TP ACP corrigé

PC & YM

28/07/2021

Packages

Plusieurs packages sont disponibles pour réaliser des ACP avec R. Dans ce TP nous utiliserons les packages `ade4` qui réalise les calculs et `factoextra` qui fournit des outils pour visualiser les résultats.

```
install.packages("ade4")
install.packages("factoextra")
```

Nous utiliserons aussi un package pour les données : `palmerpenguins`, de Horst AM, Hill AP, Gorman KB (2020) , et qui fournit des mesures de la morphologie de trois espèces de pengouins:

```
install.packages("palmerpenguins")
```

Une fois installés, on charge ces packages ainsi que d'autres déjà connus : `dplyr`, `ggplot2`

```
library(dplyr)
library(ggplot2)
library(ade4)
library(factoextra)
library(palmerpenguins)
```

Données

les données sont décrites sur la page github du package : https://allisonhorst.github.io/palmerpenguins/

Nous utiliserons l'objet `penguins` fourni par ce package , dont voici les 6 premières lignes:

```
data("penguins")
head(penguins)
```

```
## # A tibble: 6 x 8
##   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##   <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
## 1 Adelie  Torgersen           39.1          18.7               181        3750
## 2 Adelie  Torgersen           39.5          17.4               186        3800
## 3 Adelie  Torgersen           40.3          18                 195        3250
## 4 Adelie  Torgersen           NA            NA                  NA          NA
## 5 Adelie  Torgersen           36.7          19.3               193        3450
## 6 Adelie  Torgersen           39.3          20.6               190        3650
## # i 2 more variables: sex <fct>, year <int>
```

Question 1 : Préparation des données

Filtrer les observations non attribuées, de façon à ce qu'il n'y ait plus de valeurs `NA` dans le dataframe.

Indice : Utiliser les fonctions `anyNA` et `na.omit`

```r
# vérifie la présence dee valeurs NA dans les données
anyNA(penguins)
```

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

`anyNA` permet de vérifier s'il y a des valeurs NA dans les données, `na.omit` renvoie un jeu de données sans les lignes qui contiennent au moins une valeur NA.

```r
# vérifie la présence dee valeurs NA dans les données
penguins_noNA <- na.omit(penguins)
```

Question 2 : Affichage des données

Commencer par afficher les noms et les types des variables du dataset `penguins` → on utilise la fonction `names()`

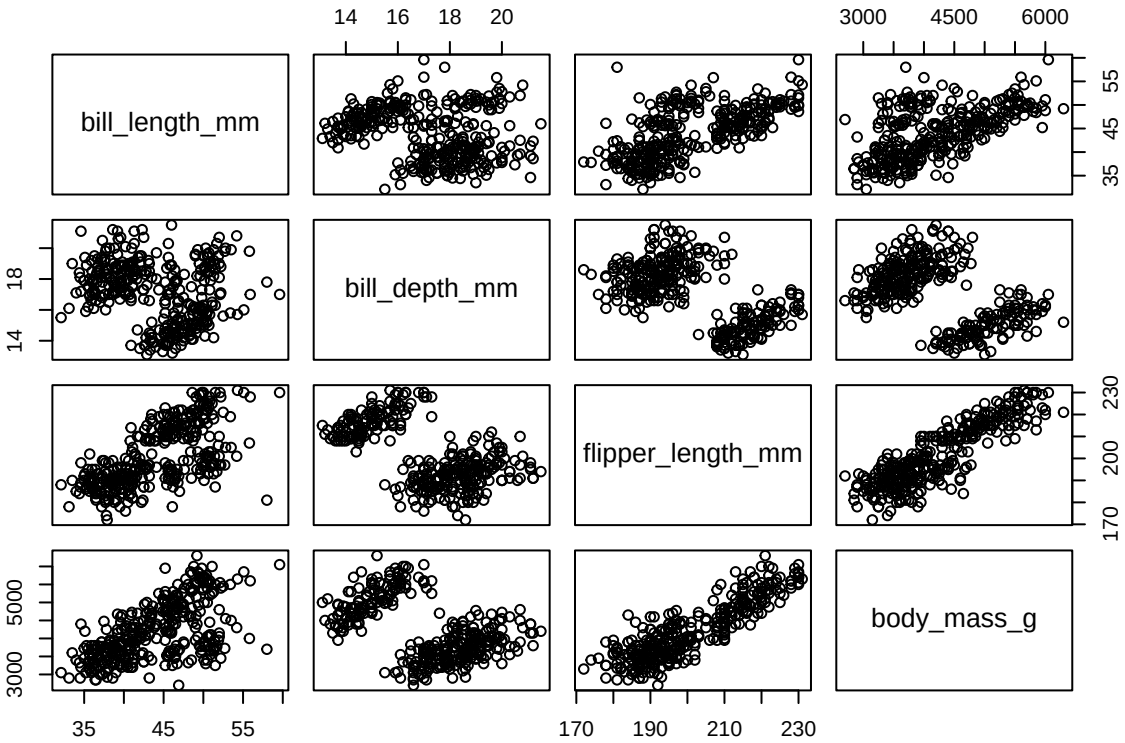
```r
names(penguins_noNA)
```

```
## [1] "species"           "island"            "bill_length_mm"
## [4] "bill_depth_mm"     "flipper_length_mm" "body_mass_g"
## [7] "sex"               "year"
```

