Homework 7: Unsupervised Learning

Wen Li Teng

March 15 2020

```r
library(tidyverse)
library(ggplot2)
library(gridExtra)
library(factoextra)
library(Rtsne)
library(ggfortify)
library(cluster)
set.seed(1234)
```

k-Means Clustering "By Hand"

```r
resp <- 1:10
input_1 <- c(5,8,7,8,3,4,2,3,4,5)
input_2 <- c(8,6,5,4,3,2,2,8,9,8)
data <- as.data.frame(cbind(resp, input_1, input_2))
```

1. Imitate k-means randomization by assigning each observation to a cluster at random

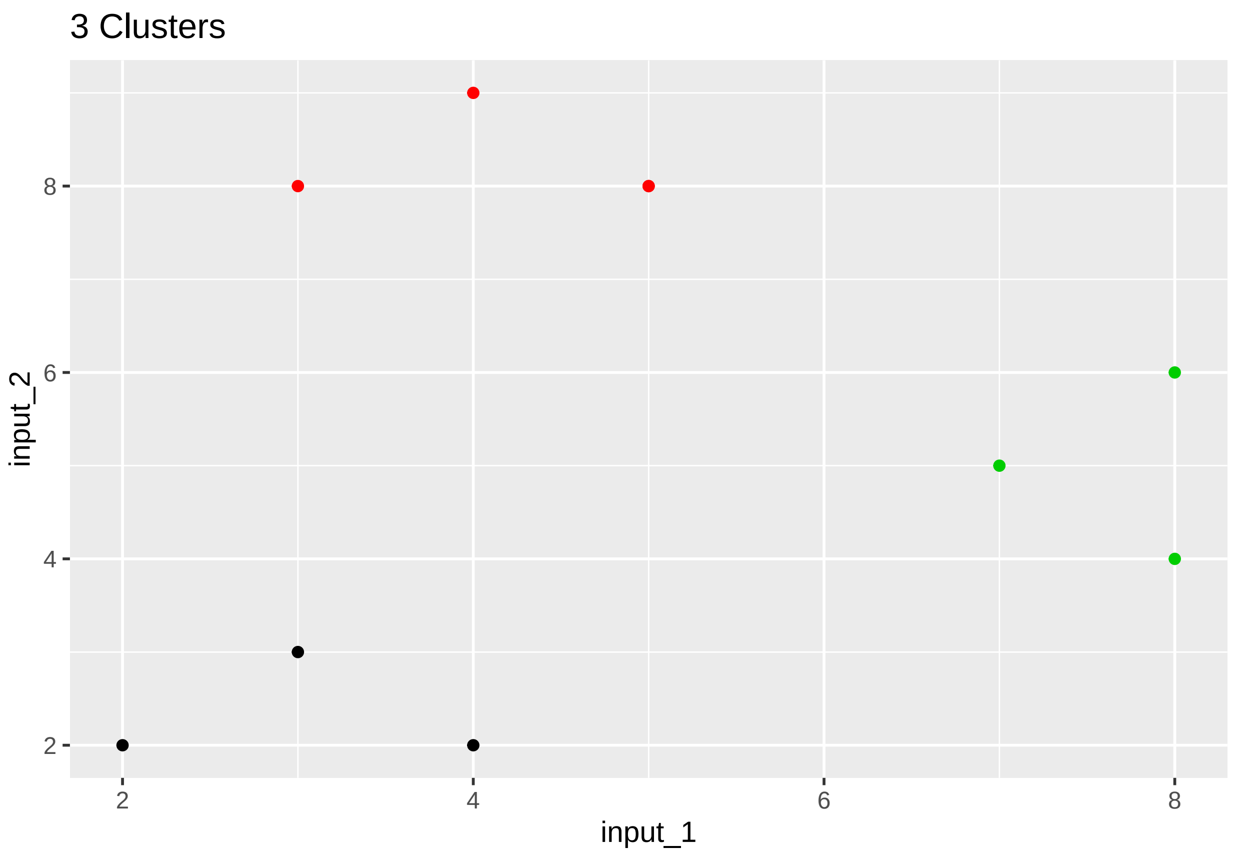
```r
clus3_1 <- sample(1:3, 10, replace = TRUE)
data <- as.data.frame(cbind(data, clus3_1))
```

2. Computer cluster centroid3 and update cluster assignments iteratively based on spatial similarity

```r
centroid3 <- data %>%
  group_by(clus3_1) %>%
  summarize(input_1 = mean(input_1),
         input_2 = mean(input_2))

assign3 <- function(x,y) {
  clust1 <- sqrt((x - centroid3$input_1[1])^2 +
                 (y - centroid3$input_2[1])^2)
  clust2 <- sqrt((x - centroid3$input_1[2])^2 +
                   (y - centroid3$input_2[2])^2)
  clust3 <- sqrt((x - centroid3$input_1[3])^2 +
                   (y - centroid3$input_2[3])^2)
  distance <- c(clust1, clust2, clust3)
  cluster <- 1:3
  distance <- as.data.frame(cbind(cluster, distance))
  new <- which.min(distance$distance)
  return(new)
}


 clus3_2 <- vector("integer", length = 10)
  for (i in 1:10){
    clus3_2[i] <- assign3(data$input_1[i], data$input_2[i])
  }
  data <- as.data.frame(cbind(data, clus3_2))

identical3 <- identical(data$clus3_1, data$clus3_2)

while(identical3 == FALSE) {
   data <- data %>%
    select(resp, input_1, input_2, clus3_2) %>%
    rename(clus3_1 = clus3_2)

  centroid3 <- data %>%
  group_by(clus3_1) %>%
  summarize(input_1 = mean(input_1),
         input_2 = mean(input_2))

  clus3_2 <- vector("integer", length = 10)
  for (i in 1:10){
    clus3_2[i] <- assign3(data$input_1[i], data$input_2[i])
  }
  data <- as.data.frame(cbind(data, clus3_2))

  identical3 <- identical(data$clus3_1, data$clus3_2)

}
```

