VID - Travail Pratique 2

Farouk Ferchichi et Hugo Huart

2023

Table des matières

Introduction

Ce travail pratique à pour thèmes les modèles de régression linéaire et pour but le perfectionnement de la rédaction de rapports industriels ainsi que de la maîtrise du logiciel **R**.

Chargement des librairies

```
library("ggplot2")
library("ggResidpanel")
library("ggrepel")
library("GGally")
library("rgl")
library("scatterplot3d")
library("leaps")
library("performance")
library("tidyverse")
library("palmerpenguins")
library("palettes")
library("MetBrewer")
library("PrettyCols")
library("harrypotter")
library("paletteer")
library("showtext")
```

Exercice 1

Cet exercice à pour but d'appliquer un modèle de régression linéaire sur des données comprenant la taille moyenne en centimètres (cm) d'un groupe d'enfants de Kalama, village en Egypte, mesurée chaque mois entre le 18-ième et le 29-ième mois.

Chargement des données dans la variable `kalama`:

```
kalama<-read.table("data/kalama.txt", header=T)
```

1 - a)

Calcul du coefficient de corrélation entre la taille et l'âge:

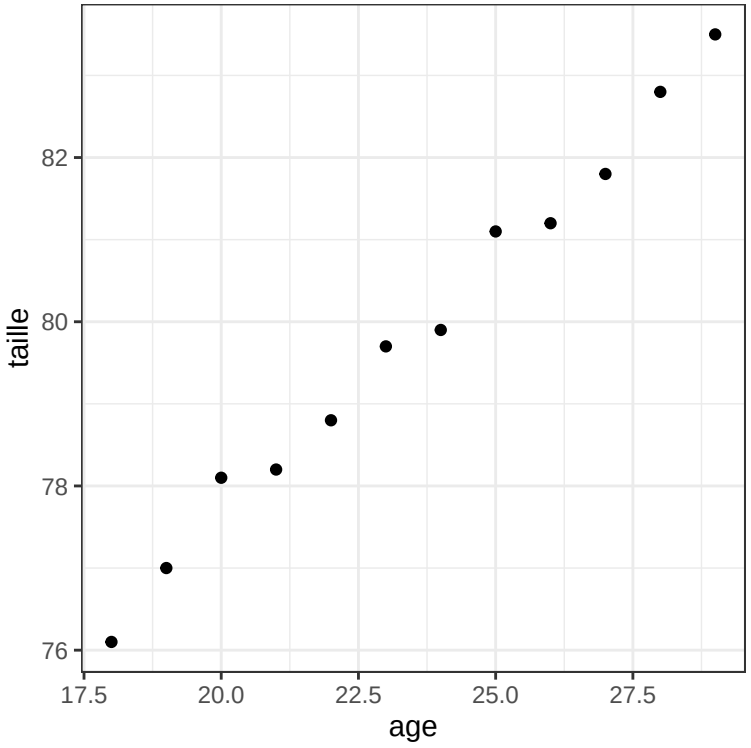
```
cor(kalama$age, kalama$taille)
```

```
## [1] 0.9943661
```

1 - b)

Affichage du graphique en nuage de points des valeurs taille versus âge:

```
ggplot(kalama, aes(x=age, y=taille)) +
  geom_point() +
  theme_bw()
```



1 - c)

