SYS 6018 | Fall 2019 | University of Virginia

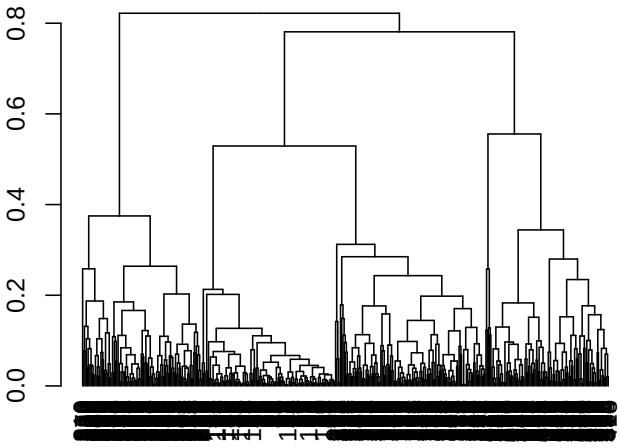
Problem 4.1: Clustering

The data for this problem are here: https://raw.githubusercontent.com/mdporter/SYS6018/master/data/clusthw.csv

```
url <- "https://raw.githubusercontent.com/mdporter/SYS6018/master/data/clusthw.csv"
destfile <- "4_1Data.csv"
curl::curl_download(url, destfile)
Cluster_Data <- read.csv(destfile)
```

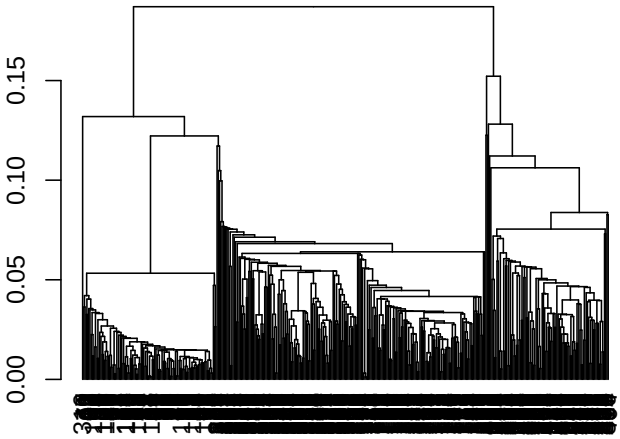
1. Run Hierarchical clustering, using Euclidean distance, and two linkage methods (of your choice). Show the resulting dendrograms.

```
d <- dist(Cluster_Data, method = "euclidean")
hc <- hclust(d, method = "average")
hc_single <- hclust(d, method = "single")
hc_complete <- hclust(d, method = "complete")
(plot(as.dendrogram(hc)))
```



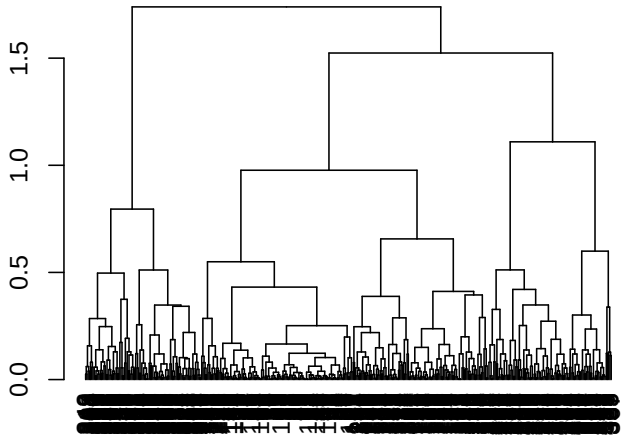

```
#> NULL
```

```
(plot(as.dendrogram(hc_single)))
```




```
#> NULL
```

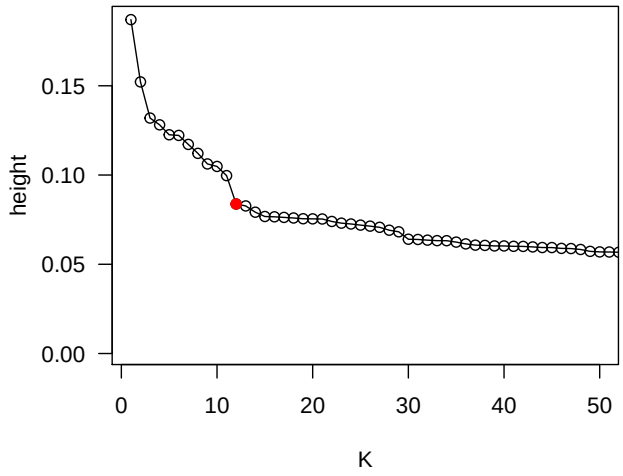
```
(plot(as.dendrogram(hc_complete)))
```




```
#> NULL
```

2. Estimate K for one of the linkage methods from part a. Explain why you chose that value of K.

```
n = length(hc_single$height)
plot(n:1, hc_single$height, type='o', xlab="K", ylab="height", las=1,
     xlim=c(1, 50))
points(12, hc_single$height[n-11], col="red", pch=19)
```




```
yhat <- cutree(hc_single, h = .06)

k <- length(unique(yhat))

k
```

```
#> [1] 43
```

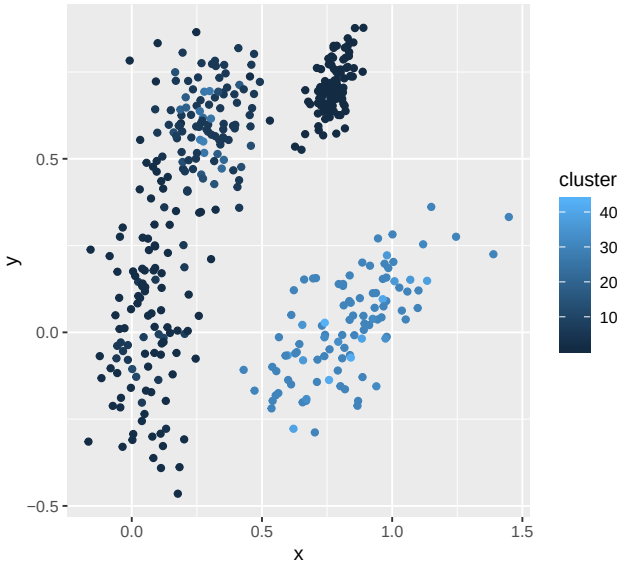
I chose K = 43 because the dissimilarity between clusters seems to spike around h = .6 based on visual analysis of the dendrogram. Cutting the tree at h = .6 yields 43 clusters.