Réaliser ensuite :

- un affichage des nuages de points des variables **numériques** deux à deux → on ne conserve que les variables numériques avec la fonction `select()` de `dplyr` est une clause `where`, on retire la colonne `year`, puis on utilise la fonction `plot()` sur le dataframe

```r
penguins_noNA_num <-  select(penguins_noNA, where(is.numeric))
penguins_noNA_num <-  select(penguins_noNA_num,-year)
plot(penguins_noNA_num)
```



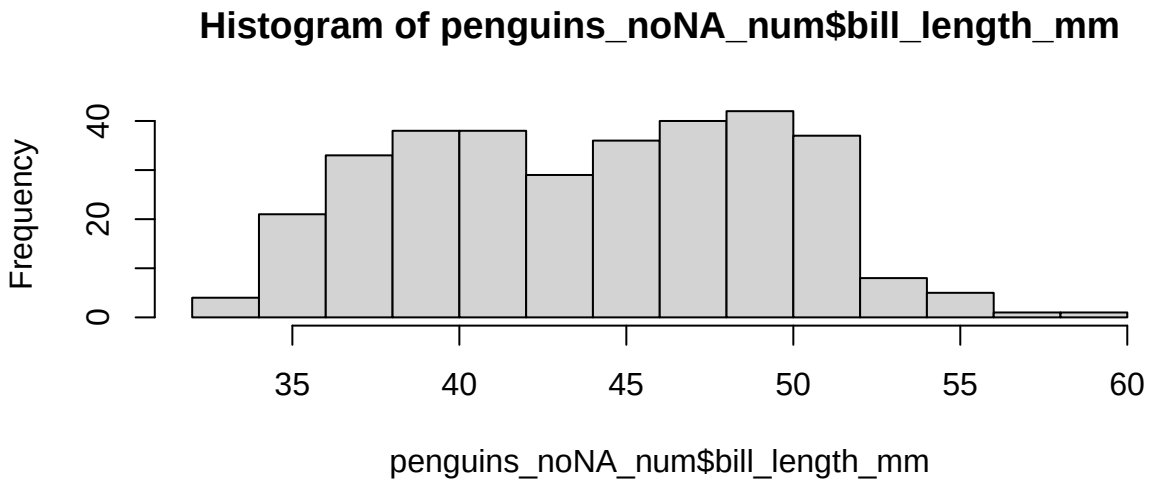
- calculer leur corrélations → on utilise la fonction `cor()`

```r
cor(penguins_noNA_num)
```