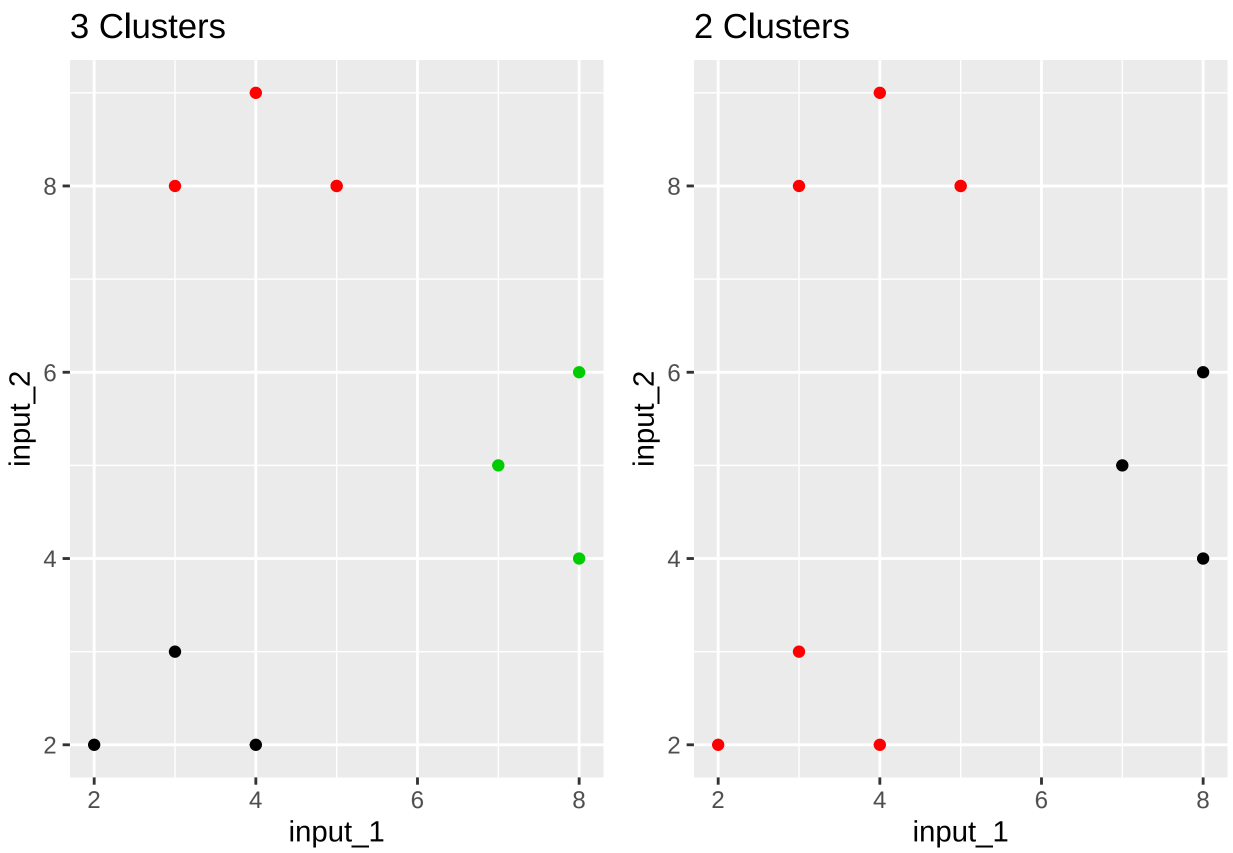
3. Present a visual description of the final, converged (stopped) cluster assignments

```
clus3 <- ggplot(data, aes(input_1, input_2)) +
  geom_point(color = clus3_2) +
  ggtitle("3 Clusters")
clus3
```



4. Repeat with `k = 2` and present final cluster assignment next to `k = 3` search

```
clus2_1 <- sample(1:2, 10, replace = TRUE)
data <- as.data.frame(cbind(data, clus2_1))

centroid2 <- data %>%
  group_by(clus2_1) %>%
  summarize(input_1 = mean(input_1),
         input_2 = mean(input_2))

assign2 <- function(x,y) {
  clust1 <- sqrt((x - centroid2$input_1[1])^2 +
                   (y - centroid2$input_2[1])^2)
  clust2 <- sqrt((x - centroid2$input_1[2])^2 +
                   (y - centroid2$input_2[2])^2)
  distance <- c(clust1, clust2)
  cluster <- 1:2
  distance <- as.data.frame(cbind(cluster, distance))
  new <- which.min(distance$distance)
  return(new)
}

clus2_2 <- vector("integer", length = 10)
  for (i in 1:10){
    clus2_2[i] <- assign2(data$input_1[i], data$input_2[i])
  }
  data <- as.data.frame(cbind(data, clus2_2))

identical2 <- identical(data$clus2_1, data$clus2_2)

while(identical2 == FALSE) {
   data <- data %>%
    select(resp, input_1, input_2, clus3_1, clus3_2, clus2_2) %>%
    rename(clus2_1 = clus2_2)

  centroid2 <- data %>%
  group_by(clus2_1) %>%
  summarize(input_1 = mean(input_1),
         input_2 = mean(input_2))

  clus2_2 <- vector("integer", length = 10)
  for (i in 1:10){
    clus2_2[i] <- assign2(data$input_1[i], data$input_2[i])
  }
  data <- as.data.frame(cbind(data, clus2_2))

  identical2 <- identical(data$clus2_1, data$clus2_2)
}

clus2 <- ggplot(data, aes(input_1, input_2)) +
  geom_point(color = clus2_2) +
  ggtitle("2 Clusters")
grid.arrange(clus3, clus2, ncol = 2)
```