3. Show a scatterplot of the data using colors to denote the K clusters. Based on a visual analysis, does it appear that K means was successful? Explain any changes to the clustering that you think should be made.

```
hc_single$order
```

```
#>   [1] 310  47  71  41  99  97  93   7  90 102   4  12  86  96   8  30  72
#>  [18] 105   1 100  40  98  49  56  81  57  11  92  24  13  88  29   5  64
#>  [35]  80 104 109  26  68  35  87  73  75  37  38 108  45  74  27  32  22
#>  [52]   2  84  14  61  18 107  44  70  31  62  79  58  52  65  77  82  23
#>  [69]  59  69  95  67  20  28  21  43   3  33  60  25  17 106  36  39  16
#>  [86]  15  53  63  83  78 103  48  54  46  50  10  42  91   9 101  51  66
#> [103]  19  94   6  34  55  85  76  89 304 260 237 167 289 224 199 311 257
#> [120] 282 188 204 269 278 279 272 264 280 293 270 315 258 241 267 230 307
#> [137] 303 318 283 245 288 240 176 317 228 294 325 185 262 298 164 313 281
#> [154] 305 255 273 320 233 218 292 253 247 277 330 266 271 238 246 263 324
#> [171] 295 252 316 226 302 219 286 323 308 327 220 285 236 251 321 249 290
#> [188] 254 234 244 296 248 284 242 319 256 261 326 276 328 314 329 221 223
#> [205] 232 231 297 301 275 299 306 309 227 300 243 229 287 312 222 265 268
#> [222] 291 250 239 259 152 143 200 134 156 177 198 322 178 118 274 149 154
#> [239] 168 172 196 131 193 125 207 175 114 208 161 183 130 180 163 169 162
#> [256] 170 139 190 151 181 192 110 201 174 120 202 146 123 187 215 147 205
#> [273] 142 173 194 212 140 150 211 186 217 158 124 203 132 182 129 133 213
#> [290] 112 121 197 155 127 159 122 184 126 214 153 115 148 191 206 166 145
#> [307] 179 111 171 117 135 136 216 189 165 119 144 113 137 141 210 116 138
#> [324] 195 128 160 157 209 225 235 413 427 363 335 367 388 383 366 379 344
#> [341] 420 397 341 425 392 431 362 415 374 403 359 400 368 357 386 354 373
#> [358] 384 418 389 399 401 406 338 429 419 375 416 396 339 409 337 360 395
#> [375] 340 380 391 364 331 414 426 390 370 358 372 346 361 371 424 411 356
#> [392] 407 410 421 369 398 382 336 342 348 422 428 332 393 343 353 423 381
#> [409] 394 405 347 355 365 376 349 387 402 378 345 412 385 351 408 333 352
#> [426] 334 430 377 417 350 404
```

```r
clusters <- data.frame(index = hc_single$order, cluster = yhat) %>% arrange(index)
Cluster_Data$cluster <- clusters$cluster

ggplot(Cluster_Data, mapping = aes(x = x, y = y, col = cluster)) +

  geom_point()
```



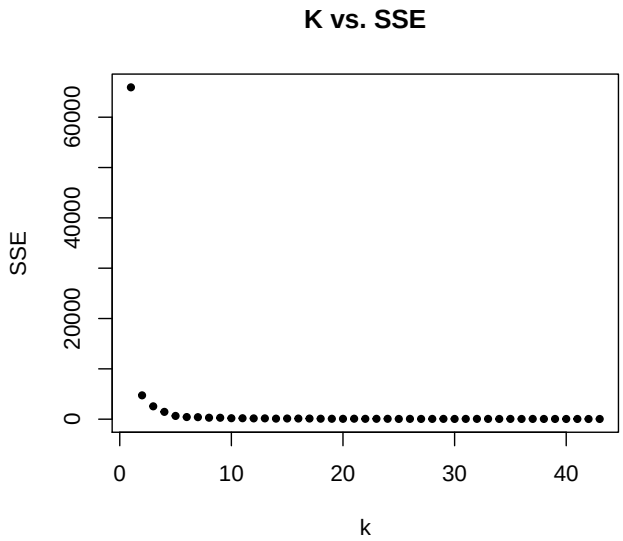
43 clusters appear to be quite excessive; the number should probably be reduced to 3 or 4.

4. Run K-means for a sequence of K values. Plot the sum of squared errors (SSE) as a function of K.

```r
K <- tibble(k = seq(1,43))

SSE <- vector()

for (k in seq(1,43)) {

  km.out = kmeans(Cluster_Data,k,nstart = 20)
  SSE <- c(SSE, km.out$tot.withinss)

}

K$SSE <- SSE

K
```

```
#> # A tibble: 43 x 2
#>        k    SSE
#>    <int>  <dbl>
#>  1     1 65935.
#>  2     2  4733.
#>  3     3  2553.
#>  4     4  1450.
#>  5     5   643.
#>  6     6   422.
#>  7     7   394.
#>  8     8   289.
#>  9     9   279.
#> 10    10   187.
#> # ... with 33 more rows
```

```r
plot(K,pch = 20, cex = 1, main = "K vs. SSE")
```



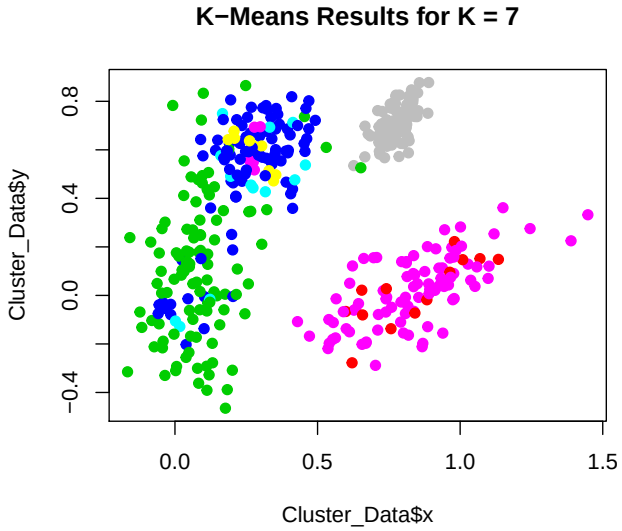
5. Estimate K. Explain why you chose that value of K.

I estimate K = 7 because the "K vs. SSE" plot above indicates that further partitions do not significantly reduce the total within-cluster SSEs.

