hw4

Collin

5/15/2021

Question 10

a)

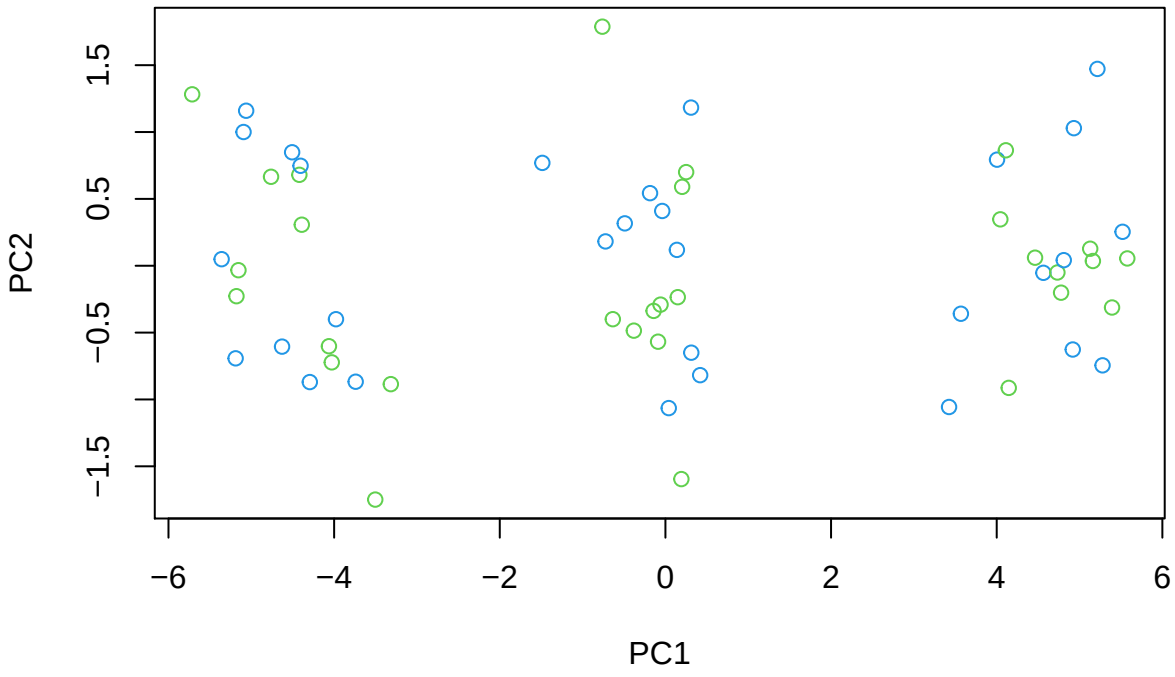
```
#pg 417 ISLR
set.seed(10101)
x_1 = matrix(rnorm(20*20,mean = 10,sd = 5),ncol = 20)
x_2 = matrix(rnorm(20*20,mean = 20,sd = 5),ncol = 20)
x_3 = matrix(rnorm(20*20,mean = 30,sd = 5),ncol = 20)

data = data.frame(x = rbind(x_1,x_2,x_3))
```

b)

```
pca = prcomp(data, scale = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4    PC5     PC6     PC7
## Standard deviation     3.845 0.75829 0.71520 0.68083 0.6648 0.64673 0.63679
## Proportion of Variance 0.739 0.02875 0.02558 0.02318 0.0221 0.02091 0.02028
## Cumulative Proportion  0.739 0.76777 0.79335 0.81652 0.8386 0.85953 0.87981
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.58368 0.56515 0.52944 0.50502 0.46663 0.42799 0.41521
## Proportion of Variance 0.01703 0.01597 0.01402 0.01275 0.01089 0.00916 0.00862
## Cumulative Proportion  0.89684 0.91281 0.92683 0.93958 0.95047 0.95962 0.96824
##                           PC15    PC16    PC17    PC18    PC19    PC20
## Standard deviation     0.38104 0.35952 0.34129 0.32593 0.28518 0.23797
## Proportion of Variance 0.00726 0.00646 0.00582 0.00531 0.00407 0.00283
## Cumulative Proportion  0.97550 0.98197 0.98779 0.99310 0.99717 1.00000
```

```
plot(pca$x[,1:2],col = 3:4)
```



c)