5. `k = 3` versus `k = 2`

The initial hunch of `k = 3` panned out. Other values like `k = 2` are unlikely to fit these data better because there is a clear separation of 3 clusters in this particular dataset. A value of `k > 3` would subdivide these clusters too finely. A value of `k < 3` would combine two or more of these clusters.

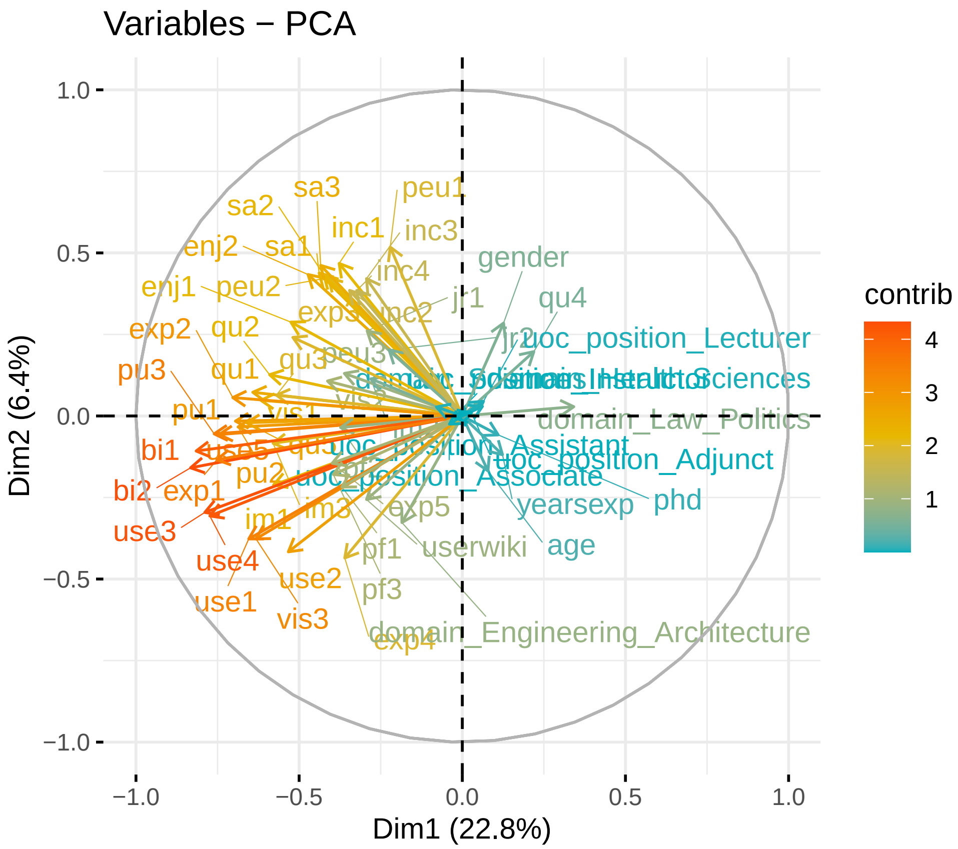
Application

```
wiki <- read.csv("data/wiki.csv")
wiki <- wiki %>%
  mutate_at(.vars = vars(age, yearsexp), scale)
```

Dimension Reduction

6. Perform PCA and describe results

```
wiki_pca <- prcomp(wiki, scale = TRUE)
wiki_pca_viz <- fviz_pca_var(wiki_pca,
                             col.var = "contrib",
                             gradient.cols = c("#00AFBB", "#E7B800","#FC4E07"),
                                        repel = TRUE)
wiki_pca_viz
```



For the first principal component, the variables `bi2`, `bi1`, `use3`, and `use4` appear to have the most weight (more than -0.75 correlation with Dim1). `bi1` and `bi2` appear to be correlated to each other. `use3` and `use4` appear to be correlated to each other as well.

For the second principal component, the variables `inc3`, `inc1`, and `sa3` appear to have the most weight (around 0.5 correlation with Dim2).

7. Calculate PVE and cumulative PVE

```
(VE <- wiki_pca$sdev^2)
```

```
##  [1] 13.00205784  3.63231049  2.86351292  2.32151607  2.14760243  1.91069275
##  [7]  1.72888851  1.45474119  1.37793331  1.36373341  1.29142071  1.18057457
## [13]  1.15595790  1.08489802  1.02172080  0.99616053  0.98400999  0.92296209
## [19]  0.87122273  0.83094092  0.81683007  0.76933871  0.73876336  0.67976548
## [25]  0.65372029  0.64370470  0.61876018  0.56324365  0.54256517  0.49376465
## [31]  0.49219629  0.47303067  0.46516274  0.45003305  0.41800729  0.41454828
## [37]  0.39425783  0.38853138  0.37658542  0.35623637  0.33197337  0.33118604
## [43]  0.31921754  0.30927548  0.30717210  0.29188432  0.28838229  0.27361923
## [49]  0.26569770  0.25822398  0.24830953  0.21906356  0.21436827  0.19281595
## [55]  0.13406607  0.11212851  0.01071333
```

```
PVE <- VE / sum(VE)
PVE[1]
```

```
## [1] 0.2281063
```

```
PVE[2]
```

```
## [1] 0.06372475
```

