Quiz 2 Solutions

March 29, 2019

0.1 Q1

Have $f(x; v) = vx^{-(v+1)}$ for $x \geq 1$. Need to solve

$$\sum_{r=1}^{R} 1\{X_r > \gamma_i\} \frac{f(X_r)}{f(X_r; v_{i-1})} \frac{\mathrm{d}}{\mathrm{d}v} \Big\{ \log\big[f(X_r; v)\big] \Big\} = 0$$

start with

$$\frac{\mathrm{d}}{\mathrm{d}v} \log(vx^{-v-1}) = \frac{\mathrm{d}}{\mathrm{d}v} \log(v) - (v+1)\log(x) = \frac{1}{v} - \log(x).$$

Then

$$\sum_{r=1}^{R} 1\{X_r > \gamma_i\} \frac{f(X_r)}{f(X_r; v_{i-1})} \Big(\frac{1}{v_i} - \log(X_r)\Big) = 0$$

$$\frac{1}{v_i} \sum_{r=1}^{R} 1\{X_r > \gamma_i\} \frac{f(X_r)}{f(X_r; v_{i-1})} = \sum_{r=1}^{R} 1\{X_r > \gamma_i\} \frac{f(X_r)}{f(X_r; v_{i-1})} \log(X_r)$$

$$\Rightarrow v_i = \frac{\sum_{r=1}^{R} 1\{X_r > \gamma_i\} \frac{f(X_r)}{f(X_r; v_{i-1})}}{\sum_{r=1}^{R} 1\{X_r > \gamma_i\} \frac{f(X_r)}{f(X_r; v_{i-1})} \log(X_r)}.$$

Compare with "maximum likelihood estimator" for Pareto on Wikipedia, which is

$$\hat{v} = \frac{R}{\sum_{r=1}^{R} \log(X_r)}.$$

0.2 Q2

```
In [1]: library(actuar, warn.conflicts = FALSE)

        set.seed(1337)

        gamma <- 100000
        maxIter <- 100
        R <- 10^6
        rho <- 0.1

        u <- c(11, 12, 13)

        v <- u

        for (i in 1:maxIter) {
            X1s <- rpareto(R, v[1], 1) + 1
            X2s <- rpareto(R, v[2], 1) + 1
```

```r
        X3s <- rpareto(R, v[3], 1) + 1
        Ss <- X1s + X2s + X3s

        gamma_i <- as.numeric(quantile(Ss, 1-rho))
        print("Threshold: ")
        print(gamma_i)

        if (gamma_i >= gamma)
            break

        indicators <- (Ss > gamma_i)

        # Stable way.
        LR1s <- dpareto(X1s - 1, u[1], 1) / dpareto(X1s - 1, v[1], 1)
        LR2s <- dpareto(X2s - 1, u[2], 1) / dpareto(X2s - 1, v[2], 1)
        LR3s <- dpareto(X3s - 1, u[3], 1) / dpareto(X3s - 1, v[3], 1)
        LRs <- LR1s * LR2s * LR3s

#       # Less numerically stable way!
#       LRNumer <- dpareto(X1s - 1, u[1], 1) *
#                  dpareto(X2s - 1, u[2], 1) *
#                  dpareto(X3s - 1, u[3], 1)
#       LRDenom <- dpareto(X1s - 1, v[1], 1) *
#                  dpareto(X2s - 1, v[2], 1) *
#                  dpareto(X3s - 1, v[3], 1)
#       LRsUnst <- LRNumer / LRDenom

#       print("Stability max error: ")
#       print(max(abs(LRs - LRsUnst)))

        v[1] <- sum(indicators * LRs) / sum(indicators * LRs * log(X1s))
        v[2] <- sum(indicators * LRs) / sum(indicators * LRs * log(X2s))
        v[3] <- sum(indicators * LRs) / sum(indicators * LRs * log(X3s))

        print("Next proposal dist: ")
        print(v)
    }
```

```
## [1] "Threshold: "
## [1] 3.500503
## [1] "Next proposal dist: "
## [1] 4.627691 5.438179 6.274371
## [1] "Threshold: "
## [1] 4.301588
## [1] "Next proposal dist: "
## [1] 2.021405 3.012076 4.216351
## [1] "Threshold: "
## [1] 6.416932
## [1] "Next proposal dist: "
## [1] 0.8451012 2.6014956 5.6327097
## [1] "Threshold: "
## [1] 18.34283
## [1] "Next proposal dist: "
## [1]  0.3675018  4.1716436 12.2280573
```

```
[1] "Threshold: "
[1] 524.6714
[1] "Next proposal dist: "
[1]  0.1575161 12.1473515 13.1947135
[1] "Threshold: "
[1] 2248143
```