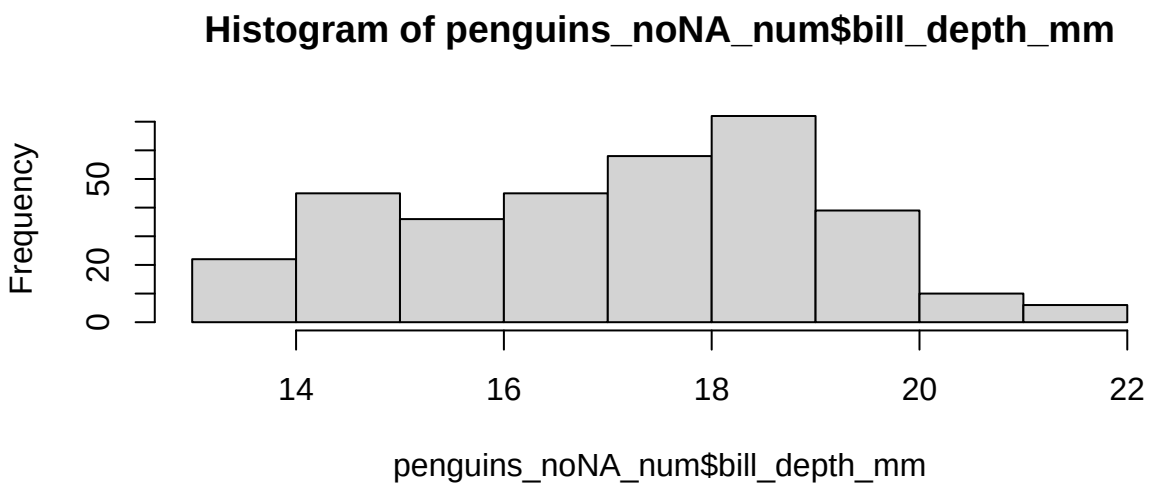
```
##                    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
## bill_length_mm          1.0000000    -0.2286256         0.6530956   0.5894511
## bill_depth_mm          -0.2286256     1.0000000        -0.5777917  -0.4720157
## flipper_length_mm       0.6530956    -0.5777917         1.0000000   0.8729789
## body_mass_g             0.5894511    -0.4720157         0.8729789   1.0000000
```

- afficher la densité ou l'histogramme de chaque variable → on peut faire les quatre histogrammes l'un après l'autre :

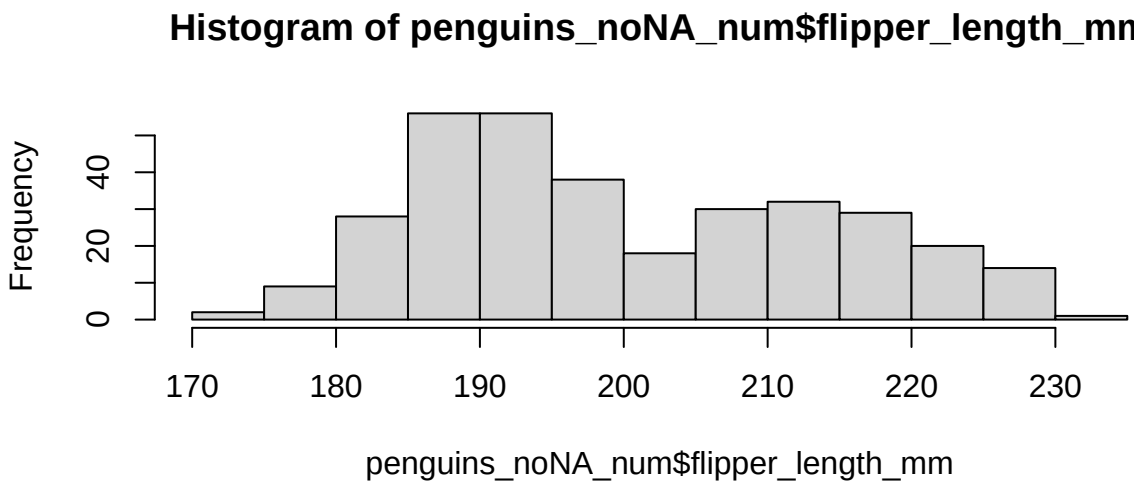
```r
hist(penguins_noNA_num$bill_length_mm)
```



