MACS30100 Homework 7

Takuyo Ozaki

3/14/2020

```r
knitr::opts_chunk$set(message = FALSE, warning = FALSE)
knitr::opts_chunk$set(fig.width = 7, fig.height = 3.5, fig.align = 'center')

# load necessary package
library(tidyverse)
library(broom)
library(rsample)
library(tibble)
library(knitr)
library(caret)
library(cluster)
library(kernlab)
library(Rtsne)
library(patchwork)
library(factoextra)
library(doParallel)
cl <- makePSOCKcluster(4)
registerDoParallel(cl)

# Set the theme as minimal
theme_set(theme_minimal())
```

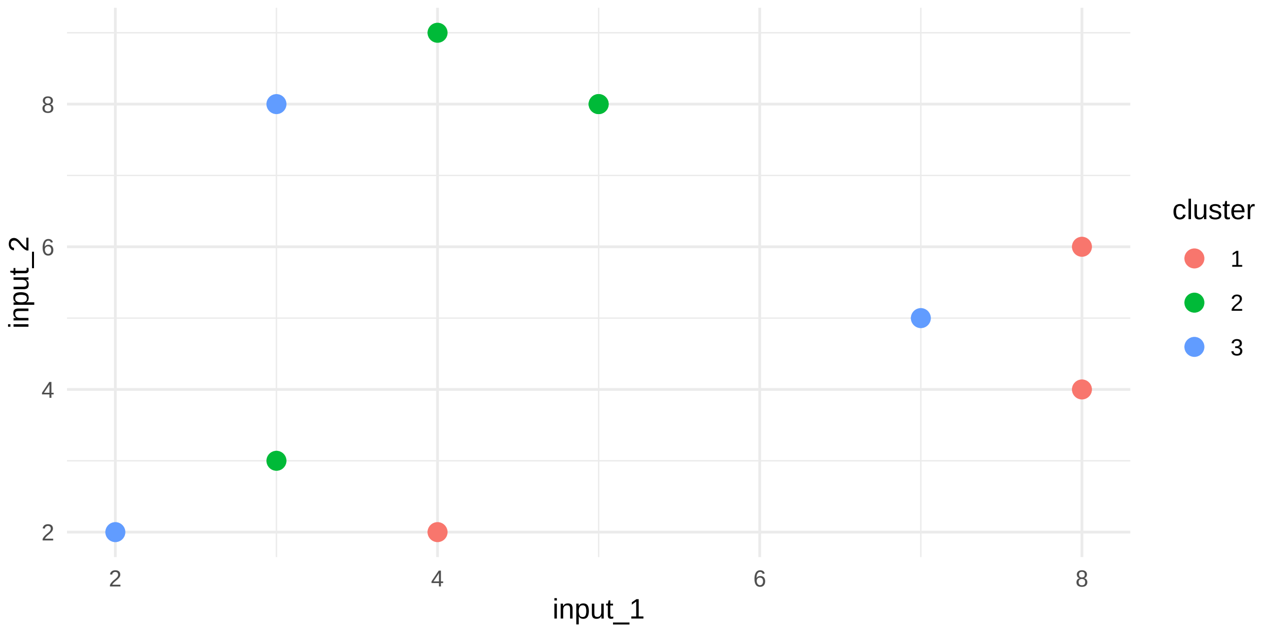
k-Means Clustering "By Hand"

Question 1

```r
set.seed(0) # Ensure the reproducibility

# Create data table by input 1 and 2
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)
x <- cbind(input_1, input_2)

# Assign random cluster number vector
(cluster <- as.factor(sample(3, nrow(x), replace = T)))
```

```
##  [1] 2 1 3 1 2 1 3 3 2 2
## Levels: 1 2 3
```

```
# Plot the cluster by the first random assingment
as.data.frame(x) %>% mutate(cluster = cluster) %>%
  ggplot(aes(x = input_1, y = input_2, color = cluster)) +
  geom_point(size = 3)
```



Question 2

```
set.seed(0) # Ensure the reproducibility

# Compute the cluster centroid
centroid1 <- c(mean(x[cluster == 1, 1]), mean(x[cluster == 1, 2]))
centroid2 <- c(mean(x[cluster == 2, 1]), mean(x[cluster == 2, 2]))
centroid3 <- c(mean(x[cluster == 3, 1]), mean(x[cluster == 3, 2]))
centroid <- c("input1", "input2")
kable(data.frame(centroid, cluster1 = centroid1, cluster2 = centroid2, cluster3 = centroid3))
```

centroid	cluster1	cluster2	cluster3
input1	6.666667	4.25	4
input2	4.000000	7.00	5

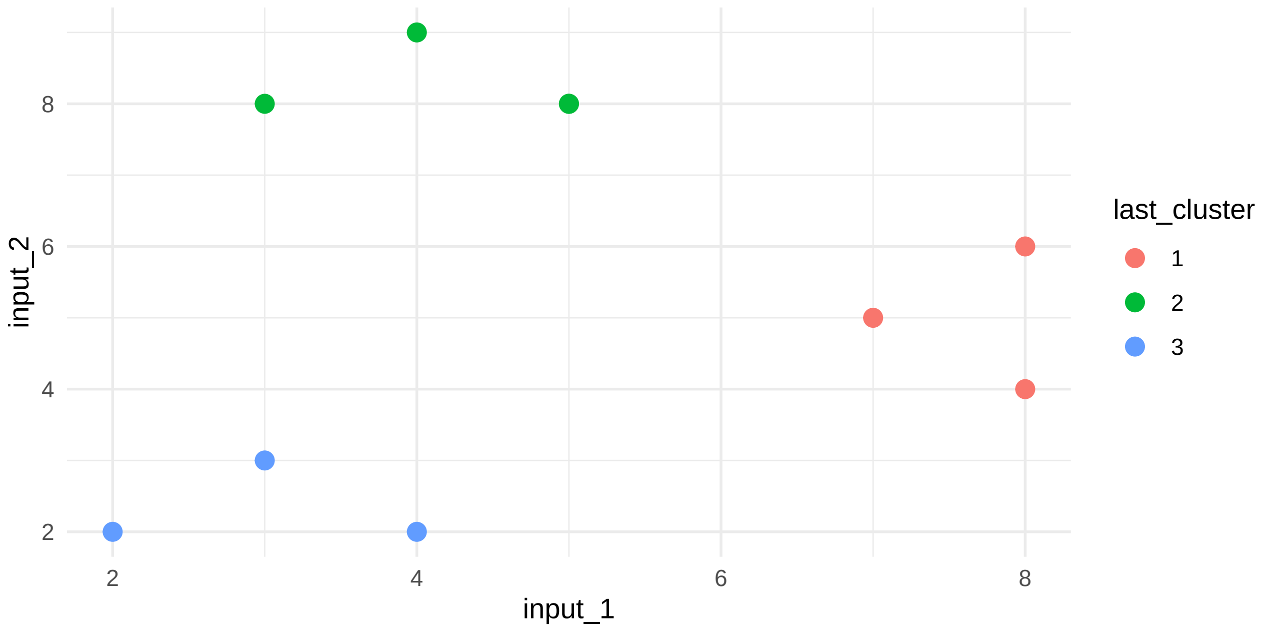
```
# Generate a function for updating clusters (three clusters ver.)
euclid <- function(a, b) {sqrt((a[1] - b[1])^2 + (a[2] - b[2])^2)}
cluster3_func = function(x, centroid1, centroid2, centroid3) {
  cluster = rep(NA, nrow(x))
  for (i in 1:nrow(x)) {
    if (euclid(x[i,], centroid1) < euclid(x[i,], centroid2)
        & euclid(x[i,], centroid1) < euclid(x[i,], centroid3))
```

```
      {cluster[i] = 1}
    else if (euclid(x[i,], centroid2) < euclid(x[i,], centroid3))
      {cluster[i] = 2}
    else {cluster[i] = 3}
  }
  return(cluster)
}

# Show the final cluster by updating iteratively
last_cluster = rep(-1, 10)
while (!all(last_cluster == cluster)) {
  last_cluster = cluster
  cluster <- cluster3_func(x, centroid1, centroid2, centroid3)
}
last_cluster
```