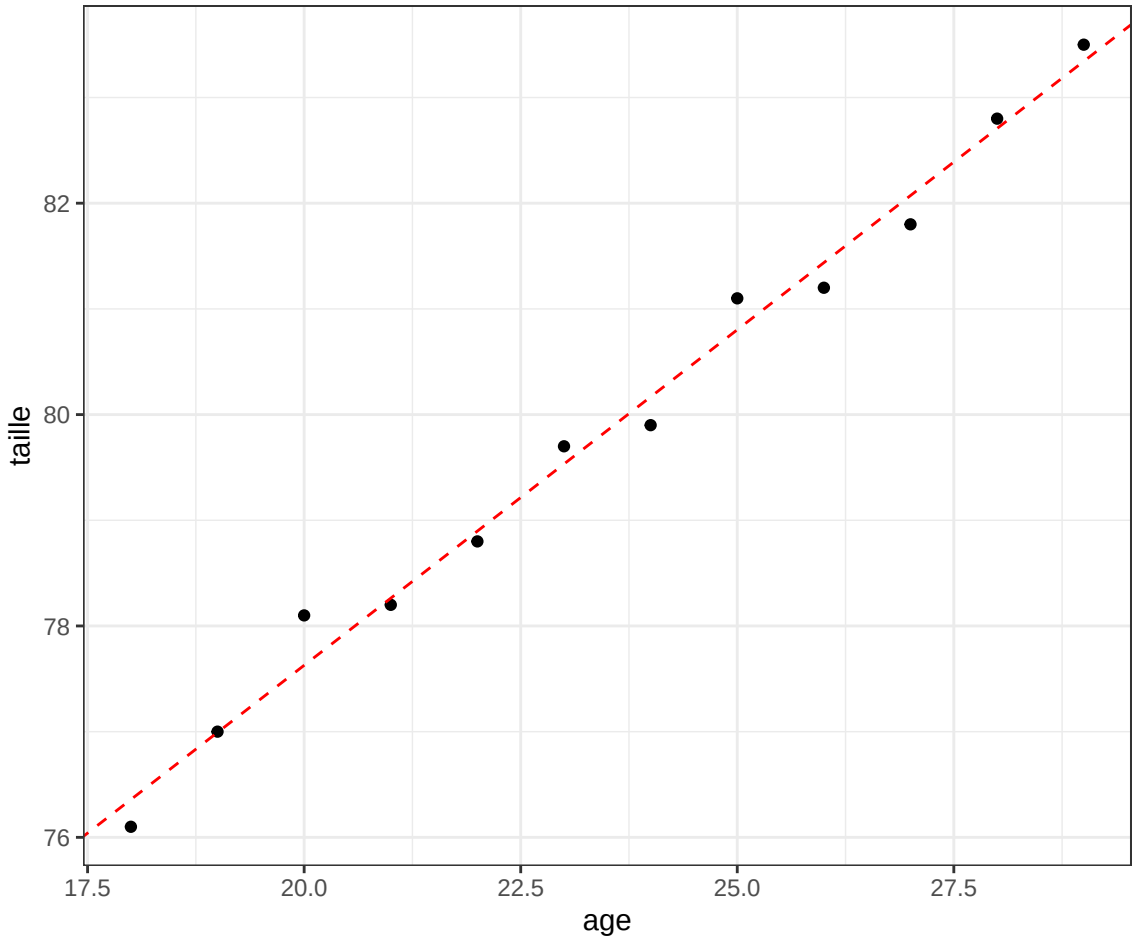
Estimation des coefficients β_0 (`intercept`) et β_1 (`slope`):

```
kalama.lm<-lm(taille~age, data=kalama)
kalama.coef<-data.frame(intercept=coef(kalama.lm)[1], slope=coef(kalama.lm)[2])
```

1 - d)

Affichage du graphique précédent modifié avec la droite des moindres carrés en rouge:

```
ggplot(data=kalama, aes(x=age, y=taille)) +
  geom_point() +
  geom_abline(data=kalama.coef, aes(intercept=intercept, slope=slope),
              colour="red", linetype=2) +
  theme_bw()
```



1 - e)

On peut estimer la variance des erreurs en calculant la variance des résidus:

```
var(resid(kalama.lm))
```

```
## [1] 0.05956135
```

1 - f)

Estimation du coefficient de détermination:

```
summary(kalama.lm)$r.squared
```

```
## [1] 0.9887639
```

Étant donné qu'il s'agit d'une régression linéaire simple, on peut également calculer le coefficient de détermination avec le carré de celui de corrélation:

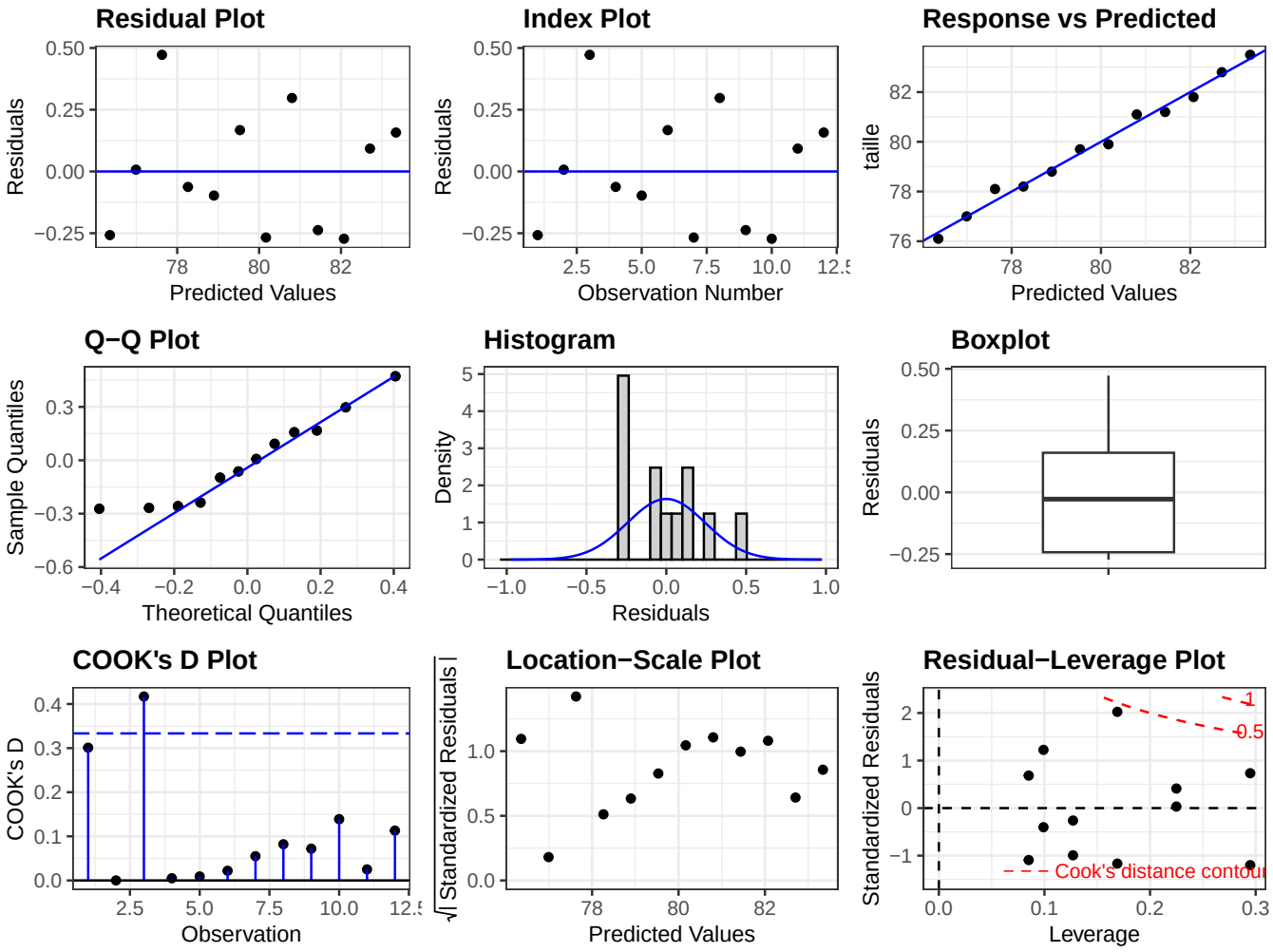
```
cor(kalama$age, kalama$taille)^2
```

```
## [1] 0.9887639
```

1 - g)

Affichage d'un diagnostic du modèle, à l'aide de la libraire `ggResidpanel`:

```
resid_panel(kalama.lm, plots="all")
```



1 - h)

On constate dans le graphique des résidus du point précédent que ces derniers ont des valeurs plutôt faibles et sont répartis de façon homogène; le modèle est donc assez bon en ce sens.

De plus, le graphique du point **d)** permet de constater facilement l'alignement des valeurs et le faible écart par rapport à la droite de régression. La variance des erreurs estimée au point **e)** est elle aussi très faible.

1 - i)

En regardant le résumé du modèle de régression linéaire:

```
summary(kalama.lm)
```

```
##
## Call:
## lm(formula = taille ~ age, data = kalama)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27238 -0.24248 -0.02762  0.16014  0.47238
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  64.9283     0.5084  127.71  < 2e-16 ***
## age           0.6350     0.0214   29.66 4.43e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.256 on 10 degrees of freedom
## Multiple R-squared:  0.9888, Adjusted R-squared:  0.9876
## F-statistic:   880 on 1 and 10 DF,  p-value: 4.428e-11
```

On constate que la *p-value* de la pente du modèle (à la ligne `age` et la colonne `Pr(>|t|)` du tableau `Coefficients`) est bien inférieure à 5%, ce qui est significatif.