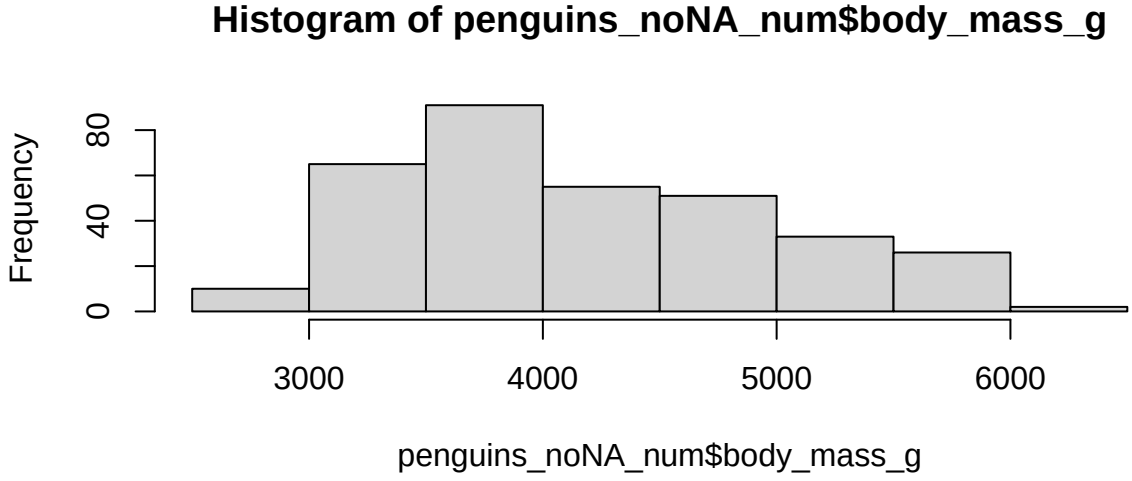

```r
hist(penguins_noNA_num$bill_depth_mm)
```




```r
hist(penguins_noNA_num$flipper_length_mm)
```

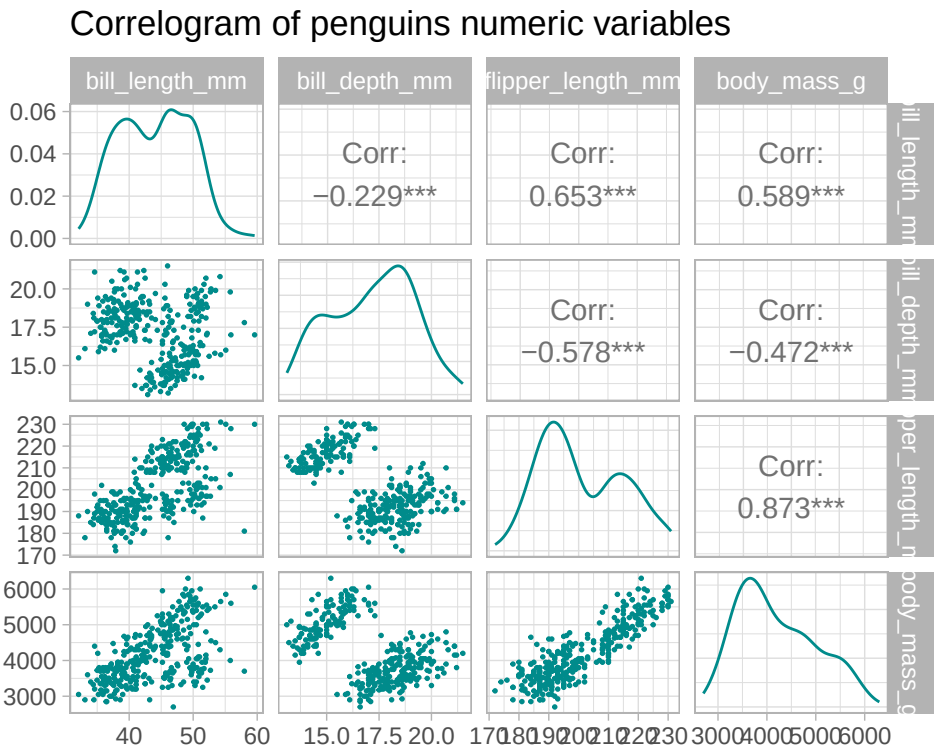



```r
hist(penguins_noNA_num$body_mass_g)
```



Commenter ces graphiques : quelle structure remarquez-vous ? → on remarque des groupes d'individus (qu'on se doute correspondre à l'espèce des manchots), deux variables sont particulièrement corrélées : la masse et la longueur des nageoires.

Voici une version synthétique de ce que vous devriez obtenir :



Question 3 : Préparation des données

Créer un dataframe nommé `dataACP` contenant uniquement les variables numériques qui dérivent la morphologie des pingouins → on a déja constitué cet objet : c'est `penguins_noNA_num`, on peut simplement affecter ce dataframe à un nouvel objet nommé `dataACP`.