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

Question 3

```
# Plot the cluster by the final result
as.data.frame(x) %>% mutate(last_cluster = as.factor(last_cluster)) %>%
  ggplot(aes(x = input_1, y = input_2, color = last_cluster)) +
  geom_point(size = 3)
```

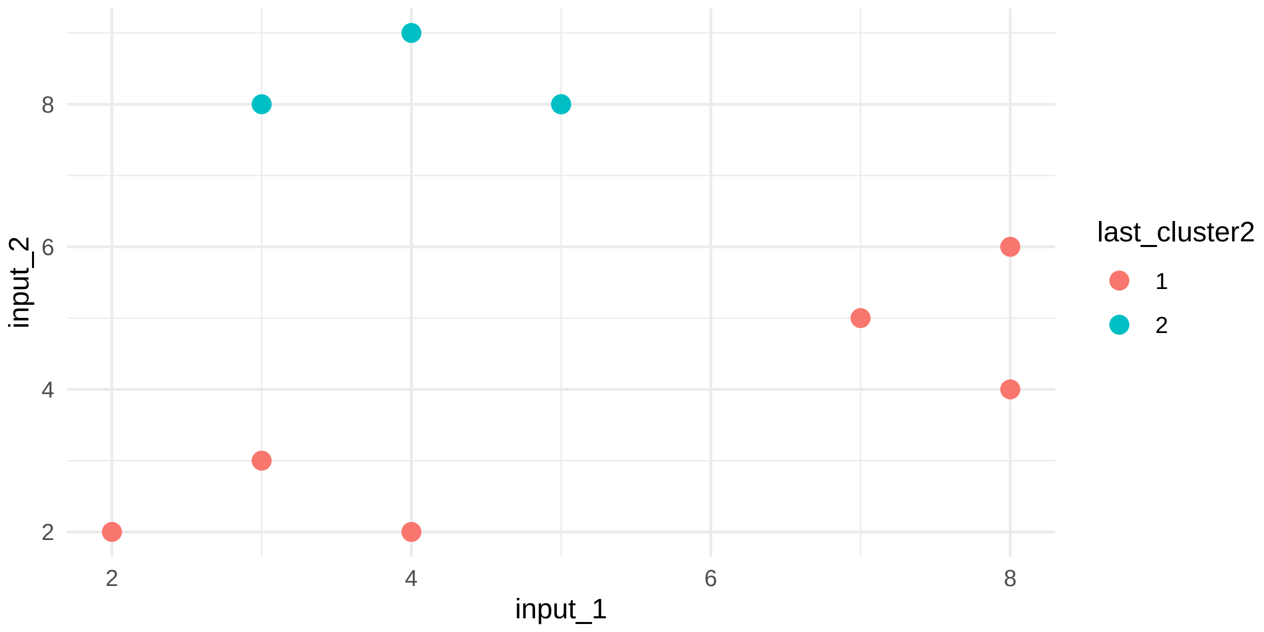


Question 4

```r
set.seed(0) # Ensure the reproducibility

# Generate a function for updating clusters (two clusters ver.)
cluster2_func = function(x, centroid1, centroid2) {
  cluster = rep(NA, nrow(x))
  for (i in 1:nrow(x)) {
    if (euclid(x[i,], centroid1) < euclid(x[i,], centroid2)) {cluster[i] = 1}
    else {cluster[i] = 2}}
  return(cluster)}

# Show the final cluster by updating iteratively
last_cluster2 = rep(-1, 10)
while (!all(last_cluster2 == cluster)) {
  last_cluster2 = cluster
  cluster <- cluster2_func(x, centroid1, centroid2)
}
last_cluster2
```

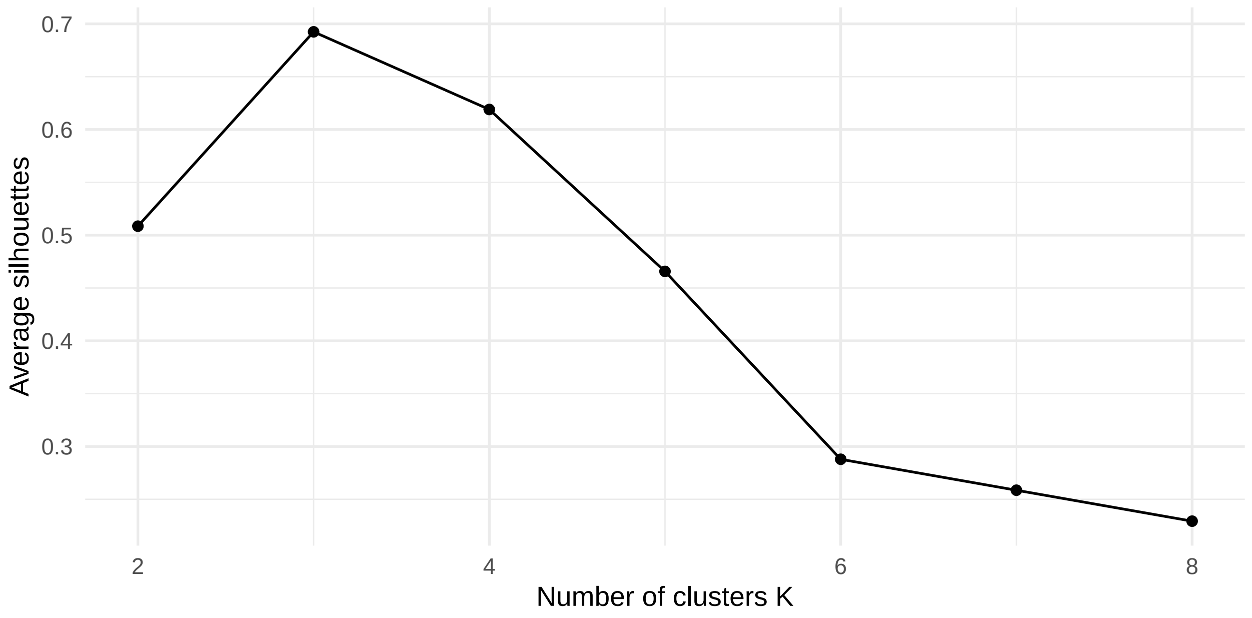
```
##  [1] 2 1 1 1 1 1 1 2 2 2
```

```r
# Plot the cluster by the final result
as.data.frame(x) %>% mutate(last_cluster2 = as.factor(last_cluster2)) %>%
  ggplot(aes(x = input_1, y = input_2, color = last_cluster2)) +
  geom_point(size = 3)
```



