Homework 08

Collin Stewart

https://github.com/collings512/BIOS512_Collin_Stewart

This homework is based on the clustering lectures. Check the lecture notes and TA notes - they should help!

In [1]:
```
library(tidyverse)
library(dplyr)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.2     ✓ readr     2.1.4
✓ forcats   1.0.0     ✓ stringr   1.5.0
✓ ggplot2   3.4.2     ✓ tibble    3.2.1
✓ lubridate 1.9.2     ✓ tidyr     1.3.0
✓ purrr     1.0.1
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

Question 1

This question will walk you through creating your own `kmeans` function.

a) What are the steps of `kmeans`?

Hint: There are 4 steps/builder functions that you'll need.

The 4 steps are:

1. Randomly assign all points to one of your clusters.
2. Compute each cluster's centroid or mean.
3. Reassign each point's class/label based on proximity to the closest cluster center.
4. Recalculate the center of each cluster.

b) Create the builder function for step 1.

In [2]:
```
label_randomly <- function(n_points, n_clusters) {
    sample(((1:n_points) %% n_clusters)+1, n_points, replace = F)
}
```

c) Create the builder function for step 2.

In [3]:
```
get_cluster_means <- function(data, labels) {
  data %>%
    mutate(label__ = labels) %>%
    group_by(label__) %>%
      summarize(across(everything(), mean), .groups = "drop") %>%
  arrange(label__)
}
```

d) Create the builder function for step 3.

Hint: There are two ways to do this part - one is significantly more efficient than the other. You can do either.

In [4]:
```
assign_to_near_cluster <- function(data, means) {

  data_mtx <- as.matrix(data)
  means_mtx <- as.matrix(means %>% dplyr::select(-label__))

    dii <- sort(rep(1:nrow(data), nrow(means)))
    mii <- rep(1:nrow(means), nrow(data))

    diff_squared <- rowSums((data_mtx[dii, ] - means_mtx[mii, ])^2)

    df <- data.frame(
      dii = dii,
      mii = mii,
      distance = diff_squared
    )
    closest <- df %>%
      group_by(dii) %>%
      slice_min(distance, n=1) %>%
      ungroup() %>%
      arrange(dii)

    means$label__[closest$mii]
}
```

e) Create the builder function for step 4.

In [5]:
```
kmeans_done <- function(old_means, new_means, eps = 1e-6) {
   om <- as.matrix(old_means)
   nm <- as.matrix(new_means)
   diff <- sqrt(rowSums((om-nm)^2))
   if (mean(diff) < eps) TRUE else FALSE
}
```

f) Combine them all into your own `kmeans` function.

In [6]:
```
collinskmeans <- function(data, n_clusters, eps=1e-6, max_it = 1000, verbose = FALS
  data <- data[, sapply(data, is.numeric)]

  labels <- label_randomly(nrow(data), n_clusters)
```

```
  old_means <- get_cluster_means(data, labels)
  done <- FALSE
  it <- 0
  while(!done & it < max_it){
    labels <- assign_to_near_cluster(data, old_means)
    new_means <- get_cluster_means(data, labels)
    if(kmeans_done(old_means, new_means)){
      done <- TRUE
    } else {
      old_means <- new_means
        it <- it + 1
        if(verbose){
            cat(sprintf("%d\n", it))
        }
    }
  }
    list(labels=labels, means = new_means)
}
```

Question 2

This is when we'll test your `kmeans` function.

a) Read in the `voltages_df.csv` data set.

In [7]:
```
voltages_df <- read.csv("voltages_df.csv")
```

b) Call your `kmeans` function with 3 clusters. Print the results with `results$labels` and `results$means`.

In [8]:
```
results <- collinskmeans(voltages_df, n_clusters = 3)
print(results$labels)
print(results$means)
```

