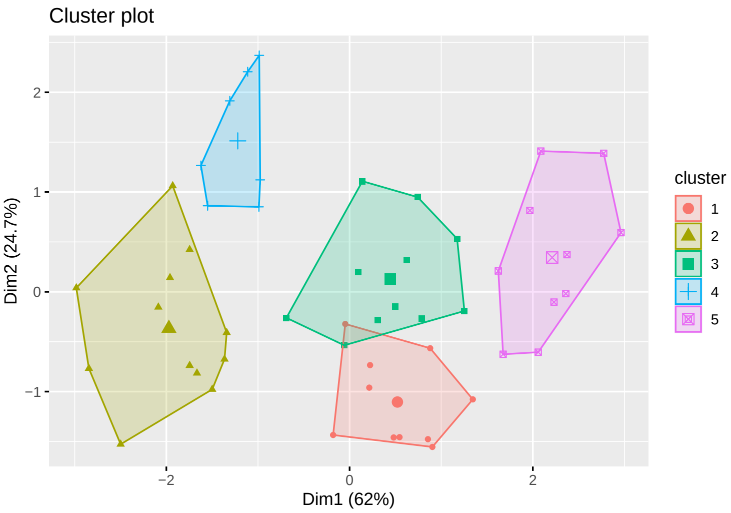
Dimensionality Reduction

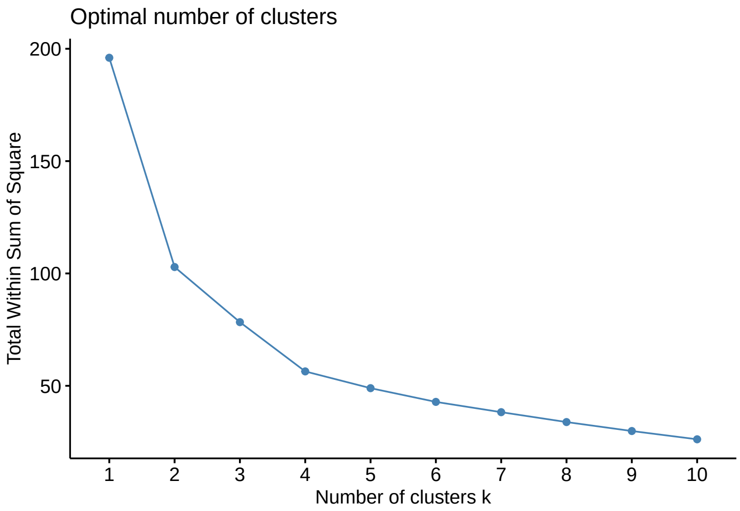
Exercise 01

Consider the code below and answer the following questions.

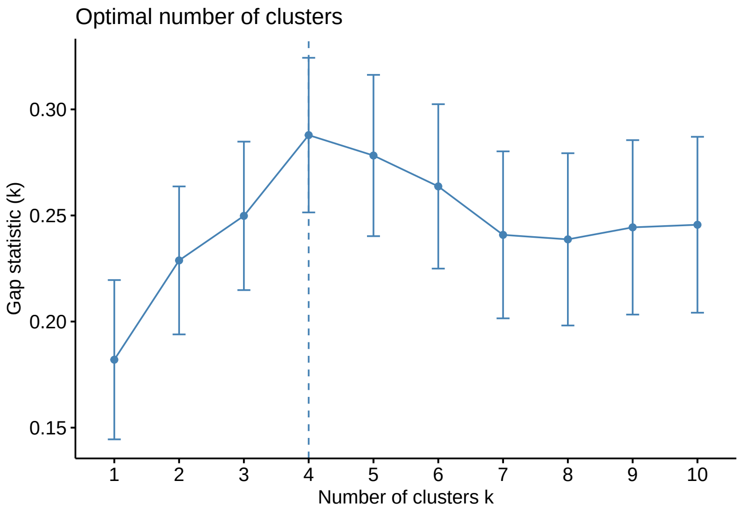
```r
library(tidyverse)
library(cluster)
library(factoextra)

set.seed(1)

# Load the data
data("USArrests")
df <- as_tibble(USArrests)

# Center and scale the data
# A recipe will not be necessary this time
df_sc <- scale(df) %>% as_tibble()

# Perform the clustering analysis with K = 5
km_cluster <- kmeans(df_sc, center = 5, nstart = 50)

# Plot the results using two Principal Components from PCA
fviz_cluster(km_cluster, geom = "point", data = df)
```



```r
# Create a plot of the Within-Cluster Sum of Squares (WSS) vs. k
# with k ranging from 1 to k.max = 10
fviz_nbclust(df_sc, kmeans, method = "wss", k.max = 10, nstart = 50)
```



```r
# Create a plot of the Gap Statistic vs. k
# with k ranging from 1 até a k.max = 10.
# nboot is the number of bootstrap samples created.
fviz_nbclust(df_sc, kmeans, method = "gap", k.max = 10, nstart = 50, nboot = 200)
```