Exercice 2

Cet exercice à pour but d'analyser le Produit National Brut par habitant (PNB) et la mortalité infantile de 14 pays européens.

2 - a)

Chargement des données dans l'objet `mortalite.pnb`:

```
mortalite.pnb <- data.frame(pays=c("Allemagne RFA","Autriche","Belgique","Danemark","Espag
"France","Grèce","Irlande","Italie","Luxembourg","Pays-Bas",
"Portugal","Royaume-Uni","Suisse"),
PNB=c(190, 128, 180, 212, 56, 192, 68, 98, 110, 197, 155, 40, 181, 233),
mortalite=c(24, 28, 24, 19, 37, 22, 34, 25, 36, 24, 14, 65, 20, 18))
print(mortalite.pnb)
```

```
##             pays PNB mortalite
## 1  Allemagne RFA 190        24
## 2       Autriche 128        28
## 3       Belgique 180        24
## 4       Danemark 212        19
## 5        Espagne  56        37
## 6         France 192        22
## 7          Grèce  68        34
## 8        Irlande  98        25
## 9         Italie 110        36
## 10    Luxembourg 197        24
## 11      Pays-Bas 155        14
## 12      Portugal  40        65
## 13   Royaume-Uni 181        20
## 14        Suisse 233        18
```

2 - b)

Affichage du résumé des variables:

```
summary(mortalite.pnb)
```

```
##      pays                PNB          mortalite
##  Length:14          Min.   : 40.0   Min.   :14.00
##  Class :character   1st Qu.:101.0   1st Qu.:20.50
##  Mode  :character   Median :167.5   Median :24.00
##                     Mean   :145.7   Mean   :27.86
##                     3rd Qu.:191.5   3rd Qu.:32.50
##                     Max.   :233.0   Max.   :65.00
```

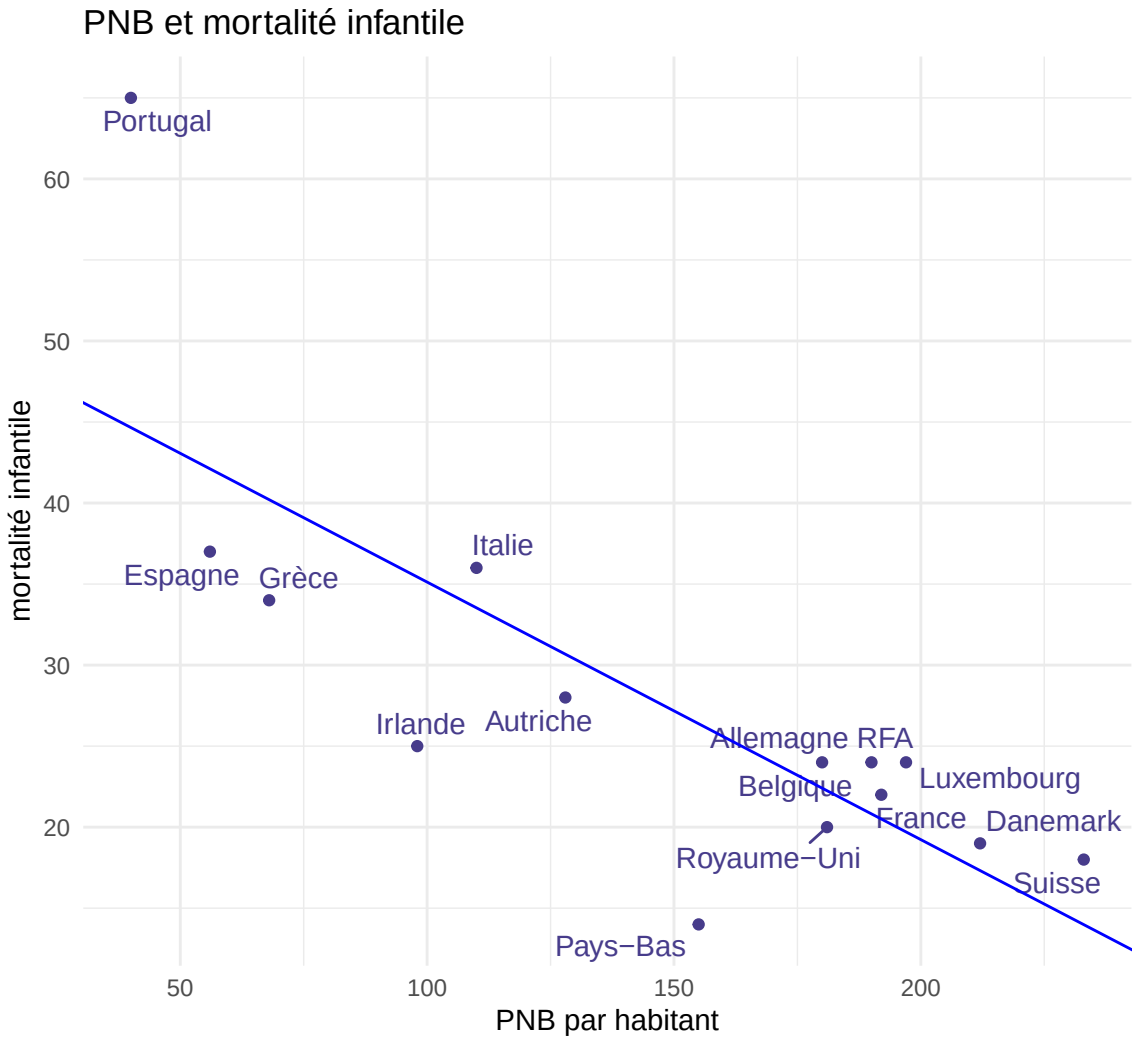
2 - c)