Question 5

```r
set.seed(0) # Ensure the reproducibility

# Generate a function to compute average silhouette for k clusters
data1 <- data.frame(input_1 = input_1, input_2 = input_2)
avg_sil <- function(k) {
  km.res <- kmeans(data1, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(data1))
  mean(ss[, 3])}

# Extract avg silhouette
tibble(k = 2:8) %>%
  mutate(avg_sil = map_dbl(k, avg_sil)) %>%
  ggplot(aes(k, avg_sil)) +
  geom_line() +
  geom_point() +
  labs(x = "Number of clusters K",
       y = "Average silhouettes")
```



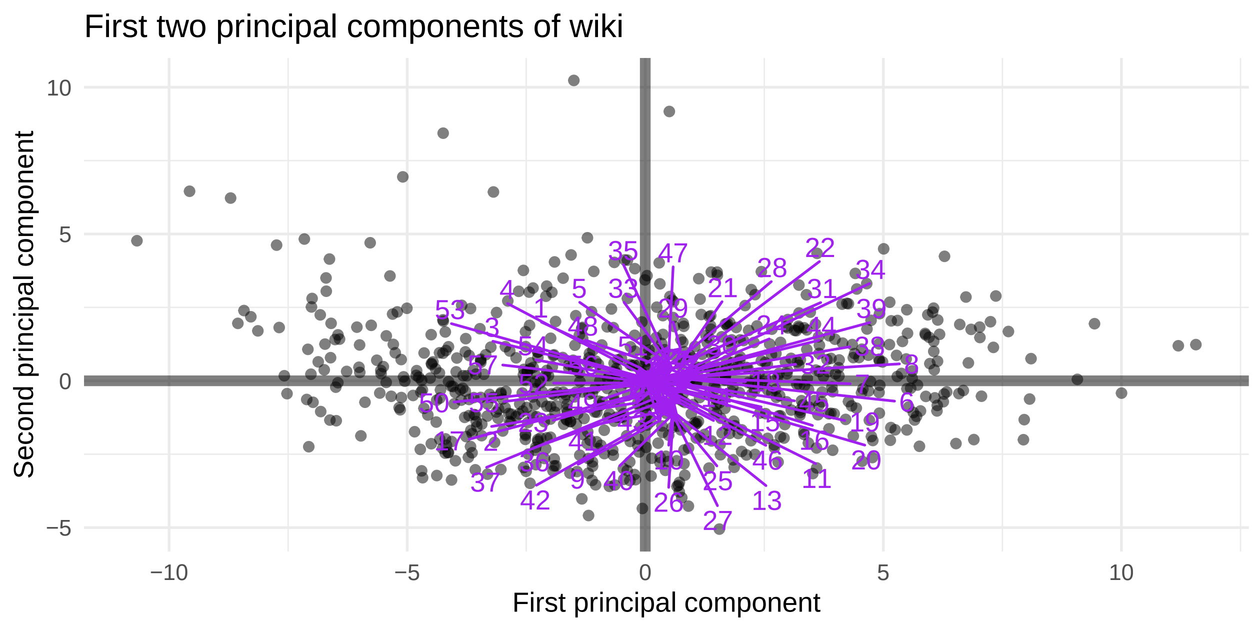
The number of clusters k = 3 has the best fit because of the maximum average silhouettes as the graph above shows.

Application

Dimension reduction

Question 6

```r
set.seed(0) # Ensure the reproducibility

# Import the data
wiki <- read_csv("data/wiki.csv")

# Scaled
scaled_wiki <- wiki %>% mutate_at(.vars = vars(colnames(wiki)), scale)

# Calculating covariance matrix
wiki_cov <- scaled_wiki %>% cov()

# Extract eigen values
wiki_eigen <- eigen(wiki_cov)

# Recover PC loading vectors and extract first two loadings (for a simple plot)
phi <- wiki_eigen$vectors[, 1:2]
colnames(phi) <- c("PC1", "PC2")

# Calculate PC scores
PC1 <- as.matrix(select_if(scaled_wiki, is.numeric)) %*% phi[,1]
PC2 <- as.matrix(select_if(scaled_wiki, is.numeric)) %*% phi[,2]

# Create data frame with PC scores
PC <- tibble(PC1 = PC1[,1], PC2 = PC2[,1])

# biplot
phi_df <- (phi * 5) %>%
  as.data.frame %>%
  rownames_to_column(var = "variable")

ggplot(PC, aes(PC1, PC2)) +
  geom_vline(xintercept = 0, size = 2, color = "black", alpha = .5) +
  geom_hline(yintercept = 0, size = 2, color = "black", alpha = .5) +
  geom_point(alpha = 0.5) +
  geom_segment(data = phi_df,
               aes(x = 0, y = 0,
                   xend = PC1,
                   yend = PC2),
               arrow = arrow(length = unit(0.03, "npc")),
               color = "purple") +
  ggrepel::geom_text_repel(data = phi_df,
                           aes(x = PC1, y = PC2, label = variable),
                           color = "purple") +
  labs(title = "First two principal components of wiki",
       x = "First principal component",
       y = "Second principal component")
```



```r
# Variable names
phi_df$variable_name <- colnames(wiki)

# Show top 5 variables that are correlated on the first principal component
kable(phi_df %>% arrange(desc(PC1)) %>% head(n = 5))
```

variable	PC1	PC2	variable_name
39	1.154620	0.4171544	bi2
38	1.130965	0.2818714	bi1
30	1.094046	0.7757579	use3
31	1.072792	0.8043226	use4
8	1.054313	0.1438814	pu3

```r
# Show top 5 variables that are correlated on the first principal component
kable(phi_df %>% arrange(desc(PC2)) %>% head(n = 5))
```

variable	PC1	PC2	variable_name
47	0.4993644	1.1424714	exp4
29	0.7392588	1.0931425	use2
28	0.9073858	0.9891375	use1
21	0.8767565	0.9881737	vis3
51	0.2565455	0.8574190	domain_Engineering_Architecture

I will show the top five variables to answer the question. “bi2”, “bi1”, “use3”, “use4”, “pu3” are strongly correlated on the first principal component. “exp4”, “use2”, “use1”, “vis3”, domain_Engineering_Architecture” are strongly correlated on the second principal component.

