Chapter 10

Lab

```r
load(file.path(here::here(), "ISLR", "10.R.RData"))
```

Suppose we want to fit a linear regression, but the number of variables is much larger than the number of observations. In some cases, we may improve the fit by reducing the dimension of the features before.

In this problem, we use a data set with n = 300 and p = 200, so we have more observations than variables, but not by much. Load the data x, y, x.test, and y.test from 10.R.RData.

First, concatenate x and x.test using the rbind functions and perform a principal components analysis on the concatenated data frame (use the "scale=TRUE" option). To within 10% relative error, what proportion of the variance is explained by the first five principal components?

```r
dat <- rbind(x, x.test)

pca.out <- prcomp(dat, scale = T)

summary(pca.out) # 0.3499
```

```
Importance of components:
                          PC1    PC2    PC3     PC4    PC5     PC6     PC7
Standard deviation     5.0565 4.5965 3.7229 2.69713 1.4631 1.16827 1.15848
Proportion of Variance 0.1278 0.1056 0.0693 0.03637 0.0107 0.00682 0.00671
Cumulative Proportion  0.1278 0.2335 0.3028 0.33915 0.3499 0.35668 0.36339
                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
Standard deviation     1.15544 1.14591 1.13933 1.13619 1.13190 1.11047 1.10810
Proportion of Variance 0.00668 0.00657 0.00649 0.00645 0.00641 0.00617 0.00614
Cumulative Proportion  0.37007 0.37663 0.38312 0.38958 0.39598 0.40215 0.40829
                          PC15    PC16    PC17   PC18    PC19    PC20    PC21
Standard deviation     1.10226 1.09960 1.09257 1.0862 1.08519 1.07475 1.06942
Proportion of Variance 0.00607 0.00605 0.00597 0.0059 0.00589 0.00578 0.00572
Cumulative Proportion  0.41436 0.42041 0.42638 0.4323 0.43816 0.44394 0.44966
                          PC22    PC23   PC24    PC25    PC26    PC27    PC28
Standard deviation     1.06265 1.05967 1.0579 1.05436 1.04973 1.04615 1.04513
Proportion of Variance 0.00565 0.00561 0.0056 0.00556 0.00551 0.00547 0.00546
Cumulative Proportion  0.45530 0.46092 0.4665 0.47207 0.47758 0.48306 0.48852
                          PC29   PC30    PC31    PC32    PC33    PC34    PC35
Standard deviation     1.03557 1.0294 1.02468 1.02362 1.02126 1.01749 1.01436
Proportion of Variance 0.00536 0.0053 0.00525 0.00524 0.00521 0.00518 0.00514
Cumulative Proportion  0.49388 0.4992 0.50443 0.50967 0.51488 0.52006 0.52520
                          PC36    PC37   PC38    PC39    PC40    PC41    PC42
Standard deviation     1.01115 1.00510 1.0002 0.99711 0.99421 0.99132 0.98549
Proportion of Variance 0.00511 0.00505 0.0050 0.00497 0.00494 0.00491 0.00486
```

```
Cumulative Proportion  0.53031 0.53536 0.5404 0.54534 0.55028 0.55519 0.56005
                          PC43    PC44    PC45    PC46    PC47    PC48    PC49
Standard deviation     0.98031 0.9797 0.97717 0.97387 0.96814 0.95762 0.95609
Proportion of Variance 0.00481 0.0048 0.00477 0.00474 0.00469 0.00459 0.00457
Cumulative Proportion  0.56485 0.5696 0.57443 0.57917 0.58386 0.58844 0.59301
                          PC50    PC51    PC52    PC53    PC54    PC55    PC56
Standard deviation     0.95320 0.95162 0.94967 0.94648 0.94083 0.93936 0.93189
Proportion of Variance 0.00454 0.00453 0.00451 0.00448 0.00443 0.00441 0.00434
Cumulative Proportion  0.59756 0.60208 0.60659 0.61107 0.61550 0.61991 0.62425
                          PC57    PC58    PC59    PC60    PC61    PC62    PC63
Standard deviation     0.92636 0.92578 0.92296 0.91955 0.91776 0.91529 0.91019
Proportion of Variance 0.00429 0.00429 0.00426 0.00423 0.00421 0.00419 0.00414
Cumulative Proportion  0.62854 0.63283 0.63709 0.64131 0.64553 0.64971 0.65386
                          PC64    PC65    PC66    PC67   PC68    PC69    PC70
Standard deviation     0.90820 0.90197 0.89812 0.89654 0.8946 0.88705 0.88248
Proportion of Variance 0.00412 0.00407 0.00403 0.00402 0.0040 0.00393 0.00389
Cumulative Proportion  0.65798 0.66205 0.66608 0.67010 0.6741 0.67804 0.68193
                          PC71    PC72    PC73    PC74    PC75    PC76    PC77
Standard deviation     0.87985 0.87815 0.87600 0.87513 0.87338 0.86931 0.86514
Proportion of Variance 0.00387 0.00386 0.00384 0.00383 0.00381 0.00378 0.00374
Cumulative Proportion  0.68580 0.68966 0.69349 0.69732 0.70114 0.70492 0.70866
                          PC78    PC79    PC80    PC81    PC82    PC83    PC84
Standard deviation     0.86104 0.85696 0.85622 0.85481 0.84987 0.84650 0.84519
Proportion of Variance 0.00371 0.00367 0.00367 0.00365 0.00361 0.00358 0.00357
Cumulative Proportion  0.71237 0.71604 0.71970 0.72336 0.72697 0.73055 0.73412
                          PC85    PC86   PC87    PC88    PC89   PC90    PC91
Standard deviation     0.84242 0.83762 0.8363 0.83387 0.82896 0.8241 0.82214
Proportion of Variance 0.00355 0.00351 0.0035 0.00348 0.00344 0.0034 0.00338
Cumulative Proportion  0.73767 0.74118 0.7447 0.74815 0.75159 0.7550 0.75836
                          PC92    PC93    PC94    PC95    PC96    PC97    PC98
Standard deviation     0.81777 0.81733 0.81102 0.81018 0.80864 0.80311 0.80097
Proportion of Variance 0.00334 0.00334 0.00329 0.00328 0.00327 0.00322 0.00321
Cumulative Proportion  0.76171 0.76505 0.76834 0.77162 0.77489 0.77811 0.78132
                          PC99   PC100   PC101   PC102   PC103   PC104   PC105
Standard deviation     0.79769 0.79601 0.79312 0.78946 0.78658 0.78492 0.78199
Proportion of Variance 0.00318 0.00317 0.00315 0.00312 0.00309 0.00308 0.00306
Cumulative Proportion  0.78450 0.78767 0.79082 0.79393 0.79702 0.80011 0.80316
                         PC106  PC107   PC108   PC109   PC110   PC111   PC112
Standard deviation     0.77723 0.7741 0.77339 0.76774 0.76684 0.76393 0.76077
Proportion of Variance 0.00302 0.0030 0.00299 0.00295 0.00294 0.00292 0.00289
Cumulative Proportion  0.80618 0.8092 0.81217 0.81512 0.81806 0.82098 0.82387
                         PC113   PC114   PC115   PC116  PC117   PC118   PC119
Standard deviation     0.75911 0.75516 0.75483 0.75024 0.7477 0.74393 0.74082
Proportion of Variance 0.00288 0.00285 0.00285 0.00281 0.0028 0.00277 0.00274
Cumulative Proportion  0.82675 0.82960 0.83245 0.83526 0.8381 0.84083 0.84357
```

