Week 8 Assignment

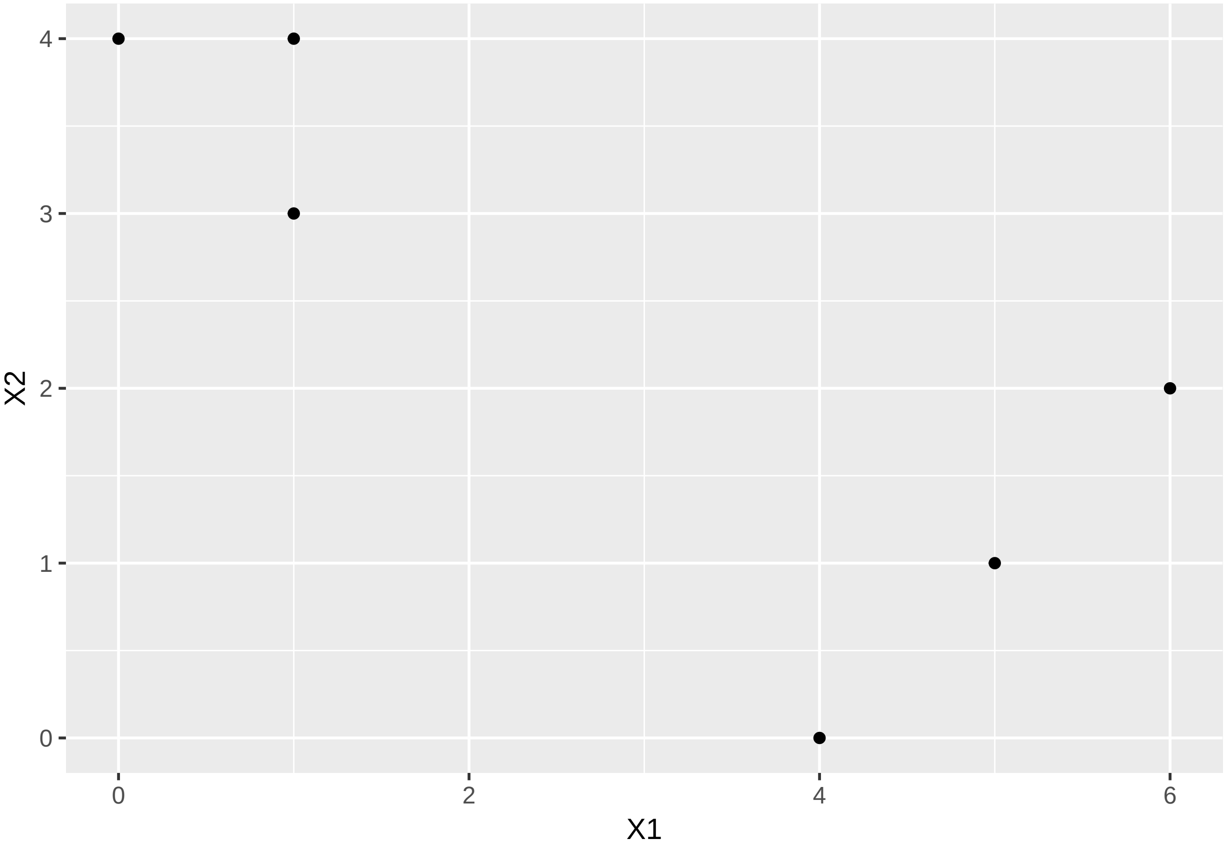
David Russo

3/9/2017

10.7 #3

- a)

```
obs <- data.frame(
  X1 = c(1, 1, 0, 5, 6, 4),
  X2 = c(4, 3, 4, 1, 2, 0)
)

obs %>%
  ggplot(aes(x = X1, y = X2)) +
  geom_point()
```



