Unsupervised_learning-HW

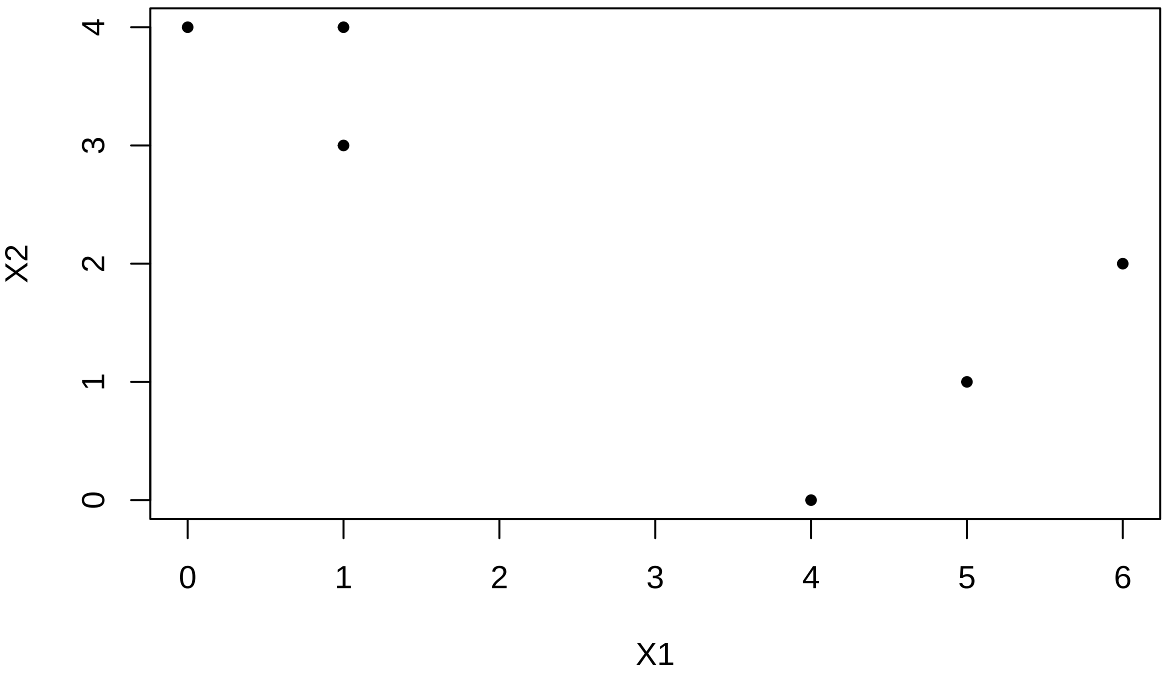
Franky Zhang

3/30/2022

12.3

(a)

```
X1 <- c(1, 1, 0, 5, 6, 4)
X2 <- c(4, 3, 4, 1, 2, 0)
dat <- cbind(X1, X2)
plot(dat, pch = 20)
```



(b)

```
set.seed(1108)
label <- sample(c(1, 2), 6, replace = TRUE)
label
```

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

(c)

```r
center1 <- apply(dat[which(label == 1), ], 2, mean)
center2 <- apply(dat[which(label == 2), ], 2, mean)
centroid <- rbind(center1, center2)
```

(d)

```r
dist <- Euclidean(vec1 = dat, vec2 = centroid)
label <- ifelse(dist[, 1] > dist[, 2], 1, 2)
```