In [2]:
```r
# Perform the importance sampling
X1s <- rpareto(R, v[1], 1) + 1
X2s <- rpareto(R, v[2], 1) + 1
X3s <- rpareto(R, v[3], 1) + 1
Ss <- X1s + X2s + X3s

indicators <- (Ss > gamma)

LR1s <- dpareto(X1s - 1, u[1], 1) / dpareto(X1s - 1, v[1], 1)
LR2s <- dpareto(X2s - 1, u[2], 1) / dpareto(X2s - 1, v[2], 1)
LR3s <- dpareto(X3s - 1, u[3], 1) / dpareto(X3s - 1, v[3], 1)
LRs <- LR1s * LR2s * LR3s

ests <- indicators * LRs
estMean <- mean(ests)
estVar <- var(ests)

print("Estimate: ")
print(estMean)

q <- abs(qnorm(0.01/2))

lowerCI <- estMean - q * sqrt(estVar / R)
upperCI <- estMean + q * sqrt(estVar / R)

print("99% CI's: ")
print(c(lowerCI, upperCI))
```

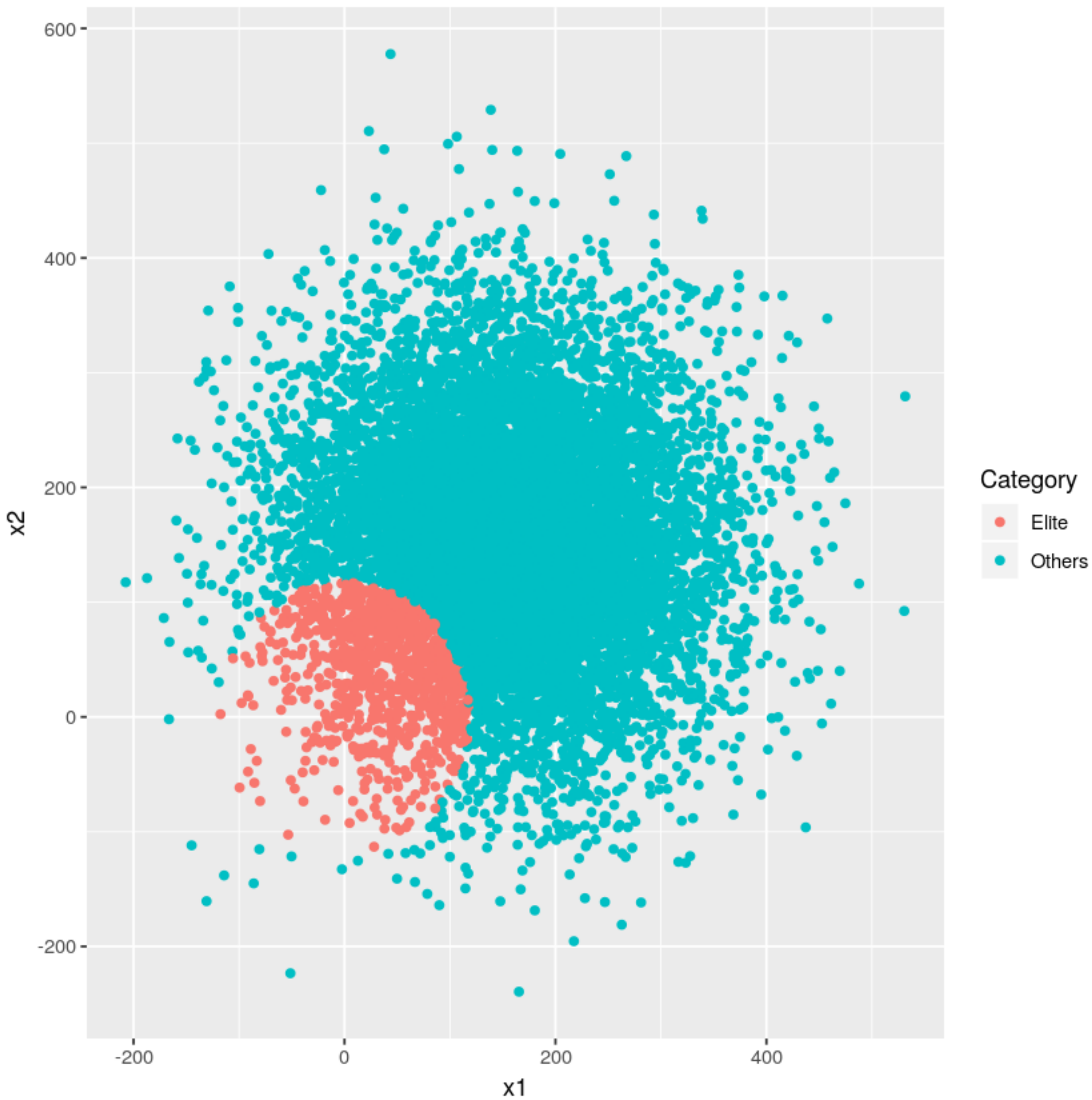
```
[1] "Estimate: "
[1] 1.023112e-55
[1] "99% CI's: "
[1] 9.844785e-56 1.061746e-55
```