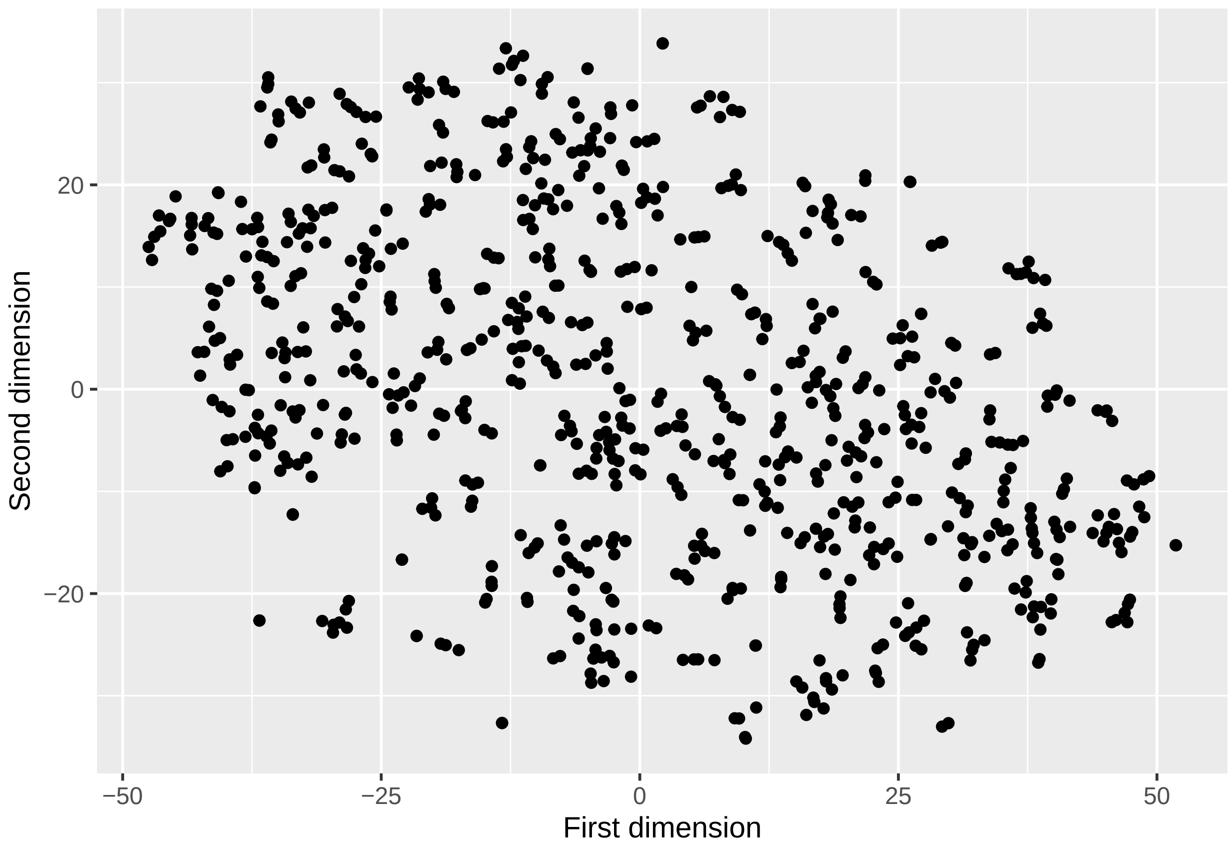
```
sum(PVE)
```

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

The first two principal components explain 22.8% and 6.4% of the variance.

8. Perform t-SNE and plot observations

```
wiki_tsne <- Rtsne(as.matrix(wiki),
                      perplexity = 5)
wiki_tsne_plot <- wiki %>%
  mutate(tsne1 = wiki_tsne$Y[,1],
         tsne2 = wiki_tsne$Y[,2]) %>%
  ggplot(aes(tsne1, tsne2)) +
  geom_point() +
  labs(x = "First dimension",
       y = "Second dimension")
wiki_tsne_plot
```



The t-SNE plot does not yield clear clusters in the data. There is a vague football shape in the data along the y = x line.