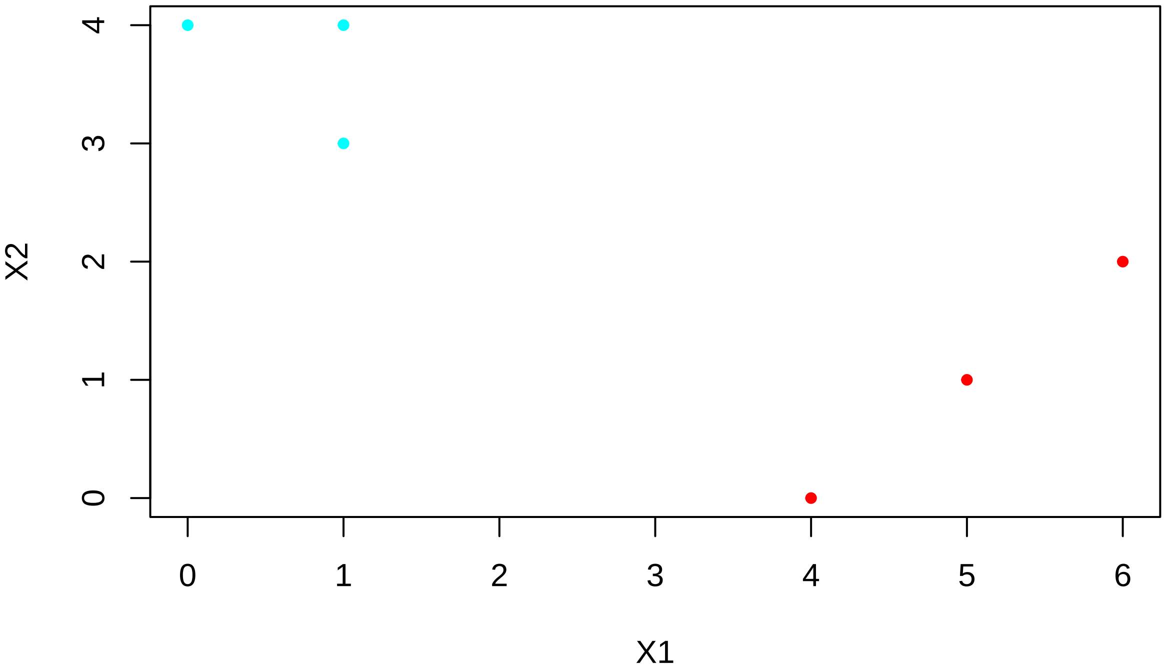
(e)

```r
set.seed(1108)
label <- sample(c(1, 2), 6, replace = TRUE)
iter = 10
for (i in 1:iter) {
  cat("iteration", i ,": ", label, "\n")
  centroid <- rbind(apply(dat[which(label == 1), ], 2, mean),
                    apply(dat[which(label == 2), ], 2, mean))
  dist <- Euclidean(vec1 = dat, vec2 = centroid)
  label <- ifelse(dist[, 1] > dist[, 2], 2, 1)
}
```

```
## iteration 1 :  1 1 2 2 1 1
## iteration 2 :  2 2 2 1 1 1
## iteration 3 :  2 2 2 1 1 1
## iteration 4 :  2 2 2 1 1 1
## iteration 5 :  2 2 2 1 1 1
## iteration 6 :  2 2 2 1 1 1
## iteration 7 :  2 2 2 1 1 1
## iteration 8 :  2 2 2 1 1 1
## iteration 9 :  2 2 2 1 1 1
## iteration 10 :  2 2 2 1 1 1
```

(f)