```
  [1] 1 3 3 3 3 3 2 2 3 1 1 1 2 1 3 1 1 1 1 1 3 3 3 3 3 1 1 1 2 2 2 2 3 2 2 3 3
 [38] 1 2 3 1 1 1 2 3 2 2 3 2 2 3 2 1 1 2 1 3 1 2 3 1 1 3 1 2 2 3 2 1 2 1 3 2 2
 [75] 3 3 1 2 2 2 2 2 3 2 2 3 3 3 1 2 2 2 3 2 3 3 2 3 3 2 2 2 3 2 1 1 2 1 1 3 1
[112] 1 3 2 1 2 1 1 3 3 3 3 1 1 2 3 3 2 1 2 3 1 3 2 1 3 3 3 1 2 1 3 3 2 1 1 3 1
[149] 1 2 1 2 3 3 2 1 2 2 3 1 2 2 1 3 1 2 3 1 3 1 1 2 3 1 2 2 3 2 1 3 2 3 2 1 2
[186] 3 3 1 3 1 1 3 1 1 2 2 1 3 1 1 2 1 2 2 3 1 3 2 3 1 3 2 1 2 2 3 2 2 3 3 3 3
[223] 3 3 3 2 2 2 2 1 3 1 2 3 3 2 2 1 1 2 2 1 2 2 3 2 1 2 2 3 2 3 1 2 3 2 2 1 1
[260] 1 3 3 2 1 3 3 3 2 2 2 1 3 3 3 2 1 2 2 3 3 2 1 2 1 1 3 2 3 3 2 2 1 2 2 1 3
[297] 1 2 2 3 3 2 1 1 1 2 1 3 3 3 1 1 1 1 1 1 3 1 3 1 1 2 3 2 1 1 1 1 3 1 3 1 2
[334] 1 3 3 2 2 1 3 2 3 2 2 1 2 3 2 1 1 2 3 3 3 1 2 3 1 2 2 3 3 3 3 1 2 1 3 1 1
[371] 1 1 1 3 3 2 2 1 1 1 1 2 2 1 1 2 2 3 3 2 1 3 3 3 3 2 3 3 2 1 1 1 1 3 3 1 1
[408] 2 2 2 3 2 2 1 3 3 3 2 3 3 3 1 1 3 1 2 3 3 3 3 2 2 3 2 1 3 1 1 1 2 2 2 2 1
[445] 3 1 1 1 2 3 1 1 1 3 3 1 3 3 1 2 3 3 1 2 3 3 3 1 2 2 1 2 1 3 3 2 1 2 1 3 3
[482] 2 1 2 2 2 3 2 2 2 2 2 3 1 1 2 2 1 3 3 3 1 1 1 3 3 2 2 1 2 1 3 3 2 2 1 2 2
[519] 3 1 1 1 3 2 1 3 2 2 1 3 2 2 2 2 1 1 3 1 1 3 3 3 3 2 1 1 2 1 1 1 3 3 1 3 2
[556] 2 3 2 2 2 3 2 2 3 2 2 3 2 2 1 2 3 2 3 2 3 3 3 1 2 2 2 2 1 3 2 2 2 3 2 2 3
[593] 3 3 2 1 3 3 1 3 2 3 3 2 1 1 2 2 2 3 2 1 3 1 2 3 1 3 3 1 2 3 1 2 2 1 3 1 2
[630] 3 1 2 1 3 3 1 3 1 1 1 1 1 3 3 1 2 1 2 2 1 2 3 2 1 2 1 1 1 3 1 1 3 3 1 1 3
[667] 1 3 2 2 3 1 3 1 1 2 3 3 1 2 1 3 2 2 1 1 3 1 3 3 2 2 2 2 3 3 3 1 2 3 1 2 2
[704] 1 1 2 3 2 2 1 3 1 2 3 1 3 3 1 3 3 1 1 3 3 1 3 2 1 1 1 1 1 3 3 1 1 3 3 1 3
[741] 1 3 2 1 1 1 3 1 2 3 3 1 1 2 1 1 2 1 2 2 1 1 3 2 1 3 2 3 3 1 2 3 2 1 3 1 1
[778] 2 1 2 3 1 1 3 2 3 1 3 1 2 1 1 3 3 3 1 3 1 2 3 3 3 3 1 1 2 3 1 1 1 1 2 2 1
[815] 2 1 3 1 3 2 3 3 3 3 3 2 2 3 2 2 2 2 2 2 3 1 1 3 2 2 1 3 2 2 2 2 1 1 3 2 3
[852] 1 2 2 2 2 1 1 2 2 2 2 2 1 3 2 3 1 3 3 2 1 2 2 1 1 3 1 1 1 2 3 2 3 3 1 1 2
[889] 2 3 1 2 2 3 3 3 3 1 2 2
# A tibble: 3 × 251
  label__     X0 X1.00401606425703 X2.00803212851406 X3.01204819277108
    <dbl> <dbl>             <dbl>             <dbl>             <dbl>
1       1 -1.03             0.938             0.762             0.363
2       2 -1.03             1.31              1.16              0.979
3       3 -1.03             1.24              1.09              0.900
# ℹ 246 more variables: X4.01606425702811 <dbl>, X5.02008032128514 <dbl>,
#   X6.02409638554217 <dbl>, X7.0281124497992 <dbl>, X8.03212851405623 <dbl>,
#   X9.03614457831325 <dbl>, X10.0401606425703 <dbl>, X11.0441767068273 <dbl>,
#   X12.0481927710843 <dbl>, X13.0522088353414 <dbl>, X14.0562248995984 <dbl>,
#   X15.0602409638554 <dbl>, X16.0642570281125 <dbl>, X17.0682730923695 <dbl>,
#   X18.0722891566265 <dbl>, X19.0763052208835 <dbl>, X20.0803212851406 <dbl>,
#   X21.0843373493976 <dbl>, X22.0883534136546 <dbl>, …
```

c) Call R's `kmeans` function with 3 clusters. Print the results with `results$labels` and `results$cluster`.

Hint: Use the `as.matrix()` function to make the `voltages_df` data frame a matrix before calling `kmeans()`.

In [9]:
```
voltages_mat <- as.matrix(voltages_df)
results2 <- kmeans(voltages_mat, centers = 3)