a) In the code above, we set **nstart = 50**. Read the documentation for the **kmeans** function and explain why we typically assign a large integer value to this parameter, usually around 20 to 50.

b) Based on the cumulative Within-Cluster Sum of Squares (WSS) plot, choose the best value for k.

c) Using Gap Statistics, what would be the optimal number of clusters?

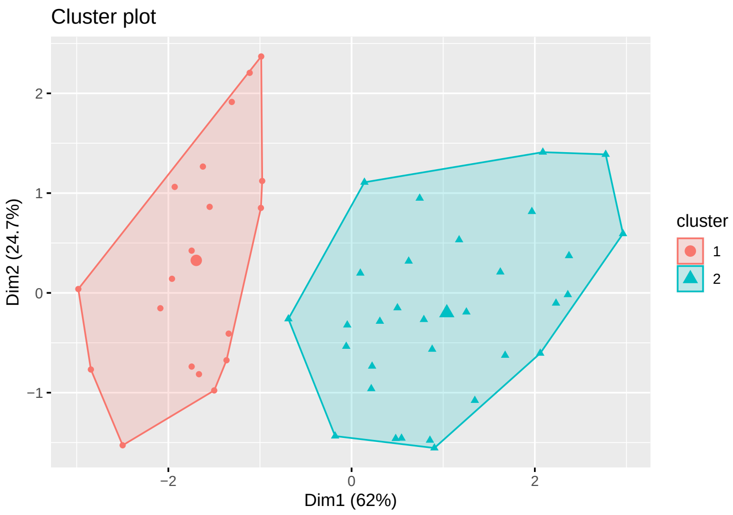
d) Repeat items b) and c) above for hierarchical clustering using complete linkage. Use the **hcut** function instead of **kmeans** to generate the clusters. Also, pass **hc_method = "complete"** as an argument instead of the **nstart** parameter, which you no longer need to specify.

a) According to the documentation, the **nstart** parameter specifies the number of random samples that the algorithm should take. This parameter should be set to a large integer value because the model may converge to different solutions due to its random initialization. Therefore, it is recommended to run the algorithm 20 to 50 times with different random initializations and select the arrangement with the lowest value for the objective function.

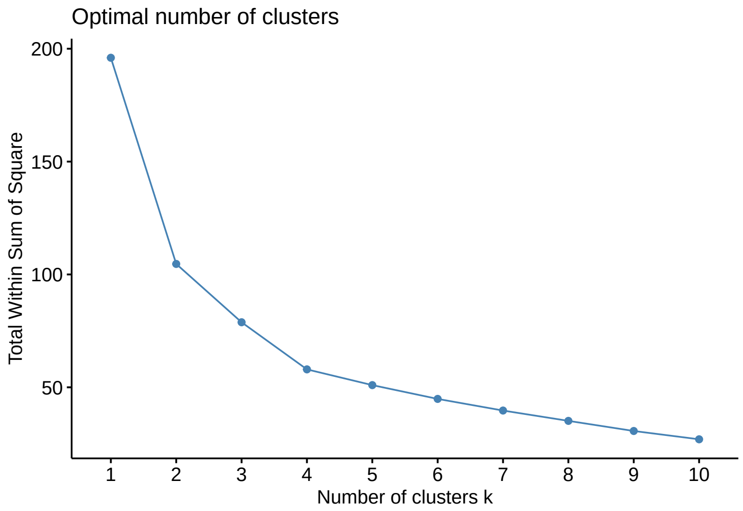
b) According to the cumulative WSS plot, the optimal number of clusters would be k = 4. This is because, based on the "elbow method," the decline in WSS becomes less steep from this point onwards.

