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.

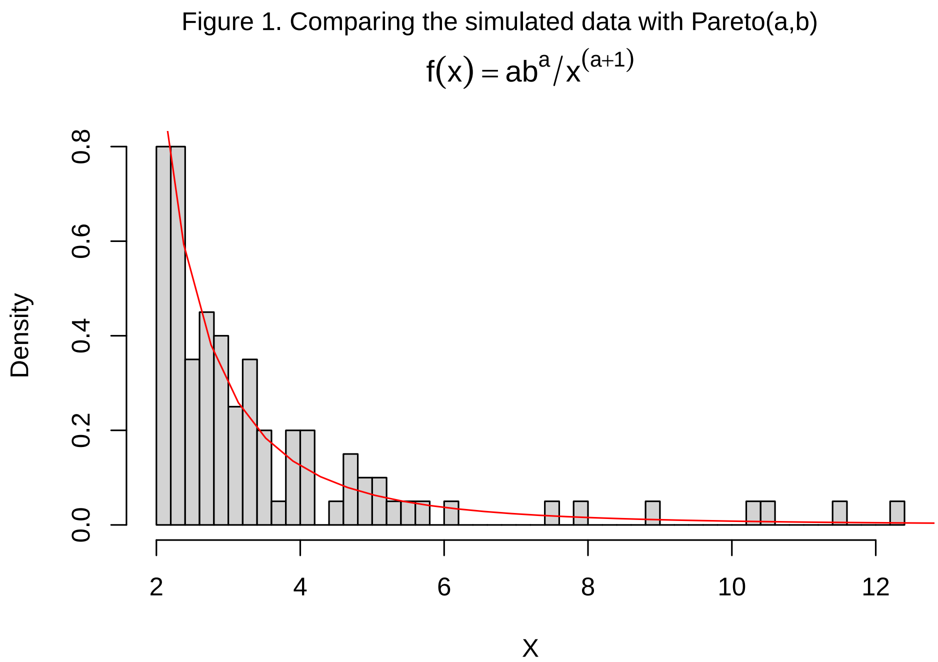
$$F^{-1}(u) = \frac{b}{(1-u)^{\frac{1}{a}}}$$

```
> # 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)
```



