Exemple d’application de l’algorithme K-means en R

Introduction

L’algorithme K-means est une méthode de classification non supervisée appartenant au domaine de l’apprentissage automatique. Il permet de regrouper des observations similaires en un nombre prédéfini de groupes appelés clusters. Dans le cadre de cette étude, nous appliquons l’algorithme K-means sur le jeu de données « iris », composé de 150 fleurs réparties équitablement en trois espèces. L’idée est de supposer que nous ne connaissons pas à l’avance les catégories réelles de ces iris, et de laisser l’algorithme regrouper les individus uniquement à partir de leurs caractéristiques mesurées : longueur et largeur des sépales et des pétales.

Objectifs

- Implémenter l’algorithme de classification K-means
- Visualiser les groupes formés par l’algorithme ainsi que les groupes réels présents dans les données
- Interpréter les clusters obtenus et les associer aux espèces d’iris correspondantes
- Comparer les résultats du clustering avec et sans réduction dimensionnelle par ACP

Mise en oeuvre de l’algorithme K-means

Dans un premier temps, nous utilisons l’algorithme K-means en ne prenant en compte que les variables les plus contributives à l’axe principal d’une analyse en composantes principales (ACP). Cela suppose d’identifier préalablement l’axe principal, puis les variables qui y contribuent le plus. Le jeu de données iris comprend les variables suivantes : Sepal.Length, Sepal.Width, Petal.Length et Petal.Width. Une ACP est donc réalisée afin d’orienter le choix des variables pertinentes.

```
library(FactoMineR)
```

```
## Warning: le package 'FactoMineR' a été compilé avec la version R 4.4.3
```

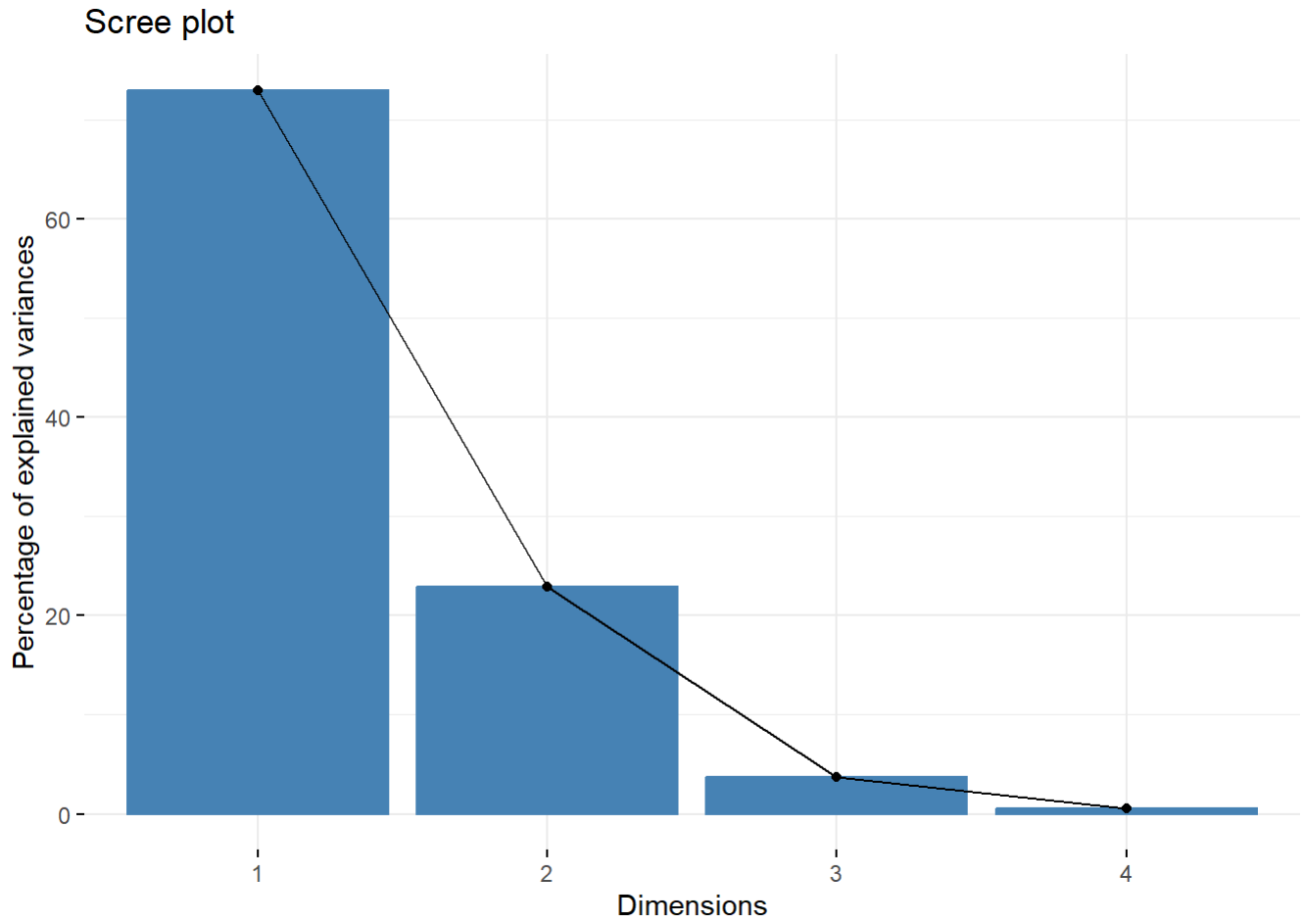
```
res.pca=PCA(iris[,1:4], graph = FALSE)

library(factoextra)
```

```
## Warning: le package 'factoextra' a été compilé avec la version R 4.4.3
```

```
## Le chargement a nécessité le package : ggplot2
```

