Week 9 Assignment

David Russo

3/12/2017

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)
```

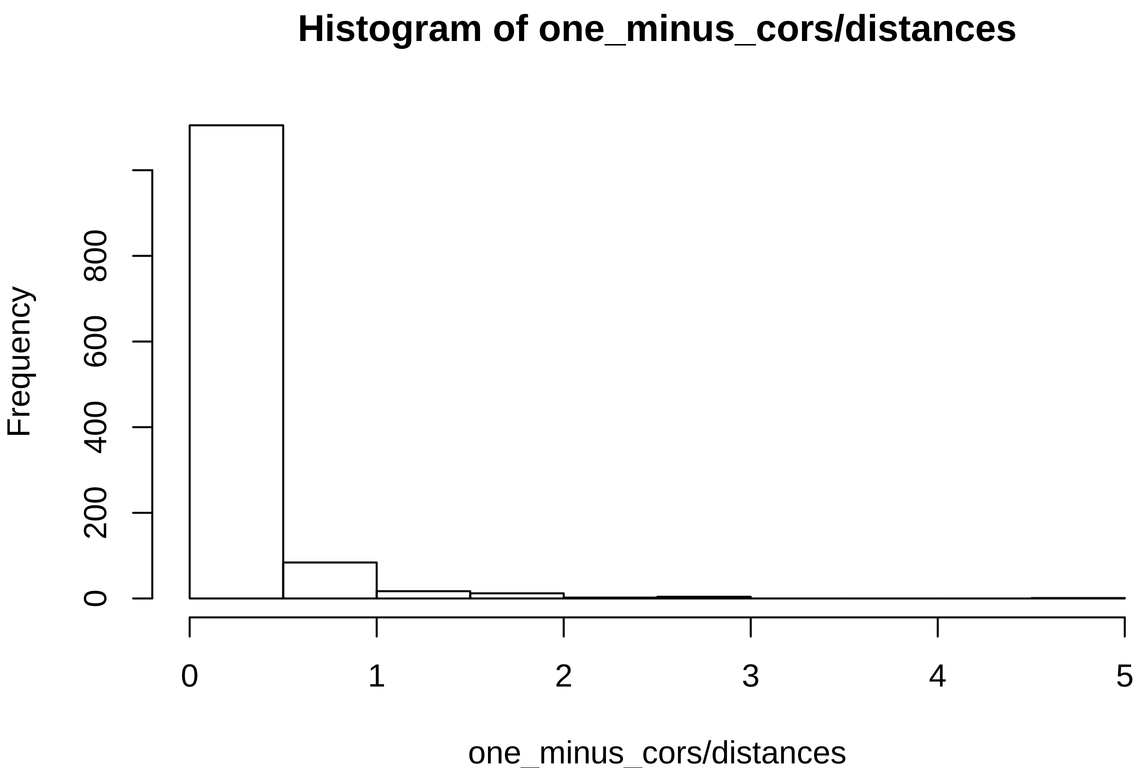
```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    3
## [3,]    1    4
## [4,]    1    5
## [5,]    1    6
## [6,]    1    7
```

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

# 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 -------------------
```

```r
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
```

10.7 #10

- a)

```r
# generate raw data matrix
set.seed(2017)
dat <- matrix(rnorm(60*50) + rep(c(2, -5, 7), each = 20), ncol = 50)
true_labels <- rep(c("group 1", "group 2", "group 3"), each = 20)

# get means for first 20 rows, representing the first class
mean(dat[1:20, ])
```

```
## [1] 1.990268
```

```r
# get means for the second 20 rows, representing the second class
mean(dat[21:40, ])
```