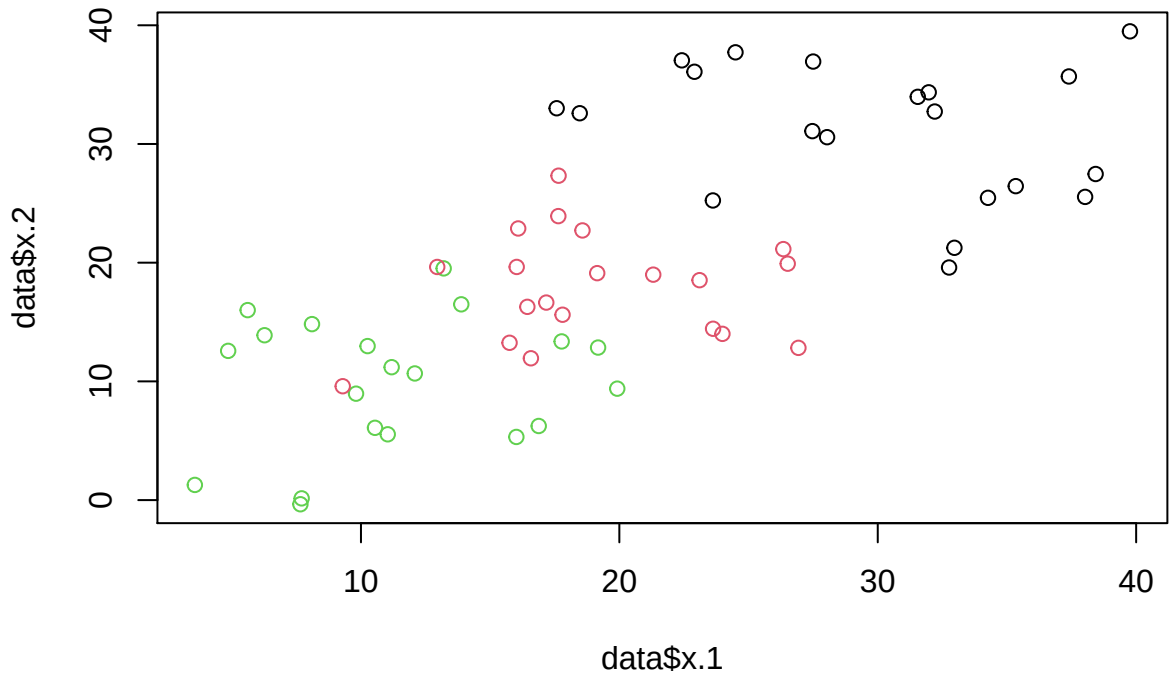
```
#k-means algorithm with k = 3
kmeans = kmeans(x = data, centers = 3, nstart = 20)
kmeans$cluster
```

```
##  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [39] 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
class = c(rep(2,20),rep(3,20),rep(1,20))
table(kmeans$cluster,class)
```

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

```
plot(data$x.1,data$x.2,col = kmeans$cluster)
```



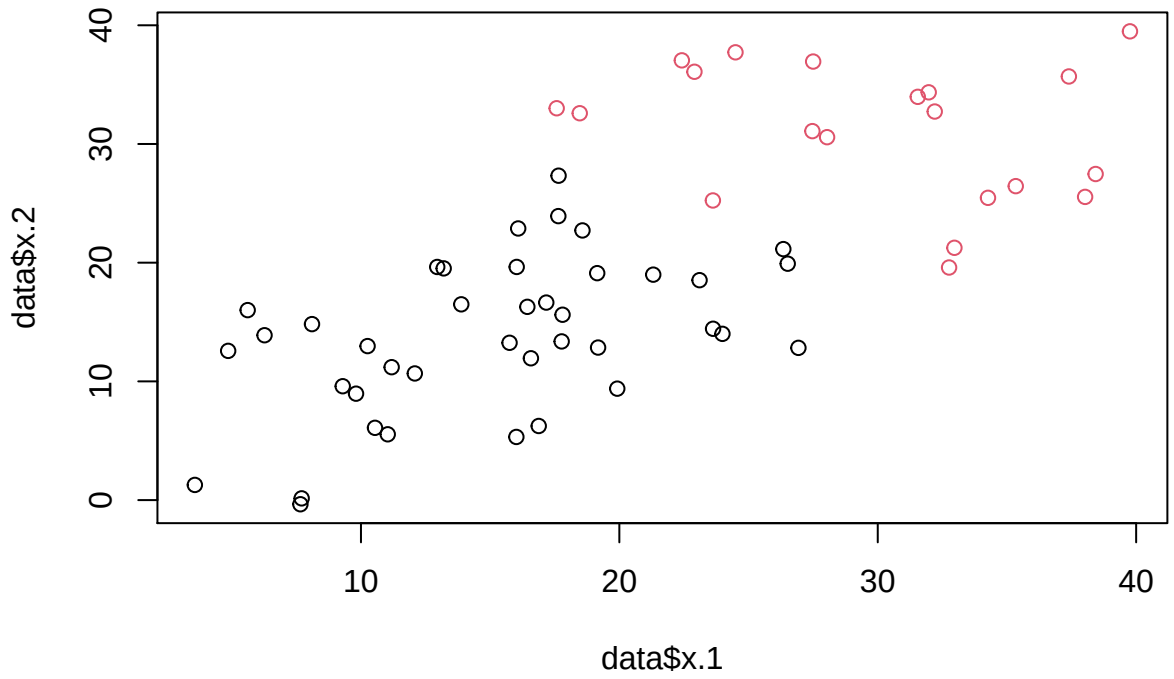
The algorithm here at least *appears* to perform decently. The data in the lower, middle, and upper right hand side of the plot, which are presumably different, are differientiated by color, indicating that the algorithm picked up on differences between them.