0.3 Q3

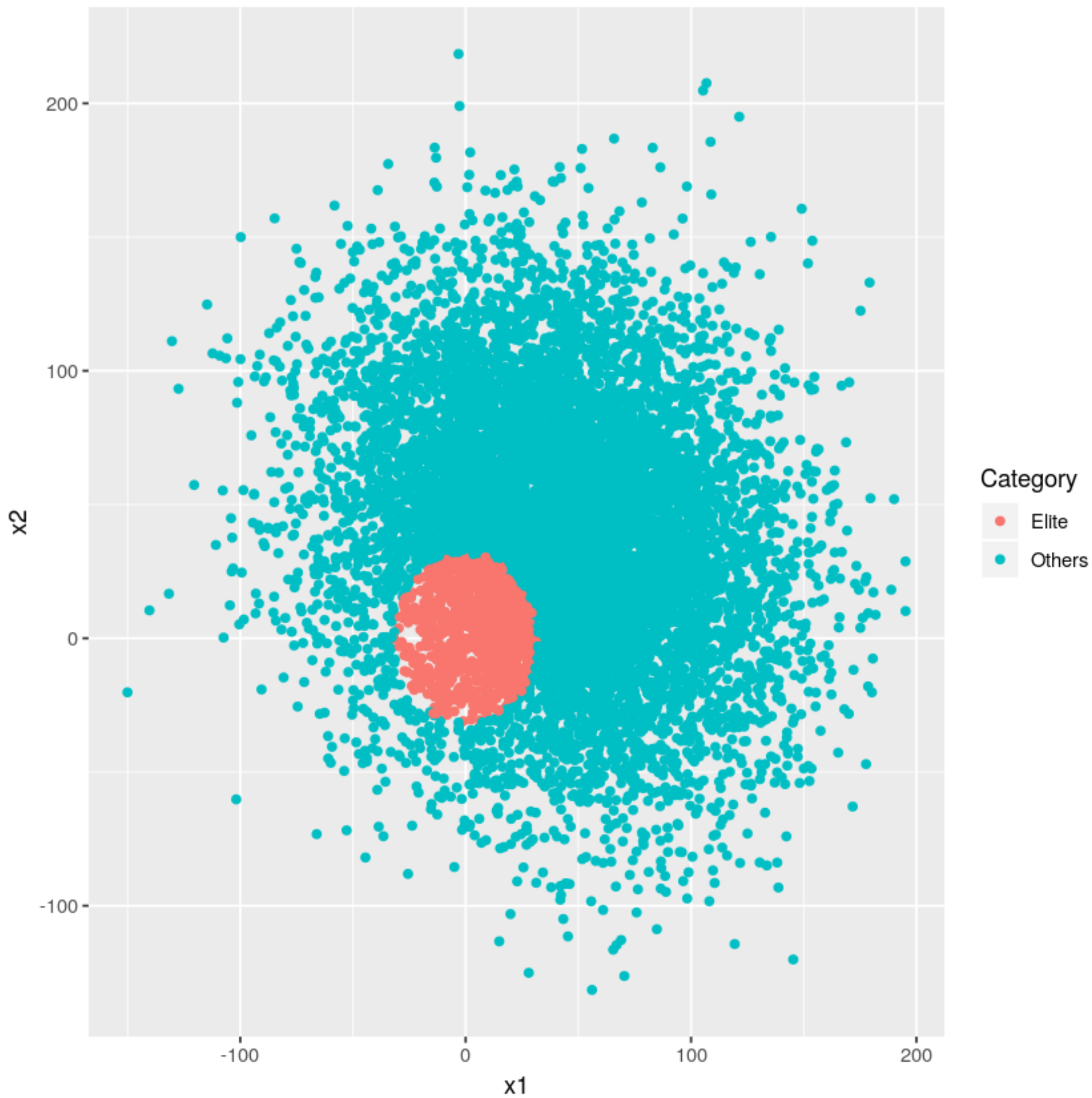
In [3]:
```r
library(ggplot2)
library(mvtnorm)

set.seed(1337)

S <- function(x) { 20 + x[,1]^2 - 10*cos(2*pi*x[,1]) + x[,2]^2 - 10*cos(2*pi*x[,2])}

R <- 10^6
rho <- 0.10
maxIter <- 20
```

```r
mu <- rep(150, 2)
sigma <- 10000*diag(2)

bestValues <- c()

for (iter in 1:maxIter) {
    Xs <- rmvnorm(R, mu, sigma)
    Ss <- S(Xs)
    bestValues <- c(bestValues, min(Ss))

    gamma <- quantile(Ss, rho)
    elites <- Xs[Ss<gamma,]

    if (iter < 5 || iter %% 10 == 0) {
        df <- data.frame(x1=Xs[,1], x2=Xs[,2], elite=(Ss<gamma))
        df$Category[Ss<gamma] <- "Elite"
        df$Category[Ss>=gamma] <- "Others"

        print(qplot(x1, x2, color=Category, data = df[1:10000,]))
    }

    mu <- c(mean(elites[,1]), mean(elites[,2]))
    sigma <- cov(elites)

    cat("Iteration: ", iter, "\tBest value: ", min(Ss),
        "\t\tMax(mu): ", max(mu), "\tMax(sigma): ", max(sigma), "\n")
}

plot(1:maxIter, bestValues, log = "y")
```