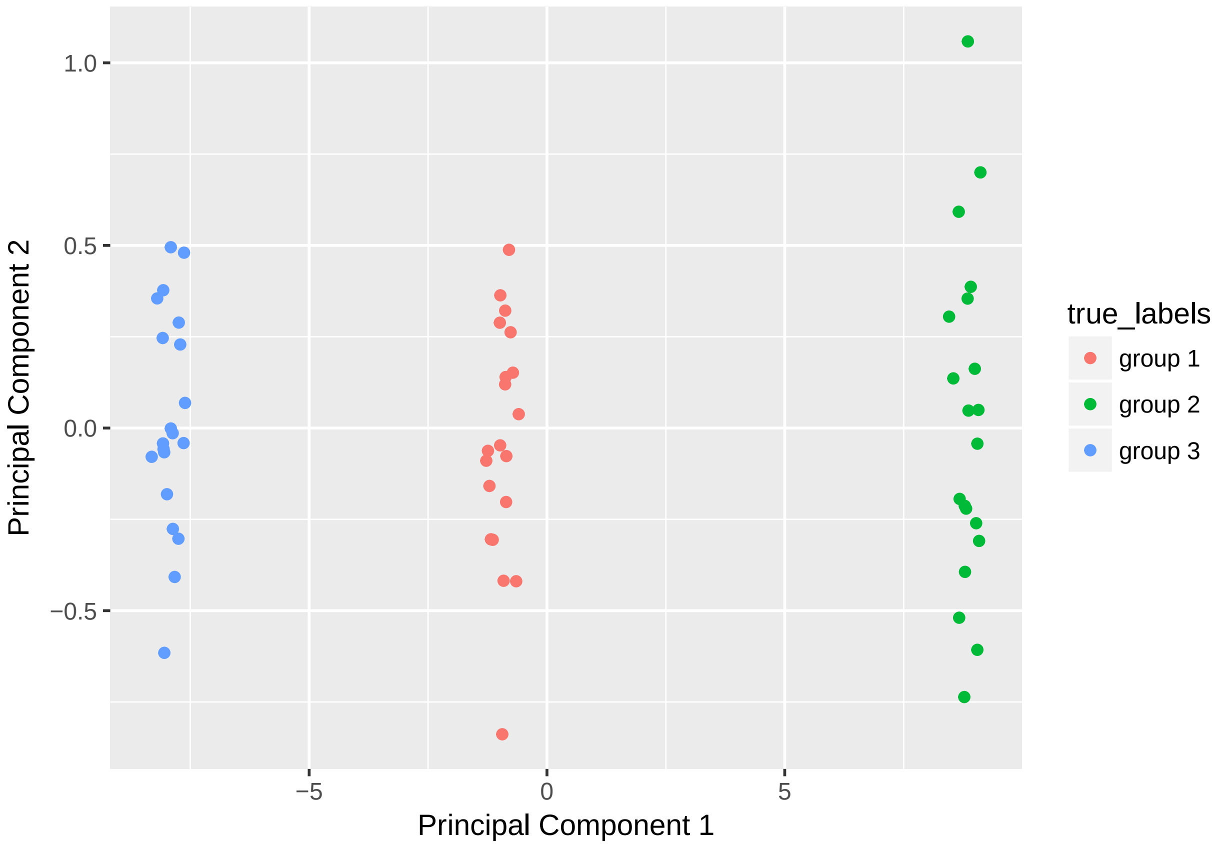
```
## [1] -5.033183
```

```r
# get means for the final 20 rows, representing the third class
mean(dat[41:60, ])
```

```
## [1] 6.998985
```

- b)

```r
# perform principal components
pr_10b <- prcomp(dat, scale = TRUE)

# plot the first two principal components
plot_dat <- as.data.frame(pr_10b$x[, c(1, 2)])
plot_dat$label <- true_labels

plot_dat %>%
  ggplot(aes(x = PC1, y = PC2)) +
  geom_point(aes(col = true_labels)) +
  xlab("Principal Component 1") +
  ylab("Principal Component 2")
```



- c)

```r
km_10c <- kmeans(dat, 3, nstart = 20)
table(true_labels, km_10c$cluster)
```

```
##
## true_labels  1  2  3
##     group 1 20  0  0
##     group 2  0 20  0
##     group 3  0  0 20
```

- d)

```r
km_10d <- kmeans(dat, 2, nstart = 20)
table(true_labels, km_10d$cluster)
```

```
##
## true_labels  1  2
##     group 1 20  0
##     group 2  0 20
##     group 3 20  0
```

When using K = 2 clusters, the clusters split cleanly between the three labels. Cluster 1 has group 1 and group 3 while Cluster 2 has group 2. Group 1 has a mean of 2 and Group 3 has a mean of 7, both corresponding to the positive means. Group 2 has a mean of -5, which is the only negative mean group. With two clusters, the groups are split into positive and negative groups.

- e)

```r
km_10e <- kmeans(dat, 4, nstart = 20)
table(true_labels, km_10e$cluster)
```

```
##
## true_labels  1  2  3  4
##     group 1 12  0  8  0
##     group 2  0 20  0  0
##     group 3  0  0  0 20
```

```r
# numeric summaries of the three true labels
summary(as.numeric(dat[1:20, ]))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -1.222   1.322   1.974   1.990   2.683   5.011
```

```r
summary(as.numeric(dat[21:40, ]))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -8.414  -5.699  -5.013  -5.033  -4.383  -1.706
```

```r
summary(as.numeric(dat[41:60, ]))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.766   6.354   7.024   6.999   7.663   9.956
```

When using K = 4 clusters, Group 1 is split between cluster 1 and cluster 3. Group 2 is assigned completely to cluster 2, and group 3 is assigned completely to cluster 4. As can be seen from the numeric summaries, Group 2 has only negative values while Group 3 has only positive values. Group 1 has mostly positive values but has some negative values as well. This is perhaps the reason why Group 1 is split into 2 clusters in this scenario.

- f)

```r
km_10f <- kmeans(pr_10b$x[, c(1, 2)],
                 3,
                 nstart = 20)
table(true_labels, km_10f$cluster)
```

```
##
## true_labels  1  2  3
##     group 1 20  0  0
##     group 2  0 20  0
##     group 3  0  0 20
```

When performing clustering on the principal components, the clusters again are perfectly divided into the three groups.

- g)

```r
km_10g <- kmeans(scale(dat), 3, nstart = 20)
table(true_labels, km_10g$cluster)
```

```
##
## true_labels  1  2  3
##     group 1 20  0  0
##     group 2  0  0 20
##     group 3  0 20  0
```

When performing clustering on the scaled data, we again see perfect separation. However, the labels are different than they have been in the past. Group 1 maps to Cluster 1, Group 2 maps to Cluster 3, and Group

3 maps to Cluster 2.