Analyse en composantes principales

Question 4 : calculer l'inertie de `dataACP` *sans les normaliser*

→ l'inertie est la somme des variances . La variance d'une série de valeurs s'obtient avec la fonction `var()`, on peut donc écrire

```
sum(var(dataACP))
```

```
## [1] 672064.7
```

Question 4 bis : Pourquoi calculer l'inertie de variables normalisées est inutile (et trivial) ?

On sait que lorsqu'on centre et réduit une variable , sa variance vaut 1. On aura donc autant de d'inertie que le nombre de variables(ici 4)

Question 5 : Réaliser une ACP sur `dataACP` et stocker le résultat dans une variable (e.g. `result_ACP`)

→ la documentation indique que par défaut les paramètres `center` et `scale` qui commandent le centrage et la reduction des variables sont mis à `TRUE` par défaut. Ici on sélectionne 4 axes .

```
resultACP <-  dudi.pca(dataACP, center = FALSE, scale = FALSE, scannf = FALSE, nf=4)
resultACP
```

```
## Duality diagramm
## class: pca dudi
```

```
## $call: dudi.pca(df = dataACP, center = FALSE, scale = FALSE, scannf = FALSE,
##     nf = 4)
##
## $nf: 4 axis-components saved
## $rank: 4
## eigen values: 18390000 750.4 16.46 5.257
##   vector length mode    content
## 1 $cw    4      numeric column weights
## 2 $lw    333    numeric row weights
## 3 $eig   4      numeric eigen values
##
##   data.frame nrow ncol content
## 1 $tab       333  4    modified array
## 2 $li        333  4    row coordinates
## 3 $l1        333  4    row normed scores
## 4 $co        4    4    column coordinates
## 5 $c1        4    4    column normed scores
## other elements: cent norm
```

Les résultats de l'ACP sont stockés dans l'objet (astucieusement nommé) `resultatACP`

Question 5 bis : quel est le pourcentage d'inertie capturée par les deux premières composantes ? Comment l'obtenir sans lire le scree-plot ?

Question 5 ter : quelle est la coordonnée de la deuxième composante dans l'espace de départ ?

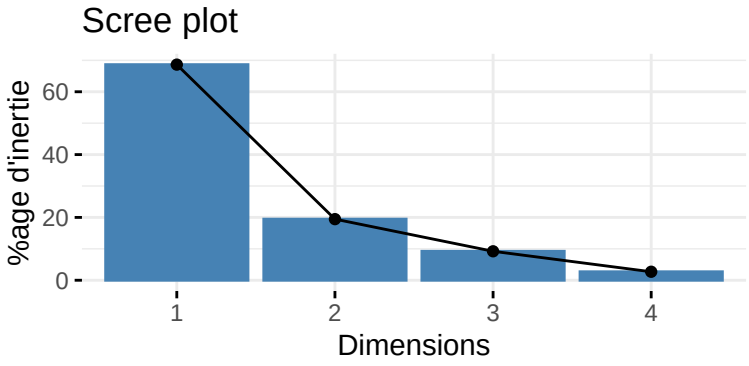
Question 6 : D'après-vous, faut-il normaliser les variables de `dataACP` lors de l'ACP ? Pourquoi ?

Pour s'en assurer, recommencer le calcul de l'ACP et comparer les résultats .