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

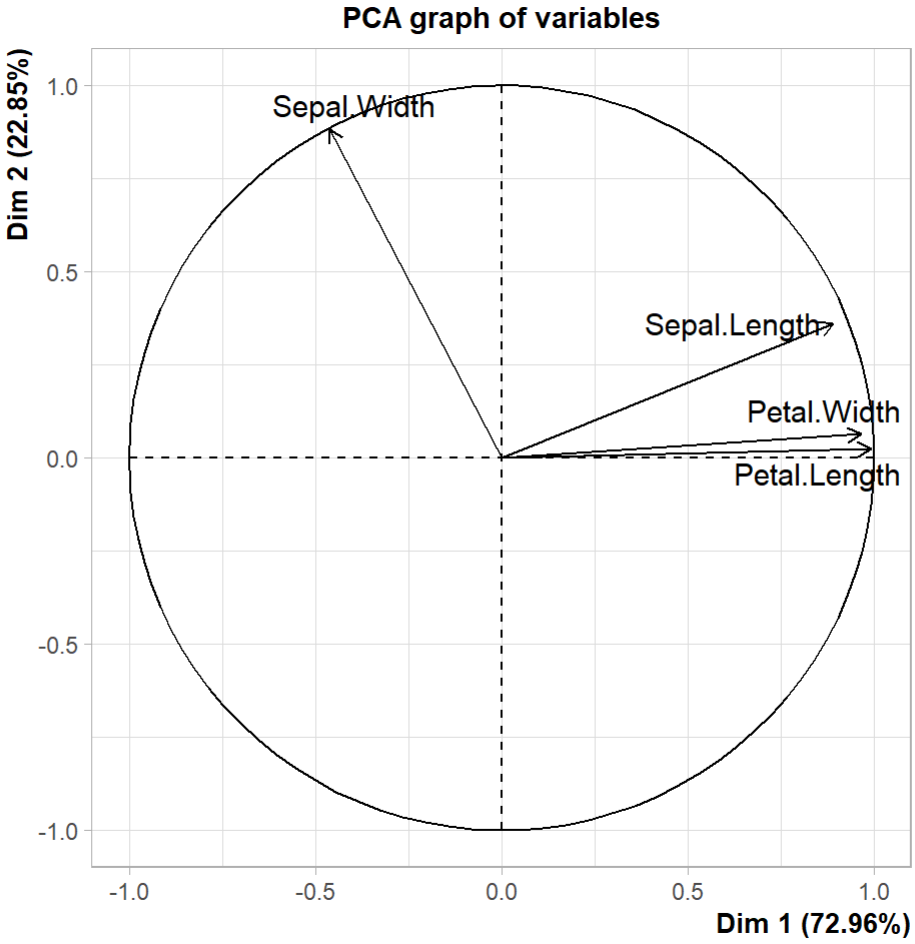
```
fviz_eig(res.pca)
```




```
res.pca$eig
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1 2.91849782             72.9624454                          72.96245
## comp 2 0.91403047             22.8507618                          95.81321
## comp 3 0.14675688              3.6689219                          99.48213
## comp 4 0.02071484              0.5178709                         100.00000
```

```
plot(res.pca,choix="var")
```