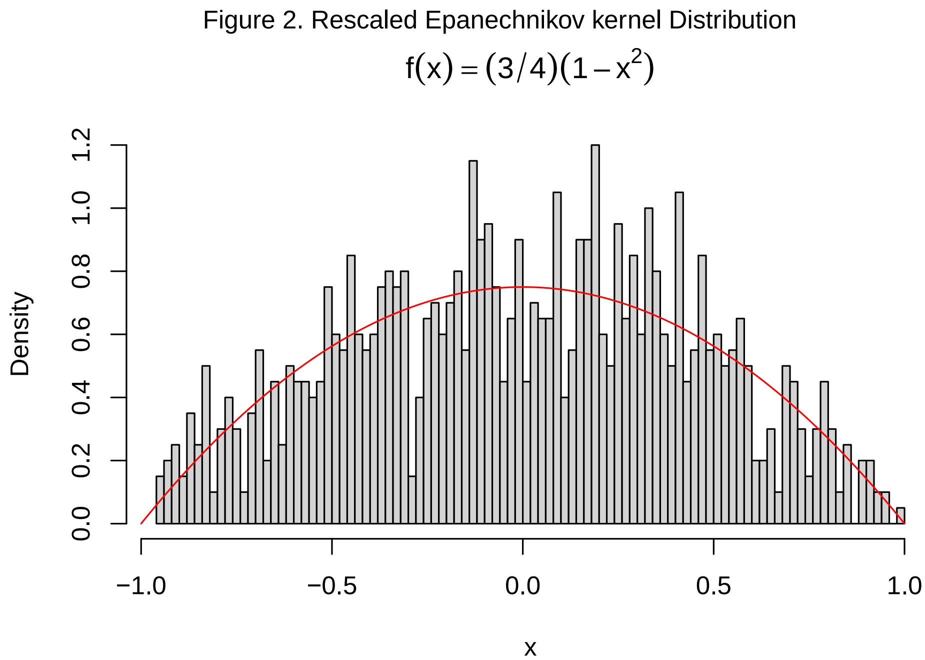
Question 02 SCR 3.9

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)
```

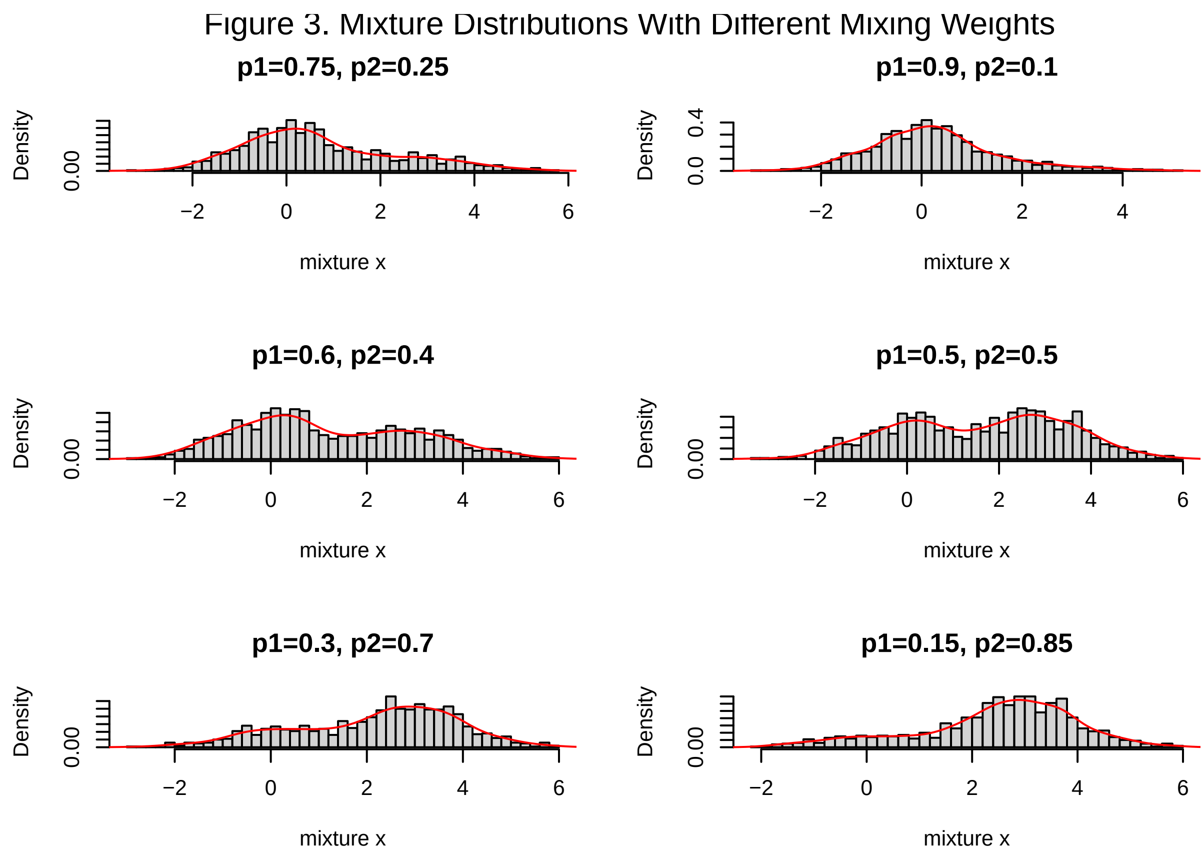


Question 03 SCR 3.11

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. Statistic-sHowTo.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.