Clustering

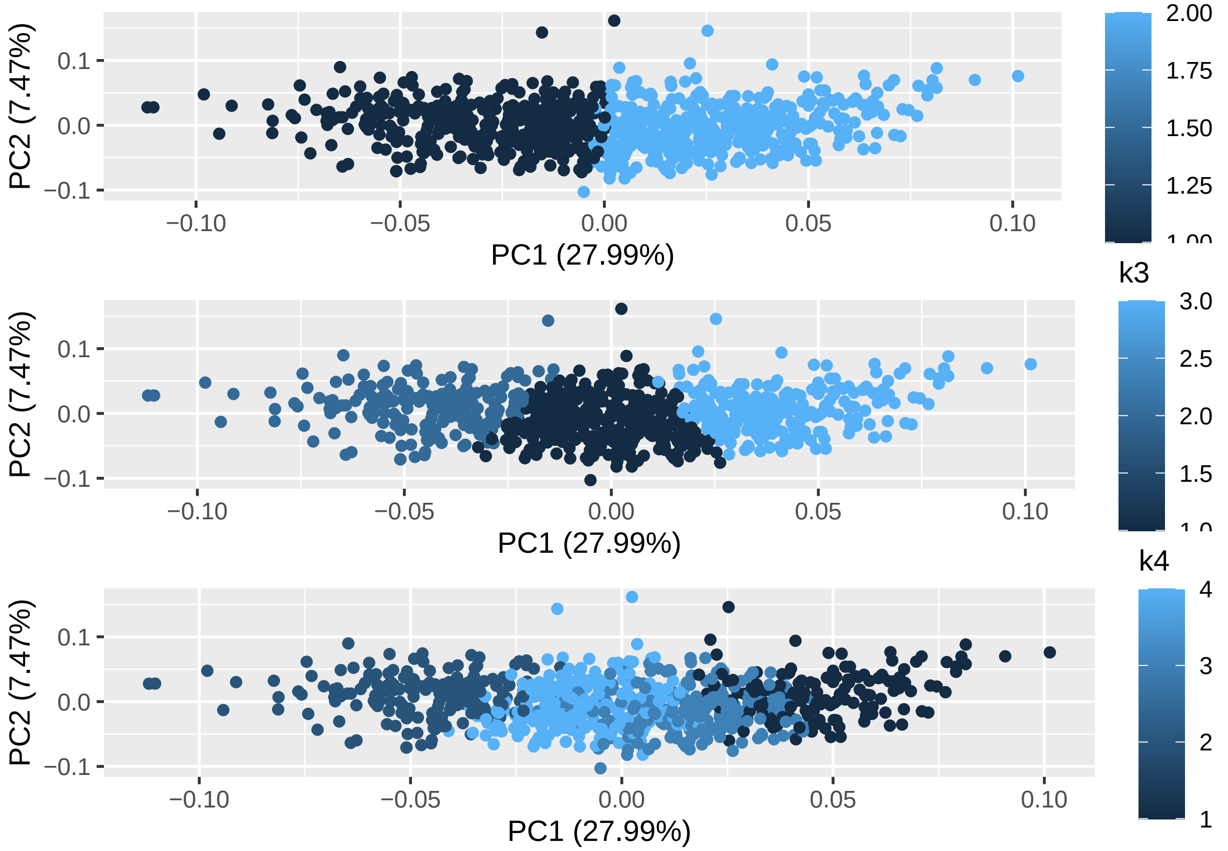
9. K-means clustering

K-means clustering with k = 2,3,4

```
k2 <- kmeans(wiki, 2, nstart = 20)$cluster
k3 <- kmeans(wiki, 3, nstart = 20)$cluster
k4 <- kmeans(wiki, 4, nstart = 20)$cluster
wiki_out <- as.data.frame(cbind(wiki, k2, k3, k4))
```

Plot observations on 1st and 2nd principal components from PCA

```
k2_plot <- autoplot(prcomp(wiki), data = wiki_out, colour = 'k2')
k3_plot <- autoplot(prcomp(wiki), data = wiki_out, colour = 'k3')
k4_plot <- autoplot(prcomp(wiki), data = wiki_out, colour = 'k4')
grid.arrange(k2_plot, k3_plot, k4_plot, ncol = 1)
```