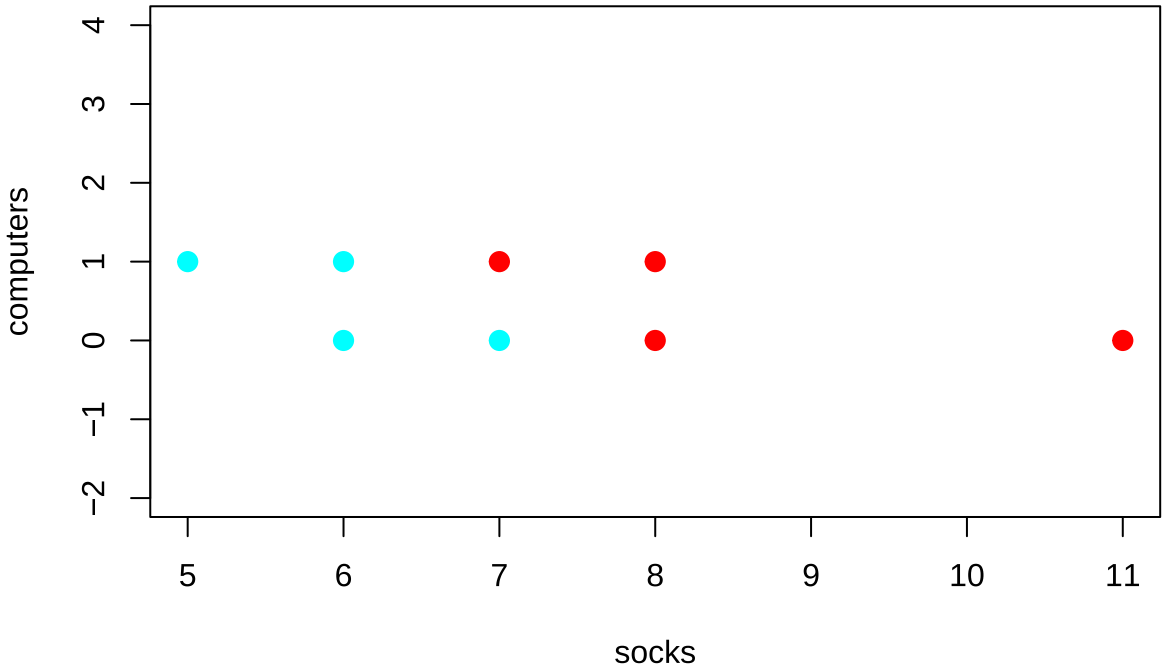
```r
plot(dat, pch = 20, col = Cols(label),
     xlab = "X1", ylab = "X2")
```



12.5

(1)

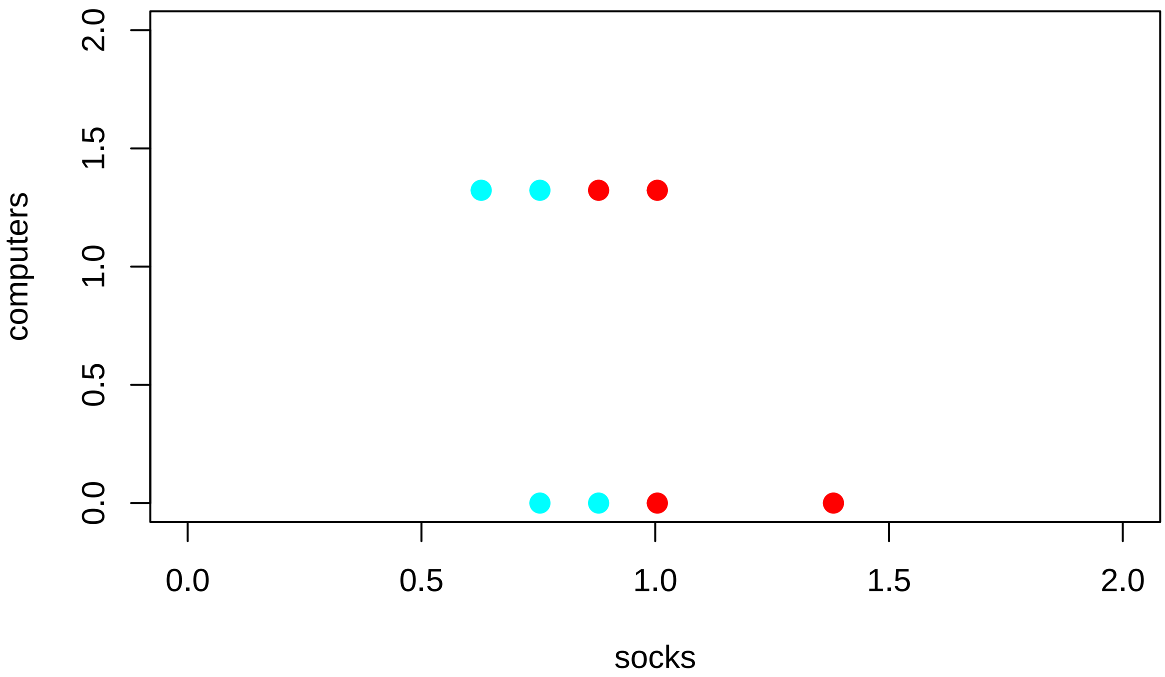
```
socks <- c(8, 11, 7, 6, 5, 6, 7, 8)
computers <- c(0, 0, 0, 0, 1, 1, 1, 1)
dat <- cbind(socks, computers)
labels <- c(1, 1, 2, 2, 2, 2, 1, 1)
plot(dat, col = Cols(labels), pch = 20, cex = 2,
     xlab = "socks", ylab = "computers", xlim = c(5, 11), ylim = c(-2, 4))
```