d)

```
#with k = 2
kmeans2 = kmeans(x = data, centers = 2, nstart = 20)



class = c(rep(1,20),rep(2,20),rep(3,20))
table(kmeans2$cluster,class)
```

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

```
plot(data$x.1,data$x.2,col = kmeans2$cluster)
```



Without knowing that this was simulated data, I could probably argue this is a promising result. The data is clearly split into a lower-left and upper-right region, and an argument could be made that the k-means clustering picked up on mean differences between the *two* groups. However, because this data was simulated, we know that there are actually 3 groups here, and that one of the groups has been absorbed/distributed amongst other two. This pretty much highlights the danger of k-means clustering, as it is highly dependent on the hyperparameter `k`.

e

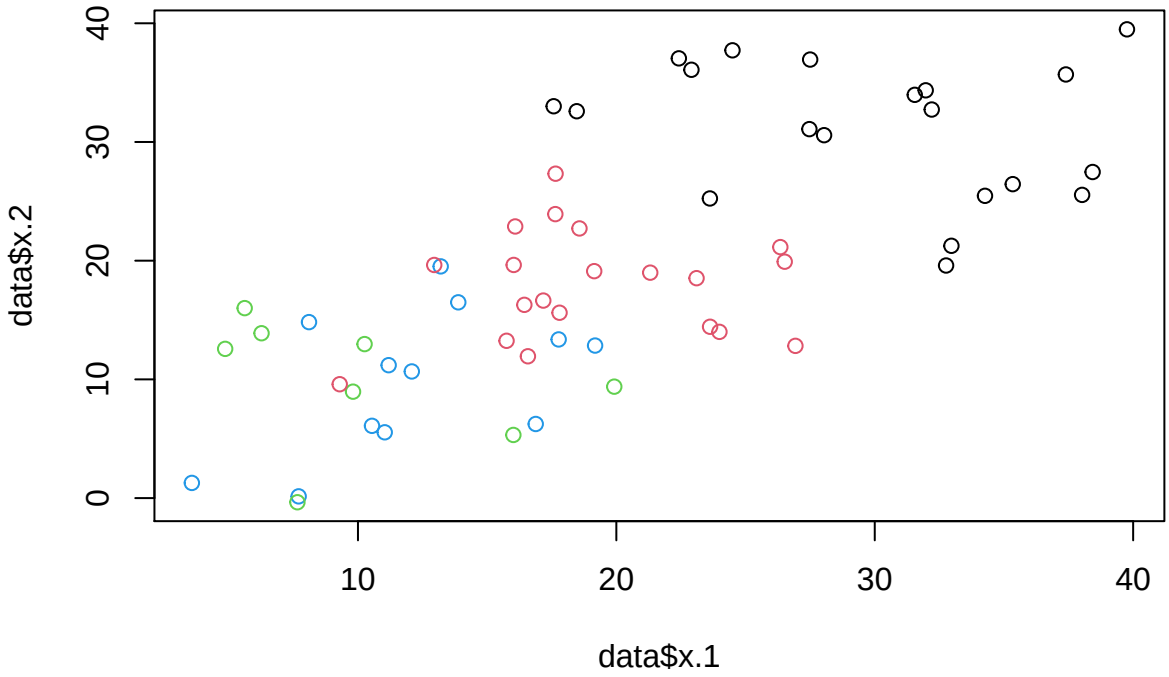
```
kmeans4 = kmeans(x = data, centers = 4, nstart = 20)
kmeans4$cluster
```

```
##  [1] 3 3 4 4 4 3 4 4 4 4 3 3 3 3 4 4 4 3 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [39] 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
class = c(rep(2,20),rep(1,20),rep(3,20))
table(kmeans4$cluster,class)
```