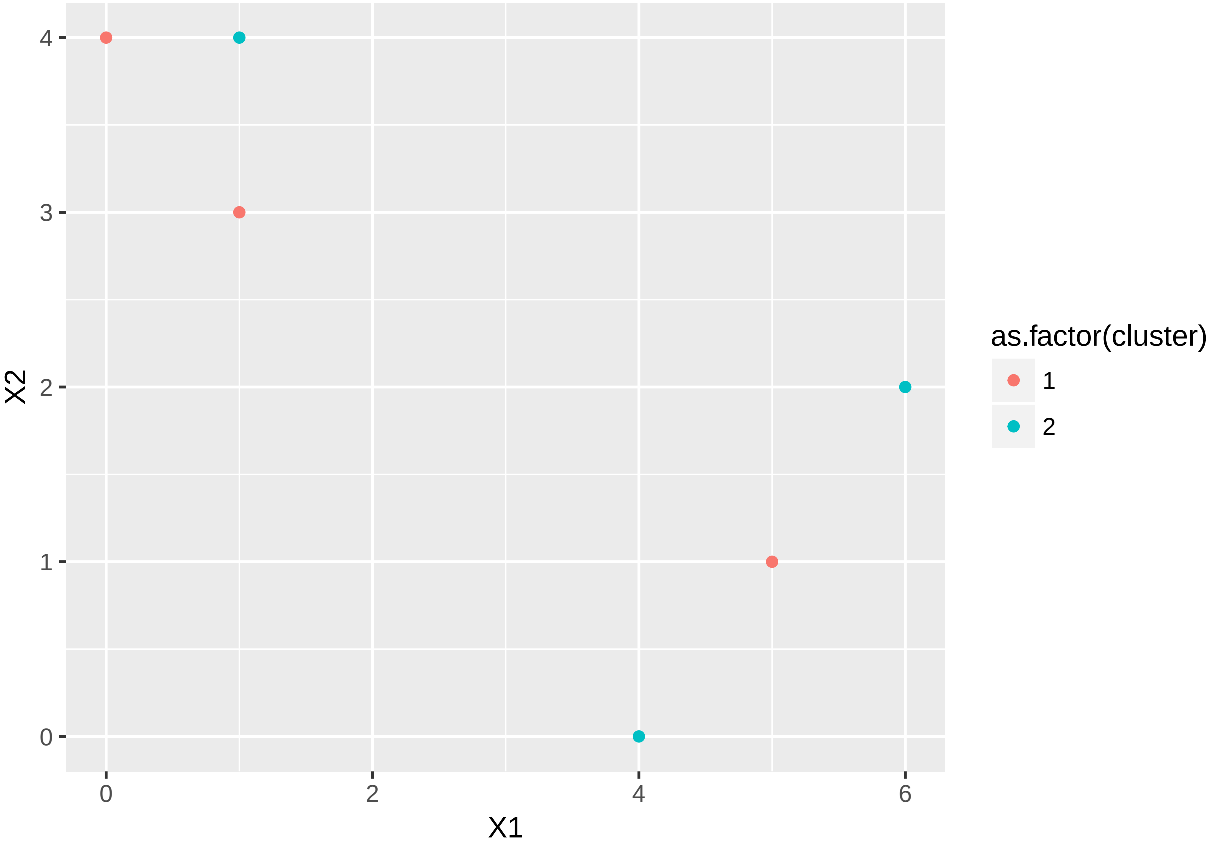
- b)

```
# set random number seed for replicable results
set.seed(2017)

# assign the labels
obs$cluster <- sample(rep(c(1, 2), each = 3), 6, replace = FALSE)
```

```
# print the labels
obs
```

```
##   X1 X2 cluster
## 1  1  4       2
## 2  1  3       1
## 3  0  4       1
## 4  5  1       1
## 5  6  2       2
## 6  4  0       2
```

```
# plot the labels
obs %>%
  dplyr::select(X1, X2, cluster) %>%
  ggplot(aes(x = X1, y = X2, color = as.factor(cluster))) +
  geom_point()
```



- c)

```
centroids <-
  obs %>%
  dplyr::group_by(cluster) %>%
  dplyr::summarise(centroid_X1 = round(mean(X1), 2), centroid_X2 = round(mean(X2), 2)) %>%
  as.data.frame()

centroids
```

```
##   cluster centroid_X1 centroid_X2
## 1       1        2.00        2.67
## 2       2        3.67        2.00
```