If take the original data into consideration, the number of socks performs to be a more important factor when clustering the data. I expect to see one cluster with less socks (obs: 3, 4, 5, 6) and another with more socks (obs: 1, 2, 7, 8)

(2)

```
socks     <- scale(c(8, 11, 7, 6, 5, 6, 7, 8), center = FALSE)
computers <- scale(c(0, 0, 0, 0, 1, 1, 1, 1) , center = FALSE)
dat <- cbind(socks, computers)
labels <- c(1, 1, 2, 2, 2, 2, 1, 1)
plot(dat, col = Cols(labels), pch = 20, cex = 2,
     xlab = "socks", ylab = "computers", xlim = c(0, 2), ylim = c(0, 2))
```



If take the centered data without scale, we prefer to cluster by the number of computers – group 1: obs 1, 2, 3, 4; group 2: obs 5, 6, 7, 8.

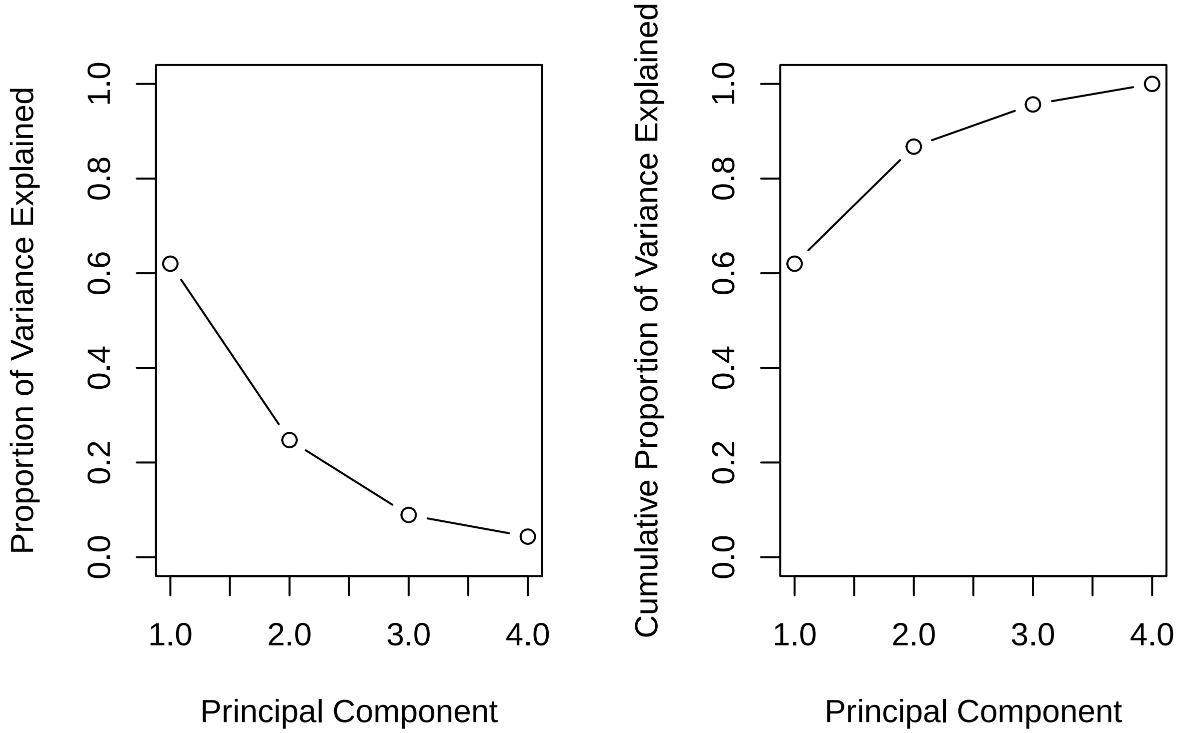
(3)

obviously, if we do the clustering by the number of dollars spent, the factor socks will be ignored and the clustering totally depends on the factor computer.