On commence par appliquer un modèle de régression linéaire sur nos données:

```
mortalite.pnb.lm<-lm(mortalite~PNB, data=mortalite.pnb)
mortalite.pnb.coef<-data.frame(intercept=coef(mortalite.pnb.lm)[1],
                               slope=coef(mortalite.pnb.lm)[2])
```

Affichage d'un nuage de point représentant la mortalité infantile *(y)* en fonction du produit national brut *(x)*:

```
ggplot(data=mortalite.pnb, aes(x=PNB, y=mortalite, label=pays)) +
  geom_point(colour="slateblue4") +
  geom_text_repel(colour="slateblue4") +
  geom_abline(data=mortalite.pnb.coef, aes(slope=slope, intercept=intercept),
              colour="blue") +
  labs(x="PNB par habitant", y="mortalité infantile",
       title="PNB et mortalité infantile") +
  theme_minimal()
```



Le Portugal est de loin le pays le plus éloigné de la tendance générale. Dans une moindre mesure,

les Pays-Bas et l'Irlande se démarquent également, bien que plus légèrement.

2 - d)

En prenant les valeurs suivantes du modèle:

```
mortalite.pnb.lm
```

```
##
## Call:
## lm(formula = mortalite ~ PNB, data = mortalite.pnb)
##
## Coefficients:
## (Intercept)          PNB
##     51.0133      -0.1589
```

On peut écrire l'équation $y = \beta_0 + \beta_1 x1 + e$ où e correspond à l'erreur, β_0 à la valeur de `(Intercept)` (51.0133) et β_1 à la valeur de -0.1589 et x_1 la variable `PNB`.

2 - e)

Calcul du coefficient de corrélation r:

```
mortalite.pnb.cor<-cor(mortalite.pnb$PNB, mortalite.pnb$mortalite)
mortalite.pnb.cor
```

```
## [1] -0.7814228
```

2 - f)

Calcul du coefficient de détermination R^2 égal à r^2:

```
mortalite.pnb.cor^2
```