- d)

```
# create distance from cluster1 centroid
obs$dist_from_cluster1_centroid <- round(
  sqrt((obs$X1 - centroids$centroid_X1[centroids$cluster == 1])^2 +
       (obs$X2 - centroids$centroid_X2[centroids$cluster == 1])^2), 4)

# create distance from cluster2 centroid
obs$dist_from_cluster2_centroid <- round(
  sqrt((obs$X1 - centroids$centroid_X1[centroids$cluster == 2])^2 +
       (obs$X2 - centroids$centroid_X2[centroids$cluster == 2])^2), 4)

obs$new_cluster <-
  ifelse(obs$dist_from_cluster1_centroid <= obs$dist_from_cluster2_centroid, 1, 2)

obs
```

```
##   X1 X2 cluster dist_from_cluster1_centroid dist_from_cluster2_centroid
## 1  1  4       2                      1.6640                      3.3360
## 2  1  3       1                      1.0530                      2.8511
## 3  0  4       1                      2.4019                      4.1796
## 4  5  1       1                      3.4335                      1.6640
## 5  6  2       2                      4.0557                      2.3300
## 6  4  0       2                      3.3360                      2.0270
##   new_cluster
## 1           1
## 2           1
## 3           1
## 4           2
## 5           2
## 6           2
```

- e)

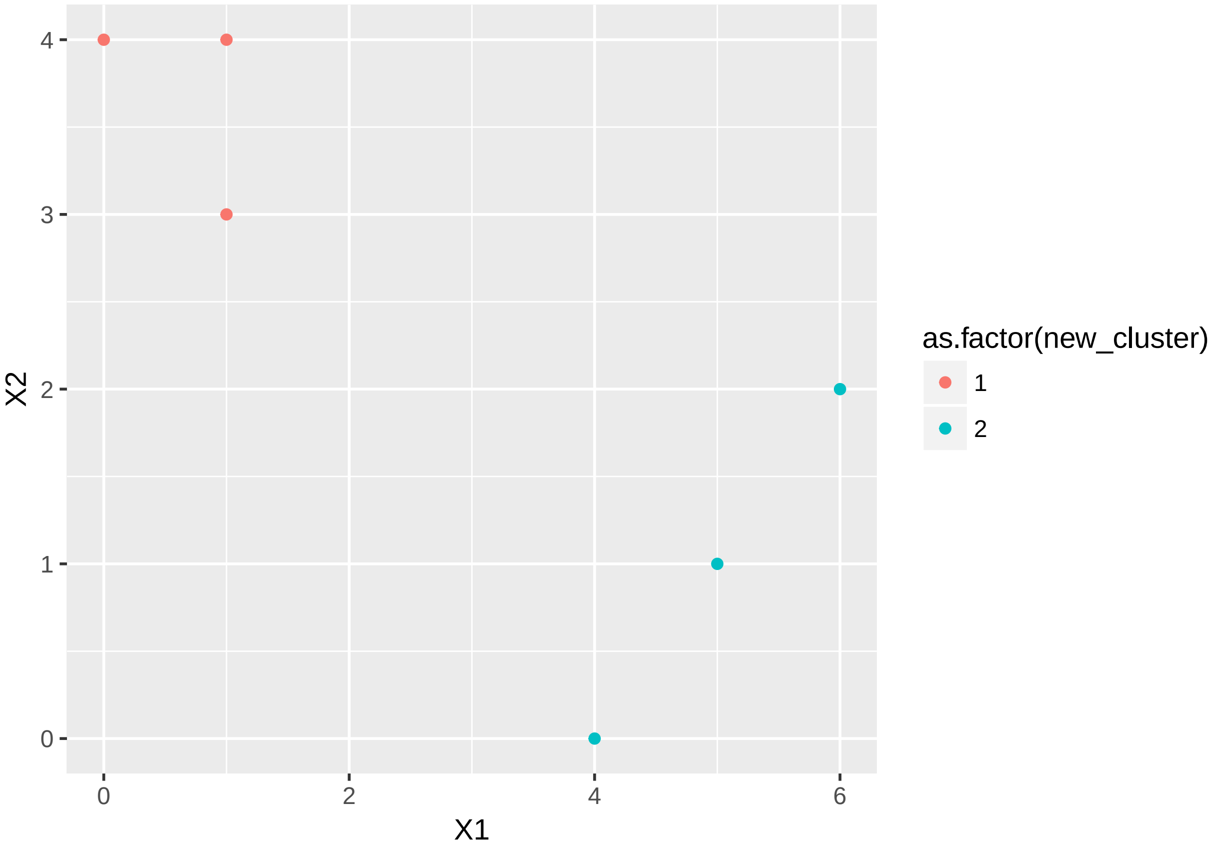
```
cluster_difference <- FALSE

while(cluster_difference == FALSE){