12.8

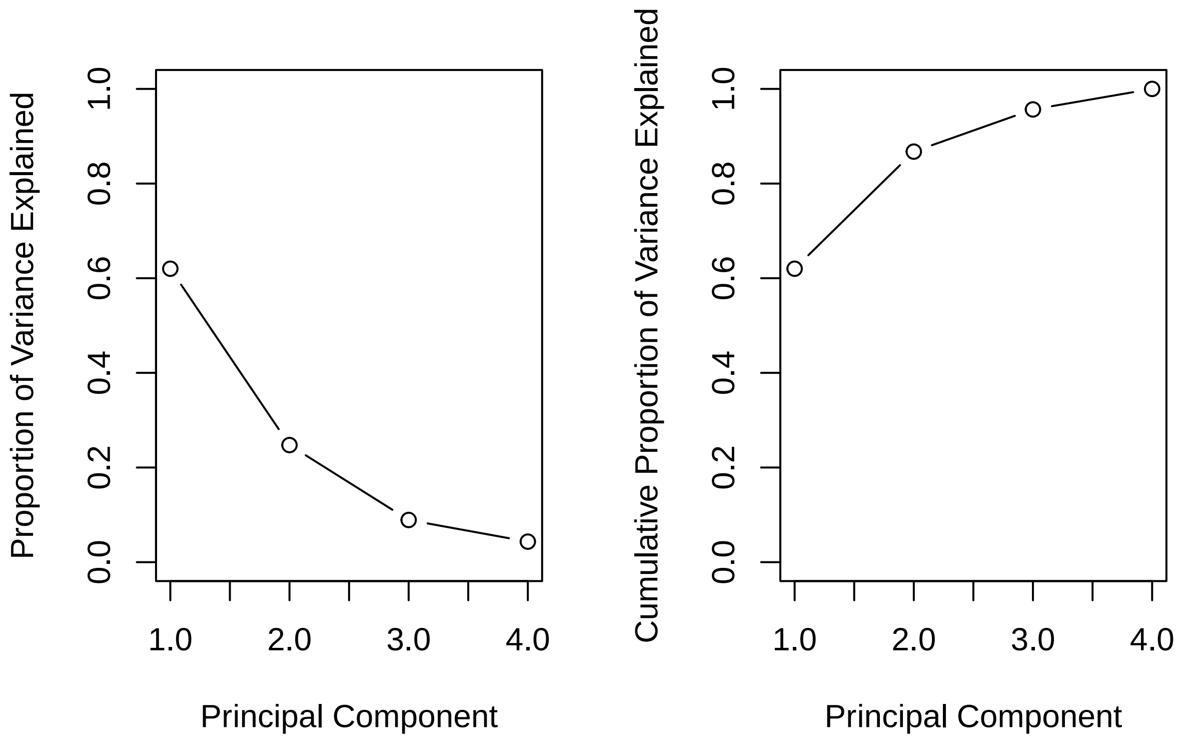
(a)

```
pr.out <- prcomp(USArrests, scale = TRUE)
pr.var <- pr.out$sdev^2
pve <- pr.var / sum(pr.var)
par(mfrow = c(1, 2))
plot(pve, xlab = "Principal Component",
    ylab = "Proportion of Variance Explained", ylim = c(0, 1),
    type = "b")
plot(cumsum(pve), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained", ylim = c(0, 1), type = "b")
```



(b)

```
dat <- scale(USArrests)
# apply(dat, 2, mean)
# apply(dat, 2, var)
z_score <- dat %*% pr.out$rotation
pve <- apply(z_score, 2, var)/sum(apply(dat, 2, var)) # eq 12.10
# sum(pve)
par(mfrow = c(1, 2))
plot(pve, xlab = "Principal Component",
    ylab = "Proportion of Variance Explained", ylim = c(0, 1),
    type = "b")
plot(cumsum(pve), xlab = "Principal Component",
     ylab = "Cumulative Proportion of Variance Explained", ylim = c(0, 1), type = "b")
```