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

```
plot(data$x.1,data$x.2,col = kmeans4$cluster)
```



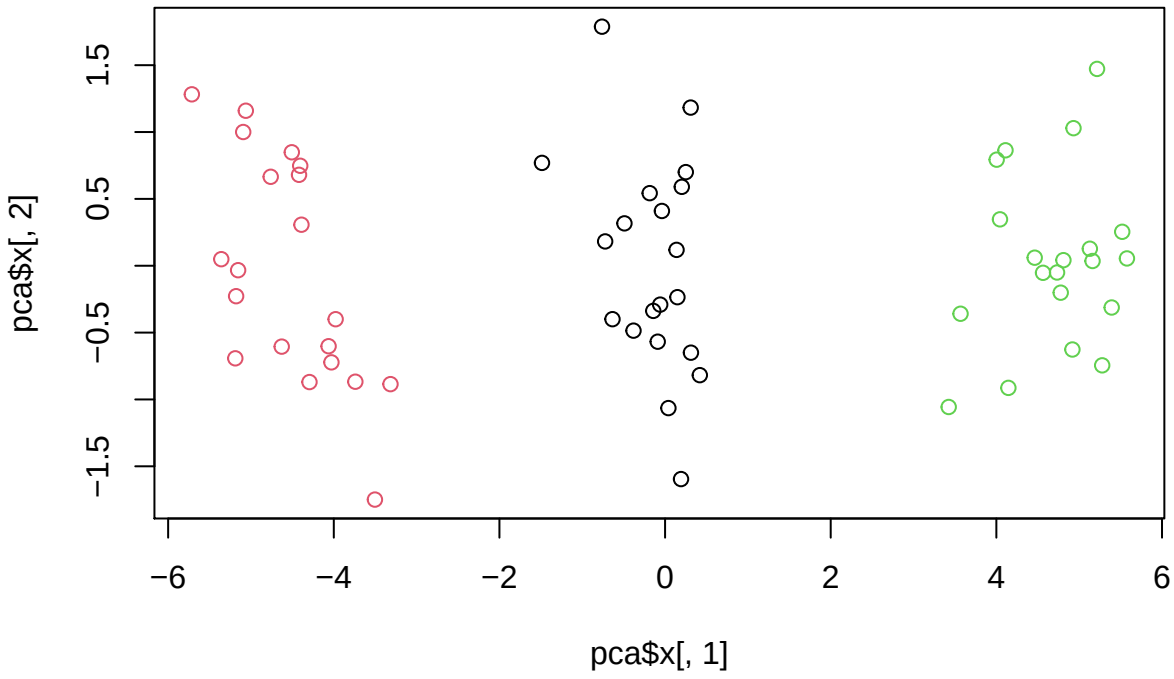
Here we make a similar error but in the opposite direction of the one we made in the previous problem. Whereas in the previous problem we had two few groups, Here we group into 1 too many groups. The fact that it is "wrong" is perhaps a little more obvious though since there is significant overlap amongst the data in the lower-left region of the plot.

f)

```
#k-means on PCs
kmeans_pca = kmeans(x = pca$x[,1:2], centers = 3, nstart = 20)

# kmeans_pca$cluster
table(kmeans_pca$cluster,c(rep(3,20),rep(2,20),rep(1,20)))
```

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

```
plot(pca$x[,1], pca$x[,2], col = kmeans_pca$cluster)
```



Having performed principal component analysis on the `data`, the correlation amongst the variables, which are now principal components, has been eliminated. The algorithm performs admirably here since the principal components are very different which is easy for the algorithm to discern.