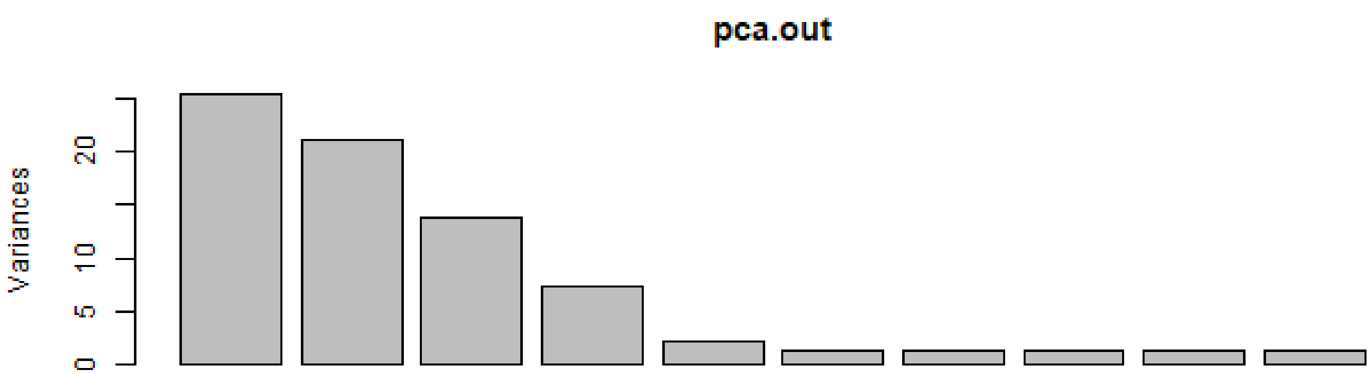
```
                          PC120   PC121   PC122   PC123   PC124   PC125   PC126
Standard deviation     0.73801 0.73356 0.73208 0.72765 0.72670 0.72333 0.72183
Proportion of Variance 0.00272 0.00269 0.00268 0.00265 0.00264 0.00262 0.00261
Cumulative Proportion  0.84629 0.84899 0.85166 0.85431 0.85695 0.85957 0.86217
                          PC127   PC128   PC129   PC130   PC131   PC132   PC133
Standard deviation     0.71917 0.71638 0.71056 0.70956 0.70381 0.70299 0.69930
Proportion of Variance 0.00259 0.00257 0.00252 0.00252 0.00248 0.00247 0.00245
Cumulative Proportion  0.86476 0.86733 0.86985 0.87237 0.87484 0.87732 0.87976
                          PC134   PC135   PC136   PC137   PC138   PC139   PC140
Standard deviation     0.69733 0.69374 0.69126 0.69035 0.68850 0.68574 0.68309
Proportion of Variance 0.00243 0.00241 0.00239 0.00238 0.00237 0.00235 0.00233
Cumulative Proportion  0.88219 0.88460 0.88699 0.88937 0.89174 0.89409 0.89643
                          PC141  PC142   PC143   PC144   PC145   PC146   PC147
Standard deviation     0.68201 0.6776 0.67488 0.67340 0.66606 0.66008 0.65796
Proportion of Variance 0.00233 0.0023 0.00228 0.00227 0.00222 0.00218 0.00216
Cumulative Proportion  0.89875 0.9011 0.90332 0.90559 0.90781 0.90999 0.91215
                          PC148   PC149  PC150   PC151   PC152   PC153   PC154
Standard deviation     0.65512 0.64990 0.6486 0.64713 0.64557 0.64445 0.64010
Proportion of Variance 0.00215 0.00211 0.0021 0.00209 0.00208 0.00208 0.00205
Cumulative Proportion  0.91430 0.91641 0.9185 0.92061 0.92269 0.92477 0.92682
                          PC155   PC156   PC157   PC158   PC159   PC160   PC161
Standard deviation     0.63631 0.63460 0.63149 0.63044 0.62557 0.62011 0.61820
Proportion of Variance 0.00202 0.00201 0.00199 0.00199 0.00196 0.00192 0.00191
Cumulative Proportion  0.92884 0.93085 0.93285 0.93484 0.93679 0.93872 0.94063
                          PC162   PC163   PC164   PC165   PC166   PC167   PC168
Standard deviation     0.61503 0.61174 0.61031 0.60601 0.60566 0.60124 0.59915
Proportion of Variance 0.00189 0.00187 0.00186 0.00184 0.00183 0.00181 0.00179
Cumulative Proportion  0.94252 0.94439 0.94625 0.94809 0.94992 0.95173 0.95352
                          PC169   PC170   PC171  PC172   PC173   PC174   PC175
Standard deviation     0.59483 0.59060 0.58598 0.5837 0.58113 0.57460 0.57292
Proportion of Variance 0.00177 0.00174 0.00172 0.0017 0.00169 0.00165 0.00164
Cumulative Proportion  0.95529 0.95704 0.95875 0.9605 0.96215 0.96380 0.96544
                          PC176   PC177  PC178   PC179   PC180   PC181   PC182
Standard deviation     0.57060 0.56745 0.5648 0.56069 0.55737 0.55029 0.54912
Proportion of Variance 0.00163 0.00161 0.0016 0.00157 0.00155 0.00151 0.00151
Cumulative Proportion  0.96707 0.96868 0.9703 0.97184 0.97340 0.97491 0.97642
                          PC183   PC184   PC185   PC186   PC187   PC188   PC189
Standard deviation     0.54679 0.54492 0.54137 0.53964 0.53536 0.52646 0.52444
Proportion of Variance 0.00149 0.00148 0.00147 0.00146 0.00143 0.00139 0.00138
Cumulative Proportion  0.97791 0.97940 0.98086 0.98232 0.98375 0.98514 0.98651
                          PC190   PC191   PC192  PC193   PC194   PC195   PC196
Standard deviation     0.52070 0.51532 0.51244 0.5106 0.50301 0.50114 0.49116
Proportion of Variance 0.00136 0.00133 0.00131 0.0013 0.00127 0.00126 0.00121
Cumulative Proportion  0.98787 0.98920 0.99051 0.9918 0.99308 0.99433 0.99554
                          PC197   PC198   PC199   PC200
```

```
Standard deviation     0.48723 0.48489 0.48099 0.43382
Proportion of Variance 0.00119 0.00118 0.00116 0.00094
Cumulative Proportion  0.99673 0.99790 0.99906 1.00000
```

```
sum(head((pca.out$sdev)^2/ sum(pca.out$sdev^2), length = 5)) # 0.3566807
```

```
[1] 0.3566807
```

```
par(mfrow = c(1,1))
plot(pca.out)
```



The previous answer suggests that a relatively small number of "latent variables" account for a substantial fraction of the features' variability. We might believe that these latent variables are more important than linear combinations of the features that have low variance.

We can try forgetting about the raw features and using the first five principal components (computed on rbind(x,x.test)) instead as low-dimensional derived features. What is the mean-squared test error if we regress y on the first five principal components, and use the resulting model to predict y.test?

```
xols <- pca.out$x[1:300,1:5]
fit0 <- lm(y ~ xols)
summary(fit0)
```

```
Call:
lm(formula = y ~ xols)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3289 -0.6992  0.0319  0.8075  2.5240

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.09541    0.06107   1.562 0.119314
xolsPC1      0.07608    0.01159   6.564 2.36e-10 ***
xolsPC2     -0.02276    0.01314  -1.732 0.084309 .
xolsPC3     -0.04023    0.01538  -2.616 0.009352 **
xolsPC4     -0.06368    0.02237  -2.847 0.004722 **
xolsPC5     -0.16069    0.04299  -3.738 0.000223 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.056 on 294 degrees of freedom
Multiple R-squared:  0.1906,    Adjusted R-squared:  0.1769
F-statistic: 13.85 on 5 and 294 DF,  p-value: 3.704e-12
```

```
yhat0 <- predict(fit0, x.test)
```

```
Warning: 'newdata' had 1000 rows but variables found have 300 rows
```

```
mean((yhat0-y.test)^2)
```

```
Warning in yhat0 - y.test: longer object length is not a multiple of shorter
object length
```

```
[1] 1.413063
```

Lab

```
states <- row.names(USArrests)
states
```

```
 [1] "Alabama"        "Alaska"         "Arizona"        "Arkansas"
 [5] "California"     "Colorado"       "Connecticut"    "Delaware"
 [9] "Florida"        "Georgia"        "Hawaii"         "Idaho"
[13] "Illinois"       "Indiana"        "Iowa"           "Kansas"
[17] "Kentucky"       "Louisiana"      "Maine"          "Maryland"
[21] "Massachusetts"  "Michigan"       "Minnesota"      "Mississippi"
[25] "Missouri"       "Montana"        "Nebraska"       "Nevada"
[29] "New Hampshire"  "New Jersey"     "New Mexico"     "New York"
[33] "North Carolina" "North Dakota"   "Ohio"           "Oklahoma"
[37] "Oregon"         "Pennsylvania"   "Rhode Island"   "South Carolina"
[41] "South Dakota"   "Tennessee"      "Texas"          "Utah"
[45] "Vermont"        "Virginia"       "Washington"     "West Virginia"
[49] "Wisconsin"      "Wyoming"
```

```
names(USArrests)
```

```
[1] "Murder"   "Assault"  "UrbanPop" "Rape"
```

```r
apply(USArrests, 2, mean)
```

```
  Murder  Assault UrbanPop     Rape 
   7.788  170.760   65.540   21.232 
```

```r
apply(USArrests, 2, var)
```

```
    Murder    Assault   UrbanPop       Rape 
  18.97047 6945.16571  209.51878   87.72916 
```

```r
pr.out <- prcomp(USArrests, scale = T)
```

```r
names(pr.out)
```

```
[1] "sdev"     "rotation" "center"   "scale"    "x"
```

```r
pr.out$center
```

```
  Murder  Assault UrbanPop     Rape 
   7.788  170.760   65.540   21.232 
```

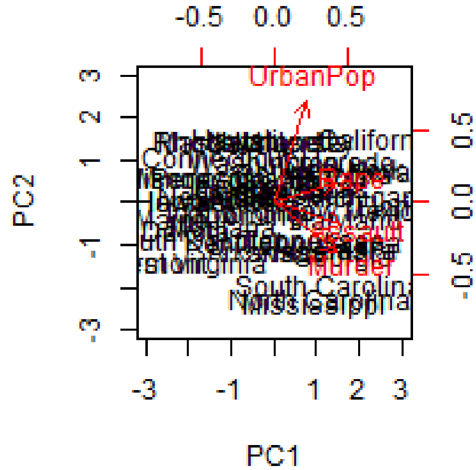
```r
pr.out$rotation
```

```
                PC1        PC2        PC3         PC4
Murder   -0.5358995  0.4181809 -0.3412327  0.64922780
Assault  -0.5831836  0.1879856 -0.2681484 -0.74340748
UrbanPop -0.2781909 -0.8728062 -0.3780158  0.13387773
Rape     -0.5434321 -0.1673186  0.8177779  0.08902432
```

```r
dim(pr.out$x)
```

```
[1] 50  4
```

```r
pr.out$rotation =-pr.out$rotation
pr.out$x =-pr.out$x
biplot(pr.out, scale = 0)
```




```r
pr.out$sdev
```

```
[1] 1.5748783 0.9948694 0.5971291 0.4164494
```

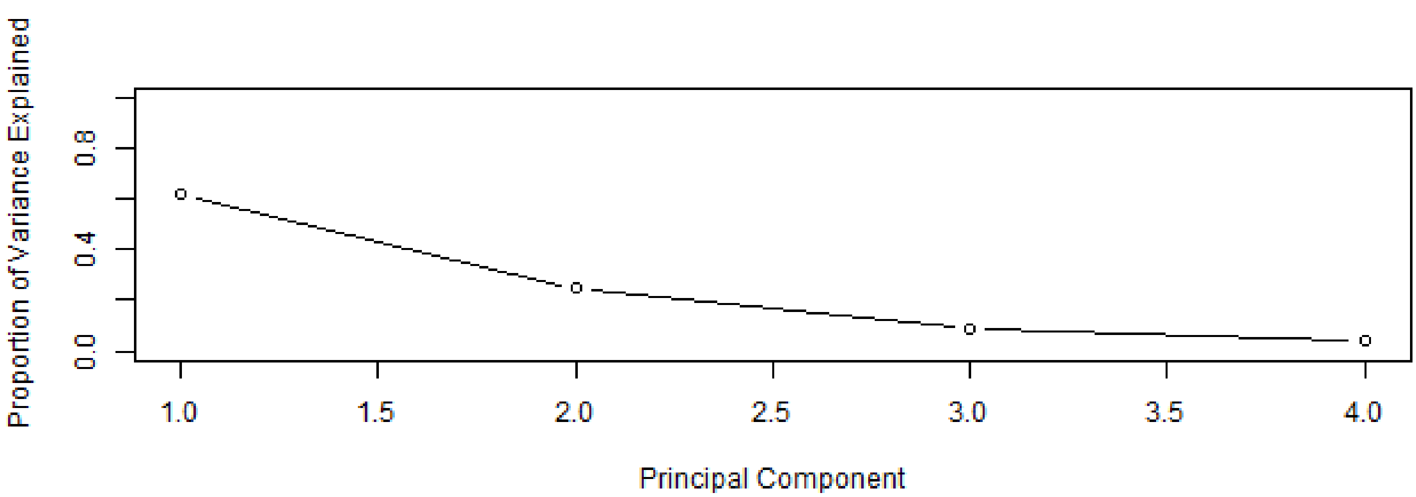
```r
pr.var = pr.out$sdev^2
pr.out$sdev
```

```
[1] 1.5748783 0.9948694 0.5971291 0.4164494
```