La première composante principale (axe 1) explique à elle seule environ 73 % de la variance totale, ce qui en fait l’axe le plus informatif. L’analyse graphique des contributions des variables montre que Petal.Length et Petal.Width sont les deux variables qui influencent le plus cet axe. Par conséquent, ces deux variables seront retenues pour exécuter l’algorithme K-means comme suit :

```
iriscluster1=kmeans(iris[,3:4],centers=3,nstart=100,iter.max=50)

iriscluster1
```

```
## K-means clustering with 3 clusters of sizes 50, 52, 48
##
## Cluster means:
##   Petal.Length Petal.Width
## 1     1.462000    0.246000
## 2     4.269231    1.342308
## 3     5.595833    2.037500
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 2 3 3 3 3
## [112] 3 3 3 3 3 3 3 3 2 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3
## [149] 3 3
##
## Within cluster sum of squares by cluster:
## [1]  2.02200 13.05769 16.29167
##  (between_SS / total_SS =  94.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
#Matrice de confusion
mc1<-table(iriscluster1$cluster,iris$Species)

mc1
```

```
##
##     setosa versicolor virginica
##   1     50          0         0
##   2      0         48         4
##   3      0          2        46
```

```
#erreur de prédiction
er1=1-sum(diag(mc1))/sum(mc1)
er1
```

```
## [1] 0.04
```

La matrice de confusion ci-dessus compare les groupes détectés par K-means aux catégories réelles des iris. Cela permet de mesurer la pertinence du regroupement effectué.

Les lignes représentent les clusters attribués par l'algorithme k-means. Dans notre cas, il y a trois clusters numérotés 1, 2, et 3 conformément au souhait formulé précedemment par "centers=3" groupe au total.

Cluster 1 : 50 iris setosa, 0 iris versicolor, et 0 iris virginica ont été classés dans ce cluster. Ce cluster a donc parfaitement identifié les iris de type setosa.

Cluster 2 : 0 iris setosa, 48 iris versicolor, et 4 iris virginica ont été classés dans ce cluster. Cela suggère que ce cluster représente principalement des iris de type versicolor.

Cluster 3 : 0 iris setosa, 2 iris versicolor, et 46 iris virginica ont été classés dans ce cluster. Cela suggère que ce cluster représente principalement des iris de type virginica.