2: Repeating Analysis using Empirical Project Data

b repeated)

```
library(AER)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

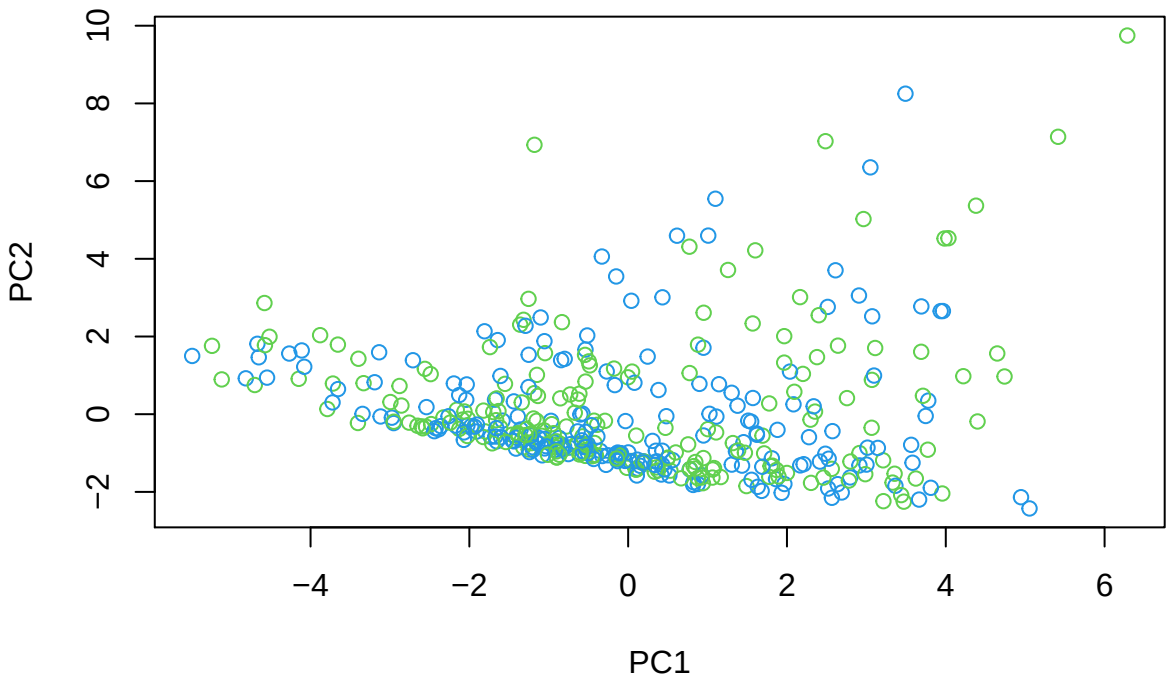
```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

```
data("CASchools")


#drop all non-numerical variables from dataset
CASchools = CASchools %>%
  select(-c(district,school,county,grades))

pca2 = prcomp(CASchools, scale = TRUE)
summary(pca2)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6    PC7
## Standard deviation     2.121 1.6940 1.0537 0.81896 0.59608 0.49545 0.3332
## Proportion of Variance 0.450 0.2870 0.1110 0.06707 0.03553 0.02455 0.0111
## Cumulative Proportion  0.450 0.7369 0.8479 0.91502 0.95055 0.97509 0.9862
##                           PC8     PC9    PC10
## Standard deviation     0.28967 0.22733 0.04930
## Proportion of Variance 0.00839 0.00517 0.00024
## Cumulative Proportion  0.99459 0.99976 1.00000
```

```
plot(pca2$x[,1:2],col = 3:4)
```



The first principal component explains about 45% of the total variation. It's not until the 4th Principal component is considered that about 91% of the proportion of variance is explained.