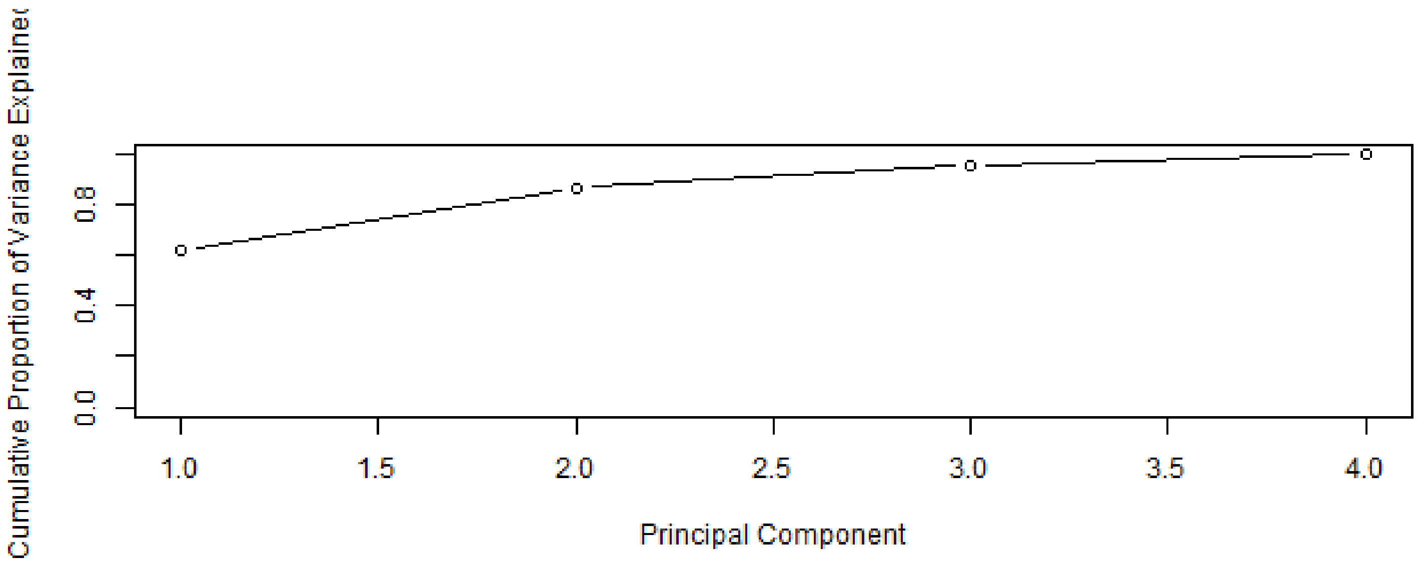
```r
pve <- pr.var/sum(pr.var)
pve
```

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

```r
plot(pve, xlab = "Principal Component", ylab = "Proportion of Variance Explained", ylim = c(0,
```



```
plot(cumsum(pve), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance Expla
```



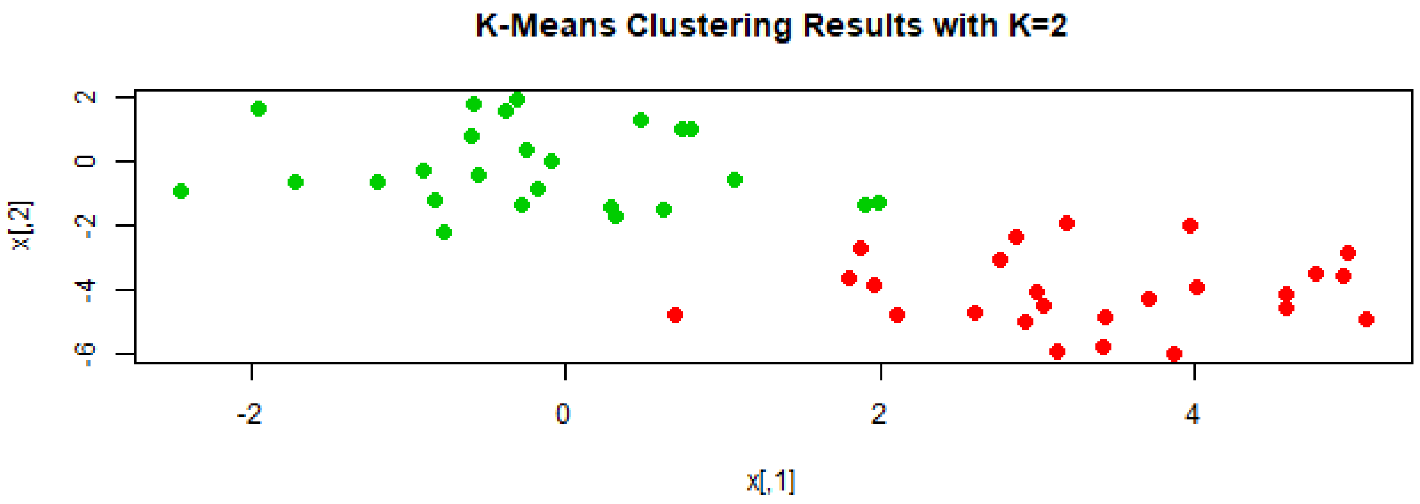

```
set.seed(2)
x <- matrix(rnorm(50*2), ncol = 2)
x[1:25, 1] = x[1:25, 1] + 3
x[1:25, 2] = x[1:25, 2] - 4
```

```
km.out <- kmeans(x, 2, nstart = 20)
```

```
km.out$cluster
```

```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 2
```

```
plot(x, col = (km.out$cluster + 1), main = "K-Means Clustering Results with K=2", pch = 20, cex
```



```r
set.seed(4)

km.out <- kmeans(x, 3, nstart = 20)
km.out
```

```
K-means clustering with 3 clusters of sizes 17, 23, 10

Cluster means:
        [,1]        [,2]
1  3.7789567 -4.56200798
2 -0.3820397 -0.08740753
3  2.3001545 -2.69622023

Clustering vector:
 [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 3 2 3 2 2 2 2

Within cluster sum of squares by cluster:
[1] 25.74089 52.67700 19.56137
 (between_SS / total_SS =  79.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

```r
set.seed(3)

km.out <- kmeans(x, 3, nstart = 1)
km.out$tot.withinss
```

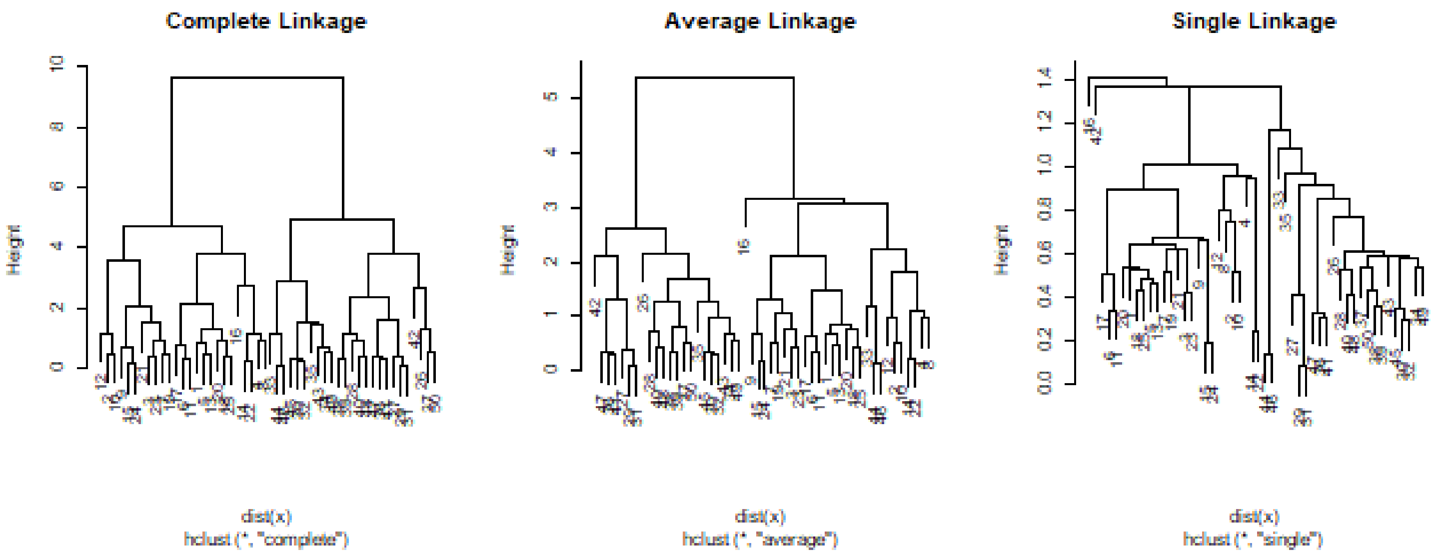
```
[1] 97.97927
```