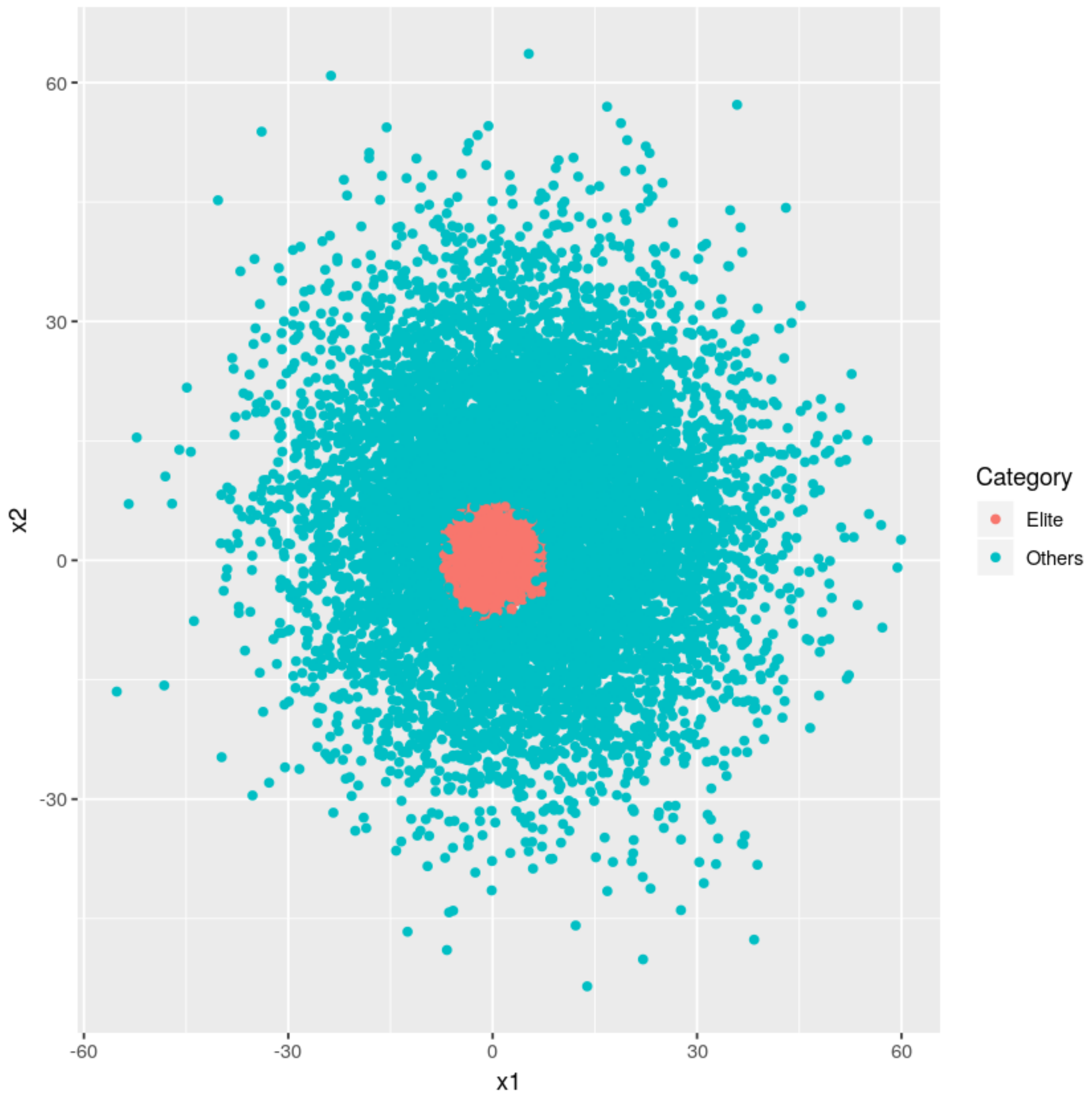
```
Iteration:  1         Best value:  1.900643                 Max(mu):  39.23646         Max(sigma):  2235
```