12.9

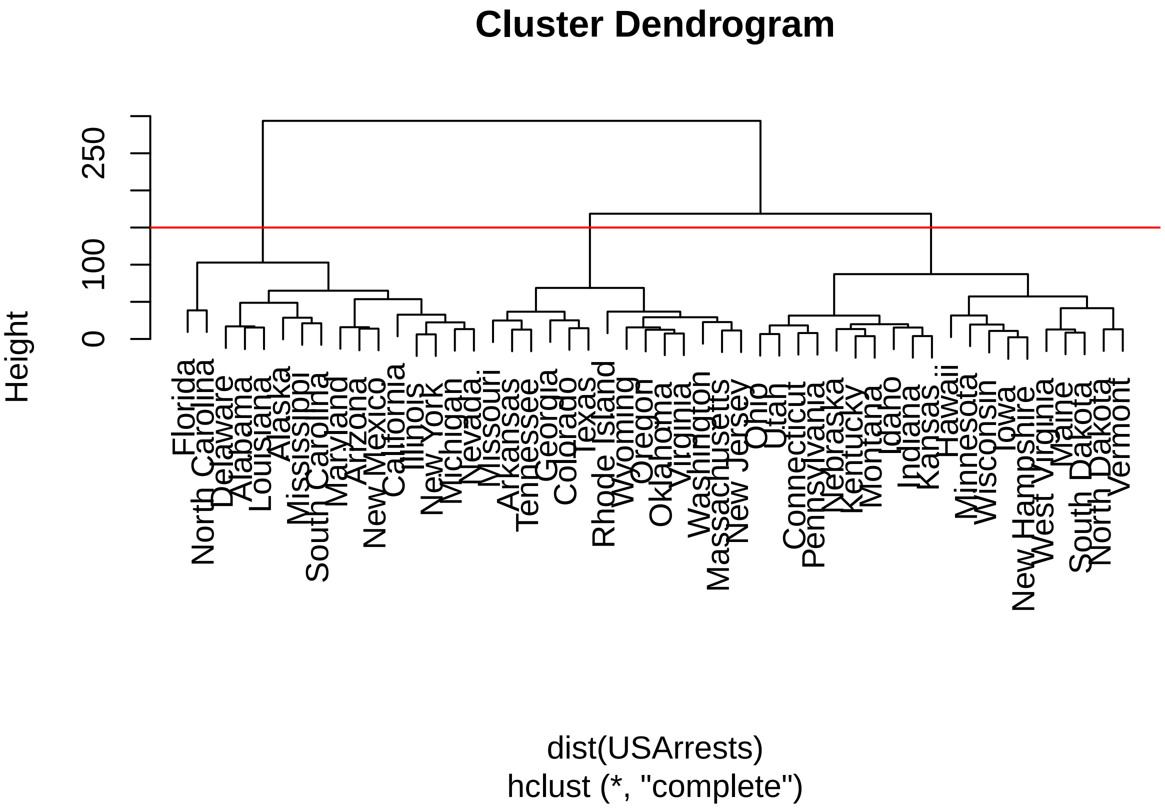
(a)

```
hc.complete <- hclust(dist(USArrests), method = "complete") # Euclidean distance, complete linkage
```

(b)

Cut at the height 150, and the clustering results are as follow:

```
hc.cluster <- cutree(hc.complete, 3)
plot(hc.complete)
abline(h = 150, col = "red")
```



```
hc.cluster
```

```
##        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
```

(c)

```r
hc.complete.scaled <- hclust(dist(dat), method = "complete") # centered and scaled data, Euclidean dist
```

(d)

```r
table(cutree(hc.complete, 3), cutree(hc.complete.scaled, 3))
```

```
##
##      1  2  3
##   1  6  9  1
##   2  2  2 10
##   3  0  0 20
```

after scaling the data, complete linkage hierarchical clustering assigns more obs to 3rd cluster. To my consideration, the variables should be scaled beforehand because variables are in different units.