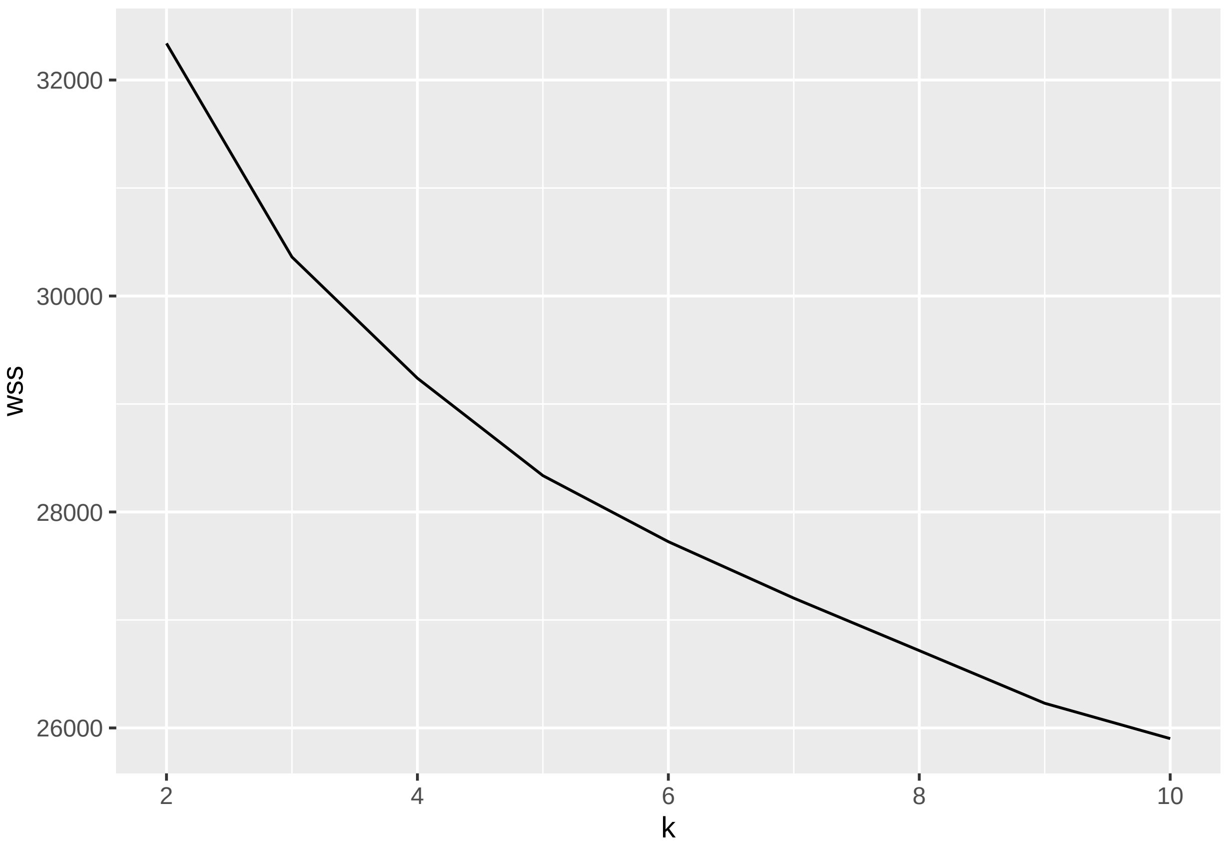


Discuss results

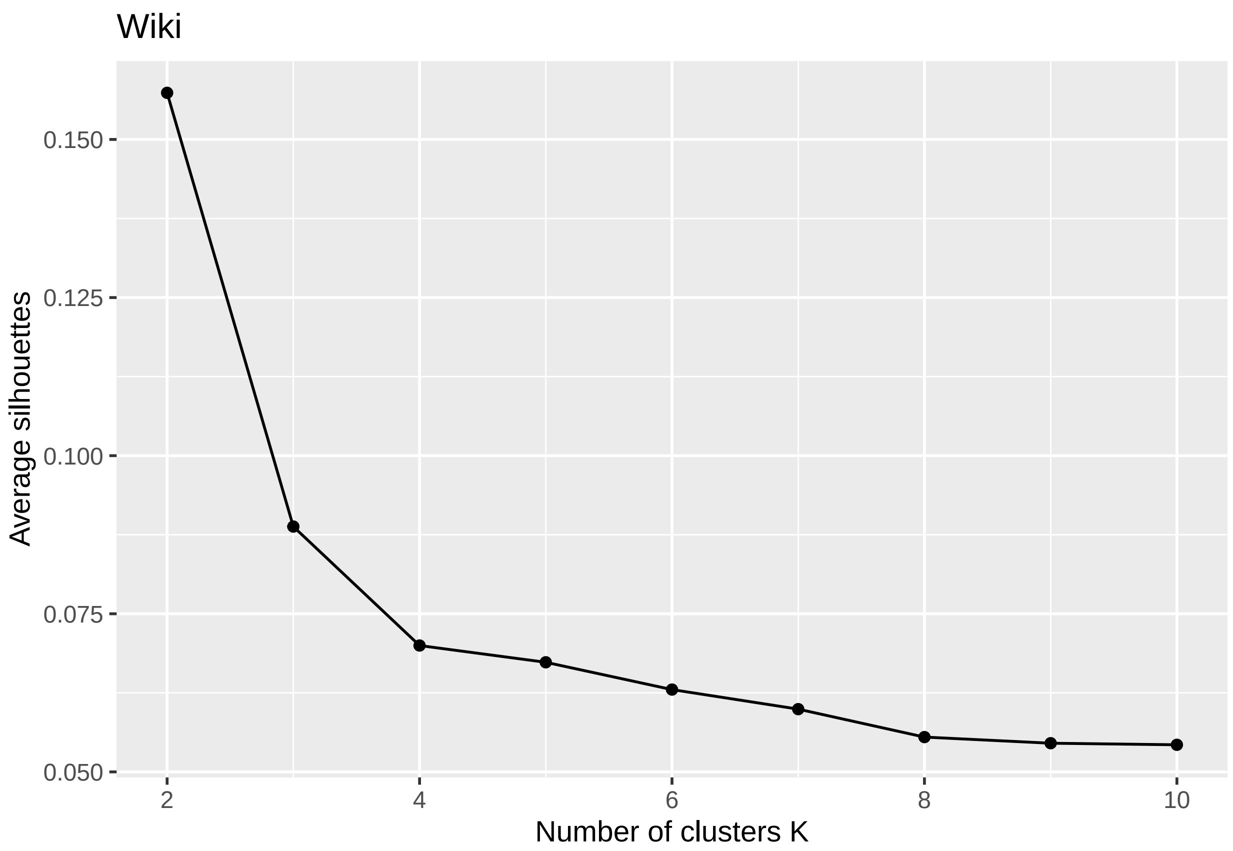
Given that the clusters when `k = 4` overlap, the data suggest that the ideal number of k is either 2 or 3.