6. Show a scatterplot of the data using colors to denote the K clusters. Based on a visual analysis, does it

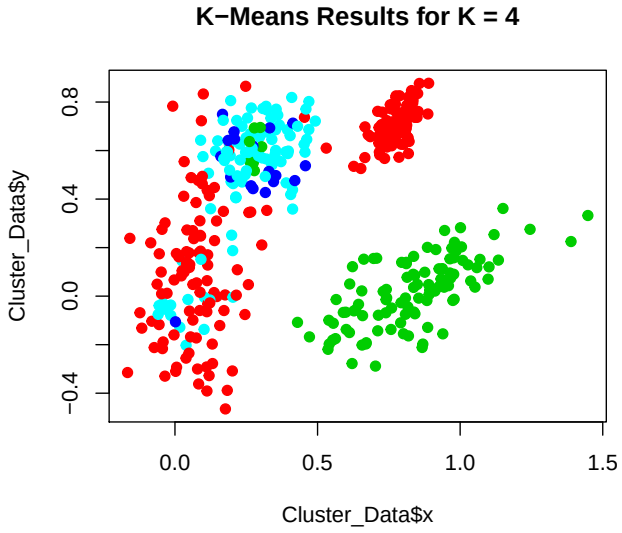
appear that K means was successful? Explain any changes to the clustering that you think should be made.

```
km.out = kmeans(Cluster_Data,7,nstart = 20)
plot(x = Cluster_Data$x, y = Cluster_Data$y, col=(km.out$cluster + 1), main = "K-Means Results for K = 
```



K means with K = 7 was somewhat successful – it correctly segmented the small, dense cluster – but seems to have oversegmented the data in some places. The clusters in lower end of x space overlap. There are a few outliers in x space which may be skewing results. I think the true clustering is probably closer to K = 3 or K = 4, but K means fails to accurately partition the data at that level (see additional graph below)

```
km.out = kmeans(Cluster_Data,4,nstart = 20)
plot(x = Cluster_Data$x, y = Cluster_Data$y, col=(km.out$cluster + 1), main = "K-Means Results for K = 
```



Problem 4.2: Activity Recognition Challenge

A current engineering challenge is to identify/classify human activity (e.g., walking, in car, on bike, eating, smoking, falling) from smartphones and other wearable devices. More specifically, the embedded sensors (e.g., accelerometers and gyroscopes) produce a time series of position, velocity, and acceleration measurements. These time series are then processed to produce a set of *features* that can be used for activity recognition. We will use a subset of such features to cluster an activity dataset. The dataset `activity.csv` contains six

features that correspond to K human activities. Your challenge is to cluster these data.

You can use any clustering method you like. You are free to transform or pre-process the data.

This will be a contest, so you will submit your cluster scores and we will evaluate how closely your clusters match the true activities. The reported clusters will be evaluated by the *Adjusted Rand index (ARI)*. You will receive credit for a proper submission and code; the top five scores will receive 2 bonus points.

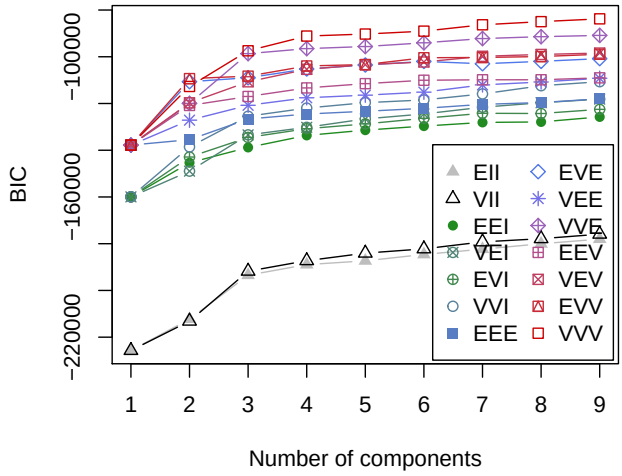
`activity.csv`: https://raw.githubusercontent.com/mdporter/SYS6018/master/data/activity.csv/activity.csv

a. Submit cluster labels for all observations. Your file should be a .txt with no header and no row numbers (i.e., 5000 rows, 1 column). Name the file `lastname_firstname.txt`. We will use automated evaluation, so the format must be exact.

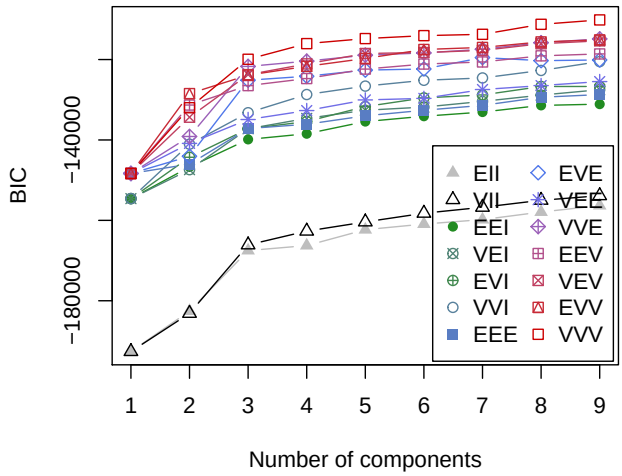
```
#-- Example of making submission data.
# est.label is a vector of labels (length 5000)
#write.table(est.l, file="save.dir/lastname_firstname.txt",
         # row.names=FALSE, col.names=NA)
```