```r
km.out <- kmeans(x, 3, nstart = 20)
km.out$tot.withinss
```

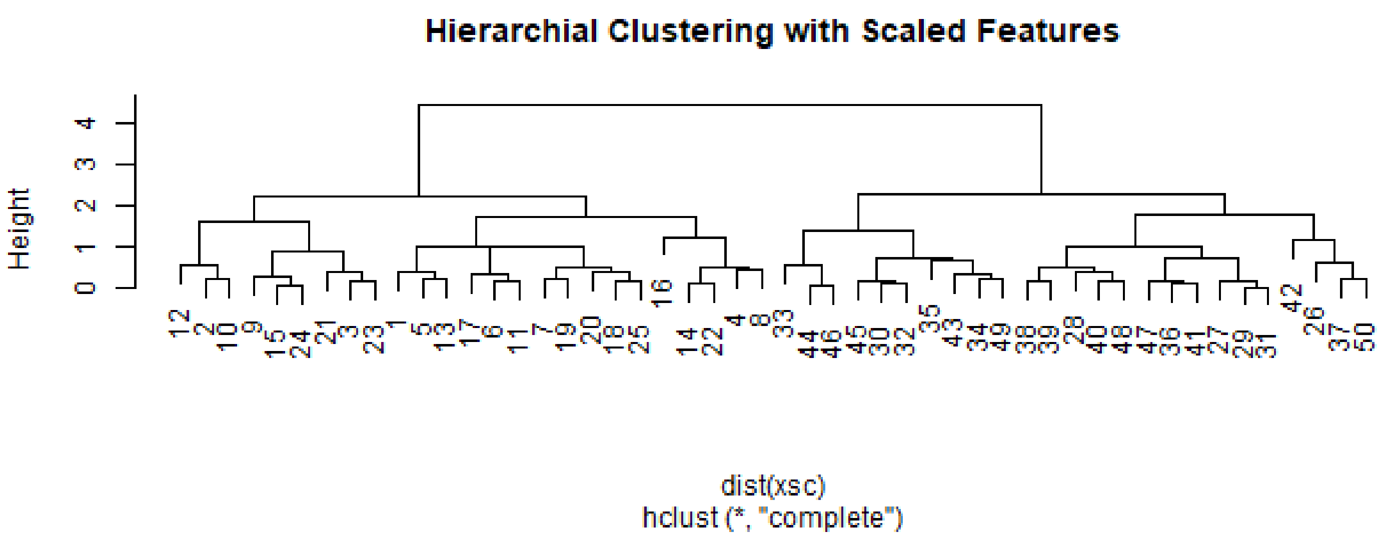
```
[1] 97.97927
```

```r
hc.complete <- hclust(dist(x), method = "complete")
hc.average <- hclust(dist(x), method = "average")
hc.single <- hclust(dist(x), method = "single")
```

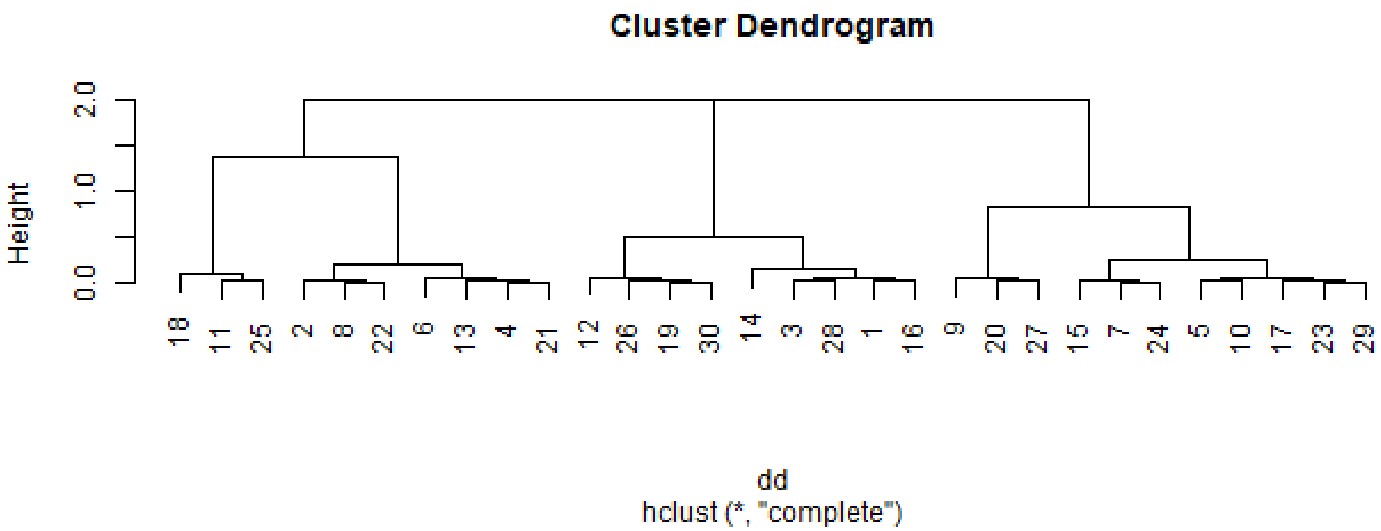
```r
par(mfrow = c(1, 3))
plot(hc.complete, main = "Complete Linkage", cex = .9)
plot(hc.average, main = "Average Linkage", cex = .9)
plot(hc.single, main = "Single Linkage", cex = .9)
```




```r
xsc <- scale(x)

par(mfrow = c(1, 1))
plot(hclust(dist(xsc), method = "complete"), main = "Hierarchial Clustering with Scaled Feature
```



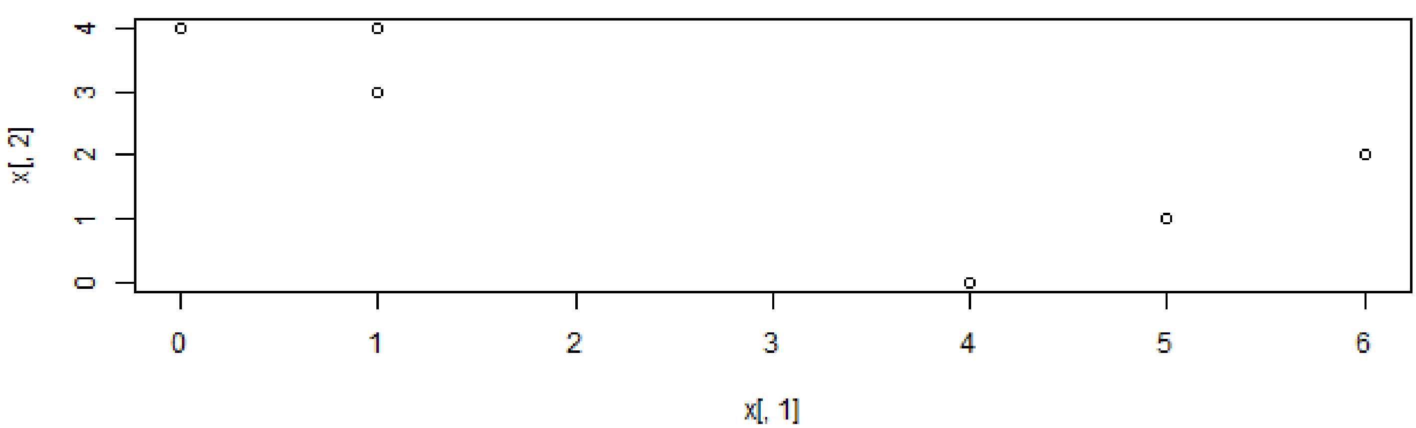

```r
x <- matrix(rnorm(30*3), ncol = 3)
dd <- as.dist(1 - cor(t(x)))

plot(hclust(dd, method = "complete"))
```



3.) In this problem, you will perform K-means clustering manually, with K=2, on a small example with n=6 observations and p=2 features. The observations are as follows.

a.) Plot the observations.

```
x <- cbind(c(1, 1, 0, 5, 6, 4), c(4, 3, 4, 1, 2, 0))
plot(x[,1], x[,2])
```

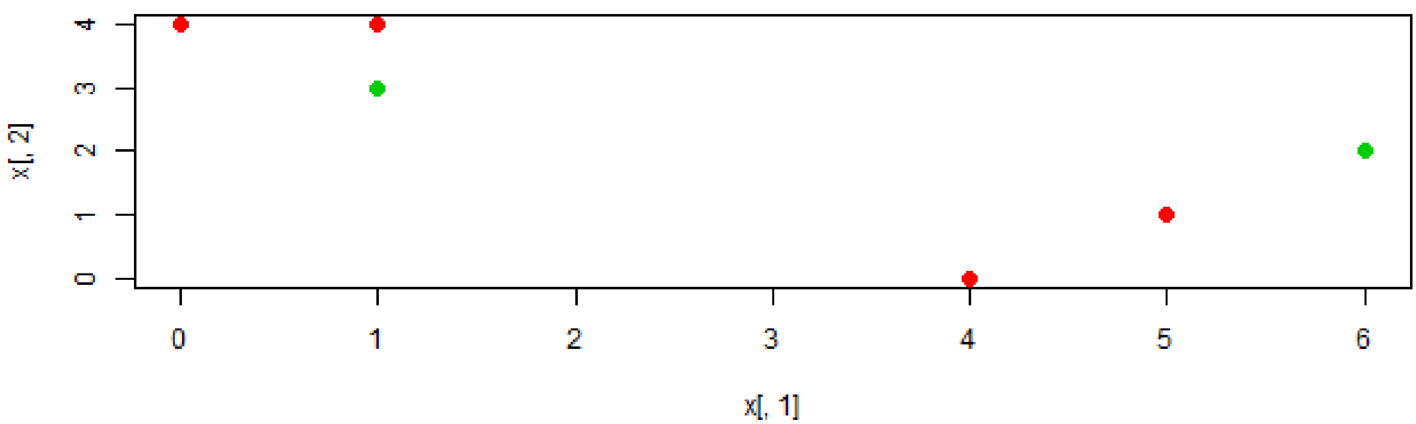


b.) Randomly assign a cluster label to each observation. Report the cluster labels for each observation.