print(results2$centers)
print(results2$cluster)
```

```
         X0 X1.00401606425703 X2.00803212851406 X3.01204819277108
1 -1.031463          1.123724         0.9618318         0.6709520
2 -1.031463          1.245873         1.0950402         0.9049394
3 -1.031463          1.243124         1.0913149         0.8984242
  X4.01606425702811 X5.02008032128514 X6.02409638554217 X7.0281124497992
1        -0.2348958         -1.109877         -1.048146        -0.964286
2         0.3511373         -1.160528         -1.110436        -1.069043
3         0.2787287         -1.159348         -1.109533        -1.068326
  X8.03212851405623 X9.03614457831325 X10.0401606425703 X11.0441767068273
1        -0.8417775        -0.7449379        -0.5477344       -0.07723852
2        -1.0343093        -1.0027217        -0.9705836       -0.93519388
3        -1.0336653        -1.0020231        -0.9697003       -0.93399945
  X12.0481927710843 X13.0522088353414 X14.0562248995984 X15.0602409638554
1        0.03795671         0.1033652        -0.1775692        -0.2028226
2       -0.89548682        -0.8522194        -0.8079892        -0.7671976
3       -0.89385445        -0.8500225        -0.8051168        -0.7636351
  X16.0642570281125 X17.0682730923695 X18.0722891566265 X19.0763052208835
1        -0.4467330        -0.9255488        -0.9929924        -0.9628079
2        -0.7356490        -0.7184454        -0.7161193        -0.7231093
3        -0.7317237        -0.7153140        -0.7152378        -0.7228370
  X20.0803212851406 X21.0843373493976 X22.0883534136546 X23.0923694779116
1        -0.9256890        -0.8930918        -0.8629138        -0.8302993
2        -0.7319168        -0.7408835        -0.7486848        -0.7545907
3        -0.7327732        -0.7425159        -0.7508725        -0.7572732
  X24.0963855421687 X25.1004016064257 X26.1044176706827 X27.1084337349398
1        -0.7820849        -0.6223898        -0.1490071        0.04858315
2        -0.7583186        -0.7599498        -0.7599961       -0.75928422
3        -0.7614730        -0.7635672        -0.7640633       -0.76375787
  X28.1124497991968 X29.1164658634538 X30.1204819277108 X31.1244979919679
1       -0.02904835        -0.4780363        -0.4865584        -0.1165327
2       -0.75846696        -0.7572927        -0.7541178        -0.7460515
3       -0.76325735        -0.7622897        -0.7592482        -0.7513164
  X32.1285140562249 X33.1325301204819 X34.136546184739 X35.140562248996
1       -0.06235668        -0.3217896       -0.8803266       -0.9704336
2       -0.72955517        -0.7008369       -0.6544317       -0.4872291
3       -0.73501097        -0.7064079       -0.6525963       -0.4932954
  X36.144578313253 X37.1485943775101 X38.1526104417671 X39.1566265060241
1      -0.92626689        -0.8770299        -0.8301983       -0.78933660
2       0.03396238         0.4863937         0.5379486       -0.05735043
3      -0.03390026         0.3772924         0.4406407       -0.09479697
  X40.1606425702811 X41.1646586345382 X42.1686746987952 X43.1726907630522
1        -0.7606095        -0.7481990        -0.7556403        -0.7739032
2        -0.6683566        -0.9242637        -0.9502679        -0.8532296
3        -0.6518143        -0.9039474        -0.8765867        -0.6841762
  X44.1767068273092 X45.1807228915663 X46.1847389558233 X47.1887550200803
1        -0.7946469        -0.8108161        -0.8220337        -0.8279381
2        -0.7315426        -0.5476880        -0.1892358         0.4063077
3        -0.5540286        -0.3958403        -0.2032288         0.2905789
  X48.1927710843374 X49.1967871485944 X50.2008032128514 X51.2048192771084
1        -0.8293054        -0.8270369        -0.8214857        -0.8128255
2         0.5753620         0.2704433        -0.5260020        -0.9128389
3         0.4022096         0.0966771        -0.6131835        -0.9277620
  X52.2088353413655 X53.2128514056225 X54.2168674698795 X55.2208835341365
1        -0.8019307        -0.7861022        -0.7897399        -0.8075403
2        -1.0353220        -1.0004920        -0.9676514        -0.9380902
3        -1.0058768        -0.9725550        -0.9449534        -0.9199306
```