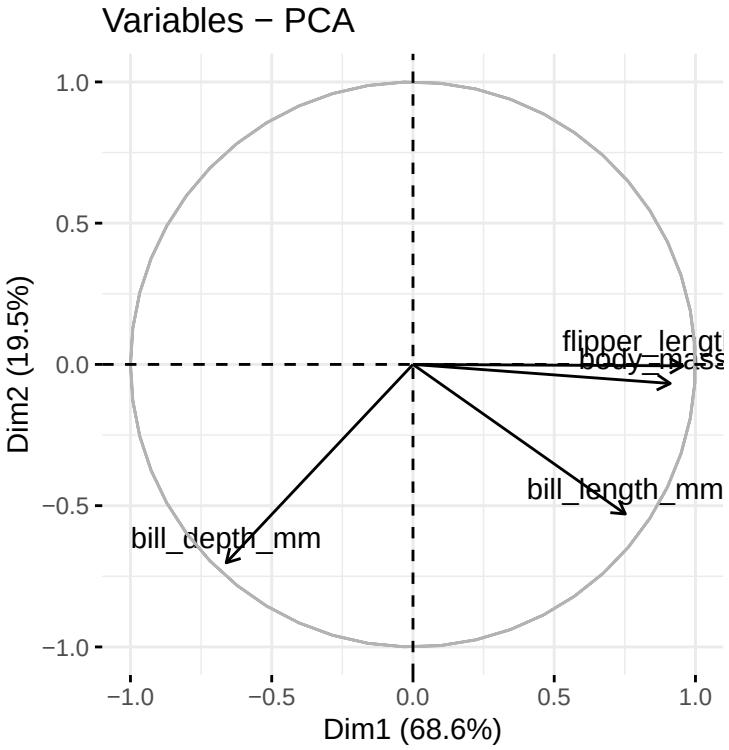
Interprétation des résultats

Question 7 : l'ACP s'est-elle bien passée ? Justifier .

Vous devriez obtenir à la question 5 un graphique à l'allure suivante :



Question 8 : Que dire des variables projetées dans le plan formé par les deux premières composantes ?



Question 9 : Quelle est la contribution des variables `bill_length` et `bill_depth` à la 3ème composante ?

Question 10 : projeter les individus dans le plan formé par les deux premières composantes . Interpréter.

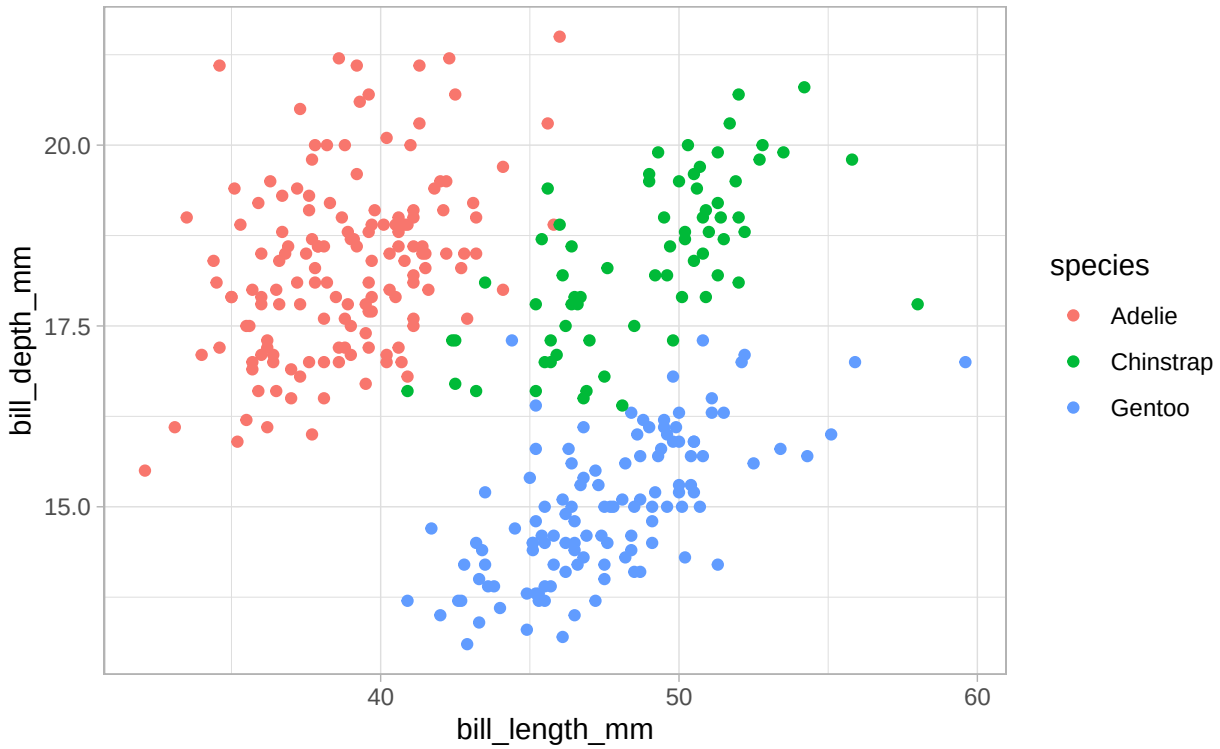
Indice : cf. la liste des fonctions de `factoextra`

Regrouppements

Nous cherchons maintenant à trouver une projection qui permette de séparer visuellement les trois espèces de pengouins.