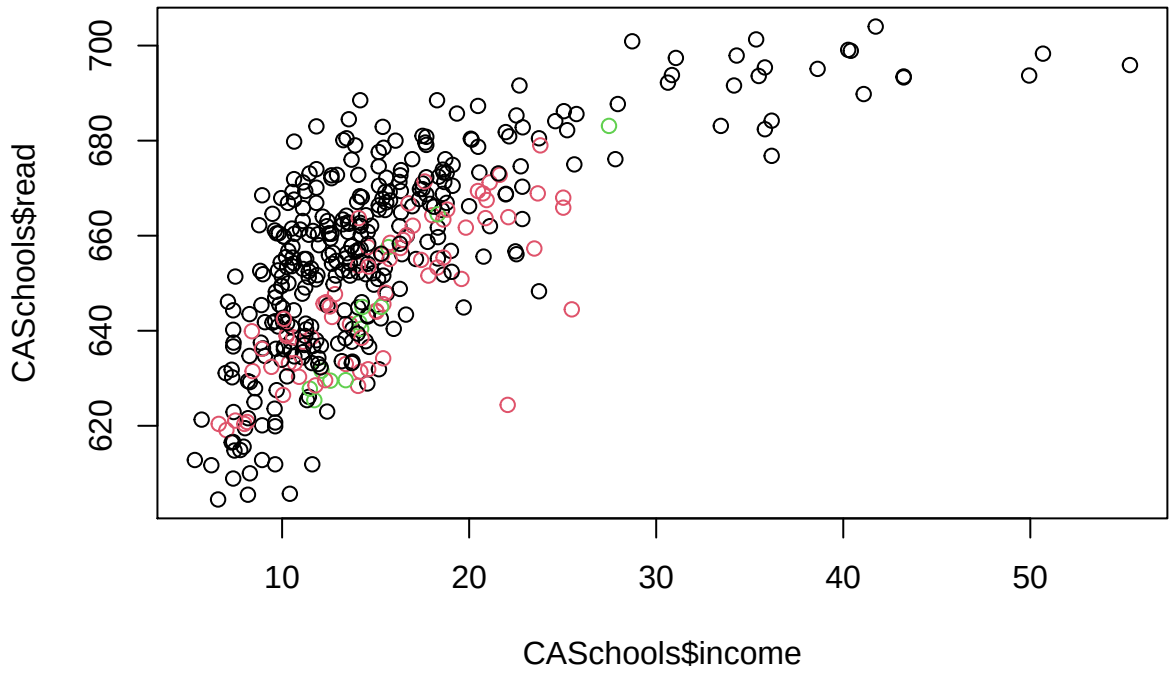
repeat c)

```
kmeans = kmeans(x = CASchools, centers = 3, nstart = 20)


class = c(rep(1,420/3),rep(2,420/3),rep(3,420/3))
table(kmeans$cluster,class)
```

```
##    class
```

```
##        1   2   3
##   1 101 112 122
##   2  32  25  16
##   3   7   3   2
```

```
plot(CASchools$income,CASchools$read,col = kmeans$cluster)
```

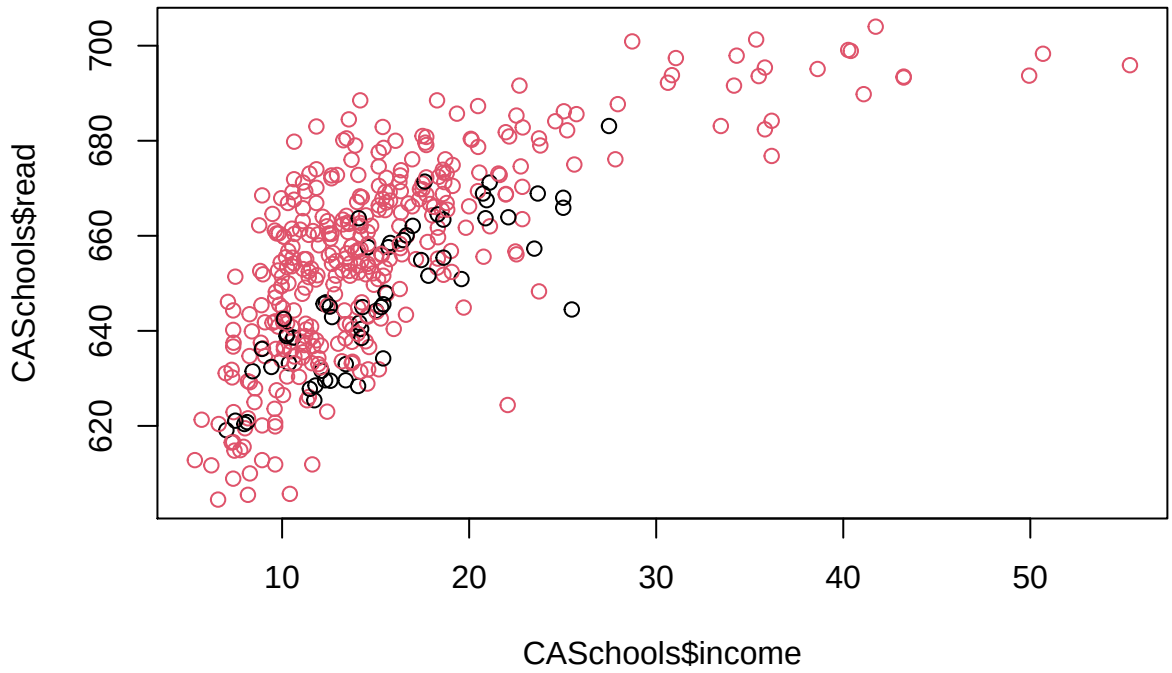


Using my data from the empirical project, I perform kmeans clustering on the entire dataset (minus any non-numerical variables). Clearly, it is somewhat of a mess. This may be due to the fact we are trying to establish 3 groups in this data, and that may not be the actual case. repeat d)

```
#with k = 2
kmeans2 = kmeans(x = CASchools, centers = 2, nstart = 20)



class = c(rep(1,210),rep(2,210))
table(kmeans2$cluster,class)
```

```
##    class
##       1   2
##   1  37  22
##   2 173 188
```

```
plot(CASchools$income,CASchools$read,col = kmeans2$cluster)
```