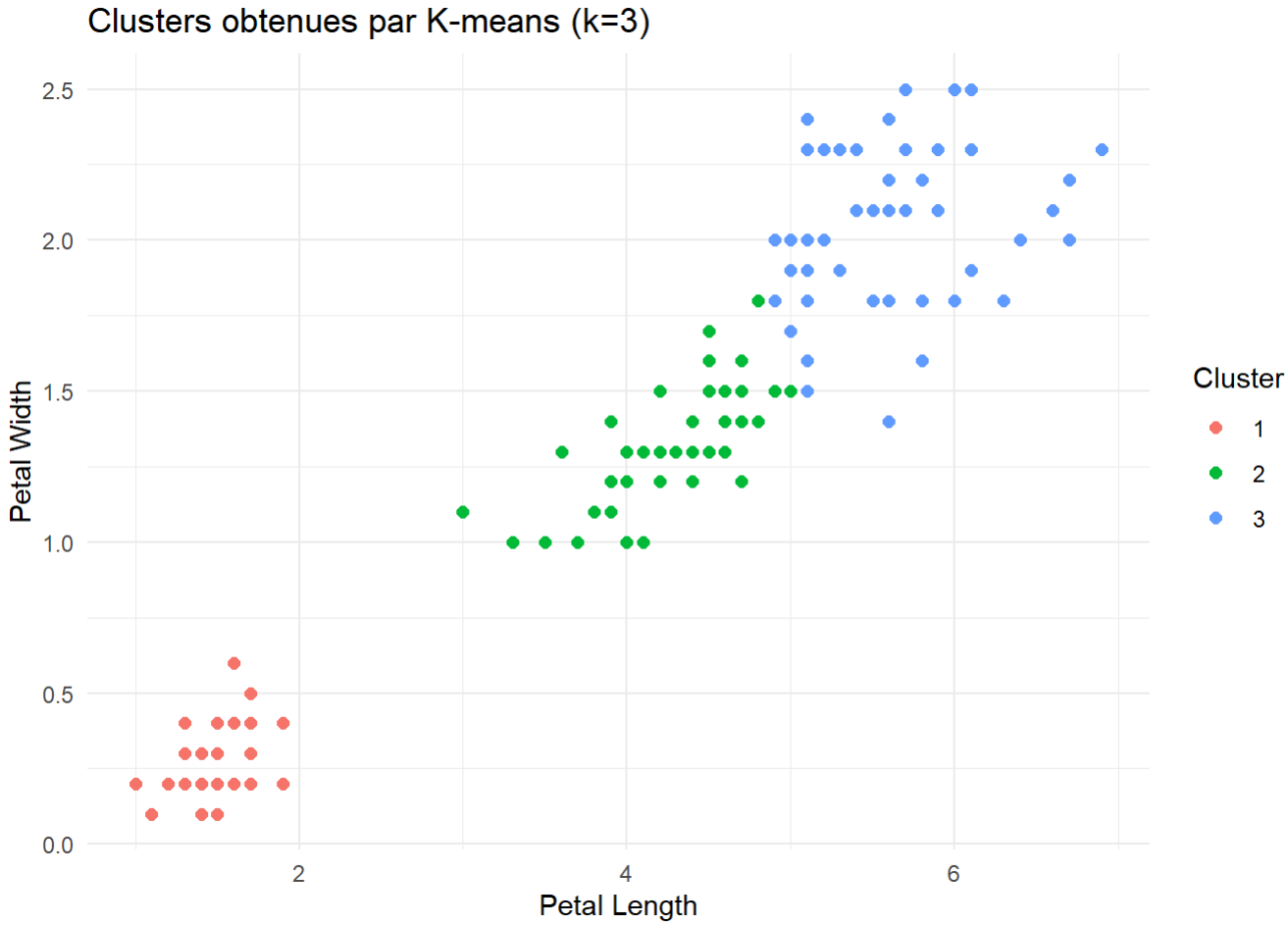
Visualisation des résultats de K-means et comparaison avec les espèces réelles

Pour visualiser les résultats, les numéros de clusters sont ajoutés au jeu de données iris. Deux graphes sont ensuite tracés :

