CS CM226, PS3, Prob. 3

Niko Darci-Maher, UID #504924547

11/11/2021

```r
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
setwd("~/Google Drive/UCLA Fall 2021/bioinfo226/ps3")

# keep random number generator the same for grading
set.seed(0)

# import data
data = read_tsv('data/Q3/q3.data')
```

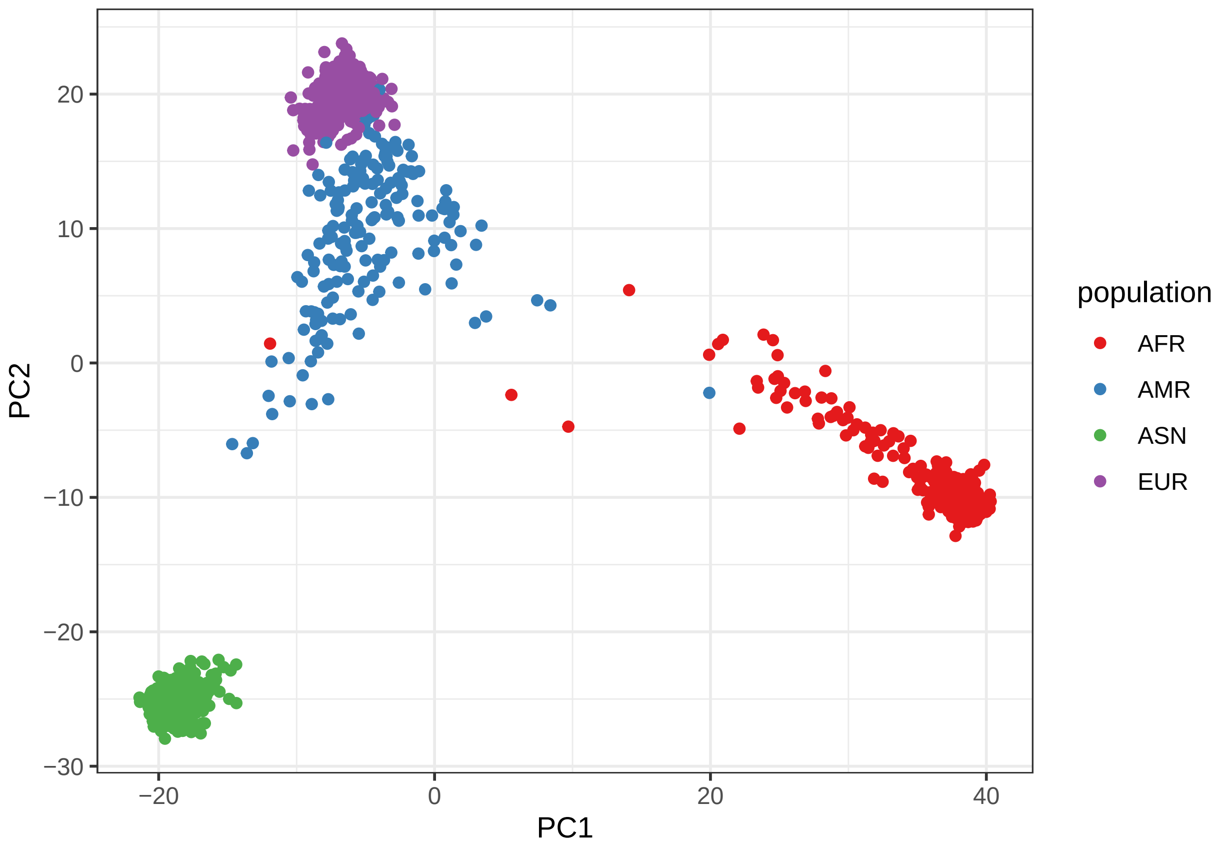
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   individual = col_character(),
##   population = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```r
snps = data %>% select(starts_with('rs'))
indiv = data[,1:2]
```

Part (a)

```r
# run PCA on the genotype data
pca = prcomp(snps, center = T, scale. = T, rank. = 2)

# plot PC1 vs PC2 and color by population
indiv_pc = cbind(indiv, pca$x)
pc12_pop = ggplot(indiv_pc, aes(x = PC1, y = PC2, colour = population)) + geom_point() +
  scale_color_brewer(palette ='Set1') +
  theme_bw()
pc12_pop
```



