ISLR_CH10

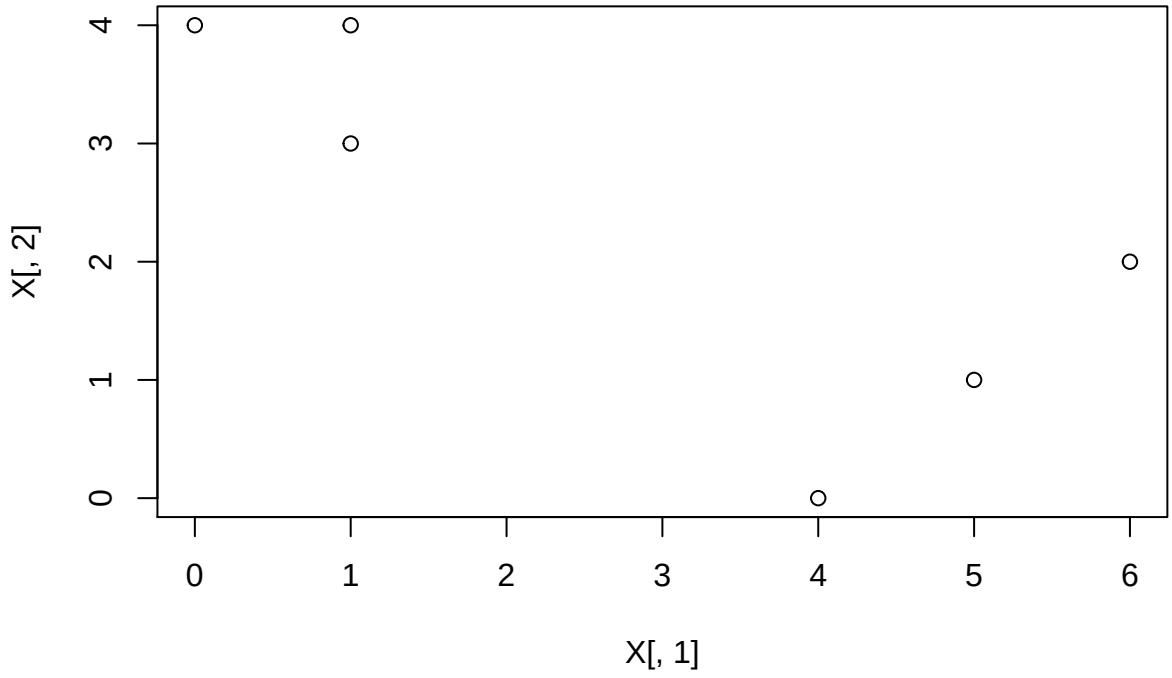
Sky Liu

3/28/2019

10.7.3

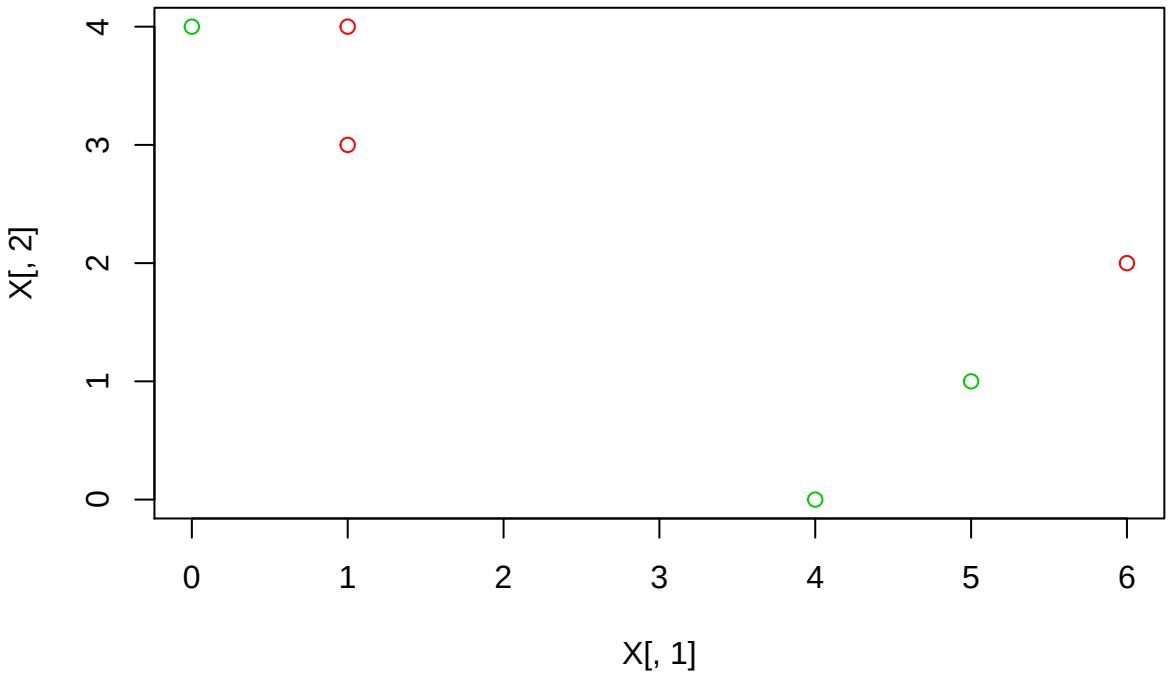
part a

```
set.seed(1)
X = cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
plot(X[,1], X[,2])
```