b. Show your code.

```
BIC <- mclustBIC(tracking)
BIC_2 <- mclustBIC(dplyr::select(tracking, X1, X2, X3, X4, X5))
plot(BIC)
```




```
plot(BIC_2)
```



```r
mod <- Mclust(tracking, x = BIC)
mod_2 <- Mclust(tracking, x = BIC_2)
mod_3 <- Mclust(tracking, G = 11)
?Mclust()
summary(mod, parameters = TRUE)
```

```
#> ----------------------------------------------------
#> Gaussian finite mixture model fitted by EM algorithm
#> ----------------------------------------------------
#>
#> Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model
#> with 9 components:
#>
#>  log-likelihood    n  df    BIC       ICL
#>       -40816.09 5000 251 -83770 -84597.21
#>
#> Clustering table:
#>   1   2   3   4   5   6   7   8   9
#> 517 334 820 861 870 316 556 412 314
#>
#> Mixing probabilities:
#>         1         2         3         4         5         6
#> 0.10852156 0.06744501 0.17104498 0.16415490 0.16771913 0.06354846
#>         7         8         9
#> 0.11257677 0.08227870 0.06271048
#>
#> Means:
#>          [,1]       [,2]       [,3]        [,4]      [,5]       [,6]
#> X1  6.0193586 17.7111097 -20.843942 -14.5566351  4.055529 16.6718552
#> X2 12.6392576 -3.8487425  -8.117756  -0.9236948 16.754807  5.1581506
#> X3 -3.3095239  2.9098063  -2.986847  -3.4845797  1.616987 -1.9466420
#> X4 -0.9006493  6.9661104  -1.817429   3.5670881 -2.813193 -5.4667794
#> X5  1.0075218  0.3558314   1.099868   0.8926152  1.002064  0.8836346
#> X6  1.0073750  0.3554221   1.087230   0.9046378  1.002002  0.8837848
#>           [,7]         [,8]         [,9]
#> X1 -23.5958383  34.13673215  34.16720313
#> X2  -5.3225949 -13.42025085 -14.03682671
#> X3   7.3528290   8.07940868  -6.44726840
#> X4   1.8854043  -0.56922049   0.68623515
#> X5   0.9544854   0.06590634   0.09881940
#> X6   0.9568064   0.06619007   0.09887486
#>
#> Variances:
#> [,,1]
#>              X1           X2          X3           X4            X5
#> X1  14.22655967 -3.109768771 -1.41124797 -13.53873405 -0.0245423563
#> X2  -3.10976877  4.424725908  1.47445903  -0.17835465  0.0102574269
#> X3  -1.41124797  1.474459028  8.28813546  -0.66561960  0.0288094154
#> X4 -13.53873405 -0.178354647 -0.66561960  23.42975236  0.0239519740
#> X5  -0.02454236  0.010257427  0.02880942   0.02395197  0.0002928617
#> X6  -0.02520590  0.009776162  0.02814066   0.02658546  0.0002909248
#>               X6
#> X1 -0.0252058952
#> X2  0.0097761623
```

```
#> X3  0.0281406596
#> X4  0.0265854599
#> X5  0.0002909248
#> X6  0.0003197951
#> [,,2]
#>             X1          X2          X3         X4         X5         X6
#> X1  147.421780 -144.327690 -49.4241579   1.556161 -3.2988926 -3.2895036
#> X2 -144.327690  178.894937  67.2673327 -27.839669  3.7614761  3.7647480
#> X3  -49.424158   67.267333 225.7797423  13.512717 -0.6037209 -0.6020508
#> X4    1.556161  -27.839669  13.5127168  46.143433 -1.4932266 -1.4910134
#> X5   -3.298893    3.761476  -0.6037209  -1.493227  0.1896321  0.1891918
#> X6   -3.289504    3.764748  -0.6020508  -1.491013  0.1891918  0.1891061
#> [,,3]
#>            X1         X2          X3         X4          X5          X6
#> X1 34.6516259 29.2390353 -8.24912568 20.5762442  0.38052739  0.28213047
#> X2 29.2390353 38.2177263 -3.46415649 26.1217053  0.42615223  0.30904315
#> X3 -8.2491257 -3.4641565  7.16718165 -6.2123601 -0.05982603 -0.08155951
#> X4 20.5762442 26.1217053 -6.21236013 27.8159820  0.26220416  0.21496800
#> X5  0.3805274  0.4261522 -0.05982603  0.2622042  0.10891992  0.09643403
#> X6  0.2821305  0.3090431 -0.08155951  0.2149680  0.09643403  0.11345639
#> [,,4]
#>             X1         X2           X3          X4          X5          X6
#> X1  23.2124359  5.7837537 -12.98155110  4.78686758  0.23387881  0.20267602
#> X2   5.7837537 13.8115212   1.05256176 11.13333773  0.18350229  0.17341173
#> X3 -12.9815511  1.0525618  14.68889825 -4.45530792 -0.02649743 -0.01459618
#> X4   4.7868676 11.1333377  -4.45530792 21.57309544  0.08082378  0.05804870
#> X5   0.2338788  0.1835023  -0.02649743  0.08082378  0.03443475  0.03323407
#> X6   0.2026760  0.1734117  -0.01459618  0.05804870  0.03323407  0.03527865
#> [,,5]
#>             X1         X2          X3          X4            X5
#> X1  7.46171647 -4.2747561 -3.91499184 -6.83327344  0.0537948482
#> X2 -4.27475608  3.7638312  3.67252883  1.68748641 -0.0369765967
#> X3 -3.91499184  3.6725288 15.60380632  0.54542424 -0.0467055015
#> X4 -6.83327344  1.6874864  0.54542424 14.56808607 -0.0412032483
#> X5  0.05379485 -0.0369766 -0.04670550 -0.04120325  0.0006212405
#> X6  0.05363322 -0.0367824 -0.04640812 -0.04110200  0.0006156194
#>               X6
#> X1  0.0536332204
#> X2 -0.0367824037
#> X3 -0.0464081177
#> X4 -0.0411019971
#> X5  0.0006156194
#> X6  0.0006143390
#> [,,6]
#>             X1          X2          X3          X4          X5          X6
#> X1  17.0903208 -15.2039824 -2.27352704 -1.04433927 -0.41959607 -0.41895718
#> X2 -15.2039824  15.6863091  3.26575747 -3.19968274  0.40646362  0.40535462
#> X3  -2.2735270   3.2657575  8.42028568 -4.77268297  0.07014124  0.06926230
#> X4  -1.0443393  -3.1996827 -4.77268297 15.44550572 -0.04256352 -0.04137978
#> X5  -0.4195961   0.4064636  0.07014124 -0.04256352  0.01105416  0.01103191
#> X6  -0.4189572   0.4053546  0.06926230 -0.04137978  0.01103191  0.01101395
#> [,,7]
#>            X1          X2         X3         X4          X5          X6
#> X1 82.3554036  81.6126243 -8.6259633 43.7798216  0.26072000  0.20152520
```

```
#> X2 81.6126243 100.3737927 -4.6031655 67.4650312  0.20364127  0.13356151
#> X3 -8.6259633  -4.6031655  7.2572076  4.2069397 -0.11953640 -0.12311621
#> X4 43.7798216  67.4650312  4.2069397 69.3774417 -0.10885146 -0.16431795
#> X5  0.2607200   0.2036413 -0.1195364 -0.1088515  0.09192514  0.08433968
#> X6  0.2015252   0.1335615 -0.1231162 -0.1643180  0.08433968  0.09160002
#> [,,8]
#>            X1         X2         X3         X4          X5          X6
#> X1  6.5310337 -6.1178361 -9.9118622 -1.6665732 -0.23848722 -0.23878867
#> X2 -6.1178361  8.8105375 10.6064086 -4.0560672  0.35217353  0.35221571
#> X3 -9.9118622 10.6064086 37.7954339 -1.6208462  0.11817672  0.11870423
#> X4 -1.6665732 -4.0560672 -1.6208462 15.1170618 -0.17112979 -0.17105159
#> X5 -0.2384872  0.3521735  0.1181767 -0.1711298  0.01920367  0.01920379
#> X6 -0.2387887  0.3522157  0.1187042 -0.1710516  0.01920379  0.01921183
#> [,,9]
#>            X1         X2          X3           X4           X5
#> X1  3.8186613 -2.2670648  3.66721243 -2.861192548 -0.180952877
#> X2 -2.2670648  3.1276544 -1.51705757 -0.931193681  0.190888070
#> X3  3.6672124 -1.5170576 10.89084995 -6.212403906 -0.056207702
#> X4 -2.8611925 -0.9311937 -6.21240391 12.438775928 -0.008256577
#> X5 -0.1809529  0.1908881 -0.05620770 -0.008256577  0.014289529
#> X6 -0.1809044  0.1908947 -0.05655685 -0.007500226  0.014277579
#>             X6
#> X1 -0.180904385
#> X2  0.190894687
#> X3 -0.056556848
#> X4 -0.007500226
#> X5  0.014277579
#> X6  0.014272744
```

```