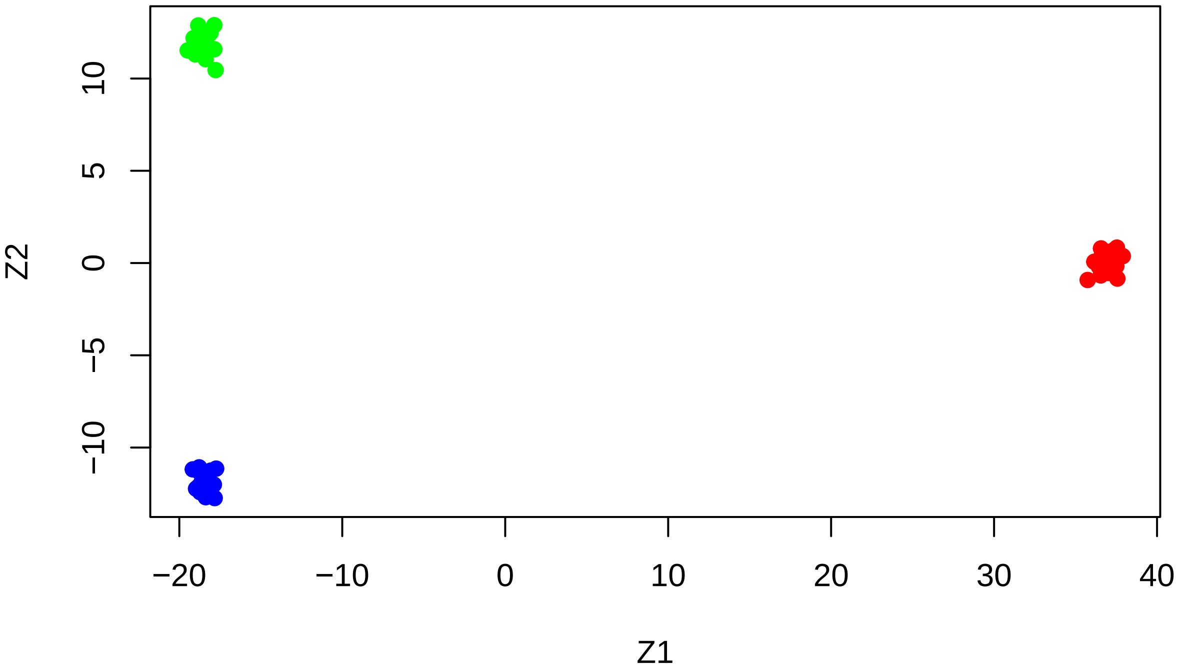
12.10

(a)

```r
set.seed(1754)
dat <- matrix(rnorm(20*3*50, sd = .5), ncol = 50)
dat[1:20, 1] <- dat[1:20, 1] + 2
dat[1:20, 2] <- dat[1:20, 2] + 13
dat[21:40, 1] <- dat[21:40, 1] + 22
# dat[21:40, 2] <- dat[21:40, 2] + 33
dat[41:60, 1] <- dat[41:60, 1] + 42
dat[41:60, 2] <- dat[41:60, 2] + 53
labels <- c(rep(3, 20), rep(2, 20), rep(1, 20))
```

(b)

```r
pr.out <- prcomp(dat)
plot(pr.out$x[, 1:2], col = Cols(labels),
     xlab = "Z1", ylab = "Z2", pch = 19)
```



(c)

```
set.seed(2)
km.out <- kmeans(dat, 3, nstart = 20)
table(labels, km.out$cluster)
```

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

the observations are perfectly clustered

(d)

```
set.seed(2)
km.out <- kmeans(dat, 2, nstart = 20)
table(labels, km.out$cluster)
```

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

(e)

```
set.seed(2)
km.out <- kmeans(dat, 4, nstart = 20)
table(labels, km.out$cluster)
```

```
##
## labels  1  2  3  4
##      1  0  0  0 20
##      2 20  0  0  0
##      3  0 12  8  0
```

(f)

```
set.seed(2)
km.out <- kmeans(pr.out$x[, 1:2], 3, nstart = 20)
table(labels, km.out$cluster)
```

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

still perfectly clustered

(g)

```
scales.dat <- scale(dat)
set.seed(2)
km.out <- kmeans(scales.dat, 3, nstart = 20)
table(labels, km.out$cluster)
```

```
##
## labels  1  2  3
##      1  0  6 14
##      2  5 14  1
##      3  8 11  1
```

the result is worse since differences are mitigated after scaling and hard to cluster.