# reset obs$cluster to obs$new_cluster
obs$cluster <- obs$new_cluster

# create distance from cluster1 centroid
obs$dist_from_cluster1_centroid <- round(
  sqrt((obs$X1 - centroids$centroid_X1[centroids$cluster == 1])^2 +
       (obs$X2 - centroids$centroid_X2[centroids$cluster == 1])^2), 4)

# create distance from cluster2 centroid
obs$dist_from_cluster2_centroid <- round(
  sqrt((obs$X1 - centroids$centroid_X1[centroids$cluster == 2])^2 +
       (obs$X2 - centroids$centroid_X2[centroids$cluster == 2])^2), 4)

obs$new_cluster <-
  ifelse(obs$dist_from_cluster1_centroid <= obs$dist_from_cluster2_centroid, 1, 2)
cluster_difference <- all(obs$cluster == obs$new_cluster)

print(obs)

}
```

```
##   X1 X2 cluster dist_from_cluster1_centroid dist_from_cluster2_centroid
## 1  1  4       1                      1.6640                      3.3360
## 2  1  3       1                      1.0530                      2.8511
## 3  0  4       1                      2.4019                      4.1796
## 4  5  1       2                      3.4335                      1.6640
## 5  6  2       2                      4.0557                      2.3300
## 6  4  0       2                      3.3360                      2.0270
##   new_cluster
## 1           1
## 2           1
## 3           1
## 4           2
## 5           2
## 6           2
```