```r
ggsave("pc12_colbypopulation.png", pc12_pop)
```

```
## Saving 6.5 x 4.5 in image
```

Part (b)

The first two PCs cluster individuals by population because we are working with genotype data. By the nature of genetic drift, and the fact that in an evolutionary-scale time perspective people generally have children with someone who lives in a similar area to them, we know that a large factor driving which alleles a person will have at their genetic variant sites is their ancestry, AKA the population they come from.

Hence, the largest factor driving the variance in the covariance matrix of the genotype data will be caused by population membership, and the first two PCs will capture this variance because they are the eigenvectors of the covariance matrix with the two largest eigenvalues.

Part (c)

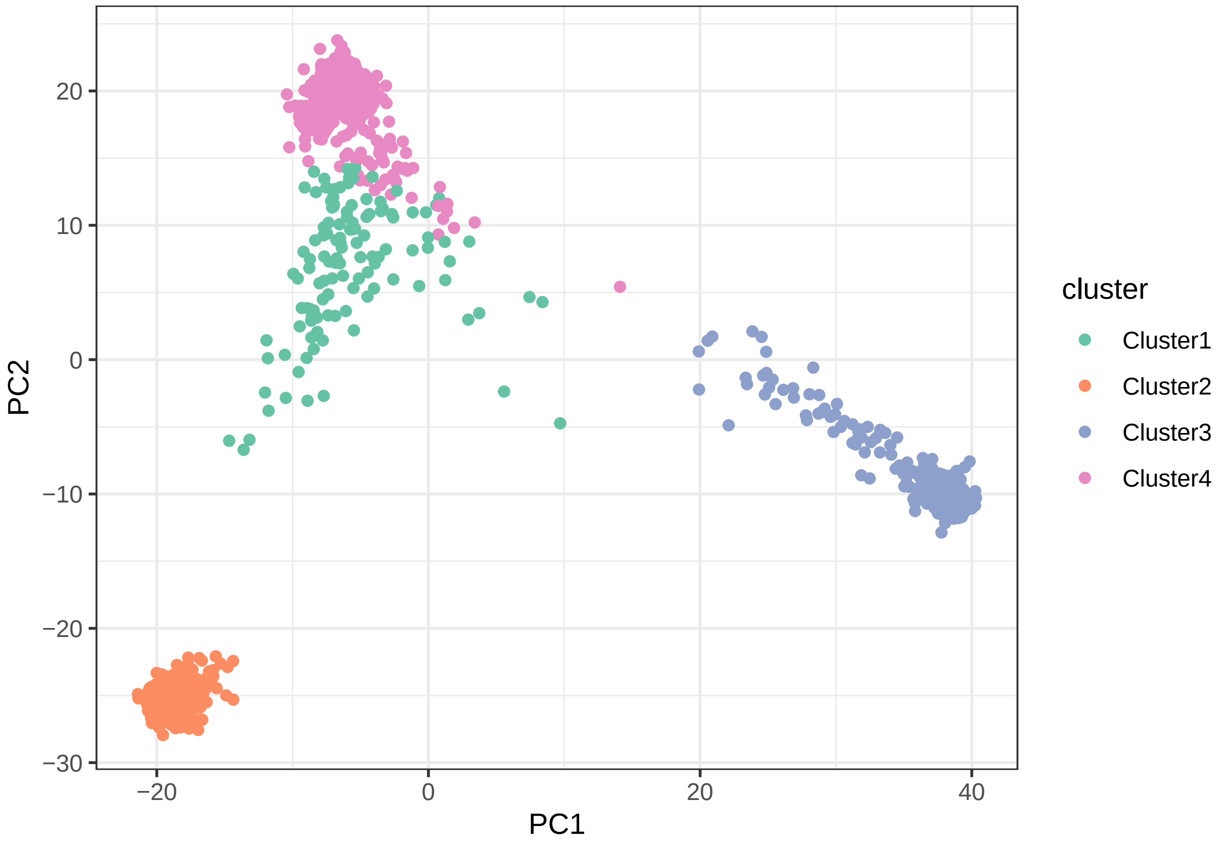
```r
# run k-means clustering
kmc = kmeans(snps, centers = 4, nstart = 5)

# check size of each cluster
table(kmc$cluster)
```

```
##
```

```
##   1   2   3   4
## 439 286 124 243
```

```
# rename the clusters by size
mapping = data.frame("old" = c(1, 2, 3, 4),
                     "new" = c("Cluster4", "Cluster2", "Cluster1", "Cluster3"))
cluster = sapply(kmc$cluster, function(x) as.character(mapping[mapping$old == x,"new"]))
indiv_pc_clust = cbind(indiv_pc, cluster)
pc12_kmc = ggplot(indiv_pc_clust, aes(x = PC1, y = PC2, colour = cluster)) + geom_point() +
  scale_color_brewer(palette ='Set2') +
  theme_bw()
pc12_kmc
```



```
ggsave("pc12_colbykmeansclust.png", pc12_kmc)
```

```
## Saving 6.5 x 4.5 in image
```

Part (d)

```
# assign cluster to pop by inspection
cluster2popDF = data.frame("cluster" = c("Cluster1", "Cluster2", "Cluster3", "Cluster4"),
                          "population" = c("AMR", "ASN", "AFR", "EUR"))
cluster2pop = function(c) {
  return(cluster2popDF[cluster2popDF$cluster == c, "population"])
}

# output my matchings of cluster to population
```

```r
print("Assignment of cluster to population by inspection:")
```

```
## [1] "Assignment of cluster to population by inspection:"
```

```r
print(cluster2popDF)
```

```
##    cluster population
## 1 Cluster1        AMR
## 2 Cluster2        ASN
## 3 Cluster3        AFR
## 4 Cluster4        EUR
```

```r
# check how much the kmeans clustering captures population
mismatch = indiv_pc_clust[indiv_pc_clust$population !=
                            sapply(indiv_pc_clust$cluster, function(x) cluster2pop(x)),]
mismatchcounts = table(mismatch[,c("population", "cluster")])
acc = (length(cluster) - sum(mismatchcounts)) / length(cluster)
print(paste0("Accuracy of kmeans clustering in predicting population: ",
             round(acc*100, 2), "%"))
```

```
## [1] "Accuracy of kmeans clustering in predicting population: 94.14%"
```