c) According to the Gap Statistics plot, the optimal number of clusters would be k = 4, as it has the highest associated Gap value. In other words, with k = 4, the clustering structure is further away from the uniform distribution of points.

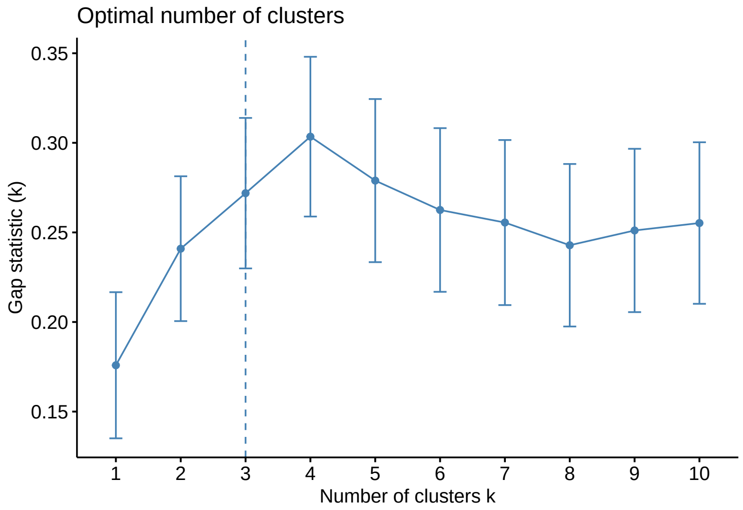
```
set.seed(1)

# Load the data
data("USArrests")
df <- as_tibble(USArrests)

# Scale the data
df_sc <- scale(df) %>% as_tibble()

# Perform clustering analysis with K = 5
hr_cluster <- hcut(df_sc, hc_method = "complete")

# Plot the results using two Principal Components from PCA
fviz_cluster(hr_cluster, geom = "point", data = df)
```



```
# Plot of the cumulative WSS
fviz_nbclust(df_sc, hcut, method = "wss", k.max = 10, nstart = 50)
```



```r
# Plot of the Gap Statistic
fviz_nbclust(df_sc, hcut, method = "gap", k.max = 10, nstart = 50, nboot = 200)
```



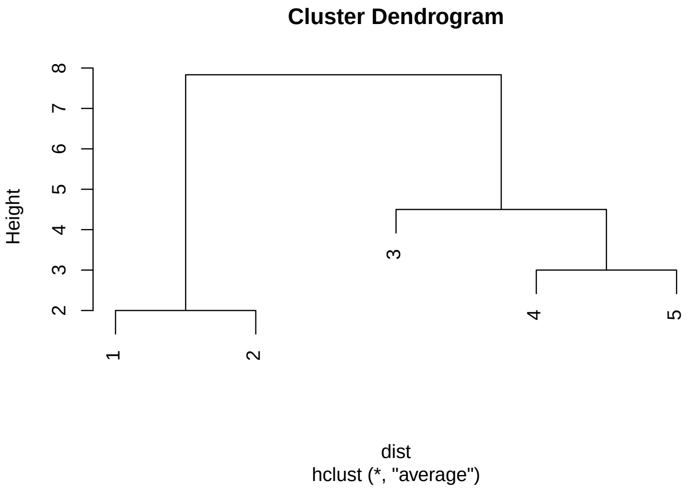
d) As with the k-means algorithm, the optimal number of clusters is k = 4 according to the two graphs above.