Voici comment obtenir des nuages de points des variables des pingouins ,colorés par espèce :

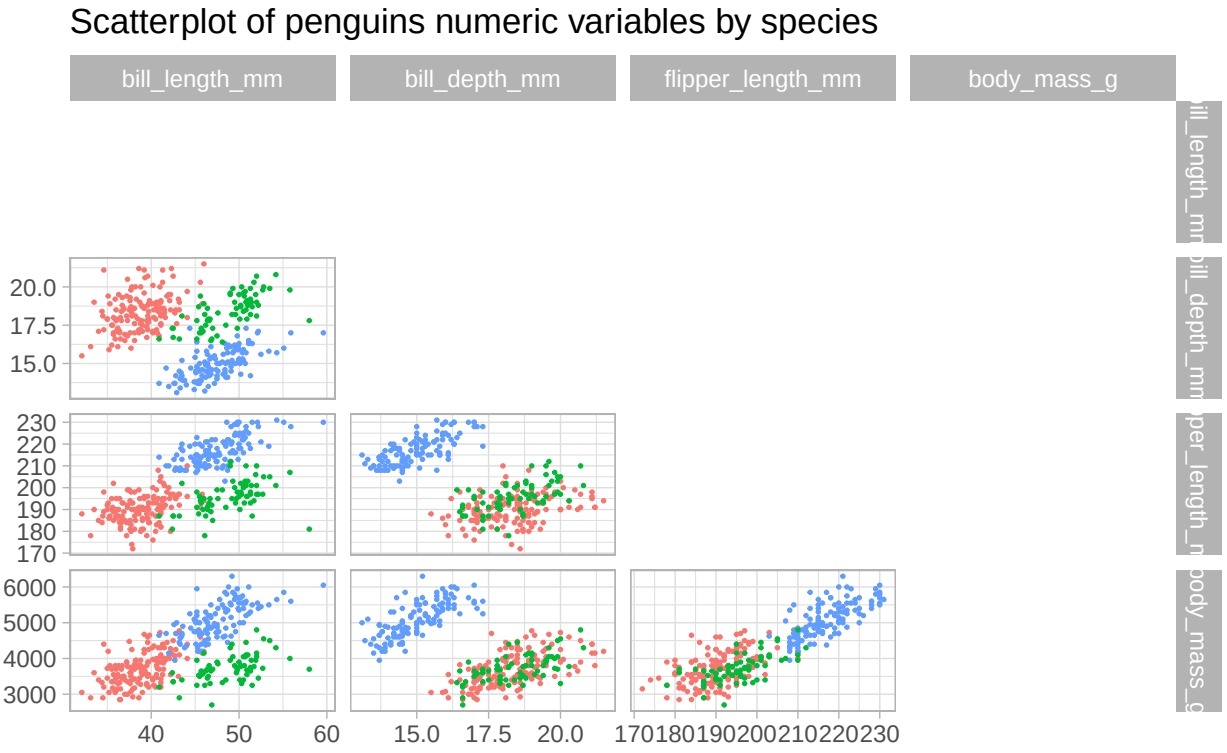
```
ggplot(penguins, aes(x = bill_length_mm, y = bill_depth_mm, colour = species)) +
  geom_point() +
  theme_light()
```



Voici une version pour tous les couples de variables :

```
ggpairs(data= penguins[,c("species" ,"bill_length_mm","bill_depth_mm", "flipper_length_mm", "body_mass_g
        columns = c("bill_length_mm","bill_depth_mm", "flipper_length_mm", "body_mass_g" ),
        #lower = list(continuous ="points", size=0.1),
        lower= list(continuous =wrap("points",  size=0.3), combo = "facethist", discrete = "facetbar", n
         #upper= list(continuous =wrap("points",  size=0.3), combo = "facethist", discrete = "facetbar"
        title="Scatterplot of penguins numeric variables by species",

        diag=NULL,
        upper=NULL,
        mapping = aes(color=species)
        ) +theme_light()
```



Question 11 : Les graphiques ci-dessus permettent-ils d'opérer cette classification visuelle ?