```
  X56.2248995983936 X57.2289156626506 X58.2329317269076 X59.2369477911647
1        -0.8329051        -0.8449075        -0.8448539        -0.8358612
2        -0.9064350        -0.8678828        -0.8188867        -0.7571809
3        -0.8917708        -0.8556405        -0.8079256        -0.7417183
  X60.2409638554217 X61.2449799196787 X62.2489959839357 X63.2530120481928
1        -0.8199875        -0.7975540        -0.7676344        -0.7102217
2        -0.6675043        -0.5059115        -0.1764873        -0.2226683
3        -0.6360597        -0.4537170        -0.2111698        -0.3013674
  X64.2570281124498 X65.2610441767068 X66.2650602409639 X67.2690763052209
1        -0.4358938        -0.1944941        -0.2300542        -0.6559918
2        -0.5325646        -0.9009078        -0.9146840        -0.9222848
3        -0.6126171        -0.8768442        -0.9023664        -0.9119603
  X68.2730923694779 X69.2771084337349 X70.281124497992 X71.285140562249
1        -0.9393542        -0.9373604       -0.9126939       -0.8838885
2        -0.9197294        -0.9031742       -0.8688966       -0.8124386
3        -0.9109706        -0.8956255       -0.8623018       -0.8067408
  X72.289156626506 X73.2931726907631 X74.2971887550201 X75.3012048192771
1       -0.8499500        -0.8091960        -0.7576007        -0.6716407
2       -0.7219536        -0.3487917         0.5214857         0.8316355
3       -0.7192026        -0.4035692         0.5145391         0.8511776
  X76.3052208835341 X77.3092369477912 X78.3132530120482 X79.3172690763052
1        -0.5862742        -0.5737616        -0.6442067        -0.7211829
2         0.5961856        -0.5660317        -1.0475440        -1.0762565
3         0.6353941        -0.6097231        -1.0689986        -1.0706910
  X80.3212851405623 X81.3253012048193 X82.3293172690763 X83.3333333333333
1        -0.7694646        -0.8233452        -0.8529900        -0.8690791
2        -1.0350857        -0.9951377        -0.9528293        -0.9068395
3        -1.0358469        -0.9955586        -0.9528177        -0.9062428
  X84.3373493975904 X85.3413654618474 X86.3453815261044 X87.3493975903614
1        -0.8711851        -0.8570312        -0.8241347        -0.7632620
2        -0.8591647        -0.8156777        -0.7852523        -0.7744626
3        -0.8577832        -0.8133310        -0.7819655        -0.7706523
  X88.3534136546185 X89.3574297188755 X90.3614457831325 X91.3654618473896
1        -0.4049884         0.1162786        0.06886896       -0.02812817
2        -0.7795446        -0.7892668       -0.79436018       -0.78991460
3        -0.7758953        -0.7862636       -0.79216160       -0.78859823
  X92.3694779116466 X93.3734939759036 X94.3775100401606 X95.3815261044177
1        -0.7480799        -0.9334328        -0.8894283        -0.8410618
2        -0.7744671        -0.7493130        -0.7184589        -0.6890976
3        -0.7741497        -0.7501763        -0.7206870        -0.6922182
  X96.3855421686747 X97.3895582329317 X98.3935742971888 X99.3975903614458
1        -0.7833918        -0.6996491        -0.6694463        -0.7252025
2        -0.6720140        -0.6749246        -0.6831933        -0.6861569
3        -0.6575499        -0.6494302        -0.6611775        -0.6967990
  X100.401606425703 X101.40562248996 X102.409638554217 X103.413654618474
1        -0.8093694       -0.8269012        -0.7987492        -0.7618758
2        -0.6817801       -0.6685226        -0.6452757        -0.5987415
3        -0.6936421       -0.6788217        -0.6538224        -0.5813017
  X104.417670682731 X105.421686746988 X106.425702811245 X107.429718875502
1        -0.7163622        -0.5956578        -0.3465539        -0.2467079
2        -0.4646349        -0.3949862        -0.3405025        -0.4965818
3        -0.5072216        -0.3964226        -0.4295125        -0.4777784
  X108.433734939759 X109.437751004016 X110.441767068273 X111.44578313253
1        -0.3514577        -0.6685083        -0.8390651       -0.9067344
2        -0.5930333        -0.7187057        -0.7293524       -0.7452987
3        -0.6076351        -0.6631021        -0.6881479       -0.7115661
```