Exercise 02

Run the hierarchical clustering algorithm on the distance matrix below using the hclust(dist, method) function with method = 'single'. Visualize the resulting plot by passing the object returned by hclust to the plot function.

$$D = \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix}$$

```r
library(Matrix)
dist <- matrix( c(0,2,6,10,9,
                  0,0,5,9,8,
                  0,0,0,4,5,
                  0,0,0,0,3,
                  0,0,0,0,0), ncol = 5)
dist <- dist + t(dist)
dist <- as.dist(dist)


hclust_avg <- hclust(dist, method = 'average')
plot(hclust_avg)
```



Exercise 03

Run the DBSCAN algorithm on the distance matrix below. Comment on the results and indicate which data points are classified as core, border, and noise points.

$$D = \begin{bmatrix} 0 & & & & & & \\ 1 & 0 & & & & & \\ 1 & 2 & 0 & & & & \\ 10 & 9 & 11 & 0 & & & \\ 11 & 10 & 12 & 2 & 0 & & \\ 21 & 20 & 22 & 18 & 19 & 0 & \\ 14 & 13 & 15 & 6 & 4 & 25 & 0 \end{bmatrix}$$

```r
library(fpc)

# Set up the distance matrix
lower_tri <- c(1, 1, 10, 11, 21, 14, 2, 9, 10, 20, 13,
11, 12, 22, 15, 2, 18, 6, 19, 4, 25, 0)
mat <- matrix(0, nrow = 7, ncol = 7)
mat[lower.tri(mat)] <- lower_tri
```

```
mat <- mat + t(mat) - diag(diag(mat))

# Run the DBSCAN algorithm
dbc <- dbscan(mat, eps = 5, MinPts = 3, method = 'dist')
dbc$cluster
```

```
## [1] 1 1 1 2 2 0 2
```

```
dbc
```

```
## dbscan Pts=7 MinPts=3 eps=5
##        0 1 2
## border 1 0 2
## seed   0 3 1
## total  1 3 3
```

As the result suggests, points 1, 2, and 3 belong to cluster 1; points 4, 5, and 7 belong to cluster 2, and point 6 is classified as an outlier. Cluster 1 consists of 3 core points (1, 2, and 3), and cluster 2 consists of one core point (5) and two border points (4 and 7).

Exercise 04

In this exercise, we illustrate the application of the PCA (Principal Component Analysis) method for dimensionality reduction on the mtcars dataset, which describes characteristics of different car models. Execute the following code and answer the questions below:

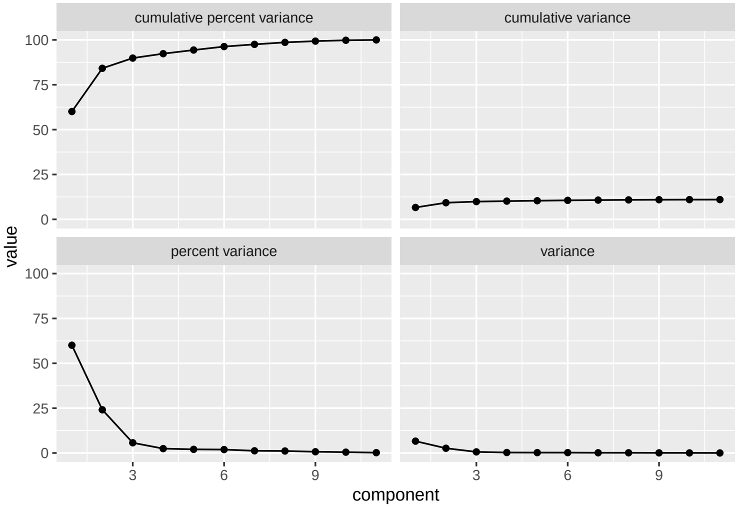
a) Using the "elbow" method, how many principal components would you choose? What percentage of data variance do they carry?

b) In the graph that represents the original variables in terms of the two principal components, which two variables contribute most to the first principal component? Why are they pointing to opposite directions?