Question 7

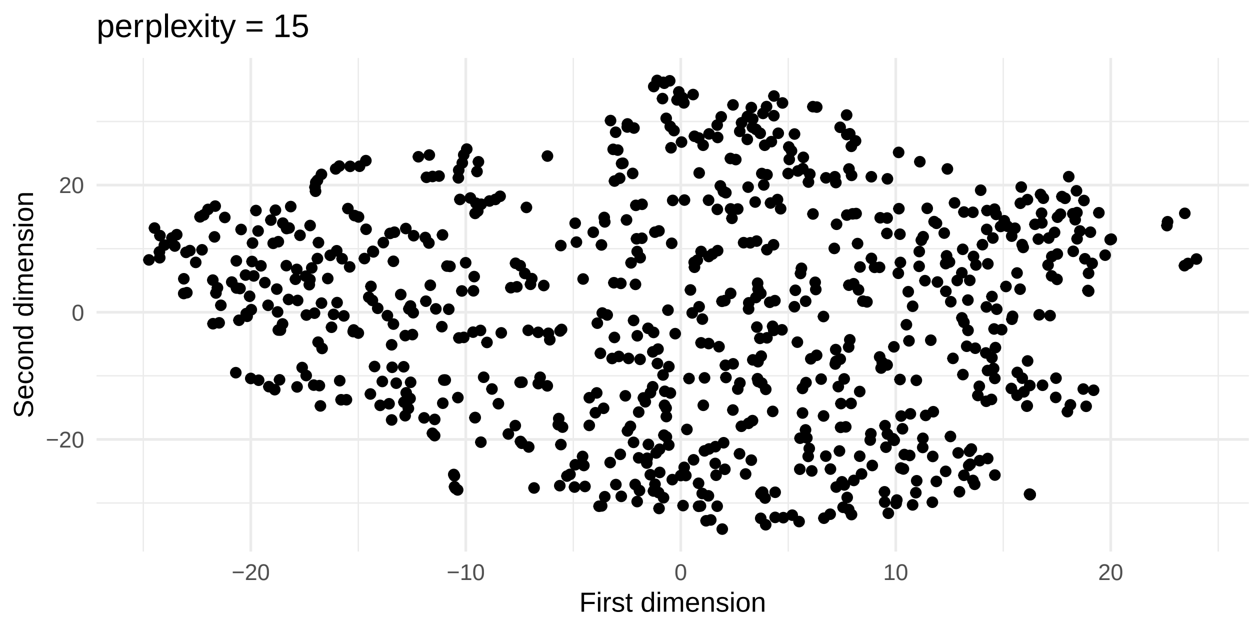
```r
# Calculate PVE and cumulative PVE for all the principal components
(wiki_pve <- tibble(
  var = wiki_eigen$values,
  var_exp = var / sum(var),
  cum_var_exp = cumsum(var_exp)) %>%
  mutate(pc = row_number()))
```

```
## # A tibble: 57 x 4
##      var var_exp cum_var_exp    pc
##    <dbl>   <dbl>       <dbl> <int>
##  1 13.0   0.228        0.228     1
##  2  3.63  0.0637       0.292     2
##  3  2.86  0.0502       0.342     3
##  4  2.32  0.0407       0.383     4
##  5  2.15  0.0377       0.420     5
##  6  1.91  0.0335       0.454     6
##  7  1.73  0.0303       0.484     7
##  8  1.45  0.0255       0.510     8
##  9  1.38  0.0242       0.534     9
## 10  1.36  0.0239       0.558    10
## # ... with 47 more rows
```

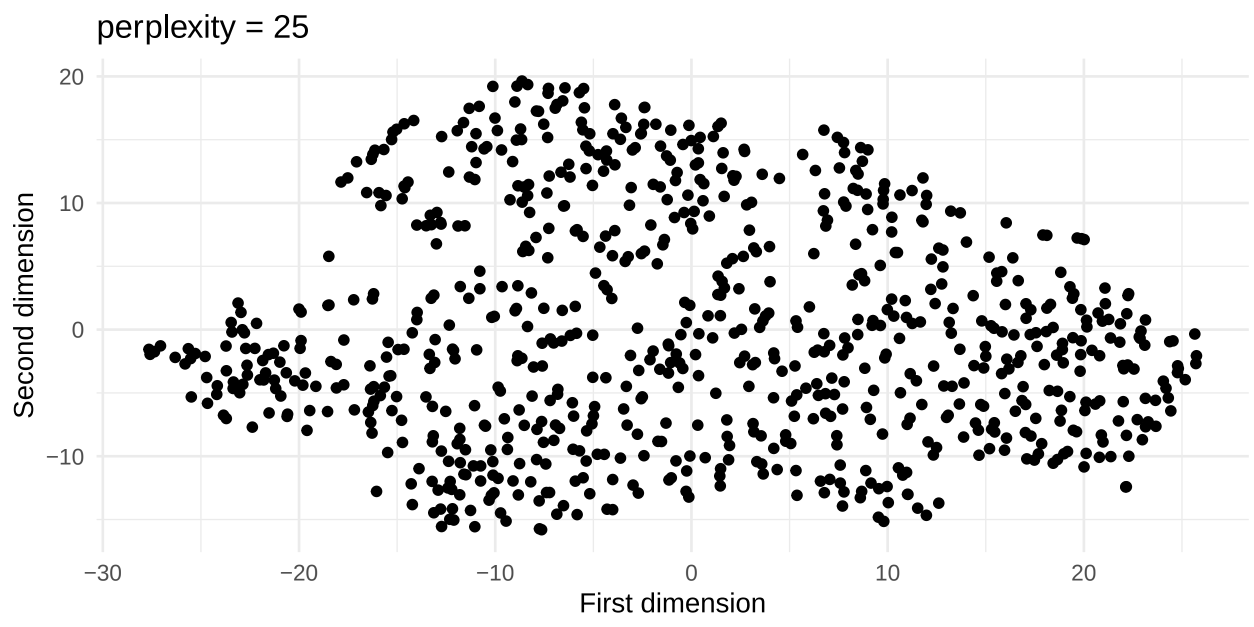
As the table above shows, the cumulative PVE at PC2 is 0.2918310. Approximately, 29.2% of the variance is explained by the first two principal components.