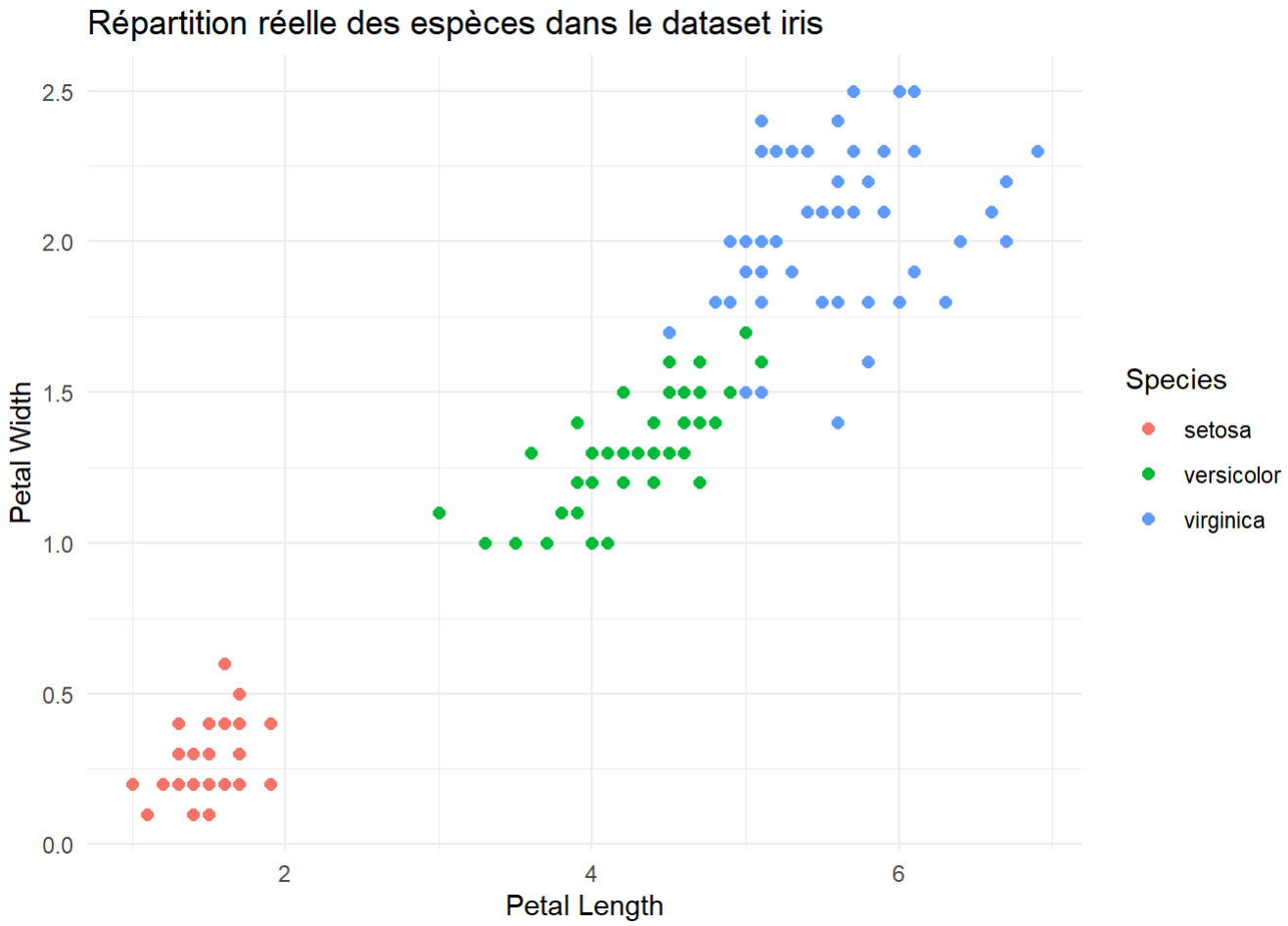
Un scatter plot montrant les clusters obtenus par K-means

Un scatter plot montrant la répartition réelle des espèces

```
# Ajoute les clusters aux données
iris$cluster1 <- iriscluster1$cluster

# scatter plot avec ggplot2
library(ggplot2)
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = as.factor(cluster1))) +
  geom_point(size = 2) +
  labs(title = "Clusters obtenues par K-means (k=3)",
       x = "Petal Length",
       y = "Petal Width",
       color = "Cluster") +
  theme_minimal()
```



```
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +
  geom_point(size = 2) +
  labs(title = "Répartition réelle des espèces dans le dataset iris",
       x = "Petal Length",
       y = "Petal Width",
       color = "Species") +
  theme_minimal()
```



Ces visualisations révèlent une bonne concordance entre les groupes formés par l'algorithme et les catégories naturelles du jeu de données. Et à l'instar de la matrice de confusion précédente, on peut également savoir à partir des deux visualisations précédentes à quelles espèces correspondent chaque cluster (très facile). Une autre méthode pour attribuer les clusters k-means aux espèces correspondantes est présentées ci-dessous.

Autre Méthode pour attribution les clusters k-means aux espèces correspondantes

Pour relier les clusters obtenus aux espèces réelles, nous analysons les centres de chaque cluster et les comparons aux moyennes des variables Petal.Length et Petal.Width pour chaque espèce.