```
  X112.449799196787 X113.453815261044 X114.457831325301 X115.461847389558
1        -0.9126229        -0.8833304        -0.8529353        -0.8289895
2        -0.7674323        -0.7912740        -0.8116082        -0.8232703
3        -0.7546820        -0.7939389        -0.8135252        -0.8244461
  X116.465863453815 X117.469879518072 X118.473895582329 X119.477911646586
1        -0.8157271        -0.8113318        -0.8079955        -0.7987436
2        -0.8219775        -0.8049175        -0.7710062        -0.7211318
3        -0.8224748        -0.8048333        -0.7704493        -0.7201484
  X120.481927710843 X121.4859437751 X122.489959839357 X123.493975903614
1        -0.7779420      -0.7176081        -0.5344082        -0.3767216
2        -0.6574683      -0.4904609        -0.4055303        -0.4741740
3        -0.6498545      -0.4705406        -0.4093369        -0.5208741
  X124.497991967871 X125.502008032129 X126.506024096386 X127.510040160643
1        -0.4080326        -0.6672944        -0.8173464        -0.8236658
2        -0.7558321        -0.8434514        -0.8569072        -0.8590319
3        -0.7687459        -0.8391254        -0.8537355        -0.8567804
  X128.5140562249 X129.518072289157 X130.522088353414 X131.526104417671
1      -0.7974207        -0.7818738        -0.7918563        -0.8268653
2      -0.8481258        -0.8244672        -0.7910987        -0.7552432
3      -0.8466009        -0.8234991        -0.7905217        -0.7548637
  X132.530120481928 X133.534136546185 X134.538152610442 X135.542168674699
1        -0.8468949        -0.8514986        -0.8468881        -0.8369852
2        -0.7300144        -0.7300478        -0.7529338        -0.7847499
3        -0.7296652        -0.7298068        -0.7529217        -0.7849602
  X136.546184738956 X137.550200803213 X138.55421686747 X139.558232931727
1        -0.8251047        -0.8162507       -0.8148012        -0.8207585
2        -0.8162638        -0.8420555       -0.8589168        -0.8649668
3        -0.8166912        -0.8427551       -0.8599653        -0.8664522
  X140.562248995984 X141.566265060241 X142.570281124498 X143.574297188755
1        -0.8297649        -0.8368127        -0.8381727        -0.8314668
2        -0.8592897        -0.8418901        -0.8138737        -0.7778305
3        -0.8613136        -0.8445748        -0.8173783        -0.7824310
  X144.578313253012 X145.582329317269 X146.586345381526 X147.590361445783
1        -0.8153693        -0.7891712        -0.7513773        -0.6514404
2        -0.7351959        -0.6920751        -0.6830996        -0.7065387
3        -0.7448509        -0.7133705        -0.6932970        -0.7056653
  X148.59437751004 X149.598393574297 X150.602409638554 X151.606425702811
1       -0.3711292        -0.2622727        -0.4114979        -0.8144104
2       -0.7431810        -0.7690228        -0.7927978        -0.8115780
3       -0.7311370        -0.7656347        -0.7891893        -0.8076282
  X152.610441767068 X153.614457831325 X154.618473895582 X155.622489959839
1        -0.8938624        -0.8809273        -0.8639022        -0.8423110
2        -0.8233287        -0.8263045        -0.8185101        -0.7970554
3        -0.8189244        -0.8213302        -0.8128264        -0.7904292
  X156.626506024096 X157.630522088353 X158.63453815261 X159.638554216867
1        -0.8170984        -0.7890960       -0.7647990        -0.7490877
2        -0.7568425        -0.6841579       -0.4045481         0.8608342
3        -0.7486963        -0.6719614       -0.2622993         0.8724486
  X160.642570281125 X161.646586345382 X162.650602409639 X163.654618473896
1        -0.7409729        -0.7383667        -0.7381180        -0.7421149
2         0.9767348         0.7849250         0.0177153        -1.1126259
3         0.9639645         0.7738459        -0.1120672        -1.1033158
  X164.658634538153 X165.66265060241 X166.666666666667 X167.670682730924
1        -0.7485001       -0.7554490        -0.7608708        -0.7619082
2        -1.0395889       -0.9734278        -0.9246694        -0.8948221
3        -1.0317032       -0.9659392        -0.9180833        -0.8896277
```