Question 8

```r
set.seed(0) # Ensure the reproducibility

# Create a function of t-SNE plot with perplexity
plot_tsne <- function(x){
  wiki_tsne <- Rtsne(as.matrix(wiki), perplexity = x)
  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",
       title = paste("perplexity =", x))
}

# Plot t-SNE when perplexity = 15
plot_tsne(15)
```



```
# Plot t-SNE when perplexity = 25
plot_tsne(25)
```



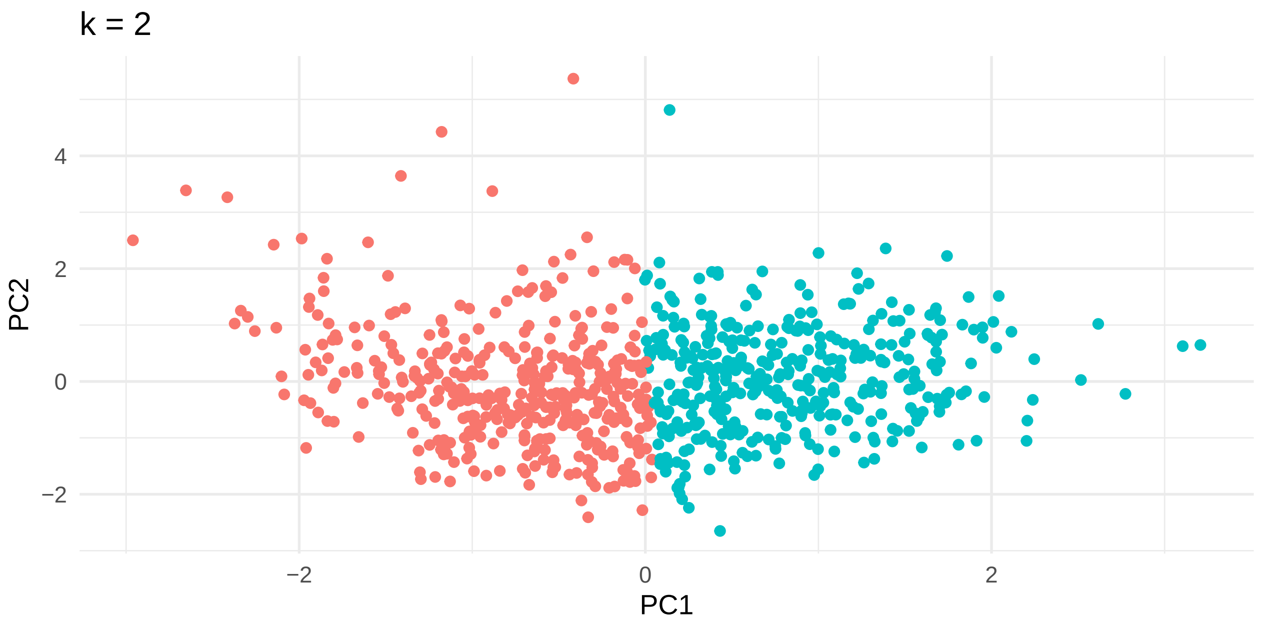
van der Maaten and Hinton suggest that perplexity is from 5 to 50. The plot of the perplexity 15 is not clear, but the plot of the perplexity 25 looks that the observations on the first and second dimensions has three clusters.

Clustering

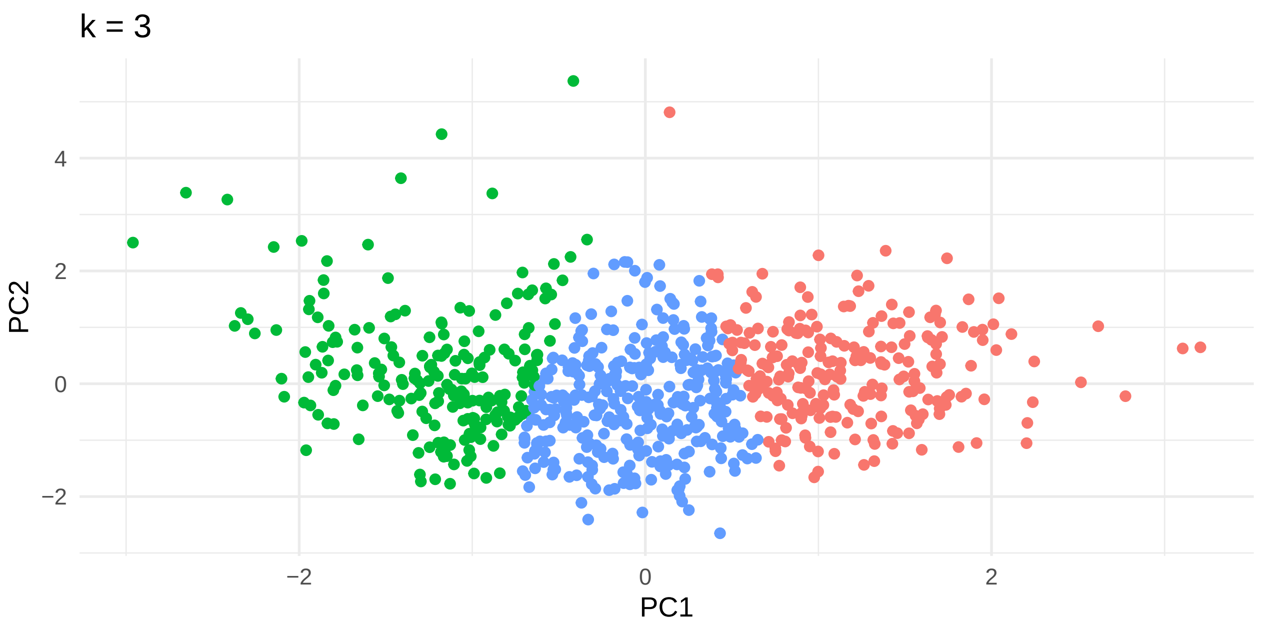
Question 9

```r
set.seed(0) # Ensure the reproducibility

# Scale each feature
PC_scale <- PC %>%  mutate_at(.vars = vars(PC1, PC2), scale)

# Generate a function by k
plot_PC <- function(k){
  PC_scale %>%
    mutate(cluster = kmeans(PC, k, nstart = 25)$cluster) %>%
    ggplot(aes(x = PC1, y = PC2, color = factor(cluster))) +
    geom_point() +
    labs(title = paste("k =", k)) + theme(legend.position = 'none')
}
```