summary(mod_2, parameters = TRUE)
```

```
#> ----------------------------------------------------
#> Gaussian finite mixture model fitted by EM algorithm
#> ----------------------------------------------------
#>
#> Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model
#> with 9 components:
#>
#>  log-likelihood    n  df       BIC       ICL
#>       -40666.55 5000 251 -83470.92 -84207.55
#>
#> Clustering table:
#>    1    2    3    4    5    6    7    8    9
#>  730  704  372  249  564  338 1149  374  520
#>
#> Mixing probabilities:
#>          1          2          3          4          5          6
#> 0.14284063 0.13971034 0.07414621 0.05096397 0.12341434 0.06869757
#>          7          8          9
#> 0.21921842 0.07670022 0.10430830
#>
#> Means:
#>         [,1]         [,2]       [,3]         [,4]       [,5]       [,6]
#> X1  6.206220  34.28708268 15.6091325  24.42021700 -23.016811  5.1325541
#> X2 14.747348 -13.68360037  6.4957162 -10.56663987 -10.318054 10.5777669
```

```
#> X3 -1.017722   1.63158722 -2.9131863   4.20968380  -2.029380 -1.9000425
#> X4 -3.304168  -0.21936184 -5.7970173   8.34729377  -3.465007  1.8891604
#> X5  1.016653   0.08271451  0.9022096   0.07043812   1.181574  0.9834005
#> X6  1.016567   0.08288973  0.9022353   0.07032736   1.160082  0.9831608
#>           [,7]       [,8]        [,9]
#> X1 -15.9258405  1.3696448 -21.8278588
#> X2  -2.6546955 18.4204719  -2.8368038
#> X3  -3.4942969  3.8705500   7.2756562
#> X4   2.2971622 -0.8498919   3.8586793
#> X5   0.9028417  0.9802618   0.9350716
#> X6   0.9118582  0.9801815   0.9422193
#>
#> Variances:
#> [,,1]
#>               X1            X2          X3           X4            X5
#> X1  3.9879574533 -1.9717493238  0.88433624 -3.767428899  4.704122e-04
#> X2 -1.9717493238  2.8009778081  1.11895874 -1.262310359  9.137598e-04
#> X3  0.8843362376  1.1189587425  9.85209197 -4.438004576  1.293489e-02
#> X4 -3.7674288993 -1.2623103593 -4.43800458 13.300164087 -3.842669e-03
#> X5  0.0004704122  0.0009137598  0.01293489 -0.003842669  7.349631e-05
#> X6  0.0003459638  0.0007962456  0.01227399 -0.003261959  6.921166e-05
#>               X6
#> X1  3.459638e-04
#> X2  7.962456e-04
#> X3  1.227399e-02
#> X4 -3.261959e-03
#> X5  6.921166e-05
#> X6  6.968760e-05
#> [,,2]
#>            X1         X2          X3         X4          X5          X6
#> X1  4.8843437 -4.5297927 -2.77153730 -1.4099953 -0.23245546 -0.23264645
#> X2 -4.5297927  6.6689208  5.80272832 -3.2807395  0.31129646  0.31135801
#> X3 -2.7715373  5.8027283 69.66179720 -8.1849831 -0.09107216 -0.09064035
#> X4 -1.4099953 -3.2807395 -8.18498307 13.9324078 -0.11291125 -0.11267168
#> X5 -0.2324555  0.3112965 -0.09107216 -0.1129113  0.01917880  0.01917832
#> X6 -0.2326464  0.3113580 -0.09064035 -0.1126717  0.01917832  0.01918462
#> [,,3]
#>              X1          X2           X3          X4           X5
#> X1  21.22827432 -20.4168819  2.434609118 -0.04365385 -0.473934862
#> X2 -20.41688190  20.7123209 -1.516398301 -2.51956375  0.473505954
#> X3   2.43460912  -1.5163983  8.820719628 -4.14579852  0.001466994
#> X4  -0.04365385  -2.5195638 -4.145798519 11.71067627 -0.047043287
#> X5  -0.47393486   0.4735060  0.001466994 -0.04704329  0.011595792
#> X6  -0.47268574   0.4720653  0.001767016 -0.04653162  0.011568597
#>              X6
#> X1 -0.472685743
#> X2  0.472065342
#> X3  0.001767016
#> X4 -0.046531619
#> X5  0.011568597
#> X6  0.011545384
#> [,,4]
#>             X1           X2           X3          X4          X5
#> X1  78.7923594 -71.13241378 -77.76465184 -25.7177306  0.37243162
```

```
#> X2 -71.1324138 105.42474200  99.58956881  -0.6212110 -0.07718221
#> X3 -77.7646518  99.58956881 304.53551238  13.9423521  0.08622707
#> X4 -25.7177306  -0.62121097  13.94235212  39.7430627 -0.49944665
#> X5   0.3724316  -0.07718221   0.08622707  -0.4994467  0.02963406
#> X6   0.3738529  -0.06431463   0.08956538  -0.5030987  0.02934316
#>             X6
#> X1  0.37385288
#> X2 -0.06431463
#> X3  0.08956538
#> X4 -0.50309869
#> X5  0.02934316
#> X6  0.02938210
#> [,,5]
#>             X1          X2          X3          X4         X5         X6
#> X1  69.8767553  65.6172122 -24.8972124  46.8804224  0.8466420  0.6973576
#> X2  65.6172122  73.3132313 -20.3459285  51.8140290  0.8794092  0.6885970
#> X3 -24.8972124 -20.3459285  16.9578363 -18.5957263 -0.2252179 -0.2568134
#> X4  46.8804224  51.8140290 -18.5957263  45.9917732  0.6227445  0.5493780
#> X5   0.8466420   0.8794092  -0.2252179   0.6227445  0.1275238  0.1109700
#> X6   0.6973576   0.6885970  -0.2568134   0.5493780  0.1109700  0.1350921
#> [,,6]
#>             X1          X2          X3           X4           X5
#> X1  28.0715201 -16.9799516 -1.03936992 -11.38078667 -0.169114714
#> X2 -16.9799516  21.0343518  6.96891885  -3.08566408  0.138701475
#> X3  -1.0393699   6.9689188 23.78012619  -5.95018897 -0.034116483
#> X4 -11.3807867  -3.0856641 -5.95018897  27.47650544  0.014577235
#> X5  -0.1691147   0.1387015 -0.03411648   0.01457723  0.002576034
#> X6  -0.1646986   0.1352132 -0.03520443   0.02138803  0.002482964
#>              X6
#> X1 -0.164698620
#> X2  0.135213210
#> X3 -0.035204434
#> X4  0.021388026
#> X5  0.002482964
#> X6  0.002535465
#> [,,7]
#>             X1         X2            X3         X4          X5         X6
#> X1  27.9158525 14.0551357 -11.103158623 10.6769411 0.242848912 0.21855601
#> X2  14.0551357 22.8350925  -0.403968050 17.2536709 0.255281951 0.25332237
#> X3 -11.1031586 -0.4039680  11.083541505 -4.9169264 0.006159081 0.01040242
#> X4  10.6769411 17.2536709  -4.916926417 25.1413135 0.118293313 0.10116123
#> X5   0.2428489  0.2552820   0.006159081  0.1182933 0.041228789 0.03966157
#> X6   0.2185560  0.2533224   0.010402417  0.1011612 0.039661570 0.04285537
#> [,,8]
#>             X1          X2          X3          X4           X5
#> X1  3.68757428 -0.32306251 -0.74652658 -6.13238378  0.0363254005
#> X2 -0.32306251  1.48728867  1.80570712 -1.03896992 -0.0197568368
#> X3 -0.74652658  1.80570712 14.48878636 -1.48873597 -0.0457852026
#> X4 -6.13238378 -1.03896992 -1.48873597 17.20317410 -0.0413204449
#> X5  0.03632540 -0.01975684 -0.04578520 -0.04132044  0.0007022371
#> X6  0.03640642 -0.01946674 -0.04506412 -0.04155243  0.0006943328
#>              X6
#> X1  0.0364064174
#> X2 -0.0194667367
```

```
#> X3 -0.0450641243
#> X4 -0.0415524278
#> X5  0.0006943328
#> X6  0.0006933616
#> [,,9]
#>            X1         X2          X3          X4          X5          X6
#> X1 51.2999212 40.7230150 -9.10691048 10.05315370  0.24189684  0.16527248
#> X2 40.7230150 47.4637769 -5.52361025 24.08137718  0.21110593  0.10267524
#> X3 -9.1069105 -5.5236102  7.77371493  4.21757110 -0.07979663 -0.09521363
#> X4 10.0531537 24.0813772  4.21757110 35.22884932 -0.07318253 -0.17287661
#> X5  0.2418968  0.2111059 -0.07979663 -0.07318253  0.06813880  0.06317762
#> X6  0.1652725  0.1026752 -0.09521363 -0.17287661  0.06317762  0.06932934
```

```
summary(mod_3, parameters = TRUE)
```