```
  X168.674698795181 X169.678714859438 X170.682730923695 X171.686746987952
1        -0.7544262        -0.7298647        -0.5699833        -0.1079854
2        -0.8810906        -0.8775295        -0.8778998        -0.8775001
3        -0.8773749        -0.8749391        -0.8759741        -0.8758909
  X172.690763052209 X173.694779116466 X174.698795180723 X175.70281124498
1        0.04722782       -0.05957484        -0.5838129       -0.8882710
2       -0.87336513       -0.86384007        -0.8482374       -0.8267569
3       -0.87185869       -0.86231396        -0.8466172       -0.8249829
  X176.706827309237 X177.710843373494 X178.714859437751 X179.718875502008
1        -0.8539757        -0.7695053        -0.6816092        -0.6408262
2        -0.8006644        -0.7727126        -0.7476321        -0.7318663
3        -0.7986672        -0.7704058        -0.7449922        -0.7294294
  X180.722891566265 X181.726907630522 X182.730923694779 X183.734939759036
1        -0.6913369        -0.7711124        -0.8108296        -0.8302308
2        -0.7304600        -0.7419756        -0.7601631        -0.7791711
3        -0.7290907        -0.7409448        -0.7589050        -0.7774115
  X184.738955823293 X185.74297188755 X186.746987951807 X187.751004016064
1        -0.8457154       -0.8548781        -0.8566039        -0.8504474
2        -0.7949314       -0.8050083        -0.8084081        -0.8056017
3        -0.7924914       -0.8016969        -0.8039880        -0.7997962
  X188.755020080321 X189.759036144578 X190.763052208835 X191.767068273092
1        -0.8361850        -0.8133510        -0.7802598        -0.7256342
2        -0.7986982        -0.7915701        -0.7893827        -0.7967554
3        -0.7912509        -0.7823894        -0.7787544        -0.7854155
  X192.771084337349 X193.775100401606 X194.779116465863 X195.78313253012
1        -0.4480129        0.07077084         0.0626212      -0.08014399
2        -0.8151266       -0.84200184        -0.8727333      -0.90241396
3        -0.8039523       -0.83163810        -0.8635567      -0.89466724
  X196.787148594378 X197.791164658635 X198.795180722892 X199.799196787149
1        -0.7790090        -0.9608427        -0.9279956        -0.8988733
2        -0.9267372        -0.9422645        -0.9465001        -0.9378980
3        -0.9206485        -0.9381266        -0.9447397        -0.9391553
  X200.803212851406 X201.807228915663 X202.81124497992 X203.815261044177
1        -0.8744666        -0.8548905       -0.8388328        -0.8236099
2        -0.9158071        -0.8803477       -0.8321912        -0.7721660
3        -0.9210513        -0.8910695       -0.8507494        -0.8024970
  X204.819277108434 X205.823293172691 X206.827309236948 X207.831325301205
1        -0.8061180        -0.7834938        -0.7412057        -0.6397551
2        -0.7002076        -0.6067468        -0.1098143         0.4207671
3        -0.7496946        -0.6971333        -0.6519524        -0.6236966
  X208.835341365462 X209.839357429719 X210.843373493976 X211.847389558233
1        -0.4576918        -0.4139470        -0.5455483       -0.75742800
2         0.7957051         0.2152144        -0.4380059       -0.93524522
3        -0.6189740        -0.6126166        -0.5514026        0.07756414
  X212.85140562249 X213.855421686747 X214.859437751004 X215.863453815261
1       -0.8068612        -0.7660377        -0.7096709        -0.6431906
2       -0.9411393        -0.8821014        -0.7003658         0.2202006
3        0.8988625         0.8884011         0.5531202        -0.5322118
  X216.867469879518 X217.871485943775 X218.875502008032 X219.879518072289
1        -0.6213680        -0.6449897        -0.6996839        -0.7247352
2         0.9652767         0.9326633         0.7176057        -0.4727933
3        -1.0160084        -0.9386326        -0.7992775        -0.5128268
  X220.883534136546 X221.887550200803 X222.89156626506 X223.895582329317
1        -0.7348560       -0.73582125       -0.7289767        -0.7152091
2        -1.0462411       -1.05011565       -1.0079484        -0.9837001
3        -0.1621612       -0.09738382       -0.4612028        -0.8843154
```