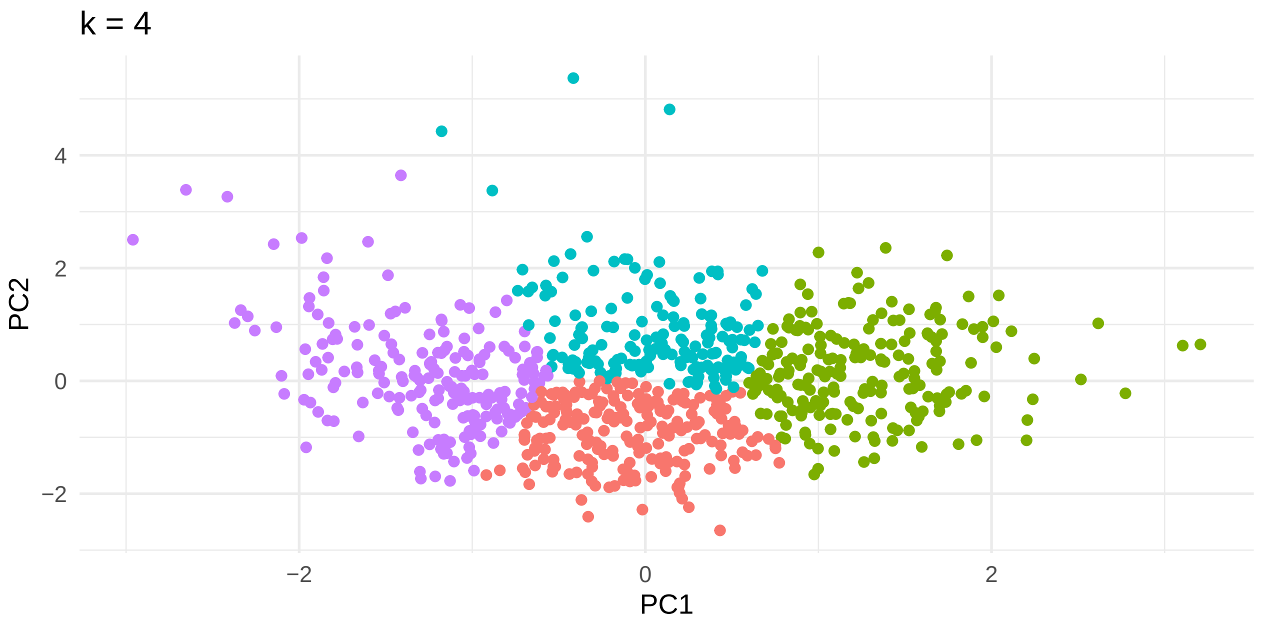
```r
plot_PC(2) # Plot k = 2
```



```r
plot_PC(3) # Plot k = 3
```



```r
plot_PC(4) # Plot k = 4
```

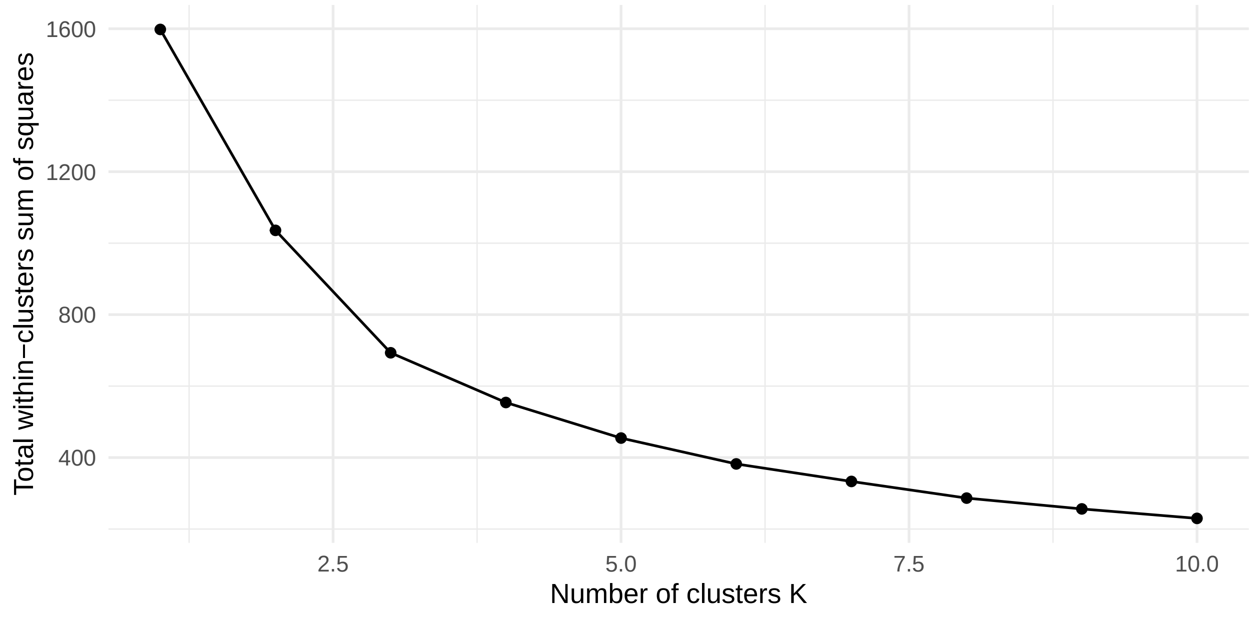


Looking at the plots of the observations on the first and second principal components in case of k = 2, 3, 4, we are not sure that we can separate clearly the observations by the clusters.

Question 10

Elbow method

```r
set.seed(0) # Ensure the reproducibility

# Elbow method
wss <- function(k) {kmeans(PC_scale, k, nstart = 10)$tot.withinss}

tibble(k = 1:10) %>%
  mutate(wss = map_dbl(k, wss)) %>%
  ggplot(aes(k, wss)) +
  geom_line() +
  geom_point() +
  labs(x = "Number of clusters K",
       y = "Total within-clusters sum of squares")
```

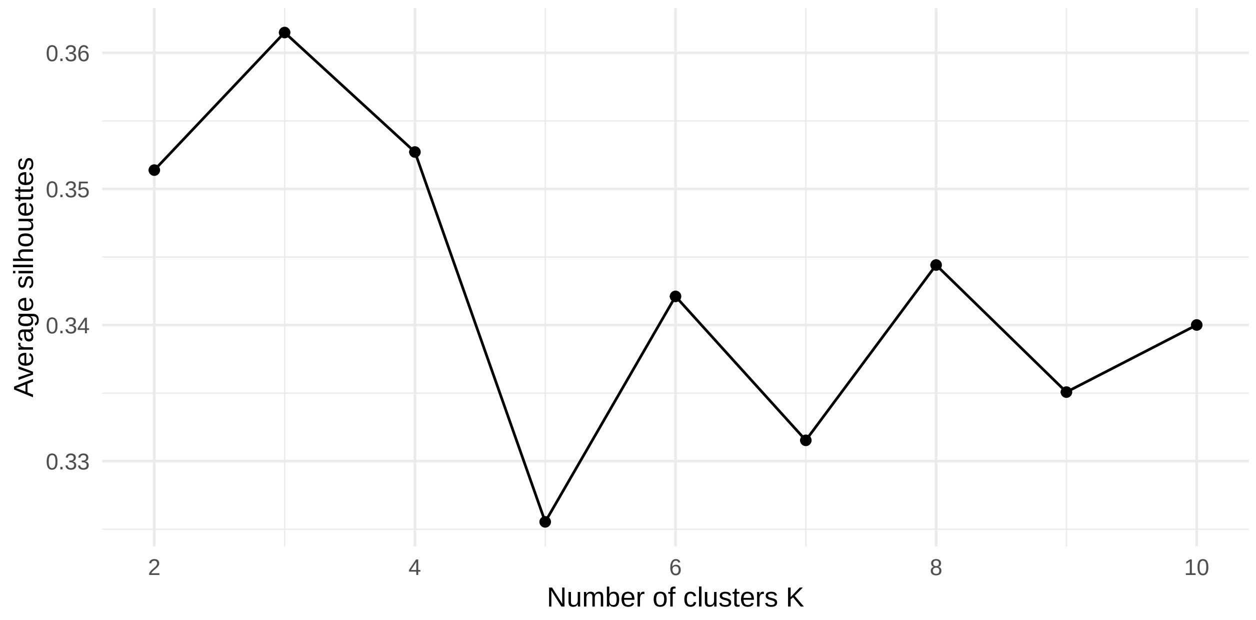


Although it seems unclear, the optimal number of clusters is k = 3.