Les centres des clusters obtenus par K-means sont :

```
iriscluster1$centers
```

```
##   Petal.Length Petal.Width
## 1     1.462000    0.246000
## 2     4.269231    1.342308
## 3     5.595833    2.037500
```

Et les moyennes par espèce sont calculées via tapply().

```
tapply(iris$Petal.Length, iris$Species, mean)
```

```
##     setosa versicolor  virginica
##      1.462      4.260      5.552
```

```
tapply(iris$Petal.Width, iris$Species,mean)
```

```
##     setosa versicolor  virginica
##      0.246      1.326      2.026
```

La correspondance est alors déduite :

- Le cluster ayant un centre proche de (1.46, 0.25) correspond à setosa
- Celui autour de (4.26, 1.34) est associé à versicolor
- Le cluster près de (5.60, 2.03) correspond à virginica

Application du K-means sans réduction dimensionnelle

Dans cette deuxième approche, l'algorithme K-means est appliqué en utilisant directement toutes les variables disponibles (Sepal.Width,Sepal.Length, Petal.Length, Petal.Width), sans passer par l'ACP.

```
iriscluster2=kmeans(iris[,2:4],3,nstart=100,iter.max=50)
iriscluster2
```

```
## K-means clustering with 3 clusters of sizes 53, 50, 47
##
## Cluster means:
##   Sepal.Width Petal.Length Petal.Width
## 1    2.754717     4.281132    1.350943
## 2    3.428000     1.462000    0.246000
## 3    3.004255     5.610638    2.042553
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 1 3 3 3 3
## [112] 3 3 3 3 3 3 3 3 1 3 3 3 1 3 3 1 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3
## [149] 3 3
##
## Within cluster sum of squares by cluster:
## [1] 18.86491  9.06280 19.93872
##  (between_SS / total_SS =  91.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

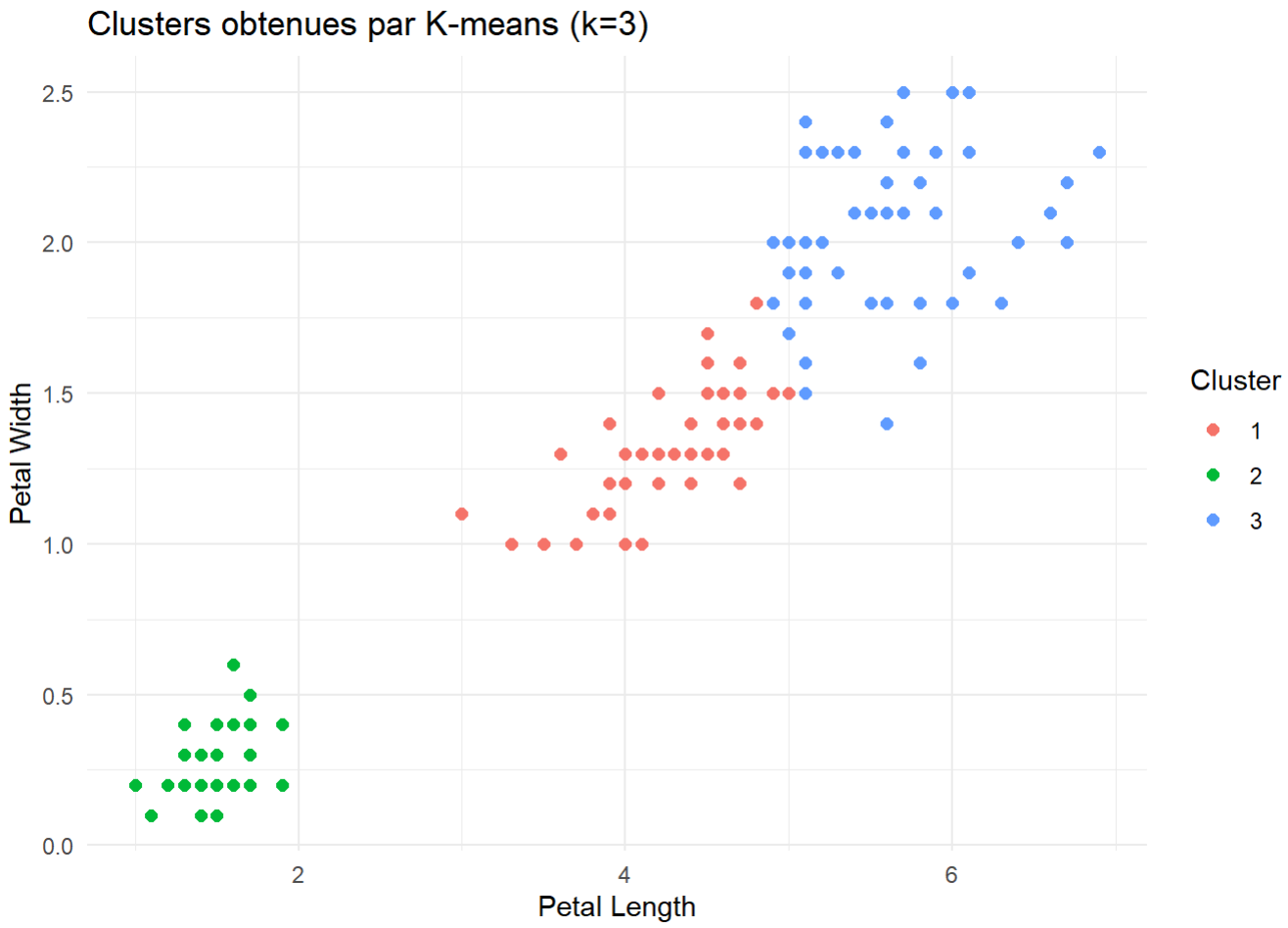
```
#matrice de confusion
mc2<-table(iriscluster2$cluster,iris$Species)