```
set.seed(1)
labels <- sample(2, nrow(x), replace = T)
labels
```

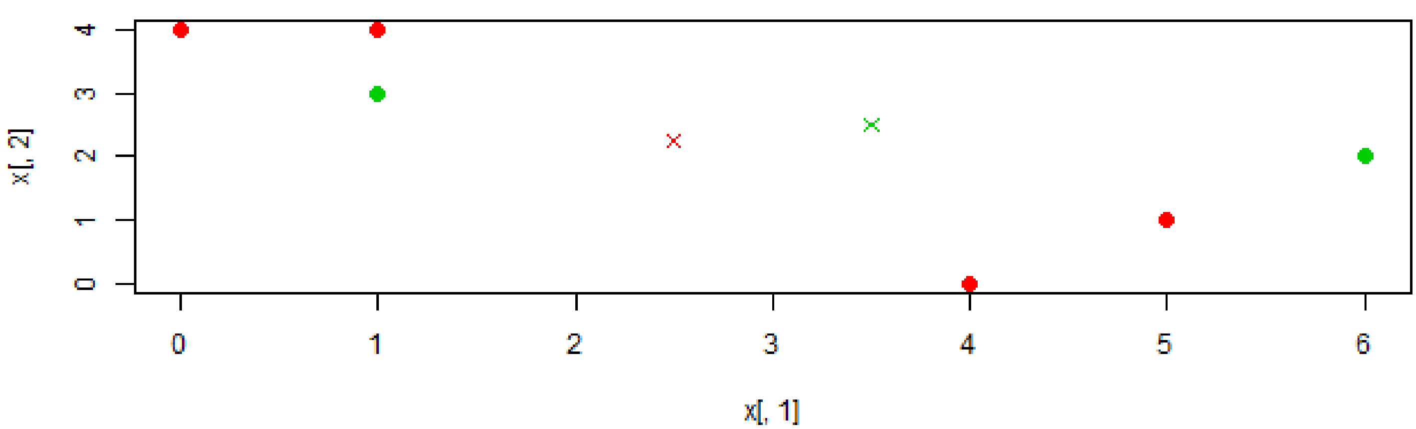
```
[1] 1 2 1 1 2 1
```

```
plot(x[, 1], x[, 2], col = (labels + 1), pch = 20, cex = 2)
```



c.) Compute the centroid for each cluster.

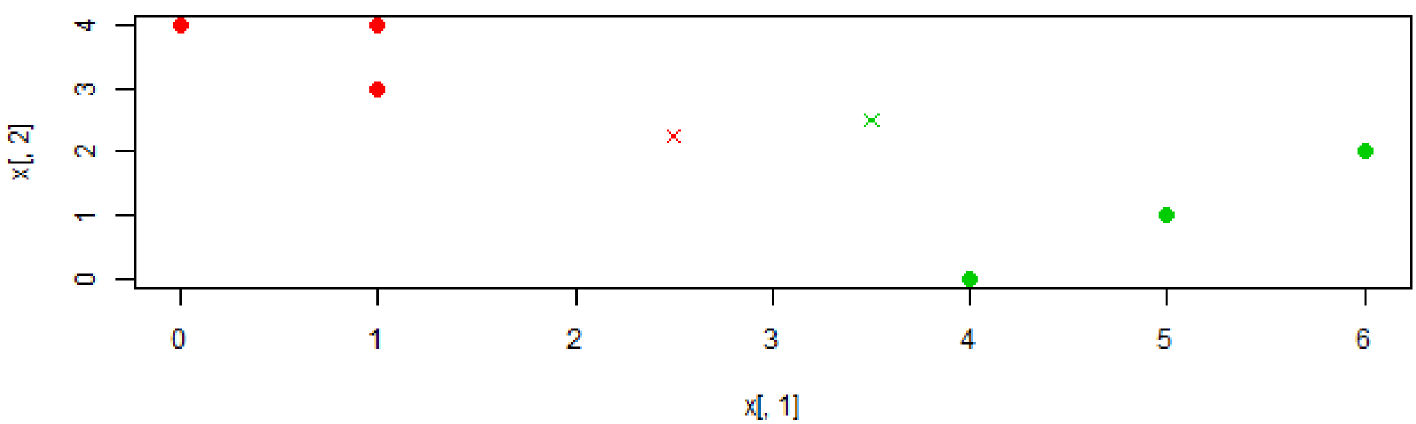
```
centroid1 <- c(mean(x[labels == 1, 1]), mean(x[labels == 1, 2]))
centroid2 <- c(mean(x[labels == 2, 1]), mean(x[labels == 2, 2]))
plot(x[,1], x[,2], col=(labels + 1), pch = 20, cex = 2)
points(centroid1[1], centroid1[2], col = 2, pch = 4)
points(centroid2[1], centroid2[2], col = 3, pch = 4)
```



d.) Assign each observation to the centroid to which it is closest, in terms of Euclidean distance. Report the cluster labels for each observation.

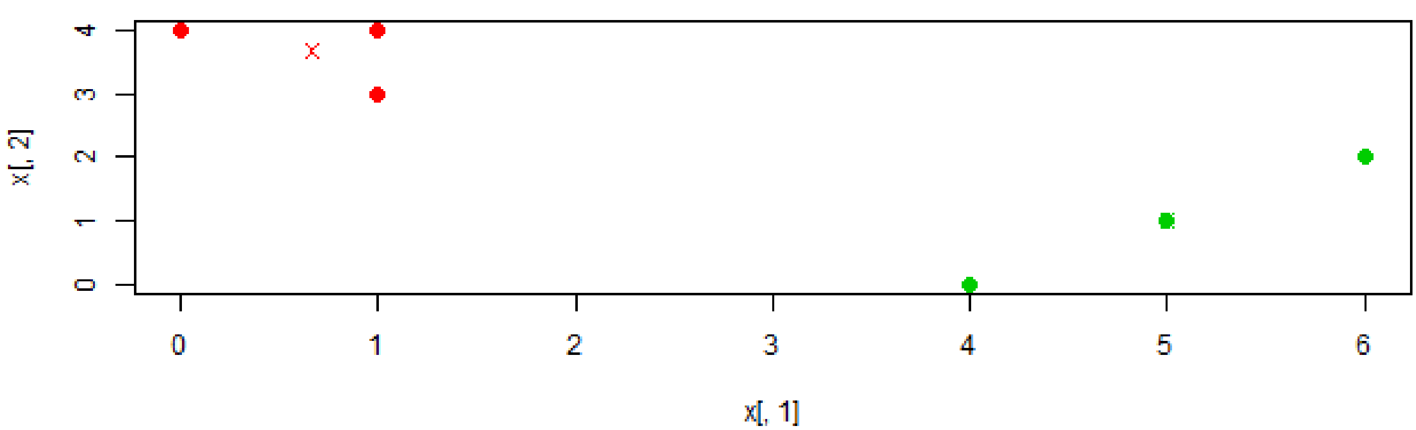
```
labels <- c(1, 1, 1, 2, 2, 2)
plot(x[, 1], x[, 2], col = (labels + 1), pch = 20, cex = 2)
```

```r
points(centroid1[1], centroid1[2], col = 2, pch = 4)
points(centroid2[1], centroid2[2], col = 3, pch = 4)
```



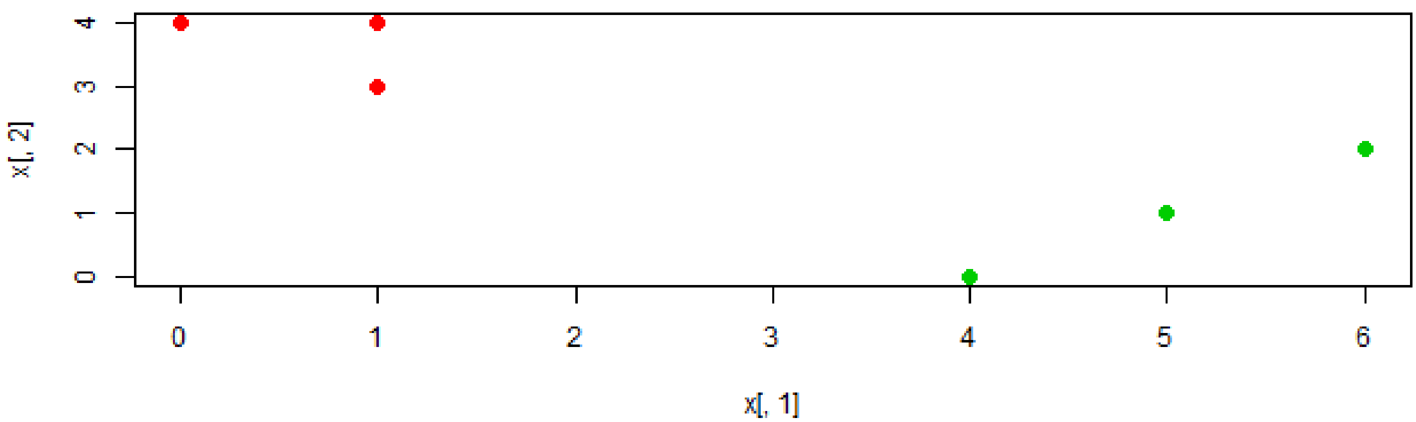
e.) Repeat (c) and (d) until the answers obtained stop changing.

```r
centroid1 <- c(mean(x[labels == 1, 1]), mean(x[labels == 1, 2]))
centroid2 <- c(mean(x[labels == 2, 1]), mean(x[labels == 2, 2]))
plot(x[,1], x[,2], col=(labels + 1), pch = 20, cex = 2)
points(centroid1[1], centroid1[2], col = 2, pch = 4)
points(centroid2[1], centroid2[2], col = 3, pch = 4)
```



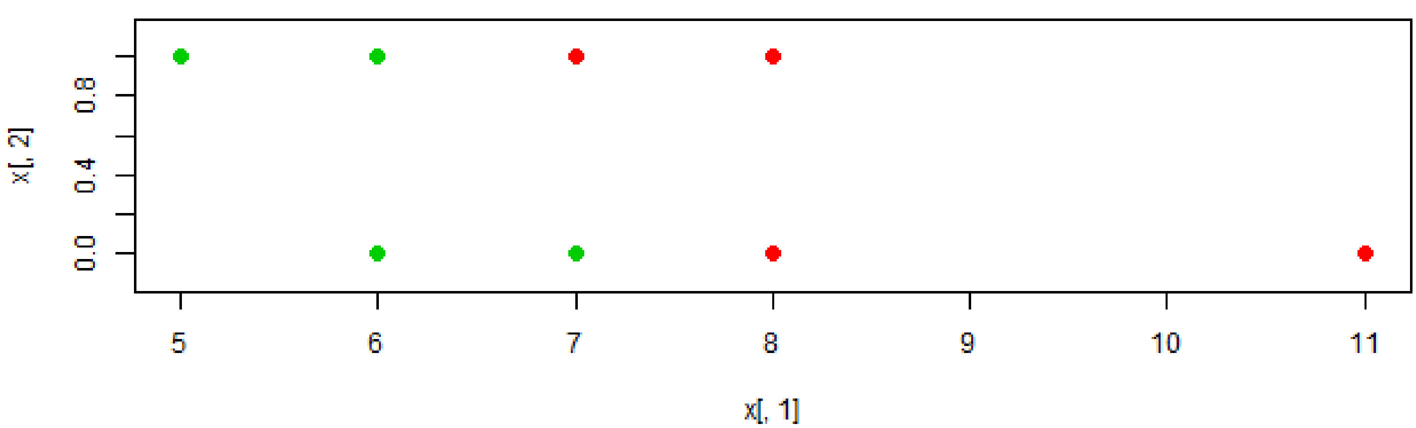
f.) In your plot from (a), color the observations according to the clusters labels obtained.