part b

```
labels = sample(1:2, nrow(X), replace=T)
plot(X[,1], X[,2],col = (labels + 1))
```



###

part c-e

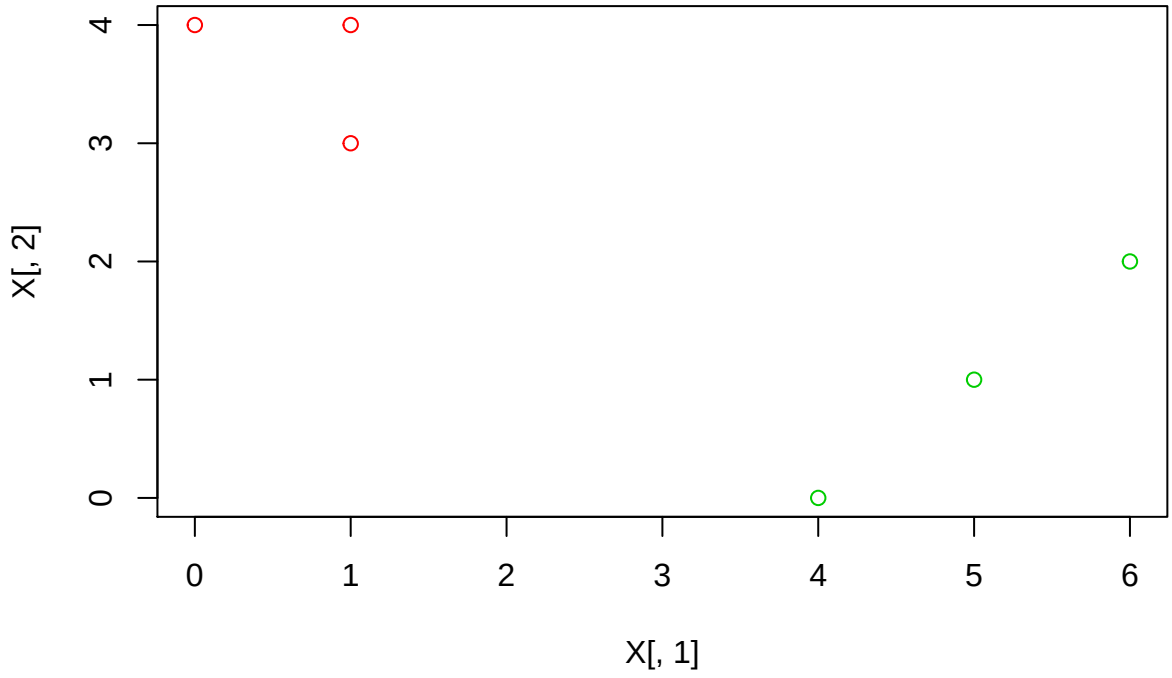
```
while (TRUE) {
    # part c
    centroid <- matrix(nrow=2,ncol=2)
    for (i in 1:2) {
        samples <- labels == i
        centroid[i, ] <- apply(X[samples, ], 2, mean)
    }

    # part d
    new_labels <- rep(NA, nrow(X))
    for (j in 1:nrow(X)) {
        smallest_norm <- +Inf
        for (i in 1:2) {
            nm <- norm(as.matrix(X[j, ] - centroid[i, ]), type = "2")
            if (nm < smallest_norm) {
                smallest_norm <- nm
                new_labels[j] <- i
            }
        }
    }

    # part e
    if (sum(new_labels == labels) == nrow(X)) {
        break
    } else {
        labels <- new_labels
    }
}
```

part f

```
plot(X[,1], X[,2],col = (labels + 1))
```



10.7.5

a) People bought more number of items (socks and computers):1,2,7,8; people bought few number of items (socks and computers):3,4,5,6;
b) Higher standardized purchase ability: 5,6,7,8; lower standardized purchase ability: 1,2,3,4;
c) People spent more money: 5,6,7,8; people spent less money: 1,2,3,4