10. Elbow method, average silhouette, and/or gap statistic

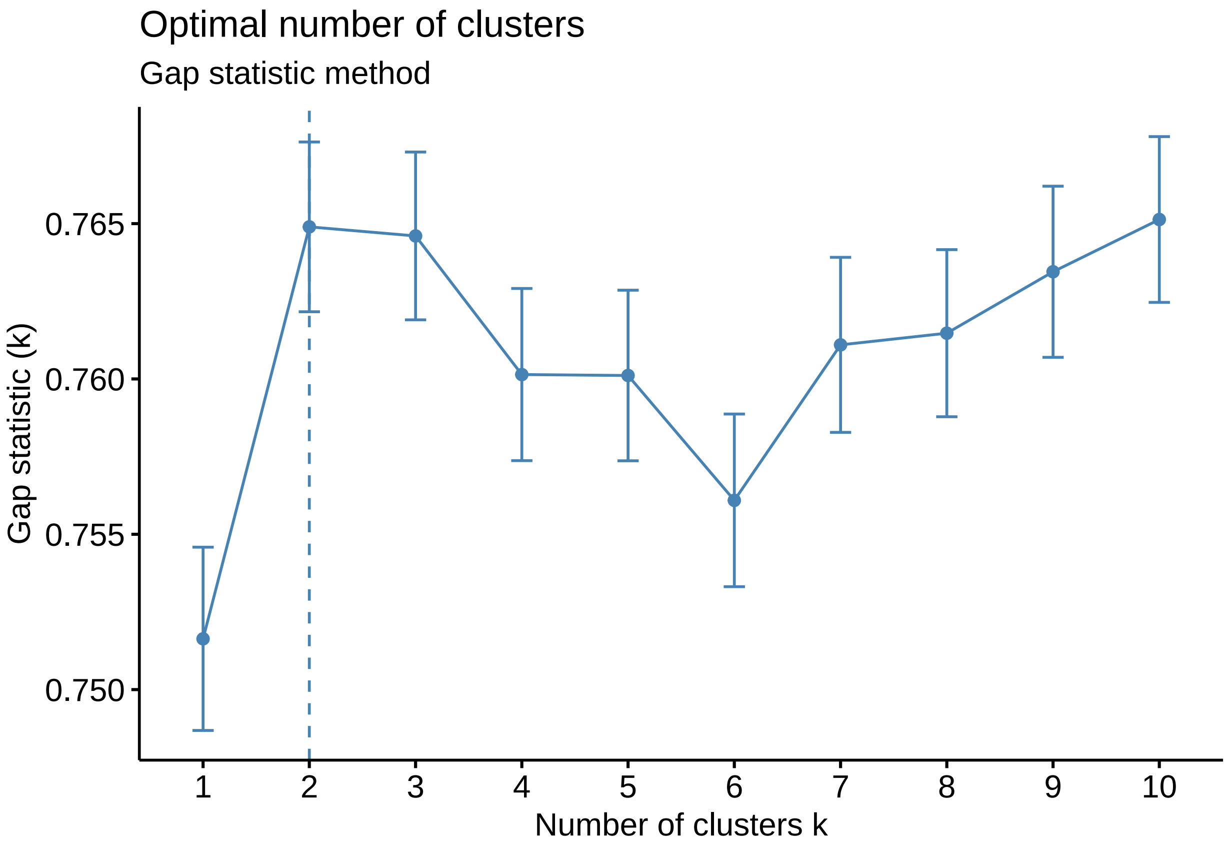
```r
wss <- 0
for (i in 2:10) {
  km.out <- kmeans(wiki, centers = i, nstart = 20)
  wss[i] <- km.out$tot.withinss
}

k <- 1:10
elbow_table <- as.data.frame(cbind(k, wss))
elbow_table <- elbow_table[-1,]
ggplot(elbow_table, aes(x = k, y = wss)) +
  geom_line()
```



```
avg_sil <- function(k) {
  km.res <- kmeans(wiki, centers = k, nstart = 20)
  ss <- silhouette(km.res$cluster, dist(wiki))
  mean(ss[, 3])
}

tibble(
  k = 2:10
) %>%
  mutate(avg_sil = map_dbl(k, avg_sil)) %>%
  ggplot(aes(k, avg_sil)) +
  geom_line() +
  geom_point() +
  labs(title = "Wiki",
       x = "Number of clusters K",
       y = "Average silhouettes")
```



```
gapstat <- fviz_nbclust(wiki, kmeans, nstart = 20,
                        method = "gap_stat", nboot = 50) +
  labs(subtitle = "Gap statistic method")
gapstat
```