```r
plot(x[, 1], x[, 2], col=(labels + 1), pch = 20, cex = 2)
```



In words, describe the results that you would expect if you performed K-means clustering of the eight shoppers in Figure 10.14, on the basis of their sock and computer purchases, with K=2. Give three answers, one for each of the variable scalings displayed. Explain.

```
socks <- c(8, 11, 7, 6, 5, 6, 7, 8)
computers <- c(0, 0, 0, 0, 1, 1, 1, 1)
x <- cbind(socks, computers)
labels <- c(1, 1, 2, 2, 2, 2, 1, 1)
plot(x[, 1], x[, 2], col=(labels + 1), pch = 20, cex = 2, asp = 1)
```

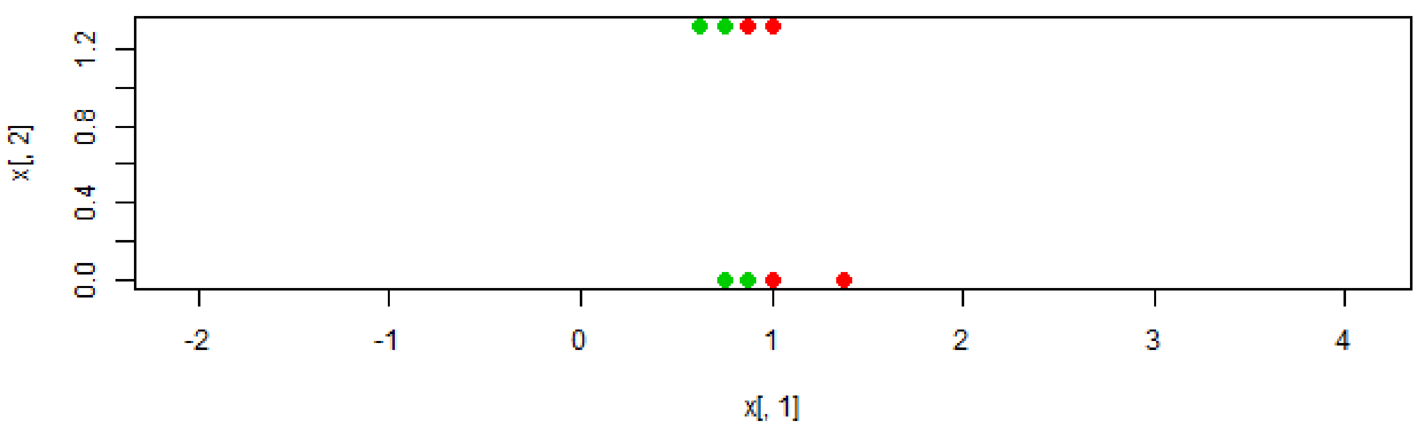


If we take into consideration the scaled variables, the number of computers plays a much larger role than the number of socks, so we have the clusters {5,6,7,8} (purchased computer) and {1,2,3,4} (no computer purchased).

```r
x <- cbind(scale(socks, center = FALSE), scale(computers, center = FALSE))
sd(computers)
```

```
[1] 0.5345225
```

```r
labels <- c(1, 1, 2, 2, 2, 2, 1, 1)
plot(x[, 1], x[, 2], col=(labels + 1), pch = 20, cex = 2, asp = 1)
```



A researcher collects expression measurements for 1000 genes in 100 tissue samples. The data can be written as a 1000x1000 matrix, which we call X, in which each row represents a gene and each column a tissue sample. Each tissue sample was processed on a different day, and the columns of X are ordered so that the samples that were processed earliest are on the left, and the samples that were processed later are on the right. The tissue samples belong to two groups : control (C) and treatment (T). The C and T samples were processed in a random order across the days. The researcher wishes to determine whether each gene's expression measurements differ between the treatment and control groups.

As a pre-analysis (before comparing T versus C), the researcher performs a principal component analysis of the data, and finds that the first principal component (a vector of length 100) has a strong linear trend from left to right, and explains 10% of the variation. The researcher now remembers that each patient sample was run on one of two machines, A and B, and machine A was used more often in the earlier times while B was used more often later. The researcher has a record of which sample was run on which machine.

Explain what it means that the first principal component "explains 10% of the variation". The first principal component "explains 10% of the variation" means 90% of the information in the gene data set is lost by projecting the tissue sample observations onto the first principal component. Another way of explaining it is 90% of the variance in the data is not contained in the first principal component.

The researcher decides to replace the (i,j)th element of X with xij−zi1ϕj1 where zi1 is the ith score, and ϕj1 is the jth loading, for the first principal component. He will then perform a two-sample t-test on each gene in this new data set in order to determine whether its expression differs between the two conditions.

Critique this idea, and suggest a better approach. Given the flaw shown in pre-analysis of a time-wise linear trend amongst the tissue samples' first principal component, I would advise the researcher to include the machine used (A vs B) as a feature of the data set. This should enhance the PVE of the first principal component before applying the two-sample t-test.

Design and run a small simulation experiment to demonstrate the superiority of your idea.

```
set.seed(1)
Control <- matrix(rnorm(50 * 1000), ncol = 50)
Treatment <- matrix(rnorm(50 * 1000), ncol = 50)
X <- cbind(Control, Treatment)
X[1, ] <- seq(-18, 18 - .36, .36) # linear trend in one dimension
pr.out <- prcomp(scale(X))
summary(pr.out)$importance[, 1]
```

```
    Standard deviation Proportion of Variance  Cumulative Proportion
              3.148148               0.099110               0.099110
```

We have 9.911% variance explained by the first principal component. Now, adding in A vs B via 10 vs 0 encoding.

```
X <- rbind(X, c(rep(10, 50), rep(0, 50)))
pr.out <- prcomp(scale(X))
summary(pr.out)$importance[, 1]
```

```
    Standard deviation Proportion of Variance  Cumulative Proportion
              3.397839               0.115450               0.115450
```

7.) In the chapter, we mentioned the use of correlation-based distance and Euclidean distance as dissimilarity measures for hierarchical clustering. It turns out that these two measures are almost equivalent : if each observation has been centered to have mean zero and standard deviation one, and if we let rij denote the correlation between the ith and jth observations, then the quantity 1−rij is proportional to the squared Euclidean distance between the ith and jth observations. On the "USArrests" data, show that this proportionality holds.

```
set.seed(1)
dsc <- scale(USArrests)
d1 <- dist(dsc)^2
d2 <- as.dist(1 - cor(t(dsc)))
summary(d2 / d1)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.000086 0.069135 0.133943 0.234193 0.262589 4.887686
```

In Section 10.2.3, a formula for calculating PVE was given in Equation 10.8. We also saw that the PVE can be obtained using the "sdev" output of the "prcomp()" function. On the "USArrests" data, calculate PVE in two ways :

a.) Using the "sdev" output of the "prcomp()" function, as was done in Section 10.2.3.

```
pr.out <- prcomp(USArrests, scale = TRUE)
pr.var <- pr.out$sdev^2
pve <- pr.var / sum(pr.var)
sum(pr.var)
```

```
[1] 4
```

b.) By applying Equation 10.8 directly. That is, use the "prcomp()" function to compute the principal component loadings. Then, use those loadings in Equation 10.8 to obtain the PVE.

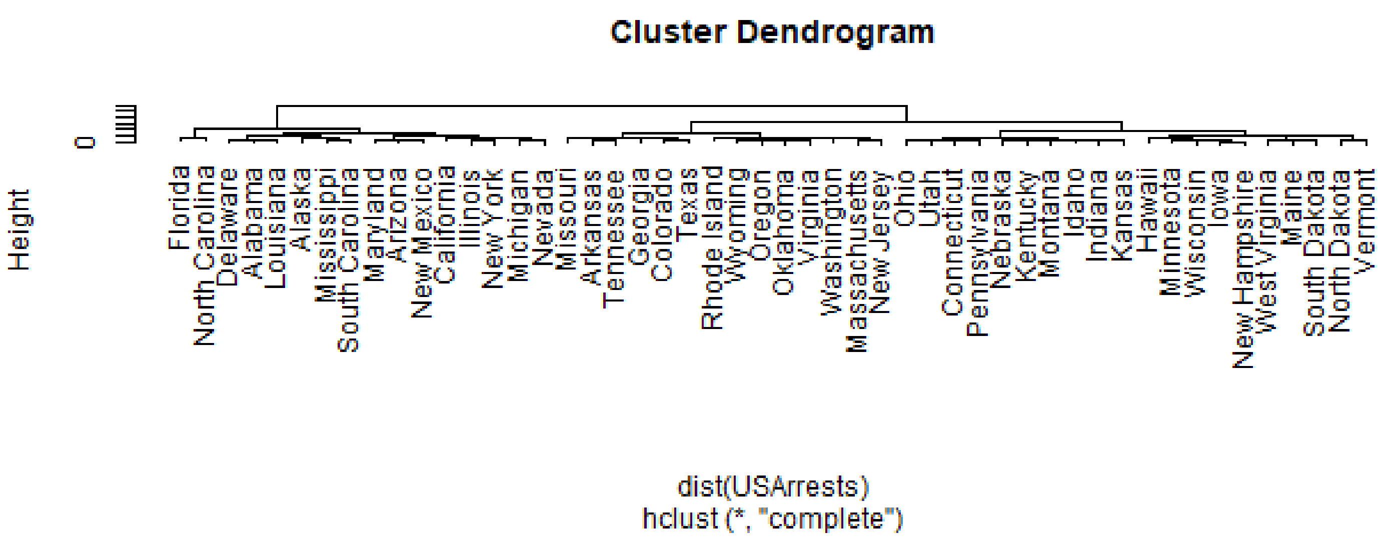
```
loadings <- pr.out$rotation
USArrests2 <- scale(USArrests)
sumvar <- sum(apply(as.matrix(USArrests2)^2, 2, sum))
apply((as.matrix(USArrests2) %*% loadings)^2, 2, sum) / sumvar
```

```
       PC1        PC2        PC3        PC4
0.62006039 0.24744129 0.08914080 0.04335752
```

9.) Consider the "USArrests" data. We will now perform hierarchical clustering on the states.

a.) Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