10.7.6

part a

90% of information is lost during the processing of projecting the original sample to the first principle component.

part b

It is obvious that there is a clear spliting of the dataset. The better way to think about the problem is to assign the machine A/B as an additional attribute of the data.

part c

```
set.seed(3)
Control1 = matrix(rnorm(50*1000), ncol=50)
Treatment1 = matrix(rnorm(50*1000), ncol=50)
```

```
d1 = cbind(Control1, Treatment1)
d1[1,] = seq(-20, 20 - .4, .4)
pr.out1 = prcomp(scale(d1))
summary(pr.out1)$importance[,1]
```

```
##     Standard deviation Proportion of Variance  Cumulative Proportion
##               3.403605               0.115850               0.115850
```

```
d2 = rbind(d1, c(rep(1, 50), rep(100, 50)))
pr.out2 = prcomp(scale(d2))
summary(pr.out2)$importance[,1]
```

```
##     Standard deviation Proportion of Variance  Cumulative Proportion
##               6.741555               0.454490               0.454490
```

Having AB machine coded as 1 and 100 other than some liner relationship y=-20+0.4x, the variance explained by the first principle component increases dramatically.

10.7.8

part a

```
set.seed(4)
pr.out = prcomp(USArrests, center=T, scale=T)
pr.var = pr.out$sdev^2
pve1 = pr.var / sum(pr.var)
pve1
```

```
## [1] 0.62006039 0.24744129 0.08914080 0.04335752
```

part b

```
n <- apply((scale(USArrests) %*% pr.out$rotation)^2, 2, sum)
d <- sum(apply(scale(USArrests)^2, 2, sum))
pve2 <- n/d
rbind(pve1,pve2,"pve1-pve2"=pve1-pve2)
```

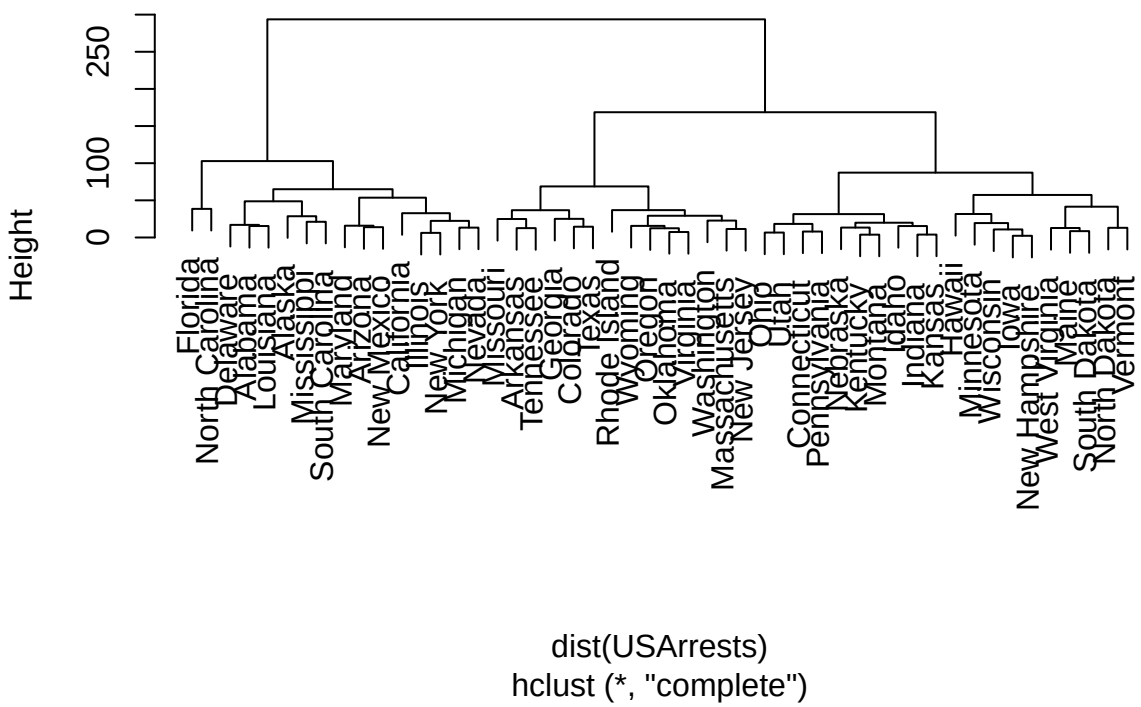
```
##                      PC1           PC2           PC3        PC4
## pve1       6.200604e-01  2.474413e-01  8.914080e-02 0.04335752
## pve2       6.200604e-01  2.474413e-01  8.914080e-02 0.04335752
## pve1-pve2 -1.110223e-16 -2.498002e-16 -4.163336e-17 0.00000000
```