Average silhouette

```r
set.seed(0) # Ensure the reproducibility

# Average silhouette
avg_sil <- function(k) {
  km.res <- kmeans(PC_scale, centers = k, nstart = 25)
  ss <- silhouette(km.res$cluster, dist(PC_scale))
  mean(ss[, 3])
}
```

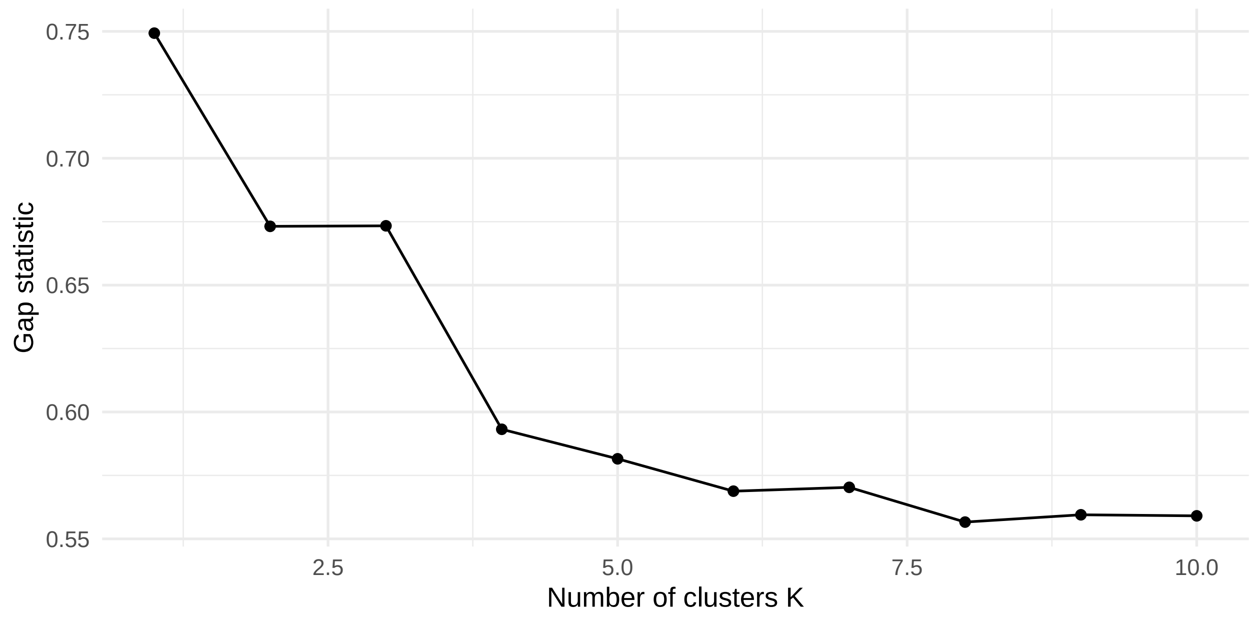
```r
tibble(k = 2:10) %>%
  mutate(avg_sil = map_dbl(k, avg_sil)) %>%
  ggplot(aes(k, avg_sil)) +
  geom_line() +
  geom_point() +
  labs(x = "Number of clusters K",
       y = "Average silhouettes")
```



The optimal number of clusters is k = 3 because of the maximum average silhouettes.

Gap statistic

```r
set.seed(0) # Ensure the reproducibility

# Gap statistic
gap_stat <- clusGap(PC_scale, FUN = kmeans, nstart = 25, K.max = 10, B = 50)

# Plot the Gap statistic
as.data.frame(gap_stat$Tab) %>%
  mutate(k = 1:10) %>%
  ggplot(aes(x = k, y = gap)) +
  geom_line() +
  geom_point() +
  labs(x = "Number of clusters K",
       y = "Gap statistic")
```



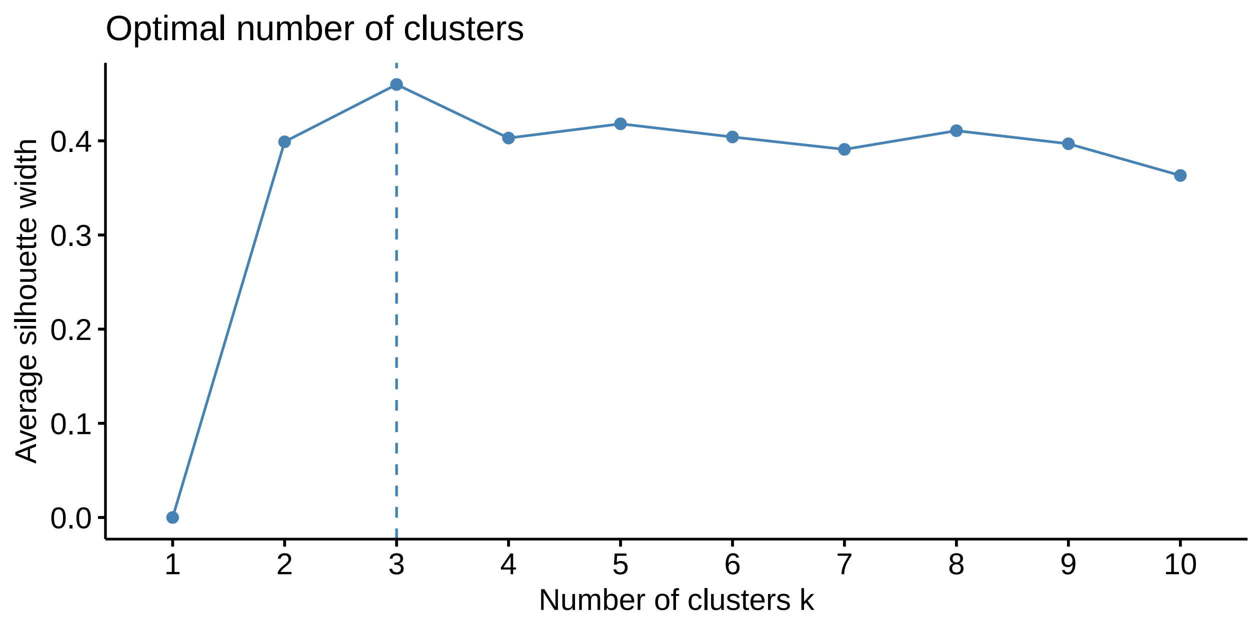
```r
# See the number because the plot is not clear if k=2 or k=3 is larger
kable(as.data.frame(gap_stat$Tab) %>%
        mutate(k = 1:10) %>%
        select(k, gap) %>%
        head(n = 5))
```

k	gap
1	0.7493151
2	0.6731725
3	0.6733750
4	0.5931839
5	0.5815528

The maximum gap statistic is k = 1, and the second gap statistic is k = 3 (a little bit larger than k = 2). Thus, the optimal number of clusters is k = 3