```
set.seed(2)
hc.complete <- hclust(dist(USArrests), method = "complete")
plot(hc.complete)
```



b.) Cut the dendrogram at a height that results in three distinct clusters. Which states belong to which clusters ?

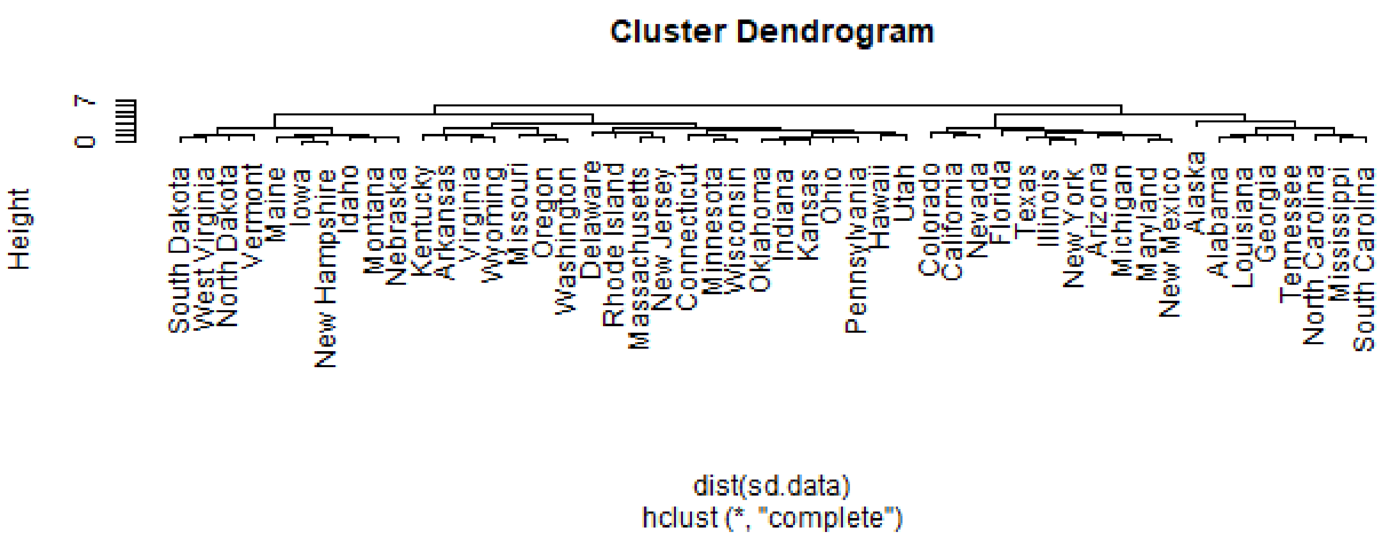
```
cutree(hc.complete, 3)
```

```
       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 
```

c.) Hierachically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one.

```r
sd.data <- scale(USArrests)
hc.complete.sd <- hclust(dist(sd.data), method = "complete")
plot(hc.complete.sd)
```



d.) What effect does scaling the variables have on the hierarchical clustering obtained ? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed ? Provide a justification for your answer.

```r
cutree(hc.complete.sd, 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
```

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

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

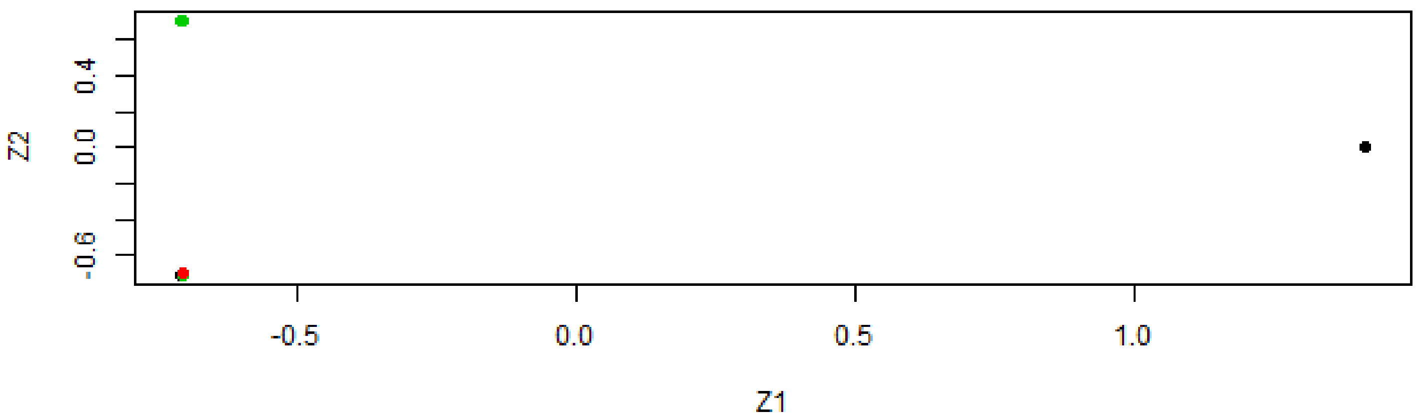
10.) In this problem, you will generate simulated data, and then perform PCA and K-means clustering on the data.

a.) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables.

```r
set.seed(2)
x <- matrix(rnorm(20 * 3 * 50, mean = 0, sd = 0.001), ncol = 50)
x[1:20, 2] <- 1
x[21:40, 1] <- 2
x[21:40, 2] <- 2
x[41:60, 1] <- 1
true.labels <- c(rep(1, 20), rep(2, 20), rep(3, 20))
```

b.) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then continue on to part (c). If not, the return to part (a) and modify the simulation so that there is greater separation between the three classes. Do not continue to part (c) until the three classes show at least some separation in the first two principal component score vectors.

```
pr.out <- prcomp(x)
plot(pr.out$x[, 1:2], col = 1:3, xlab = "Z1", ylab = "Z2", pch = 19)
```



c.) Perform K-means clustering of the observations with K=3. How well do the clusters that you obtained in K-means clustering compare to the true class labels ?

```
km.out <- kmeans(x, 3, nstart = 20)
table(true.labels, km.out$cluster)
```

```
true.labels  1  2  3
          1  0  0 20
          2 20  0  0
          3  0 20  0
```

d.) Perform K-means clustering with K=2. Describe your results.

```
km.out <- kmeans(x, 2, nstart = 20)
table(true.labels, km.out$cluster)
```

```
true.labels  1  2
          1 20  0
          2  0 20
          3 20  0
```

e.) Now perform K-means clustering with K=4, and describe your results.

```
km.out <- kmeans(x, 4, nstart = 20)
table(true.labels, km.out$cluster)
```

```
true.labels  1  2  3  4
          1 11  9  0  0
          2  0  0 20  0
          3  0  0  0 20
```

f.) Now perform K-means clustering with K=3 on the first two principal component score vectors, rather than on the raw data. That is, perform K-means clustering on the 60x2 matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.

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

```
true.labels  1  2  3
          1  0  0 20
          2  0 20  0
          3 20  0  0
```

g.) Using the “scale()” function, perform K-means clustering with K=3 on the data after scaling each variable to have standard deviation one. How do these results compare to those obtained in (b) ? Explain.

```
km.out <- kmeans(scale(x), 3, nstart = 20)
table(true.labels, km.out$cluster)
```