The differences between pve1 and pve2 are almost zero

10.7.9

part a

```
set.seed(5)
hc_complete = hclust(dist(USArrests), method="complete")
plot(hc_complete)
```



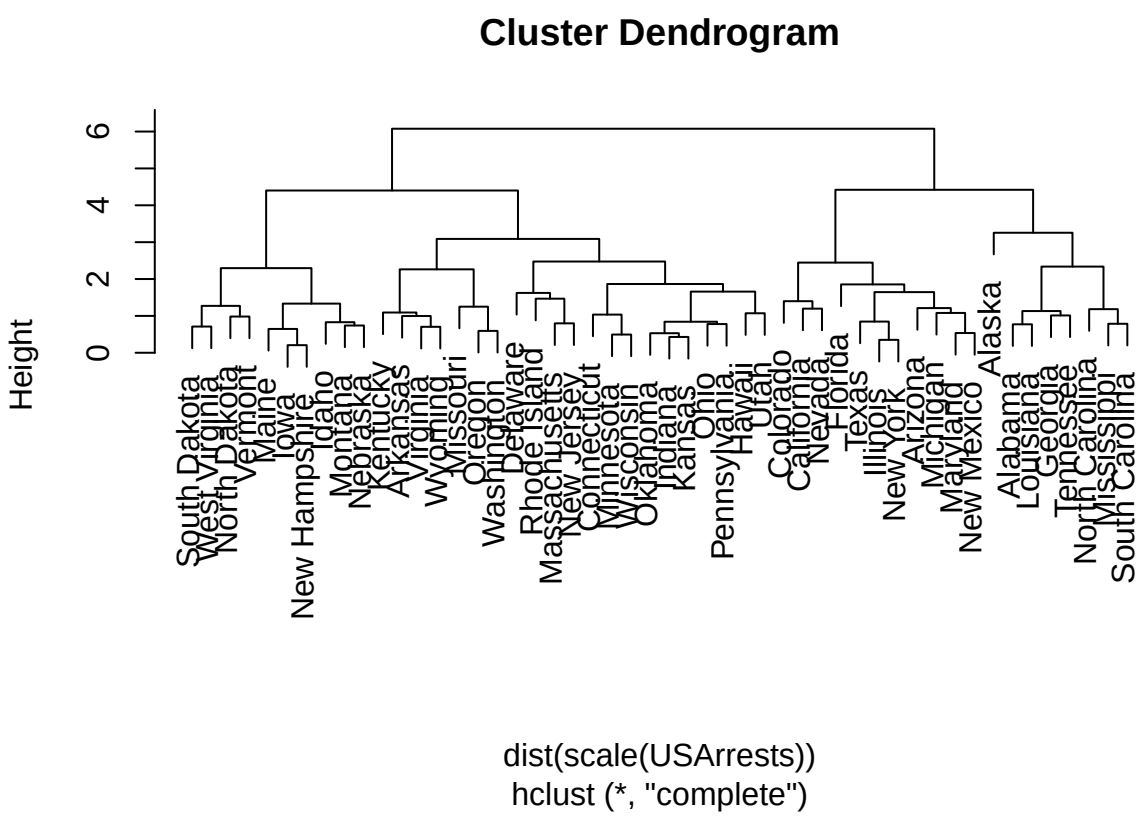
part b

```
cutree(hc_complete, 3)
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              1              2              1
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              1              1              2
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              1              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              1
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              2              1              3              1              2
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              3              3              1              3              2
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              1              1              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              2              2              3              2              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              2              2              3              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              2              2              3              3              2
```

part c

```
hc_complete_scaled = hclust(dist(scale(USArrests)), method="complete")
plot(hc_complete_scaled)
```




```
cutree(hc_complete_scaled, 3)
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              2              3              2
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              3              2              1
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              2              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              2
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              3              2              3              1              3
##        Montana       Nebraska         Nevada New Hampshire     New Jersey
##              3              3              2              3              3
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              2              2              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              3              3              3              3              1
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              1              2              3              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              3              3              3              3              3
```

part d

From this example, scaling reduced the height of the dendogram obtained from hierarchical clustering.