```
  X224.899598393574 X225.903614457831 X226.907630522088 X227.911646586345
1        -0.6877745        -0.5605799        -0.2054143       -0.07317932
2        -0.9738671        -0.9725299        -0.9735992       -0.97239260
3        -0.9501220        -0.9614418        -0.9689313       -0.97078084
  X228.915662650602 X229.919678714859 X230.923694779116 X231.927710843374
1        -0.2221838        -0.7314874        -0.9252416        -0.9229558
2        -0.9662790        -0.9548617        -0.9399930        -0.9254512
3        -0.9660888        -0.9553815        -0.9409694        -0.9268295
  X232.931726907631 X233.935742971888 X234.939759036145 X235.943775100402
1        -0.9033189        -0.8832138        -0.8616690        -0.8378087
2        -0.9158243        -0.9144223        -0.9212453        -0.9328464
3        -0.9175849        -0.9164864        -0.9234823        -0.9351429
  X236.947791164659 X237.951807228916 X238.955823293173 X239.95983935743
1        -0.8111274        -0.7808217        -0.7419944       -0.6216227
2        -0.9441382        -0.9503261        -0.9478582       -0.9345562
3        -0.9464305        -0.9525853        -0.9500650       -0.9366832
  X240.963855421687 X241.967871485944 X242.971887550201 X243.975903614458
1        -0.2268060        0.01333991       -0.05285934        -0.5433274
2        -0.9093694       -0.87200867       -0.82253175        -0.7606134
3        -0.9113662       -0.87377754       -0.82385929        -0.7608566
  X244.979919678715 X245.983935742972 X246.987951807229 X247.991967871486
1        -0.9271054        -0.9278647       -0.89463579       -0.86078516
2        -0.6785901        -0.3916473       -0.08367886        0.02967317
3        -0.6636216        -0.3140227       -0.03042208        0.02754518
  X248.995983935743       X250
1        -0.8323765 -0.8090319
2        -0.3416634 -0.7627508
3        -0.4426584 -0.8335155
  [1] 1 3 3 3 3 2 1 1 3 1 1 1 1 1 3 1 1 1 1 1 2 3 3 3 3 1 1 1 1 1 1 3 1 1 3 3
 [38] 1 1 2 1 1 1 1 2 1 1 3 1 1 2 1 1 1 1 1 2 1 1 3 1 1 3 1 1 1 2 1 1 1 1 3 1 1
 [75] 3 2 1 1 1 1 1 1 3 1 1 3 3 3 1 1 1 1 3 1 3 2 1 3 3 1 1 1 3 1 1 1 1 1 1 2 1
[112] 1 2 1 1 1 1 1 3 3 2 3 1 1 1 2 3 1 1 1 3 1 3 1 1 2 3 2 1 1 1 3 3 1 1 1 3 1
[149] 1 1 1 1 3 3 1 1 1 1 2 1 1 1 1 3 1 1 3 1 3 1 1 1 3 1 1 1 3 1 1 2 1 3 1 1 1
[186] 2 2 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 3 1 3 1 2 1 2 1 1 1 1 3 1 1 3 3 3 2
[223] 3 3 2 1 1 1 1 1 3 1 1 3 3 1 1 1 1 1 1 1 1 1 3 1 1 1 1 2 1 3 1 1 2 1 1 1 1
[260] 1 3 3 1 1 3 3 3 1 1 1 1 2 2 2 1 1 1 1 2 3 1 1 1 1 1 2 1 3 3 1 1 1 1 1 1 2
[297] 1 1 1 3 3 1 1 1 1 1 1 2 3 2 1 1 1 1 1 1 3 1 3 1 1 1 3 1 1 1 1 1 3 1 2 1 1
[334] 1 2 3 1 1 1 3 1 3 1 1 1 1 3 1 1 1 1 3 2 3 1 1 3 1 1 1 2 3 3 3 1 1 1 2 1 1
[371] 1 1 1 3 2 1 1 1 1 1 1 1 1 1 1 1 1 2 3 1 1 3 3 2 3 1 2 3 1 1 1 1 1 3 3 1 1
[408] 1 1 1 3 1 1 1 3 2 2 1 3 3 3 1 1 2 1 1 3 2 3 2 1 1 2 1 1 2 1 1 1 1 1 1 1 1
[445] 3 1 1 1 1 3 1 1 1 2 3 1 3 3 1 1 3 3 1 1 3 3 3 1 1 1 1 1 1 3 3 1 1 1 1 3 3
[482] 1 1 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1 3 3 3 1 1 1 3 3 1 1 1 1 1 3 2 1 1 1 1 1
[519] 3 1 1 1 2 1 1 2 1 1 1 3 1 1 1 1 1 1 3 1 1 3 3 3 2 1 1 1 1 1 1 1 2 3 1 3 1
[556] 1 3 1 1 1 3 1 1 3 1 1 3 1 1 1 1 2 1 3 1 2 2 3 1 1 1 1 1 1 3 1 1 1 3 1 1 3
[593] 3 2 1 1 3 3 1 3 1 3 2 1 1 1 1 1 1 2 1 1 2 1 1 3 1 3 3 1 1 3 1 1 1 1 3 1 1
[630] 2 1 1 1 2 3 1 3 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 3 1 1 3 3 1 1 3
[667] 1 2 1 1 3 1 3 1 1 1 2 3 1 1 1 3 1 1 1 1 3 1 2 3 1 1 1 1 3 2 2 1 1 3 1 1 1
[704] 1 1 1 3 1 1 1 2 1 1 3 1 2 3 1 3 3 1 1 3 2 1 2 1 1 1 1 1 1 3 3 1 1 3 3 1 2
