Week 10 K-Means

#Preamble Just some packages

```
library(dplyr)       # for data manipulation
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)     # for data visualization
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(stringr)     # for string functionality
library(gridExtra)   # for manipulaiting the grid
```

```
## Warning: package 'gridExtra' was built under R version 4.1.3

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(tidyverse)  # data manipulation
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v tibble  3.1.6     v purrr   0.3.4
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x gridExtra::combine() masks dplyr::combine()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
```

```
library(cluster)     # for general clustering algorithms
library(factoextra)  # for visualizing cluster results
```

```
## Warning: package 'factoextra' was built under R version 4.1.3

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

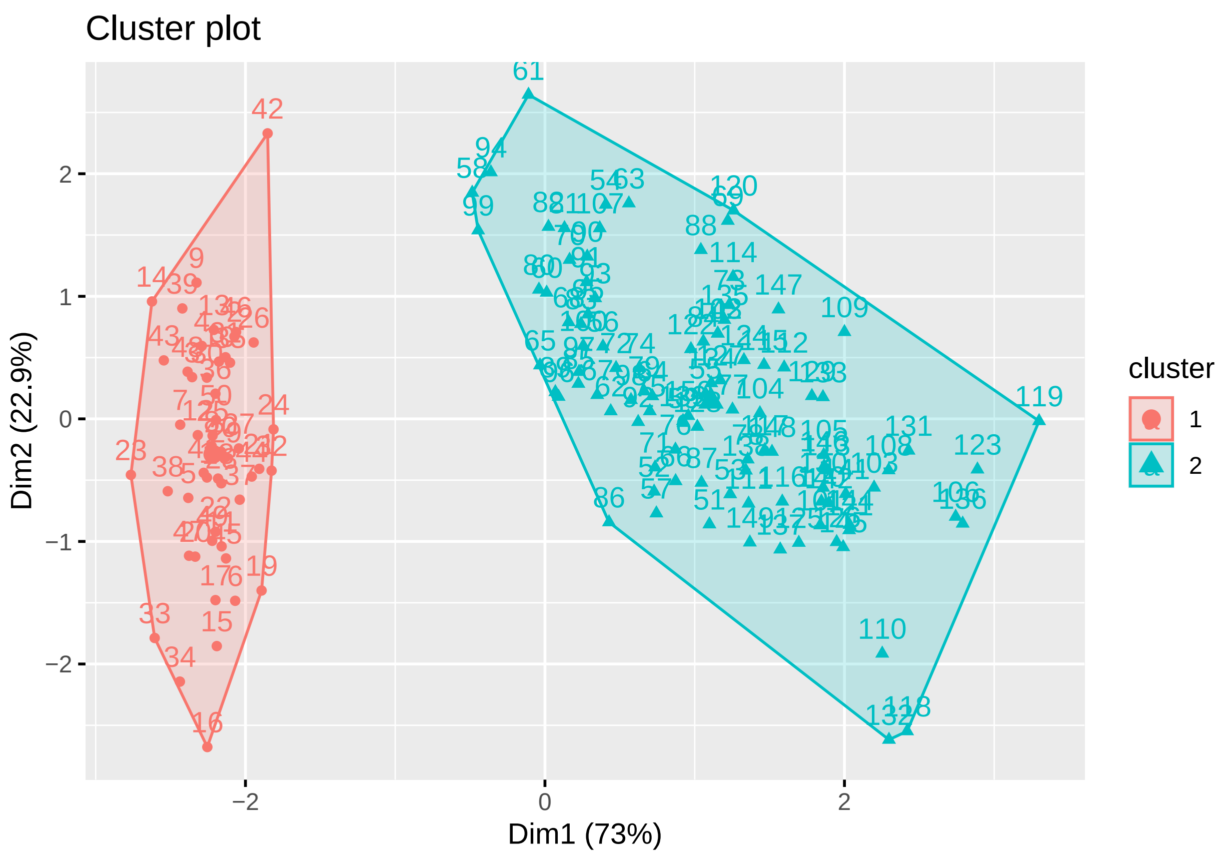
#Loading and preprocessing data

```r
data("iris")

#To remove any missing value that might be present in the data, type this:
df <- na.omit(iris)

#we start by scaling/standardizing the data
df <- scale(df[c(1:4)])
head(df)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1   -0.8976739  1.01560199    -1.335752   -1.311052
## 2   -1.1392005 -0.13153881    -1.335752   -1.311052
## 3   -1.3807271  0.32731751    -1.392399   -1.311052
## 4   -1.5014904  0.09788935    -1.279104   -1.311052
## 5   -1.0184372  1.24503015    -1.335752   -1.311052
## 6   -0.5353840  1.93331463    -1.165809   -1.048667
```

#Basic K-means Application Simple application with two centers

```r
#start at 2 clusters
k2 <- kmeans(df, centers = 2, nstart = 25)
#plot the 2 clusters
fviz_cluster(k2, data = df)
```



#get the each clsuter's data df %>% as_tibble() %>% mutate(cluster = k2$cluster, Species = row.names(iris)) %>% ggplot(aes(Sepal.Length, Sepal.Width, color = factor(cluster), label = Species)) + geom_text()

k3 <- kmeans(df, centers = 3, nstart = 25) k4 <- kmeans(df, centers = 4, nstart = 25) k5 <- kmeans(df, centers = 5, nstart = 25)

plots to compare

p1 <- fviz_cluster(k2, geom = “point”, data = df) + ggtitle(“k = 2”) p2 <- fviz_cluster(k3, geom = “point”, data = df) + ggtitle(“k = 3”) p3 <- fviz_cluster(k4, geom = “point”, data = df) + ggtitle(“k = 4”) p4 <- fviz_cluster(k5, geom = “point”, data = df) + ggtitle(“k = 5”)

grid.arrange(p1, p2, p3, p4, nrow = 2)

#Determining Optimal Number of Clusters set.seed(123)

#function to compute total within-cluster sum of square wss <- function(k) { kmeans(df, k, nstart = 10)$tot.withinss }

Compute and plot wss for k = 1 to k = 15

k.values <- 1:15

extract wss for 2-15 clusters

wss_values <- map_dbl(k.values, wss)

plot(k.values, wss_values, type=“b”, pch = 19, frame = FALSE, xlab=“Number of clusters K”, ylab=“Total within-clusters sum of squares”)

#or use this fviz_nbclust(df, kmeans, method = “silhouette”)

compute gap statistic

set.seed(123) gap_stat <- clusGap(df, FUN = kmeans, nstart = 25, K.max = 10, B = 50) # Print the result print(gap_stat, method = “firstmax”)

fviz_gap_stat(gap_stat)

Compute k-means clustering with k = 2

set.seed(123) final <- kmeans(df, 2, nstart = 25) print(final)

#final data fviz_cluster(final, data = df)