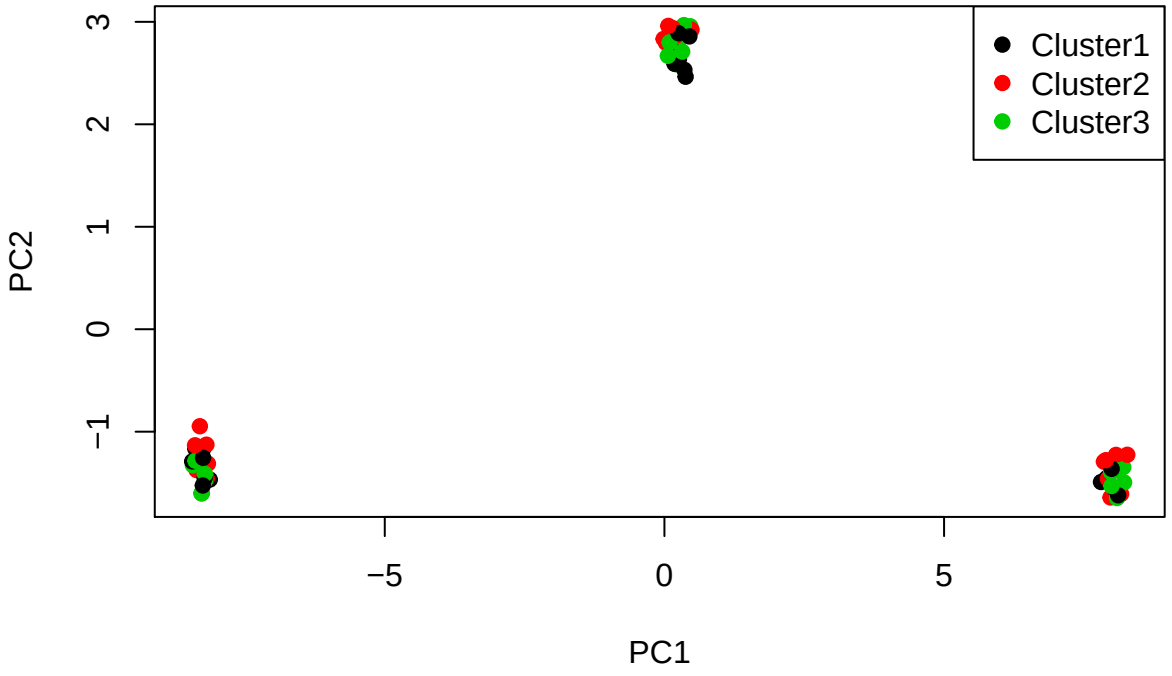
10.7.10

part a

```
set.seed(7)
D <-  matrix(rnorm(20*3*50, mean=0, sd=0.1), ncol=50)
for ( i in 1:50 ) {
  set.seed(i)
  D[1:20, i] = D[1:20, i] + runif(1, 0, 3)
  D[41:60, i] = D[41:60, i] - runif(1, 0, 3)
}
```

part b

```
pca.out = prcomp(D, scale = T)
plot(pca.out$x[, 1:2], col=1:3, pch = 19)
legend("topright", legend = c("Cluster1", "Cluster2", "Cluster3"), pch = 19, col = unique(1:3))
```



###

part c

```
km.out = kmeans(D, 3, nstart=20)
table(km.out$cluster, c(rep(1,20), rep(2,20), rep(3,20)))
```

```
##
##      1  2  3
##   1 20  0  0
##   2  0  0 20
```

```
##   3  0 20  0
```

```
km.out$cluster
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3
## [36] 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

part d

```
km.out = kmeans(D, 2, nstart=20)
km.out$cluster
```

```
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [36] 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

All previous 1,3 become 2

part e

```
km.out = kmeans(D, 4, nstart=20)
km.out$cluster
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 2 2 2 2 2 2 4 4 4 4
## [36] 2 2 2 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

All previous 2 split into 1,2,4

part f

```
km.out = kmeans(pca.out$x[,1:2], 3, nstart=20)
table(km.out$cluster, c(rep(1,20), rep(2,20), rep(3,20)))
```

```
##
##      1  2  3
##   1  0  0 20
##   2 20  0  0
##   3  0 20  0
```

```
km.out$cluster
```

```
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3
## [36] 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Matches perfectly like in part c

part g

```
km.out = kmeans(scale(D), 3, nstart=20)
table(km.out$cluster, c(rep(1,20), rep(2,20), rep(3,20)))
```

```
##
##      1  2  3
##   1  0 20  0
##   2 20  0  0
```

```
##   3  0  0 20
```

```
km.out$cluster
```

```
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [36] 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

After scaling, the result is the same