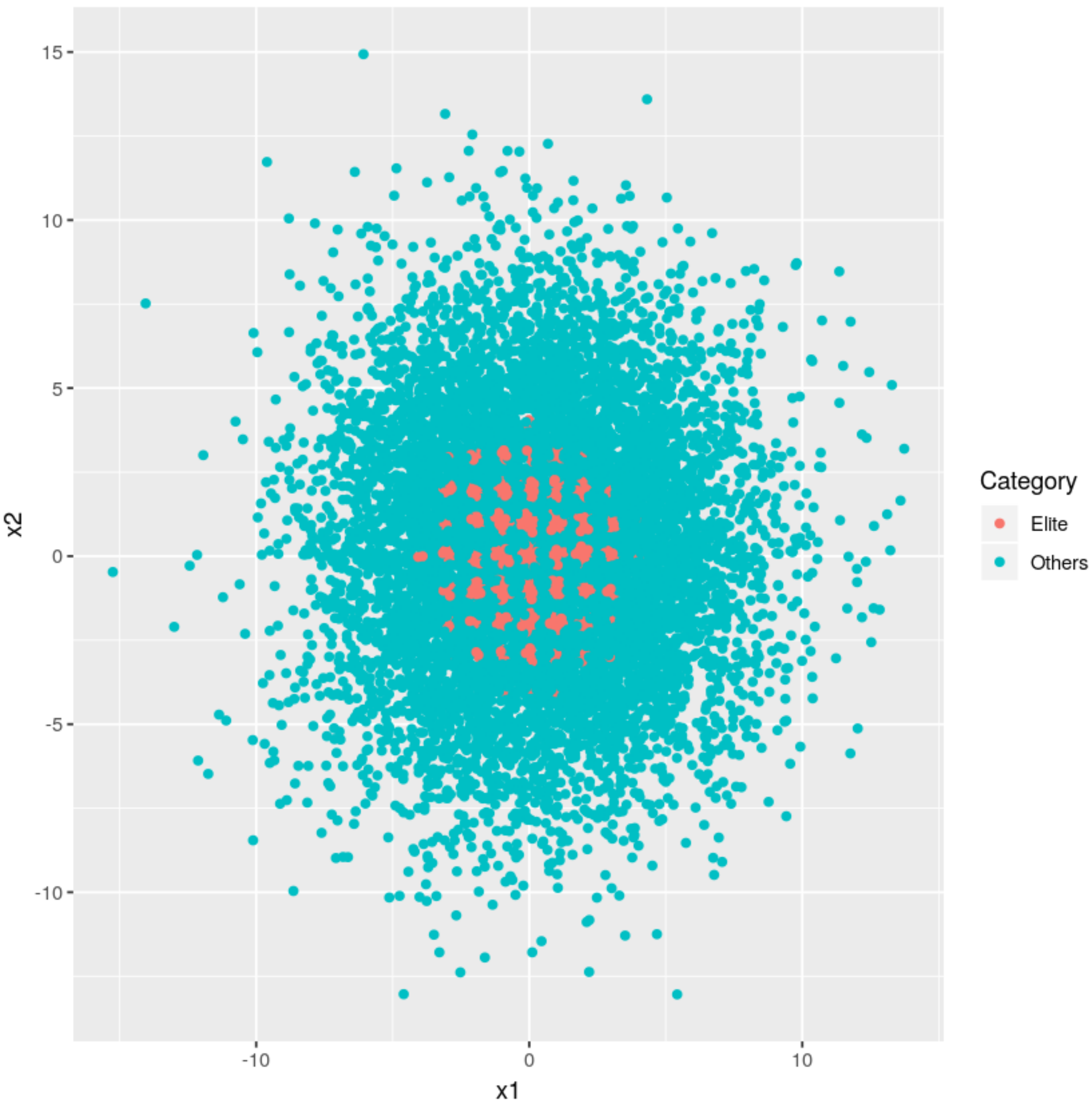

```
Iteration:  2         Best value:  1.098078                 Max(mu):  4.870495         Max(sigma):  227
```