[741] 1 2 1 1 1 1 3 1 1 3 2 1 1 1 1 1 1 1 1 1 1 1 3 1 1 2 1 2 3 1 1 3 1 1 2 1 1
[778] 1 1 1 3 1 1 3 1 2 1 2 1 1 1 1 3 3 3 1 3 1 1 3 3 3 3 1 1 1 2 1 1 1 1 1 1 1
[815] 1 1 2 1 3 1 2 3 3 3 3 1 1 3 1 1 1 1 1 1 3 1 1 3 1 1 1 3 1 1 1 1 1 1 3 1 3
[852] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 3 1 2 3 1 1 1 1 1 1 3 1 1 1 1 2 1 3 3 1 1 1
[889] 1 3 1 1 1 3 3 3 3 1 1 1
```

d) Are your labels/clusters the same? If not, why? Are your means the same?

The labels/clusters are not the same. This is normal because the the labels/clusters for each point are randomized initially in Step 1 every time I call my collinskmeans function, unless you set a seed. R's version of kmeans also does this, and without a set seed, this will lead to a different first iteration each time the function is called, meaning they won't align.

The means, are close but not identical (-1.03 vs -1.031463), meaning that the rounding is probably different between collinskmeans and R's kmeans.

Question 3

a) Explain the process of using a for loop to assign clusters for kmeans.

When using for loops to assign clusters for kmeans, we are calculating the distance from each individual point (i) to the center of each of the clusters (j). The point is assigned to the cluster of whichever centroid is closest in distance, which is relatively slow.

b) Explain the process of vectorizing the code to assign clusters for kmeans.

Instead of running a calculation for each data point, vectorization creates matrices that are able to compare all point-cluster pairs at once. By subtracting the matrices directly, the distances can all be calculated in a single operation, which is a faster approach to assigning clusters for all points.

c) State which (for loops or vectorizing) is more efficient and why.

Vectorizing is faster and more efficient. For loops require distances to be calculated between each individual point and each cluster center, which is slow for large datasets. Vectorizing, on the other hand, calculates all point-cluster distances at once, allowing faster and more efficient cluster assignment.

Question 4

When does `kmeans` fail? What assumption does `kmeans` use that causes it to fail in this situation?

Kmeans fails when clusters are not uniform in size or shape, such as the scikit-learn clustering module from the lecture notes.. The assumption that causes it to fail is the assumption that we have vectorial data which is uniformly sized and shaped.

Question 5

What assumption do Gaussian mixture models make?

Gaussian Mixture Models (GMMs) assume that data is drawn from N Gaussian distributions whose individual parameters are estimated from the data. This allows us to handle clusters of different sizes and shapes more easily.

Question 6

What assumption does spectral clustering make? Why does this help us?

Spectral clustering assumes that the two points are more likely to be in the same cluster if they are closer to each other. This helps us reducing all the assumptions we make about data into a single principle. Also, this assumption is based on the existence of vector space, which is much stronger and applicable than the assumption that multidimensional scaling makes on the existence of a metric in the data.

Question 7

Define the gap statistic method. What do we use it for?

The gap statistic method involves comparing the clustering for each value of K to a cluster of data "randomized" into the same domain as the original data. Then, we compute the dispersion of the two clustering and look at the difference. We use this to identify the "knee", which is our cluster number.