STAT7400 HW10, 2017

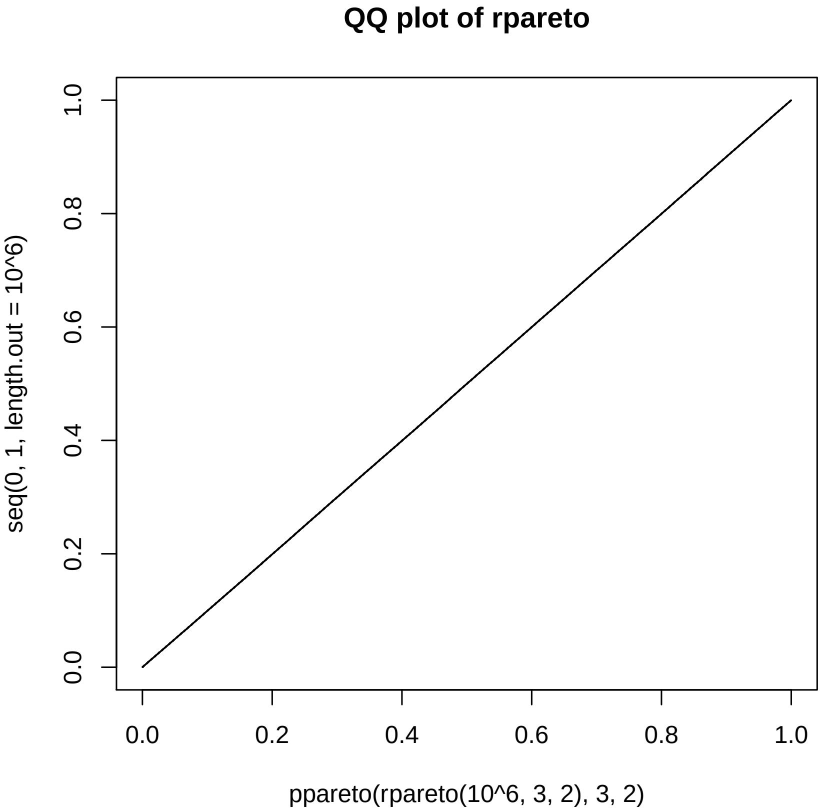
Yiheng Liu

Apr 17, 2017

Problem 1

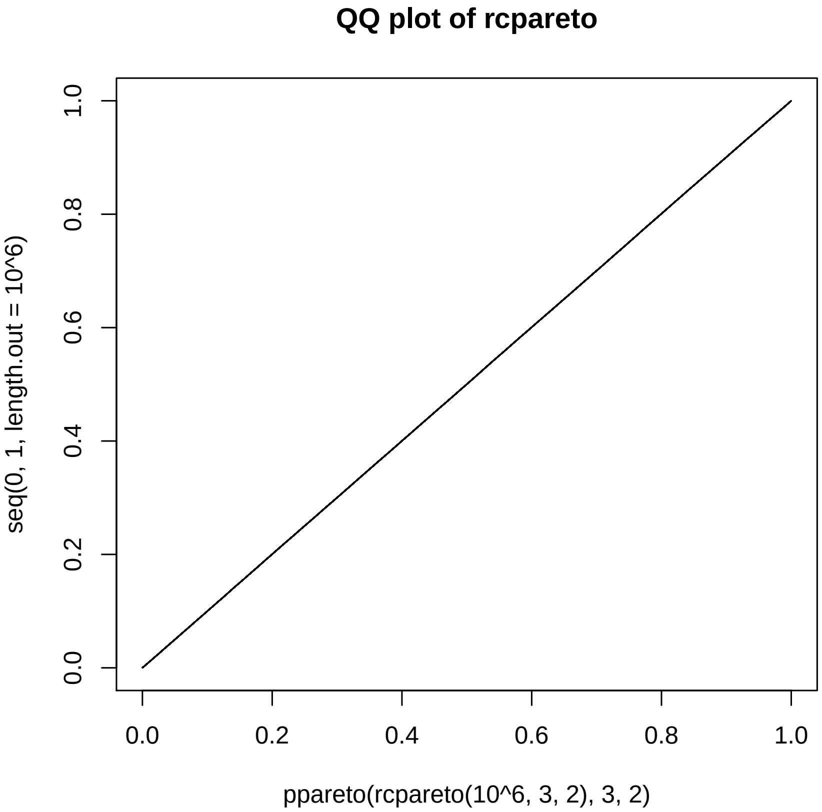
(a) With `runif`, uniform random numbers can be generated. Then I can use `qpareto` to generate pareto random numbers by Inverse CDF method. Some examples of `rpareto` are shown below.

```
> library(pareto)
> rpareto(1,3,-2)

[1] NaN

> rpareto(1,3,2)

[1] 71.59332

> rpareto(4,c(100,1),2)

[1] 338.870932   1.413489 121.032345   1.942355

> #Check reproducability
> set.seed(100)
> a <- rpareto(c(NA,1,99), 3, 2)
> set.seed(100)
> b <- rpareto(3, 3, 2)
> all.equal(a,b)

[1] TRUE

> #Check qq-plot
> qqplot(ppareto(rpareto(10^6,3,2),3,2), seq(0,1, length.out = 10^6),
+        type = "l", main = "QQ plot of rpareto")
```



(b) With `unif_rand`, uniform random numbers can be generated in C. Then I can use Inverse CDF method to generate pareto random numbers. R function `rcpareto` is defined to call my C code through .C interface. Some examples of `rcpareto` are shown below.

```
> rcpareto(1,3,-2)

[1] NaN

> rcpareto(1,3,2)

[1] 4.986479

> rcpareto(4,c(100,1),2)

[1] 137.136972   1.043180 264.620436   1.031287

> #Check reproducability
> set.seed(100)
> a <- rcpareto(c(NA,1,99), 3, 2)
> set.seed(100)
> b <- rcpareto(3, 3, 2)
> all.equal(a,b)

[1] TRUE

> #Check qq-plot
> qqplot(ppareto(rcpareto(10^6,3,2),3,2), seq(0,1, length.out = 10^6),
+        type = "l", main = "QQ plot of rcpareto")
```



It is expected that C should be more efficient than R. The comparison of two functions are as follows.

```
> #Compare efficiency of two functions
> system.time(rpareto(10^6, 3, 2))

   user  system elapsed
  0.125   0.004   0.131

> system.time(rcpareto(10^6, 3, 2))

   user  system elapsed
  0.092   0.000   0.093
```