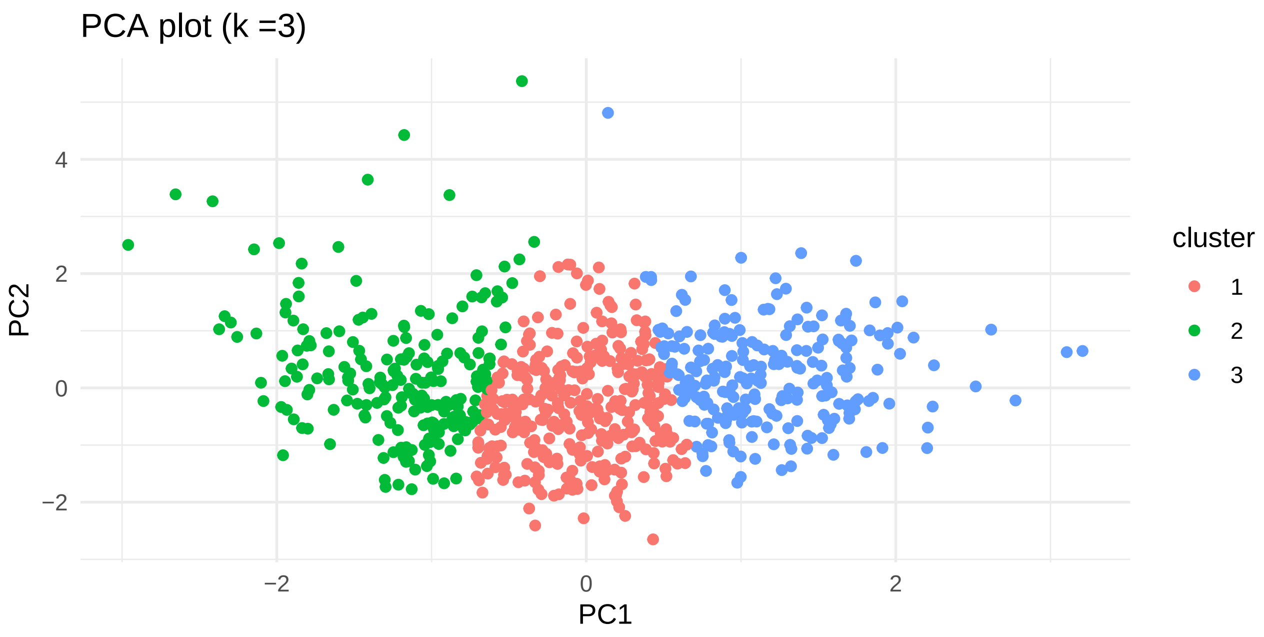
Question 11

```r
set.seed(0) # Ensure the reproducibility

# identify the optimal number of clusters in case of the first and second dimensions from t-SNE
wiki_tsne <- Rtsne(as.matrix(wiki), perplexity = 25)
tsne_scale <- data.frame(dimension1 = wiki_tsne$Y[,1], dimension2 = wiki_tsne$Y[,2]) %>%
  mutate_at(.vars = vars(dimension1, dimension2), scale)

# Use the average silhouette
fviz_nbclust(tsne_scale, kmeans, method = "silhouette")
```

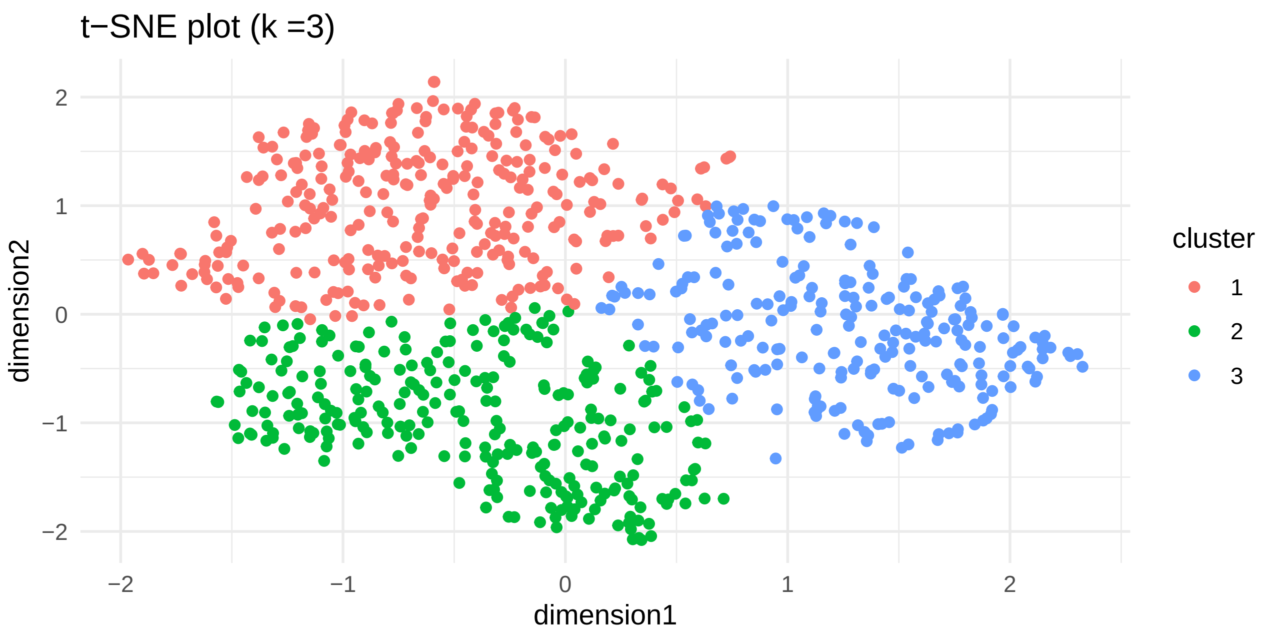


As the graph above clearly shows, the optimal number of clusters is also k = 3 even in case of using the first and second dimensions from t-SNE.

```r
# PCA plot when k = 3
PC_scale %>%
  mutate(cluster = factor(kmeans(PC, 3, nstart = 25)$cluster)) %>%
  ggplot(aes(x = PC1, y = PC2, color = cluster)) +
  geom_point() +
  labs(title = "PCA plot (k =3)")
```



```
# t-SNE plot when k =3
tsne_scale %>%
  mutate(cluster = factor(kmeans(tsne_scale, 3, nstart = 25)$cluster)) %>%
  ggplot(aes(x = dimension1, y = dimension2, color = cluster)) +
  geom_point() +
  labs(title = "t-SNE plot (k =3)")
```



When using use the first and second principal components from PCA, we are not sure that we can clearly separate the observations by the three clusters based on the plot above. When using the first and second dimensions from t-SNE (perplexity = 25), it seems clearer that we can separate it by three clusters based on the plot above. But as my answer in Question 8, it depends on the perplexity. (In case of the perplexity = 15, it seems unclear about the clusters)