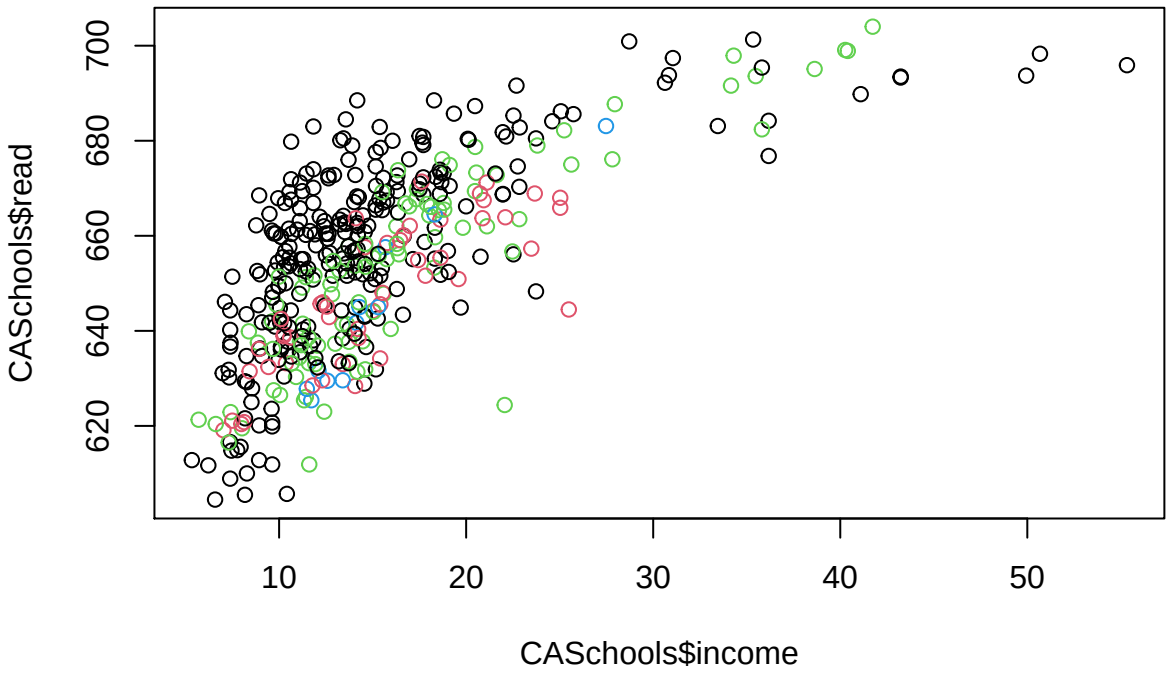
We see a similar thing here. There is lots of overlap in the data between the two groups,indicating there are no clear cut differences between groups (at least when k = 2).

repeat e)

```
kmeans4 = kmeans(x = CASchools, centers = 4, nstart = 20)



class = c(rep(2,420/4),rep(1,420/4),rep(4,420/4),rep(3,420/4))
table(kmeans4$cluster,class)
```

```
##    class
##      1  2  3  4
##   1 65 58 76 67
##   2 16 13  4 15
##   3 21 29 24 21
##   4  3  5  1  2
```

```
plot(CASchools$income,CASchools$read,col = kmeans4$cluster)
```



Here we impose k = 4 groups on the data, and the result doesnt appear much better. This may simply be due to the fact that this data cannot be so easily distributed into 4 groups. There may be too many overlapping characteristics between observations.