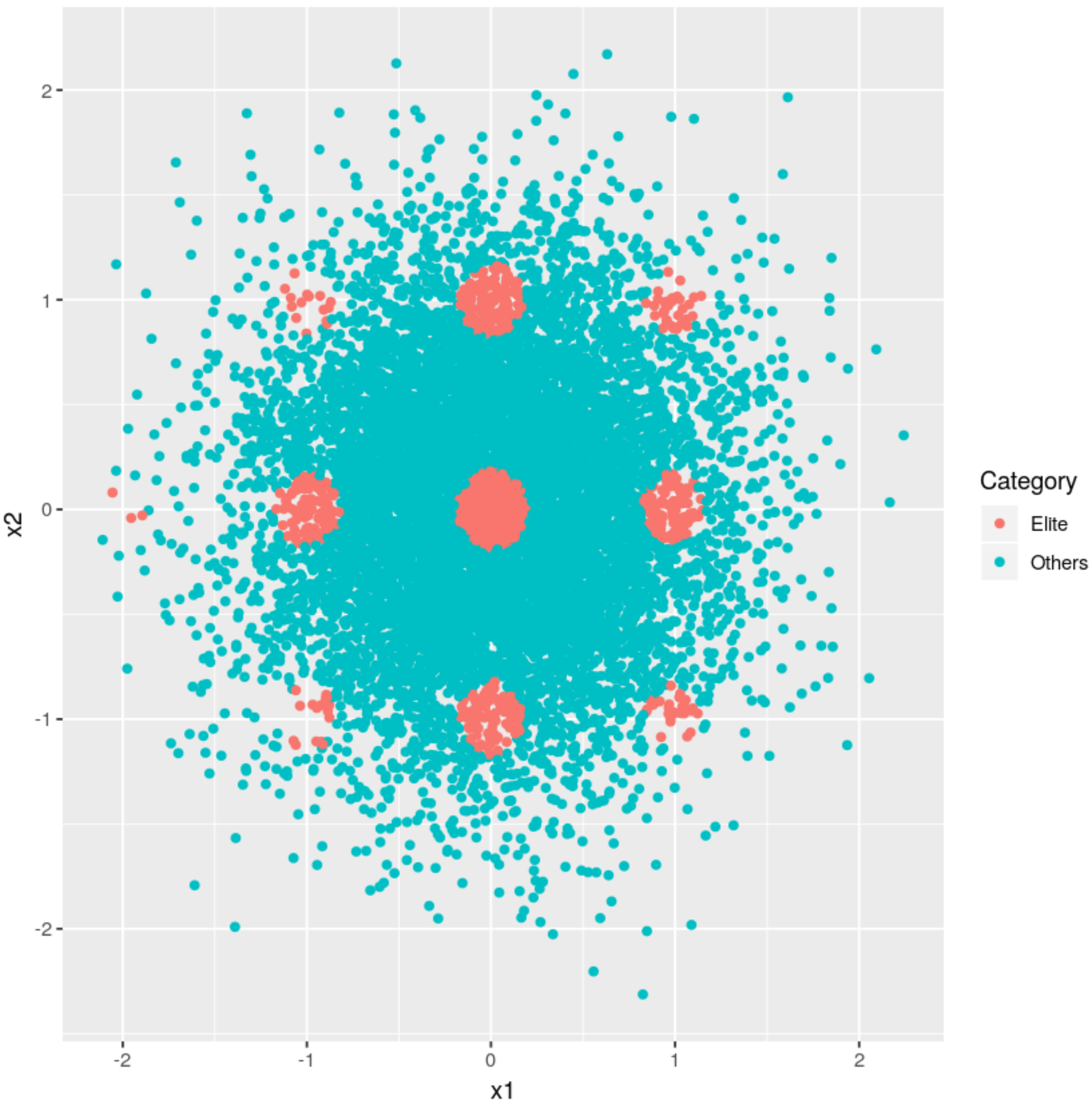

```
Iteration:  3        Best value:  0.0534268              Max(mu):  0.2886972         Max(sigma):  13
```