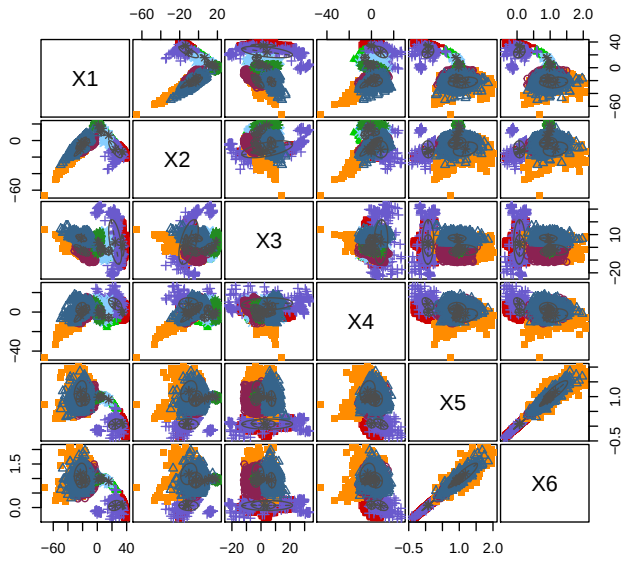
```
#> ----------------------------------------------------
#> Gaussian finite mixture model fitted by EM algorithm
#> ----------------------------------------------------
#>
#> Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model
#> with 11 components:
#>
#>  log-likelihood    n  df       BIC       ICL
#>        -40665.8 5000 307 -83946.38 -85118.73
#>
#> Clustering table:
#>   1   2   3   4   5   6   7   8   9  10  11
#> 904 416 743 213 491 394 244 382 415 132 666
#>
#> Mixing probabilities:
#>         1          2          3          4          5          6
#> 0.17636175 0.08824542 0.13509387 0.04273139 0.10238449 0.08523158
#>         7          8          9         10         11
#> 0.04994669 0.07856876 0.08170879 0.02690083 0.13282642
#>
#> Means:
#>         [,1]       [,2]        [,3]        [,4]        [,5]       [,6]
#> X1  4.170657  4.5951054 -13.9439307  25.9125419 -19.2349628 -26.649146
#> X2 16.643365 12.5360201  -0.0816628 -11.0918183  -6.0642075  -9.229931
#> X3  1.262547 -3.3877429  -3.7891908  -8.4934873  -3.4979389   5.696456
#> X4 -2.768719  0.5955959   4.5200833   5.5008951  -0.3532685  -2.303837
#> X5  1.002862  1.0052264   0.9044948   0.2507406   1.2780872   1.001058
#> X6  1.002785  1.0048672   0.9164954   0.2515225   1.2628620   1.005673
#>           [,7]        [,8]       [,9]       [,10]        [,11]
#> X1 -17.9671193 -20.3532989 14.1594834 21.15508994  34.43791470
#> X2   0.4290006  -8.6740838  7.6742112 -5.93154346 -13.77872329
#> X3   7.4797715  -2.7832551 -1.0619876 20.72606454   1.41505165
#> X4   6.3211986  -2.4176394 -5.3798066  8.04502851  -0.33155877
#> X5   0.8670347   0.8097829  0.9326177  0.08841635   0.08367615
#> X6   0.8671011   0.8047324  0.9326098  0.08778847   0.08384745
#>
#> Variances:
#> [,,1]
#>             X1          X2          X3          X4            X5
#> X1  7.11771074 -4.29208743 -4.40446559 -6.05272790  0.0518992783
```

```
#> X2 -4.29208743  3.88121267  4.28817515  1.36394575 -0.0369687367
#> X3 -4.40446559  4.28817515 16.46781529  0.25761408 -0.0484034514
#> X4 -6.05272790  1.36394575  0.25761408 13.66917047 -0.0370737675
#> X5  0.05189928 -0.03696874 -0.04840345 -0.03707377  0.0006084461
#> X6  0.05183284 -0.03680901 -0.04823723 -0.03712363  0.0006031387
#>               X6
#> X1  0.0518328426
#> X2 -0.0368090142
#> X3 -0.0482372322
#> X4 -0.0371236289
#> X5  0.0006031387
#> X6  0.0006022610
#> [,,2]
#>              X1           X2          X3           X4            X5
#> X1  15.94596925 -3.874172522 -5.52335391 -14.01073233 -0.0238714217
#> X2  -3.87417252  6.276670539  2.92790015  -0.59745578  0.0044646583
#> X3  -5.52335391  2.927900153 10.63817536   2.84923280  0.0251406340
#> X4 -14.01073233 -0.597455785  2.84923280  25.07180957  0.0282428079
#> X5  -0.02387142  0.004464658  0.02514063   0.02824281  0.0004413250
#> X6  -0.02296409  0.004083537  0.02502212   0.03087473  0.0004174288
#>               X6
#> X1 -0.0229640923
#> X2  0.0040835372
#> X3  0.0250221207
#> X4  0.0308747332
#> X5  0.0004174288
#> X6  0.0004677294
#> [,,3]
#>             X1         X2           X3           X4          X5
#> X1  23.1328327 3.21728601 -13.95223673  2.521087635  0.22850964
#> X2   3.2172860 9.26175071   1.42450712  7.328102666  0.10989073
#> X3 -13.9522367 1.42450712  15.61043110 -4.221000646 -0.02900884
#> X4   2.5210876 7.32810267  -4.22100065 18.704416058  0.01861968
#> X5   0.2285096 0.10989073  -0.02900884  0.018619676  0.03540837
#> X6   0.1922242 0.09735413  -0.01198902 -0.006615069  0.03411133
#>              X6
#> X1  0.192224155
#> X2  0.097354126
#> X3 -0.011989017
#> X4 -0.006615069
#> X5  0.034111333
#> X6  0.035875540
#> [,,4]
#>            X1         X2         X3         X4         X5         X6
#> X1  52.154503 -42.739371  5.7042928  -4.394748 -1.8504026 -1.8570662
#> X2 -42.739371  65.607726 -6.2256984 -28.320433  2.3621110  2.3705569
#> X3   5.704293  -6.225698 45.2554312  -7.300365  0.7510096  0.7528437
#> X4  -4.394748 -28.320433 -7.3003647  47.288043 -1.0275919 -1.0227010
#> X5  -1.850403   2.362111  0.7510096  -1.027592  0.1296127  0.1294993
