sum_{i=1}^{n} x_i) = \frac{1}{n^2}var(\sum_{i=1}^{n} x_i)$$

. Because the samples are independently and identically drawn from the population, the covariance among them should be zero, we can have

$$\sigma^2_{\overline{x}} = \frac{1}{n^2}\sum_{i=1}^{n} var(x) = \frac{1}{n}var(x) = \frac{1}{n}\sigma^2$$

.From the equations above, one can find the *mle* estimator of variance does not follow the rule $E(estimator) = PopulationParamter$, which means this estimator is biased. The difference between this estimator and the true population variance is $\frac{1}{n}\sigma^2$, which represents the bias.

$$E(s^2) - \sigma^2 = -\sigma^2_{\overline{x}} = -\frac{1}{n}\sigma^2$$

Reference For this Homework

Glen,S.(n.d.). *Mixture Distribution: Definition and Examples.* https://www.statisticshowto.com/mixture-distribution/

Keller, B.(2023). *HUDM 6026 Computational Statistics: Simulation and Other Monte Carlo Methods*[Lecture notes].

Liang, D. (2012). *Maximum likelihood estimator for variance is biased: Proof.* https://dawenl.github.io/files/mle_biased.pdf

Rizzo, M. L. (2019). *Statistical computing with R.* Chapman and Hall/CRC.