repeat f)

```
#k-means on PCs
kmeans_pca = kmeans(x = pca2$x[,1:2], centers = 3, nstart = 20)

kmeans_pca$cluster
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   3   3   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2   1
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   1   1   1   1   1   1   2   1   1   1   1   1   2   1   2   1   1   1   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   2   1   1   1   1   1   2   1   2   2   2   1   2   2   1   1   1   1   1   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   1   1   2   1   1   1   1   1   1   1   1   1   1   2   1   1   1   1   1   1
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   1   1   1   1   1   1   1   2   2   1   1   1   1   1   1   1   1   1   1   1
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   1   1   1   1   1   1   2   1   1   1   1   2   1   1   1   1   2   2   1   1
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   1   1   1   1   1   1   1   1   1   1   2   1   1   2   3   2   3   1   1   1   2
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   1   1   2   1   1   1   1   1   1   1   1   1   1   1   1   1   2   1   2   2
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   1   1   3   2   2   3   3   1   2   1   3   1   1   1   1   1   3   1   3   3
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   1   3   1   1   1   1   1   3   3   1   2   3   2   1   3   1   1   3   3   3
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
##   3   3   3   3   1   1   3   1   3   1   1   1   3   3   3   1   3   3   3   2
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
```

```
##   1   3   3   3   3   3   3   3   3   3   3   1   3   3   2   2   3   3   3   1
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   3   3   3   3   3   3   3   3   2   3   3   3   3   3   3   3   3   1   3   3
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280
##   3   1   3   2   3   2   3   3   3   3   3   3   1   3   3   3   3   3   3   3
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   2   3   2   3   3
## 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320
##   3   3   3   2   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   2   3
## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
```

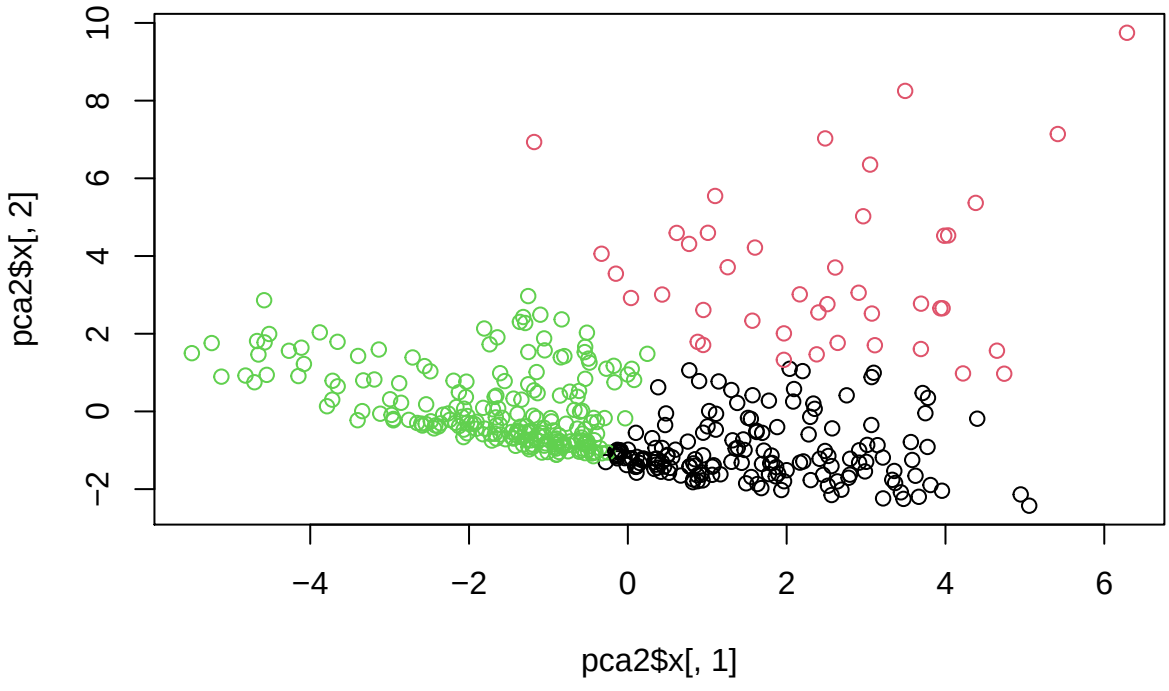
```
length(kmeans_pca$cluster)
```

```
## [1] 420
```

```
# kmeans_pca$cluster
table(kmeans_pca$cluster,c(rep(1,420/3),rep(3,420/3),rep(2,420/3)))
```

```
##
##       1   2   3
##   1 113   0  49
##   2  23   4  15
##   3   4 136  76
```

```
plot(pca2$x[,1], pca2$x[,2], col = kmeans_pca$cluster)
```



kmeans clustering on the principal components appears to work much better than in the case where PCA was not conducted. Similarly too when we did this in the simulated data case, multicollinearity between predictors is eliminated during the dimension reduction process of PCA, as the principal components are no longer correlated. While they now clearly fit into three groups, from an interpretation standpoint, this really means nothing as principal components cannot be interpreted the same way that regular predictor variables can be interpreted, as the principal components are an amalgam of the various predictors in the original data set.