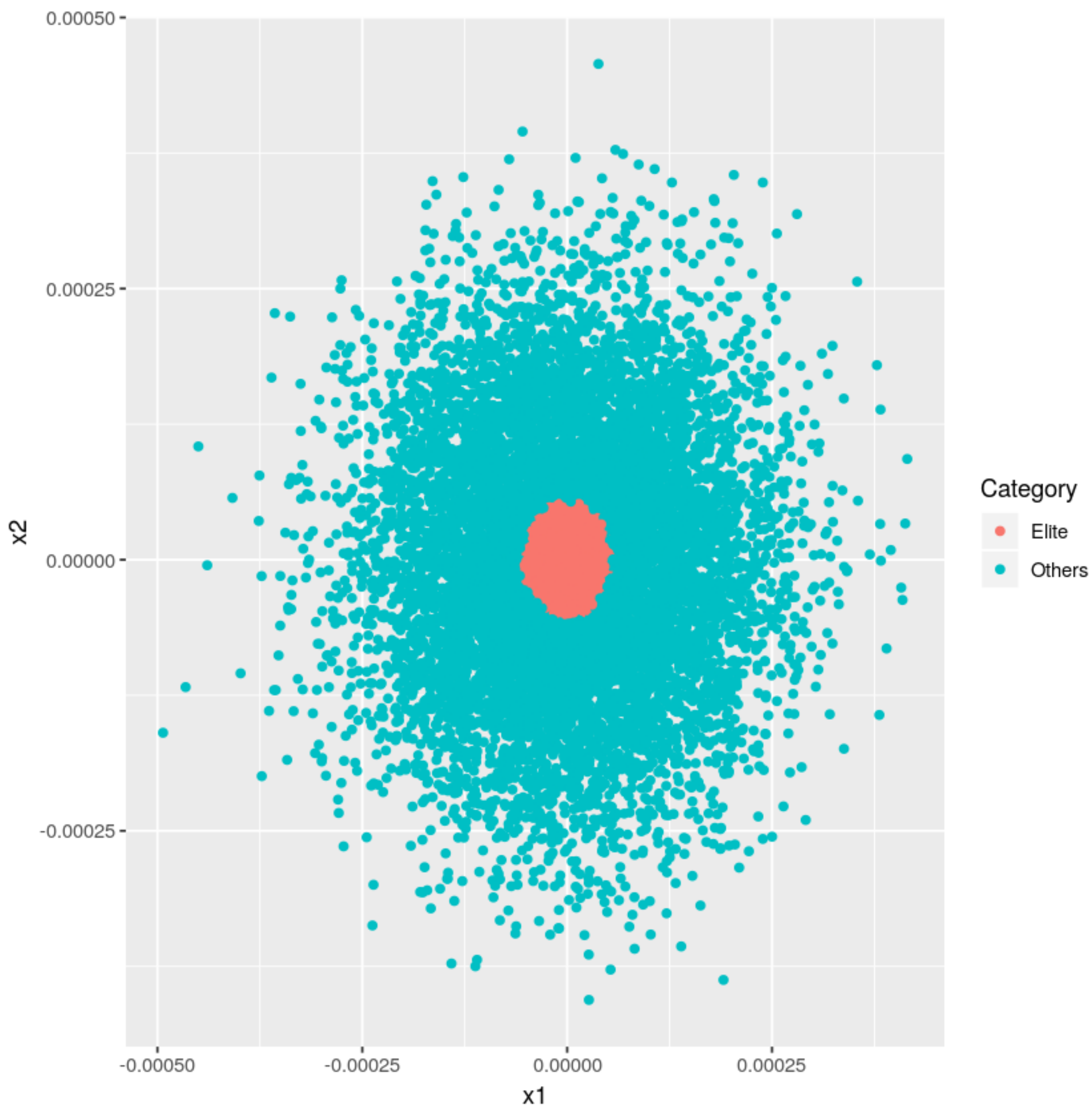

```
Iteration:  4        Best value:  0.002592227             Max(mu):  0.05956493       Max(sigma):
Iteration:  5        Best value:  0.002936498             Max(mu):  0.02623596       Max(sigma):
Iteration:  6        Best value:  0.000177803             Max(mu):  0.01264946       Max(sigma):
Iteration:  7        Best value:  3.184571e-05             Max(mu):  0.006256309       Max(sigma
Iteration:  8        Best value:  0.0003860067             Max(mu):  0.005300794       Max(sigma
Iteration:  9        Best value:  0.0004497498             Max(mu):  0.0033962       Max(sigma):
```