```
true.labels  1  2  3
          1  9  2  9
          2  2 18  0
          3  7  1 12
```

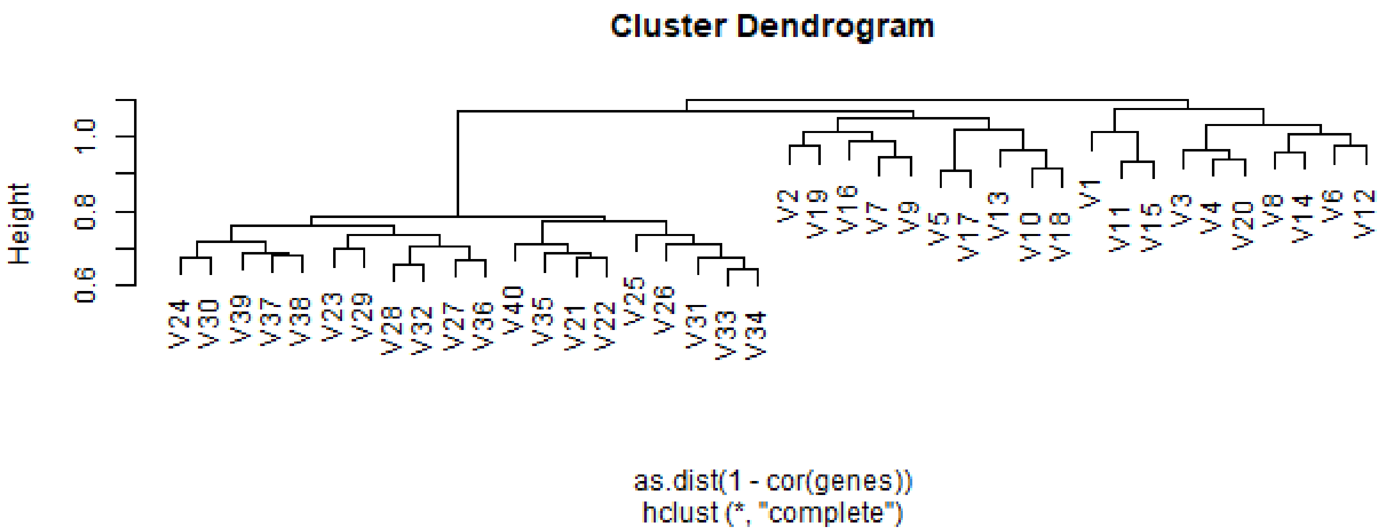
11.) On the book website, there is a gene expression data set that consists of 40 tissue samples with measurements on 1000 genes. The first 20 samples are from healthy patients, while the second 20 are from a diseased group.

a.) Load the data using “read.csv()”. You will need to select “header = F”.

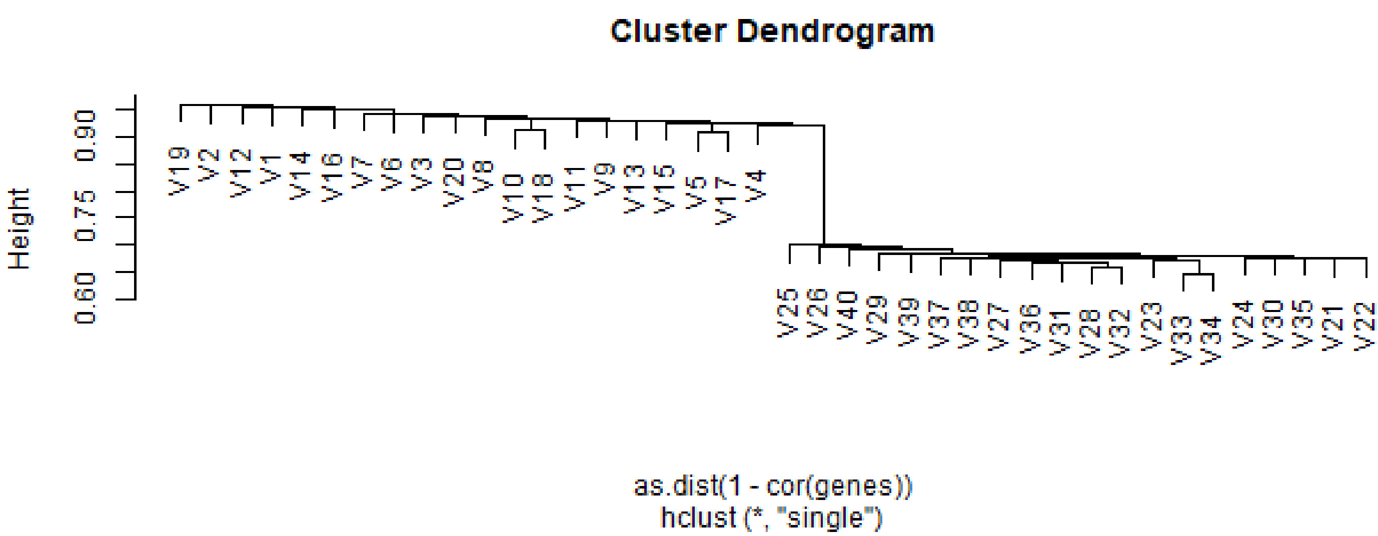
```
genes <- read.csv(file.path(here::here(), "ISLR", "Ch10Ex11.csv"), header = FALSE)
```

b.) Apply hierarchical clustering to the samples using correlation-based distance, and plot the dendrogram. Do the genes separate the samples into two groups ? Do your results depend on the type of linkage used ?

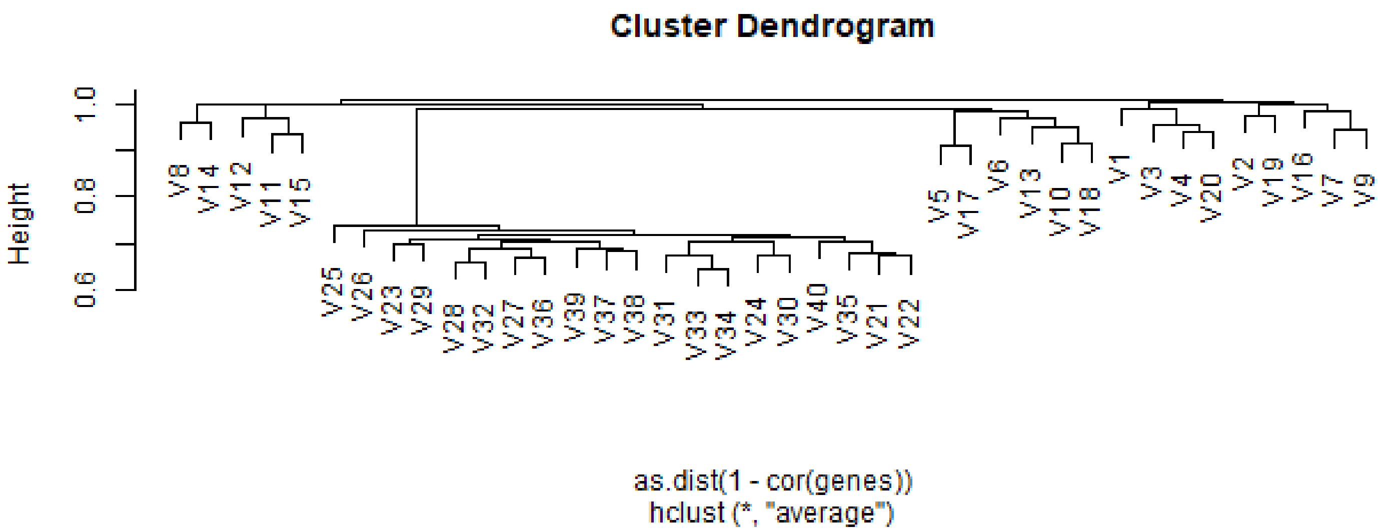
```
hc.complete <- hclust(as.dist(1 - cor(genes)), method = "complete")
plot(hc.complete)
```




```r
hc.single <- hclust(as.dist(1 - cor(genes)), method = "single")
plot(hc.single)
```




```r
hc.average <- hclust(as.dist(1 - cor(genes)), method = "average")
plot(hc.average)
```



The results are pretty different when using different linkage methods as we obtain two clusters for complete and single linkages or three clusters for average cluster.

c.) Your collaborator wants to know which genes differ the most across the two groups. Suggest a way to answer this question, and apply it here. We may use PCA to see which genes differ the most. We will examine the absolute values of the total loadings for each gene as it characterizes the weight of each gene.

```r
pr.out <- prcomp(t(genes))
head(pr.out$rotation)
```

```
              PC1          PC2          PC3          PC4          PC5
[1,] -0.002434664 -0.030745799  0.009359932  0.009699551 -0.012847866
[2,] -0.002016598 -0.025927592  0.050300983 -0.026082885  0.003488293
[3,]  0.011233842 -0.003937802  0.014564920  0.054373032 -0.020411836
[4,]  0.013912855  0.025625408  0.033998676 -0.011530298 -0.009364524
[5,]  0.007293322  0.013590353 -0.008229702 -0.001343010  0.030002978
[6,]  0.017928318 -0.026302699 -0.020728401 -0.024069152 -0.018619253
              PC6          PC7          PC8          PC9         PC10
[1,]  0.023439995  0.010152261 -0.024602570 -0.021925557 -0.035003076
[2,]  0.001605492 -0.037364376 -0.017332292  0.011319311  0.007802611
[3,]  0.025337127  0.070772412  0.047340581 -0.013963868  0.023624407
[4,]  0.029529539  0.002885764 -0.093667774 -0.008391226 -0.019226470
[5,] -0.017042934  0.003555111 -0.053227214 -0.010479774  0.008446406
[6,] -0.049103273 -0.040473304 -0.005455454 -0.003882692  0.028472950
             PC11         PC12         PC13         PC14         PC15
[1,]  0.068133070  0.002322824 -0.050042837 -0.043957087  0.007542896
[2,] -0.092523227  0.036265781  0.002951734  0.021272662 -0.040075267
[3,]  0.017649621  0.021512568  0.013587072  0.005264628 -0.002918920
[4,]  0.006695624  0.025918069 -0.081179098  0.017689681  0.045951951
[5,]  0.053250618 -0.076682641 -0.049516326 -0.003282028  0.060755699
```

```
[6,] -0.018103035  0.015433035  0.015967833 -0.006985293 -0.025237500
            PC16         PC17        PC18         PC19         PC20
[1,] -0.04567334 -0.019899716  0.02946561 -0.009362957 -0.029855408
[2,]  0.03433259  0.003735211 -0.01218600 -0.023466062 -0.005495696
[3,]  0.01881913  0.003284517  0.02597233  0.021581732  0.016808524
[4,] -0.01062858  0.018342677 -0.03334608 -0.052262385 -0.030868339
[5,] -0.02562691  0.049934804 -0.04221058 -0.012279815  0.018004932
[6,] -0.00394582  0.037319024 -0.02541592 -0.029423771 -0.012043007
             PC21          PC22         PC23         PC24         PC25
[1,] -0.009190761  0.0230209664 -0.028970518  0.033060132  0.021453017
[2,] -0.002808309  0.0079065160 -0.007921167 -0.034424716  0.011932971
[3,]  0.010683143 -0.0392265342  0.004592080  0.026463736 -0.038085712
[4,]  0.079419742 -0.0001627164  0.070396594 -0.002015954  0.006459925
[5,] -0.038364004 -0.0230993500 -0.047439556 -0.001129421 -0.001285153
[6,] -0.004522525  0.0304001071  0.016062043 -0.019329595 -0.034486284
             PC26         PC27         PC28         PC29         PC30
[1,]  0.034447853  0.017729906  0.034708970 -0.028136309 -0.009873440
[2,]  0.051079165  0.032435028 -0.006934708 -0.026307151 -0.008143422
[3,] -0.064720318 -0.004616608  0.038015189  0.006455198  0.004570640
[4,]  0.022138389 -0.017120199  0.074901678  0.015812685  0.016391804
[5,] -0.010772594  0.010889806 -0.005305488  0.015248277  0.029303828
[6,]  0.001489549  0.028082907 -0.036617970 -0.054760935  0.023337598
            PC31         PC32        PC33          PC34        PC35
[1,] -0.03576788  0.016708304 -0.01823350  0.0007957941 -0.01443692
[2,] -0.04439239  0.011968530  0.04168309  0.0123210140  0.02739196
[3,]  0.02932866  0.026066011  0.02055204 -0.0716448783  0.02726941
[4,] -0.03954720  0.014714963  0.02846397  0.0316775643  0.01866774
[5,]  0.05494446 -0.005416152  0.03476606  0.0245476439 -0.04037835
[6,]  0.01132569  0.006320203 -0.00237484  0.0061140832  0.01402898
             PC36         PC37         PC38          PC39        PC40
[1,]  0.010652118 -0.009366629 -0.012754402  0.0020214363  0.07000786
[2,] -0.002733484 -0.001318693  0.031410461 -0.0108377476 -0.06326465
[3,]  0.020891497 -0.001380233 -0.025857254  0.0008800921 -0.32824953
[4,] -0.027363133 -0.006080650 -0.025316130 -0.0235404170 -0.01675446
[5,] -0.046869227 -0.017973802  0.002917167  0.0342753219  0.04896111
[6,]  0.042083325  0.055817170 -0.010080327  0.0029965594  0.05407104
```

```r
total.load <- apply(pr.out$rotation, 1, sum)
index <- order(abs(total.load), decreasing = TRUE)
index[1:10]
```

```
 [1] 865  68 911 428 624  11 524 803 980 822
```

```r
rm(list = ls())
```