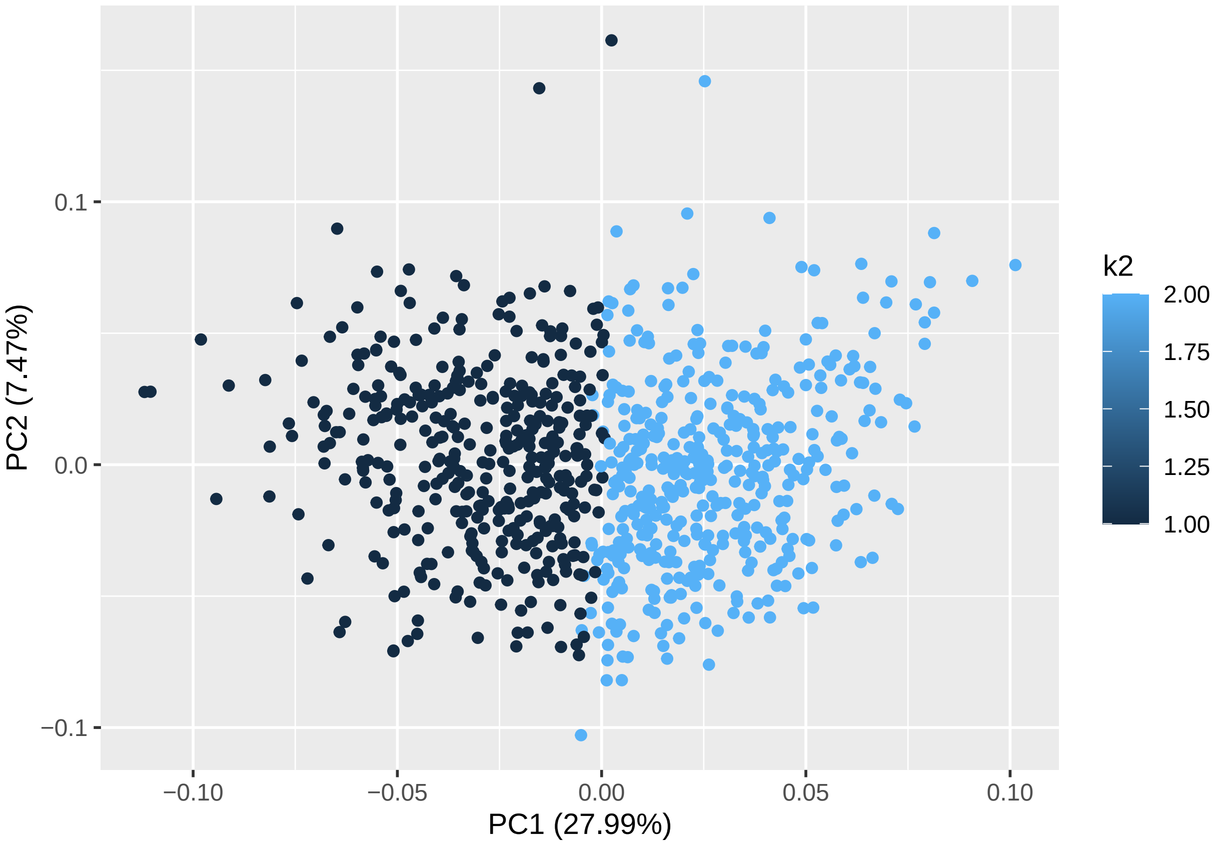


Though the elbow method does not provide an obvious number of clusters, the silhouette and gap statistic methods suggest 2 as the best number of clusters.

11. Visualize results of optimal k-means clustering model

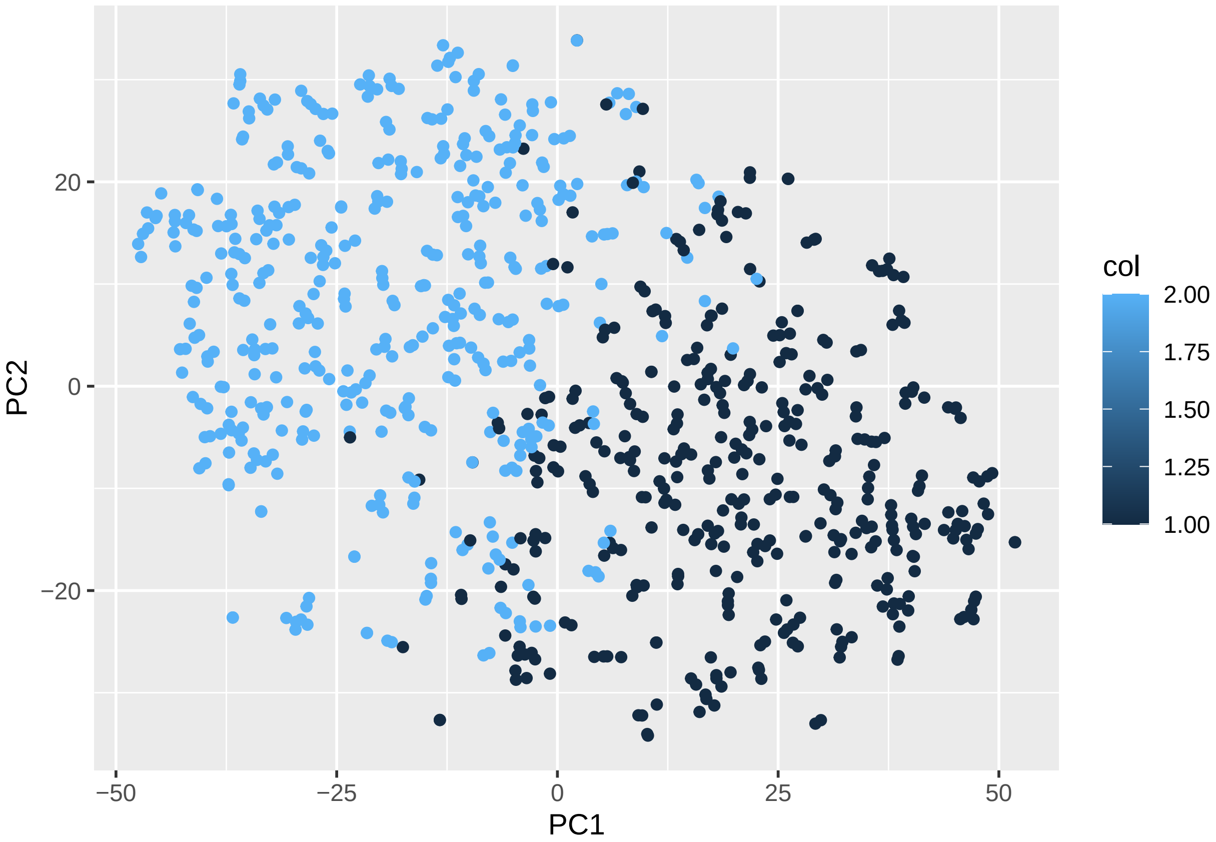
Use 1st and 2nd principal components from PCA

```
k2_plot
```



Use 1st and 2nd principal components from t-SNE

```
tsne_plot <- data.frame(x = wiki_tsne$Y[,1], y = wiki_tsne$Y[,2], col = wiki_out$k2)
ggplot(tsne_plot) + geom_point(aes(x = x, y = y, color=col)) +
  labs(x = "PC1", y = "PC2")
```



Describe results

In this instance, PCA yields clusters that are more cleanly separated and compact than t-SNE, which has clusters that are partially-overlapping and more spread out. Given that PCA works better on linear data and t-SNE works better on non-linear data, the differing results might show that there is an underlying linear relationship within the data.