The clustering only took one iteration until the algorithm converged.

- f)

```
obs %>%
  dplyr::select(X1, X2, new_cluster) %>%
  ggplot(aes(x = X1, y = X2, color = as.factor(new_cluster))) +
  geom_point()
```



10.7 #7

```
# load the data
data("USArrests")

# scale the data so that each variable has a mean of 0 and a variance of 1
usa_scaled <- as.data.frame(scale(USArrests))
# verify that the scaled data's variables all have mean 0 and variance 1
apply(usa_scaled, 2, function(x) round(c(mean(x), var(x)), 2))
```

```
##      Murder Assault UrbanPop Rape
## [1,]      0       0        0    0
## [2,]      1       1        1    1
```

```
# create data frame of combinations of rows of usa_scaled data set
# there are choose(50, 2) = 1225 ways to choose 2 observations out of 50
combs <- t(combn(nrow(usa_scaled), 2))
head(combs)
```

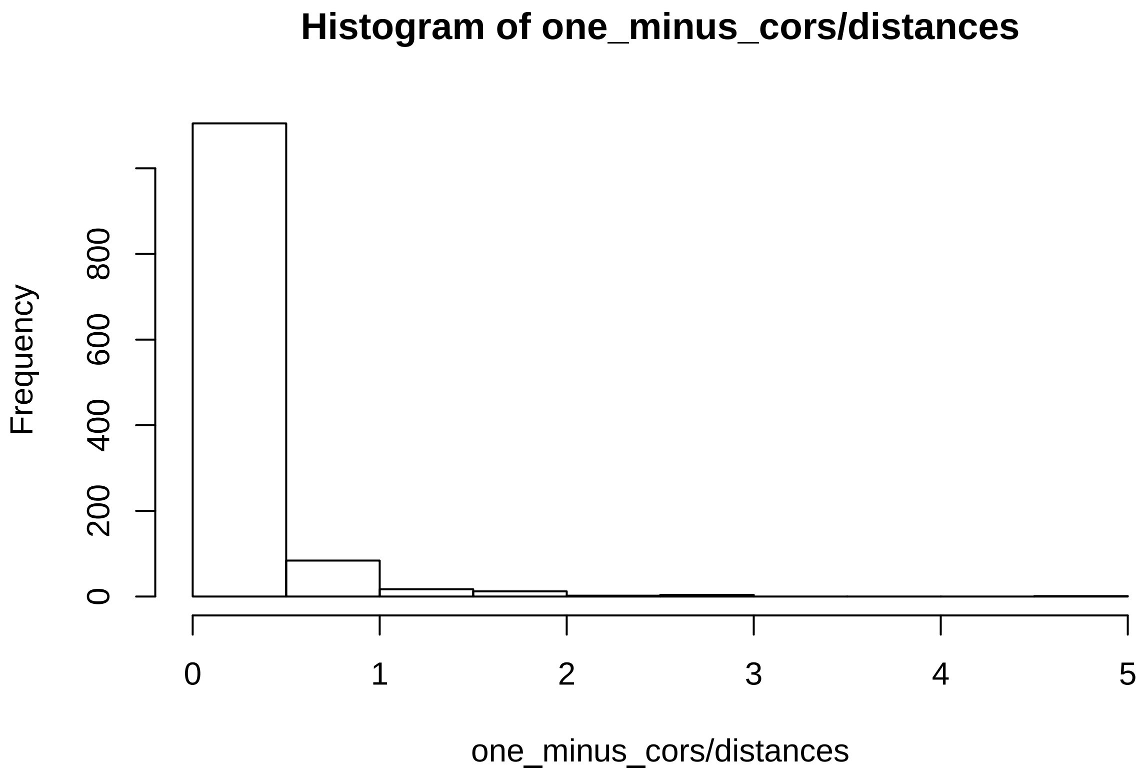
```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    3
## [3,]    1    4
## [4,]    1    5
## [5,]    1    6
## [6,]    1    7
```

```
###----------------- Determine Correlations ----------------------------
```

```r
# cycle through combs to get each correlation between rows
one_minus_cors <- rep(0, nrow(combs))

# create function for getting correlations between each observation
one_minus_cor_usa <- function(x){
  res <- cor(as.numeric(usa_scaled[combs[x, 1], ]),
             as.numeric(usa_scaled[combs[x, 2], ]))
  res <- 1 - res
  res
}

# determine 1 - r_ij for each pair of observations
one_minus_cors <- sapply(1:nrow(combs), function(x) one_minus_cor_usa(x))

###----------------- Determine Squared Euclidean Distances --------------

# cycle through combs to get distance between rows
distances <- dist(usa_scaled)^2

###----------------- Determine Proportionality -------------------
summary(one_minus_cors/distances)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000086 0.069140 0.133900 0.234200 0.262600 4.888000
```

```r
hist(one_minus_cors/distances)
```



As can be seen from the summary above, there is no evidence of proportionality. If there were proportionality, there would only be one value. I believe the wording of this question is either wrong or simply hard to understand. The question refers to distances and correlations between the i^{th} and j^{th} observations, which would lead me to believe it is referring to the rows of the data set (i.e., 50 rows; one per state). If the authors meant to refer to the distances and correlations between i^{th} and j^{th} variables (i.e., the 4 variables of Murder, Assault, UrbanPop, and Rape), then there is constant proportionality. This is shown below.

```r
# squared distance matrix
distance_matrix <- dist(t(usa_scaled))^2
distance_matrix
```

```
##            Murder  Assault UrbanPop
## Assault  19.41642
## UrbanPop 91.18188 72.63057
## Rape     42.76927 32.80636 57.68856
```

```r
# 1 - correlation matrix
one_minus_cor_mat <- as.dist(1-cor(usa_scaled))
one_minus_cor_mat
```

```
##             Murder   Assault  UrbanPop
## Assault  0.1981267
## UrbanPop 0.9304274 0.7411283
## Rape     0.4364212 0.3347588 0.5886588
```

```r
# demonstration of proportionality
distance_matrix/one_minus_cor_mat
```

```
##          Murder Assault UrbanPop
## Assault      98
## UrbanPop     98      98
## Rape         98      98       98
```