mc2
```

```
##
##     setosa versicolor virginica
##   1      0         48         5
##   2     50          0         0
##   3      0          2        45
```

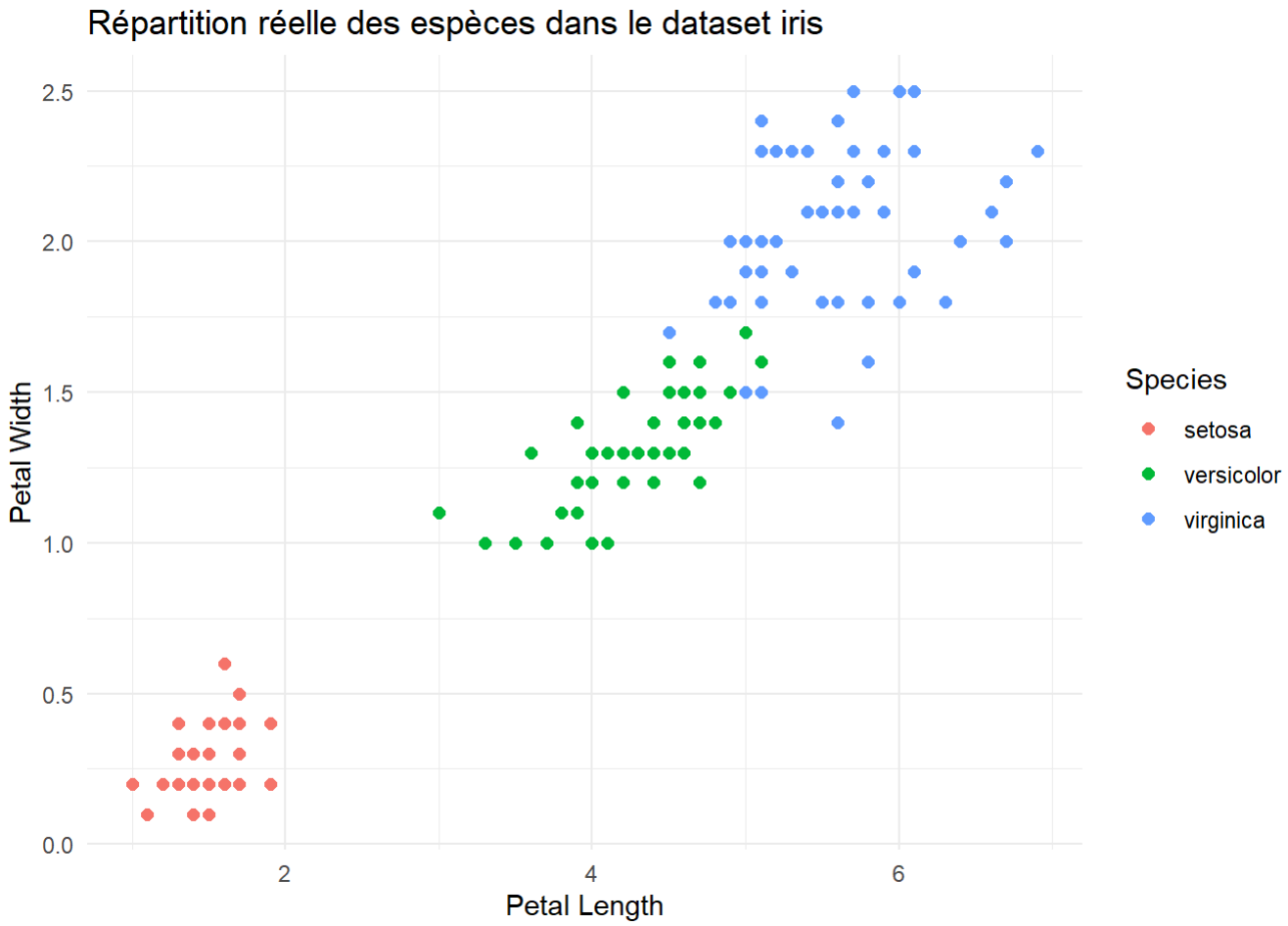
```
#erreur de prédiction
er2=1-sum(diag(mc2))/sum(mc2)
er2
```