#> X6  -1.857066   2.370557  0.7528437  -1.022701  0.1294993  0.1297605
#> [,,5]
#>            X1         X2           X3         X4          X5           X6
#> X1 26.5609597 19.0879761 -7.740413304 13.7127812 -0.15745099 -0.246567553
#> X2 19.0879761 27.1606821 -2.050894665 19.3771559 -0.33305489 -0.436439938
```

```
#> X3 -7.7404133 -2.0508947  7.574537627 -5.4591993  0.02976990  0.001897777
#> X4 13.7127812 19.3771559 -5.459199263 23.9021009 -0.21046010 -0.214480935
#> X5 -0.1574510 -0.3330549  0.029769899 -0.2104601  0.06629529  0.050643646
#> X6 -0.2465676 -0.4364399  0.001897777 -0.2144809  0.05064365  0.074058416
#> [,,6]
#>            X1          X2          X3         X4          X5          X6
#> X1 79.2984945  94.2199570 -6.82443203 58.1179713  0.66483420  0.56159837
#> X2 94.2199570 126.2900448  0.93104278 89.3696631  0.77865259  0.67245914
#> X3 -6.8244320   0.9310428 14.71373191  7.0632201 -0.03520316 -0.03431749
#> X4 58.1179713  89.3696631  7.06322014 76.8313957  0.46605017  0.40939775
#> X5  0.6648342   0.7786526 -0.03520316  0.4660502  0.11315288  0.10141646
#> X6  0.5615984   0.6724591 -0.03431749  0.4093978  0.10141646  0.11070759
#> [,,7]
#>             X1         X2          X3          X4          X5          X6
#> X1  60.6404790 41.3891357 -9.40097083 -15.0887028  0.41113985  0.40309470
#> X2  41.3891357 47.9576493 -3.46034590   1.4111484  0.17050995  0.12051211
#> X3  -9.4009708 -3.4603459  7.99020524   8.9620189 -0.06923748 -0.08277483
#> X4 -15.0887028  1.4111484  8.96201888  24.3969763 -0.19102213 -0.26347198
#> X5   0.4111399  0.1705099 -0.06923748  -0.1910221  0.04433071  0.04245610
#> X6   0.4030947  0.1205121 -0.08277483  -0.2634720  0.04245610  0.04754351
#> [,,8]
#>            X1         X2          X3          X4         X5         X6
#> X1 23.4319065 15.4470440 -3.13582966  7.26306770 0.12489862 0.12333324
#> X2 15.4470440 20.9585599  1.41545073  8.30980092 0.20720950 0.19880888
#> X3 -3.1358297  1.4154507  4.67623195 -2.62776096 0.10645042 0.07916489
#> X4  7.2630677  8.3098009 -2.62776096 11.01799737 0.04793059 0.01906488
#> X5  0.1248986  0.2072095  0.10645042  0.04793059 0.02773248 0.02465738
#> X6  0.1233332  0.1988089  0.07916489  0.01906488 0.02465738 0.03053897
#> [,,9]
#>             X1          X2          X3           X4           X5
#> X1  15.4777000 -14.1396941 -3.53613503 -2.770817996 -0.276403463
#> X2 -14.1396941  14.8696786  4.04772703 -1.128925299  0.276713730
#> X3  -3.5361350   4.0477270 13.71382479 -2.879584341  0.022440353
#> X4  -2.7708180  -1.1289253 -2.87958434 14.439911012 -0.007846653
#> X5  -0.2764035   0.2767137  0.02244035 -0.007846653  0.005791863
#> X6  -0.2753645   0.2754504  0.02147175 -0.007127131  0.005768366
#>              X6
#> X1 -0.275364548
#> X2  0.275450370
#> X3  0.021471748
#> X4 -0.007127131
#> X5  0.005768366
#> X6  0.005748906
#> [,,10]
#>             X1          X2          X3          X4          X5          X6
#> X1 106.0313885 -87.8957435 -49.9189966 -43.9771601  0.89634894  0.89868143
#> X2 -87.8957435 121.3375814  84.2487309  15.5589697 -0.69729714 -0.67448818
#> X3 -49.9189966  84.2487309  83.6065659   0.5821942 -0.33449072 -0.31530662
#> X4 -43.9771601  15.5589697   0.5821942  43.2964725 -0.44171137 -0.45027139
#> X5   0.8963489  -0.6972971  -0.3344907  -0.4417114  0.02440428  0.02436442
#> X6   0.8986814  -0.6744882  -0.3153066  -0.4502714  0.02436442  0.02446102
#> [,,11]
#>            X1         X2         X3         X4          X5          X6
#> X1  4.3914052 -4.1043674 -0.7606950 -1.2624745 -0.23657742 -0.23678319
```

```
#> X2 -4.1043674  6.1960807  3.3004266 -3.3139797  0.31337721  0.31343593
#> X3 -0.7606950  3.3004266 59.0643831 -7.3914606 -0.13995672 -0.13929001
#> X4 -1.2624745 -3.3139797 -7.3914606 13.8386543 -0.10888381 -0.10867377
#> X5 -0.2365774  0.3133772 -0.1399567 -0.1088838  0.01955229  0.01955113
#> X6 -0.2367832  0.3134359 -0.1392900 -0.1086738  0.01955113  0.01955658
```

```
plot(mod_2, what = "classification")
```




```
classes <- mod_2$classification
write.table(classes, file="Stein_Rebecca.txt", col.names= FALSE, row.names = FALSE)
```