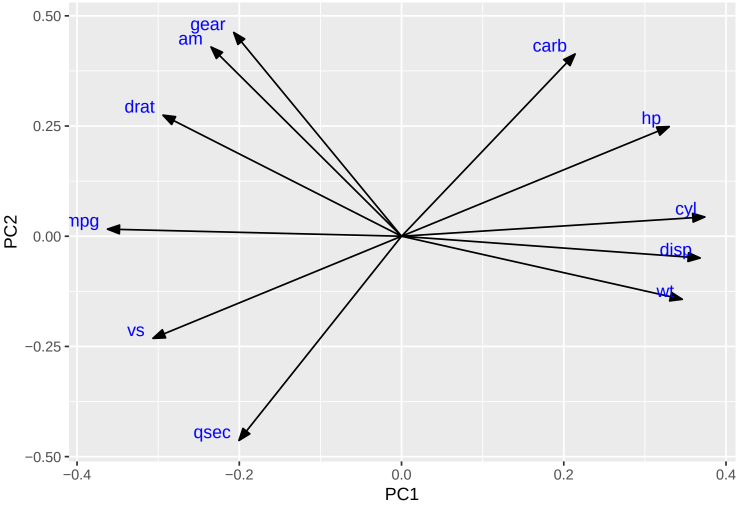
c) After analyzing the code below, measure the effect of "hp" on the two principal components. In other words, provide the exact value of the vector $V4 = (\phi_{41}, \phi_{42})^T$, where j = 4 represents the "hp" feature, which is the fourth variable in the table.

```
library(tidymodels)
# Load the data
df <- as_tibble(mtcars,rownames="model")

# Recipe for dimensionality reduction
pca_rec <- recipe(df) %>%
step_center(all_numeric()) %>%
step_scale(all_numeric()) %>%
step_pca(all_numeric(),num_comp = 10, id = "pca") %>%
prep()

# Retrieving the variance for each principal component
var_info <- tidy(pca_rec,id="pca",type="variance")
```

```r
# Plot the explained variance
var_info %>%
ggplot(aes(x = component, y = value )) +
geom_path() +
geom_point() +
ylim(c(0,100))+
facet_wrap(~terms)
```



```r
# Plot representing each original component in terms
# of the first two principal components
coefs_info <- tidy(pca_rec,id="pca",type="coef")
coefs_info %>%
pivot_wider(names_from = component, values_from=value) %>%
ggplot(aes( x = PC1, y = PC2)) +
geom_segment( xend = 0, yend = 0, arrow = arrow(angle = 20,
ends = "first",
type = "closed",
length = grid::unit(8, "pt"))) +
geom_text(aes(label = terms), hjust = 1, nudge_x = -0.01,
nudge_y = +0.02, color = "blue")
```



```
coefs_info
```

```
## # A tibble: 121 x 4
##    terms  value component id   
##    <chr>  <dbl> <chr>     <chr>
##  1 mpg   -0.363 PC1       pca  
##  2 cyl    0.374 PC1       pca  
##  3 disp   0.368 PC1       pca  
##  4 hp     0.330 PC1       pca  
##  5 drat  -0.294 PC1       pca  
##  6 wt     0.346 PC1       pca  
##  7 qsec  -0.200 PC1       pca  
##  8 vs    -0.307 PC1       pca  
##  9 am    -0.235 PC1       pca  
## 10 gear  -0.207 PC1       pca  
## # i 111 more rows
```

a) The optimal number of principal components is 3. They account for approximately 6.25% of the variance in the data.

b) The first principal component has very high loadings for the mpg (-0.362530504) and cyl (0.373916027) components. Their respective vectors are pointing to opposite directions because, although both features have a strong influence over PC1, they are negatively correlated.

c) V4 = (0.330056925, 0.248784020).