```
## [1] 0.7
```

```
# Ajoute les clusters aux données
iris$cluster2 <- iriscluster2$cluster

# scatter plot avec ggplot2
library(ggplot2)
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = as.factor(cluster2))) +
  geom_point(size = 2) +
  labs(title = "Clusters obtenues par K-means (k=3)",
       x = "Petal Length",
       y = "Petal Width",
       color = "Cluster") +
  theme_minimal()
```



```
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +
  geom_point(size = 2) +
  labs(title = "Répartition réelle des espèces dans le dataset iris",
       x = "Petal Length",
       y = "Petal Width",
       color = "Species") +
  theme_minimal()
```



On constate ici que l'algorithme a attribué des étiquettes de clusters (1, 2, 3) à des espèces différentes de celles obtenues lorsqu'une Analyse en Composantes Principales (ACP) avait été réalisée au préalable.

Dans cette nouvelle analyse sans ACP, l'algorithme assigne :

- le cluster 1 à l'espèce versicolor,
- le cluster 2 à l'espèce setosa,
- et le cluster 3 à l'espèce virginica.

Cela contraste avec l'analyse avec ACP, où :

- le cluster 1 correspond à setosa,
- le cluster 2 à versicolor,
- et le cluster 3 à virginica.

Cependant, les visualisations et les sorties des vecteurs de clustering (dans iriscluster1 et iriscluster2) montrent clairement que les groupes de données associés à chaque espèce sont identiques dans les deux cas. Ce n'est que le numéro des clusters qui change d'un cas à l'autre.

En effet, dans l’analyse avec ACP, les 50 premières observations sont associées au cluster 1, qui représente bien l’espèce setosa (cela se confirme aussi via la matrice de confusion). Dans l’analyse sans ACP, ces mêmes 50 premières observations sont associées au cluster 2, qui correspond également à l’espèce setosa.

Le même phénomène est observé pour les deux autres espèces. Ainsi, malgré des numéros de clusters différents, l’algorithme regroupe bien les mêmes espèces dans les deux approches. L’attribution reste cohérente avec la répartition réelle des espèces dans les données, avec un taux d’erreur de prédiction très faible dans la méthode avec ACP.

En conclusion, les deux méthodes (avec ou sans ACP) produisent des résultats similaires avec l’algorithme K-means, en termes de qualité de clustering. On note toutefois une erreur de prédiction plus élevé dans l’approche sans ACP.