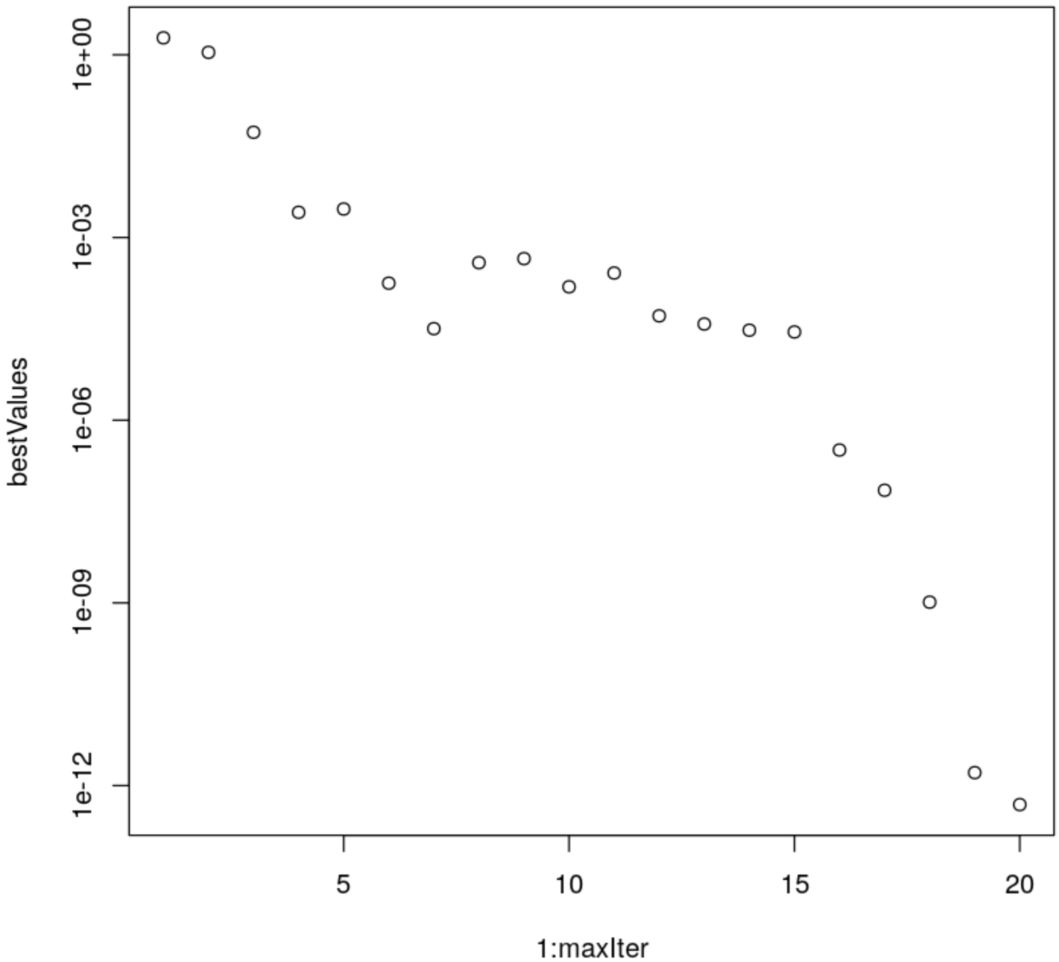



```
Iteration:  10        Best value:  0.0001549733              Max(mu):  0.004923299        Max(sigma
Iteration:  11        Best value:  0.0002611206              Max(mu):  0.005293373        Max(sigma
Iteration:  12        Best value:  5.168094e-05              Max(mu):  0.004298656        Max(sigma
Iteration:  13        Best value:  3.796178e-05              Max(mu):  0.00240521        Max(sigma)
Iteration:  14        Best value:  3.004398e-05              Max(mu):  0.001145529        Max(sigma
Iteration:  15        Best value:  2.811228e-05              Max(mu):  0.0004379217        Max(sigm
Iteration:  16        Best value:  3.238833e-07              Max(mu):  7.360588e-05        Max(sigm
Iteration:  17        Best value:  7.06562e-08              Max(mu):  8.785794e-06        Max(sigma
Iteration:  18        Best value:  1.028054e-09              Max(mu):  5.739931e-07        Max(sig
Iteration:  19        Best value:  1.630696e-12              Max(mu):  5.687833e-08        Max(sig
```




```
Iteration:  20        Best value:  4.867218e-13              Max(mu):  1.168593e-07          Max(sig
```


0.00050
0.00025
0.00000
-0.00025
x2
-0.00050
-0.00025
0.00000
0.00025
x1
Category
Elite
Others




```
In [4]: bestValues
```

1. 1.90064338916066 2. 1.09807799737755 3. 0.0534267970766731 4. 0.00259222739365761 5. 0.00293649829586329 6. 0.000177802981061959 7. 3.18457131260885e-05 8. 0.000386006712775 9. 0.000449749756164408 10. 0.000154973292351812 11. 0.000261120617020438 12. 5.16809429953469e-05 13. 3.79617845194247e-05 14. 3.00439844309608e-05 15. 2.81122816474522e-05 16. 3.23883327268959e-07 17. 7.06561991137278e-08 18. 1.02805408630502e-09 19. 1.63069557856943e-12 20. 4.86721773995669e-13