Problem 4.3: Poisson Mixture Model

The pmf of a Poisson random variable is:

$$f_k(x;\lambda_k) = \frac{\lambda_k^x e^{-\lambda_k}}{x!}$$

A two-component Poisson mixture model can be written:

$$f(x;\theta) = \pi\frac{\lambda_1^x e^{-\lambda_1}}{x!} + (1-\pi)\frac{\lambda_2^x e^{-\lambda_2}}{x!}$$

a. What are the parameters of the model?

$\theta = (\lambda_1, \lambda_2, \pi_1, \pi_2)$

b. Write down the log-likelihood for n observations $(x_1, x_2, \ldots, x_n)$.

$log(L(X:\theta) = \sum_1^n log([(\pi_1\lambda_1^x exp(-\lambda_1) + ((1-\pi_1)\lambda_2^x exp(-\lambda_2)]\ /x!)$

c. Suppose we have an initial value of the parameters. Write down the equation for updating the *responsibilities*.

$r(xk) = \lambda_k^x exp(-\lambda_k)/\sum_k \lambda_k^x exp(-\lambda_k)$

d. Suppose we have responsibilities, r_{ik} for all $i = 1, 2, \ldots, n$ and $k = 1, 2$. Write down the equations for updating the parameters.

$n_k = \sum_i rsubik\ \pi_k = n_k/n\ \lambda_k = argmax(log(L(X:\lambda_k)$

e. Fit a two-component Poisson mixture model, report the estimated parameter values, and show a plot of the estimated mixture pmf for the following data:

```r
#-- Run this code to generate the data
set.seed(123)             # set seed for reproducibility
n = 200                   # sample size
z = sample(1:2, size=n, replace=TRUE, prob=c(.25, .75)) # sample the latent class
theta = c(8, 16)          # true parameters
y = ifelse(z==1, rpois(n, lambda=theta[1]), rpois(n, lambda=theta[2]))
```

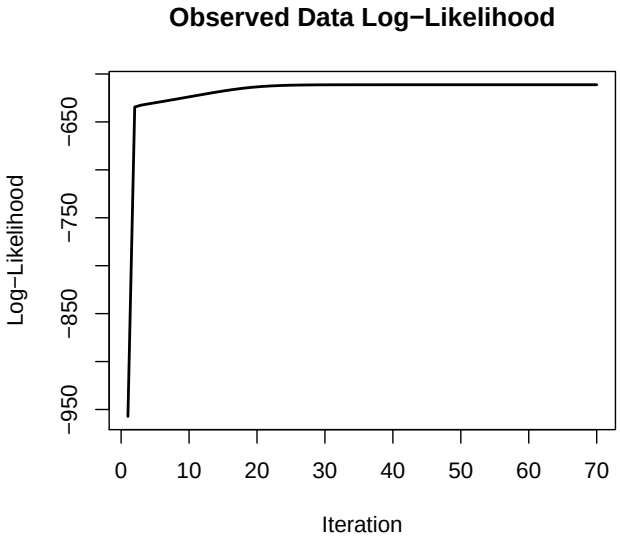
Note: The function `poisregmixEM()` in the R package `mixtools` is designed to estimate a mixture of *Poisson regression* models. We can still use this function for our problem of density estimation if it is recast as an intercept-only regression. To do so, set the x argument (predictors) to `x = rep(1, length(y))` and `addintercept = FALSE`.

Look carefully at the output from this model. The `beta` values (regression coefficients) are on the log scale.

```r
twocomppois <- poisregmixEM(y = y, x = rep(1,length(y)), k = 2, addintercept = FALSE)
```

```
#> number of iterations= 69
```

```r
plot.mixEM(twocomppois)
```



f. **2 pts Extra Credit**: Write a function that estimates this two-component mixture model using the EM approach. Show that it gives the same result as part *e*.
   - Note: you are not permitted to copy code. Write everything from scratch and use comments to indicate how the code works (e.g., the E-step, M-step, initialization strategy, and convergence should be clear)

SOLUTION HERE