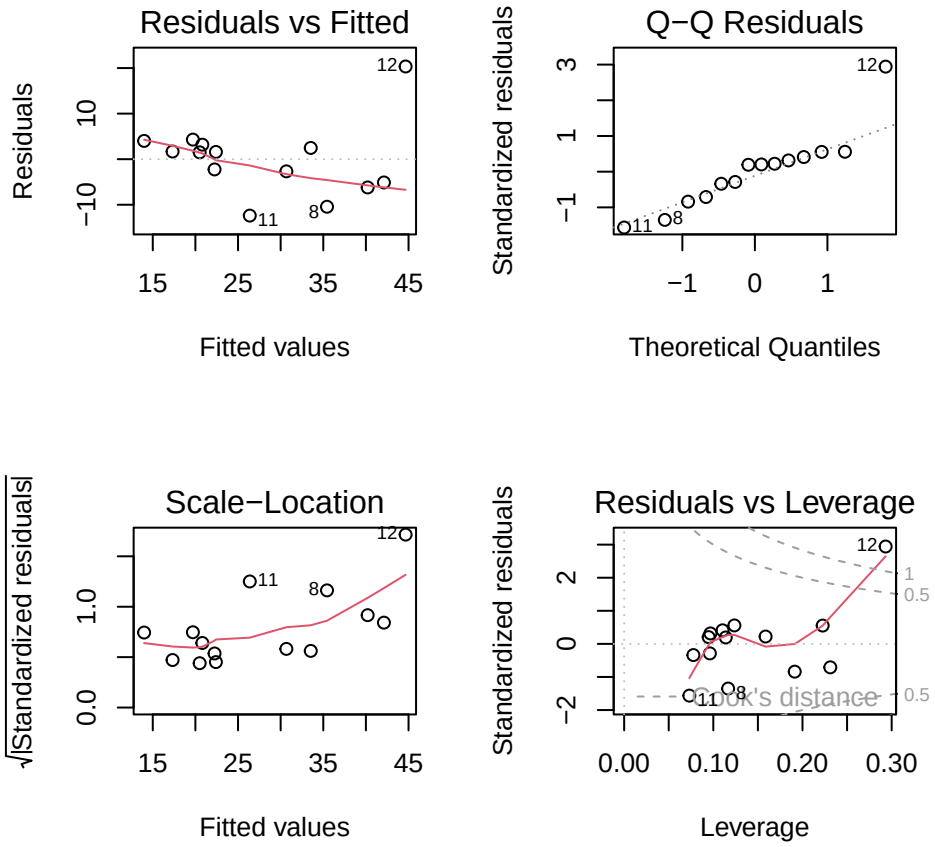
```
## [1] 0.6106217
```

2 - g)

Affichage des 4 graphiques aidant à la vérification des hypothèses inhérentes au modèle, c'est-à-dire:

1. La linéarité de la relation;
2. la nullité de l'espérance des erreurs et leur variance constante;
3. La normalité des variables aléatoires erreurs.

```
par(mfrow=c(2, 2))
plot(mortalite.pnb.lm)
```



Dans ces graphiques, **R** a relevé les 3 pays aux valeurs les plus anormales identifiées précédemment en affichant leur numéro respectifs. Il s'agit de l'Irlande (**8**), des Pays-Bas (**11**) et du Portugal (**12**).

Ci-dessous se trouvent les informations relevées par chaque graphique:

Nuage de points des résidus (`Residuals vs Fitted`)

On constate une variabilité assez proche autour de 0 malgré une légère tendance à la baisse. Le modèle reste assez satisfaisant mais pas totalement idéal en ce sens. L'hypothèse d'espérance des erreurs nulle semble être vérifiée.

Nuage de points de la racine carrée de la valeur absolue des résidus contre les valeurs ajustées (`Scale-Location`)

Hormis les valeurs plutôt atypiques déjà connues, les valeurs ont toutes une variance assez semblable. L'hypothèse de variance constante est vérifiée.

Graphique des résidus observés standardisés contre les résidus théoriques (`Q-Q Residuals`)

Hormis la valeur du Portugal, les valeurs restent proches de la droite. L'hypothèse de normalité des erreurs est vérifiée.

Graphique représentant la distance de Cook (`Residuals vs Leverage`)

On retrouve la valeur du Portugal au delà des traitillés, ce qui confirme encore une fois son statut de valeur atypique.

2 - h)

Prédiction de la mortalité infantile pour un PNB de 100.

```
xnew <- matrix(c(100), nrow=1)
colnames(xnew) <- c("PNB")
xnew <- as.data.frame(xnew)
predict(mortalite.pnb.lm, xnew, interval="pred")
```

```
##        fit      lwr      upr
## 1 35.12181 16.22173 54.02189
```

La prédiction nous donne une valeur d'environ 35.12.

2 - i)

Le nouveau modèle possédant un coefficient de détermination de 0.74 est sensiblement meilleur au modèle existant, possédant un coefficient d'environ 0.61. Celui du nouveau modèle étant plus proche de 1, on peut considérer ce dernier comme étant meilleur.

Exercice 3

Cet exercice a pour but d'analyser la relation entre la taille de la portée, le poids du corps ainsi que le poids du cerveau d'un échantillon de 20 souris.

Chargement des données nécessaires dans un objet `litters`:

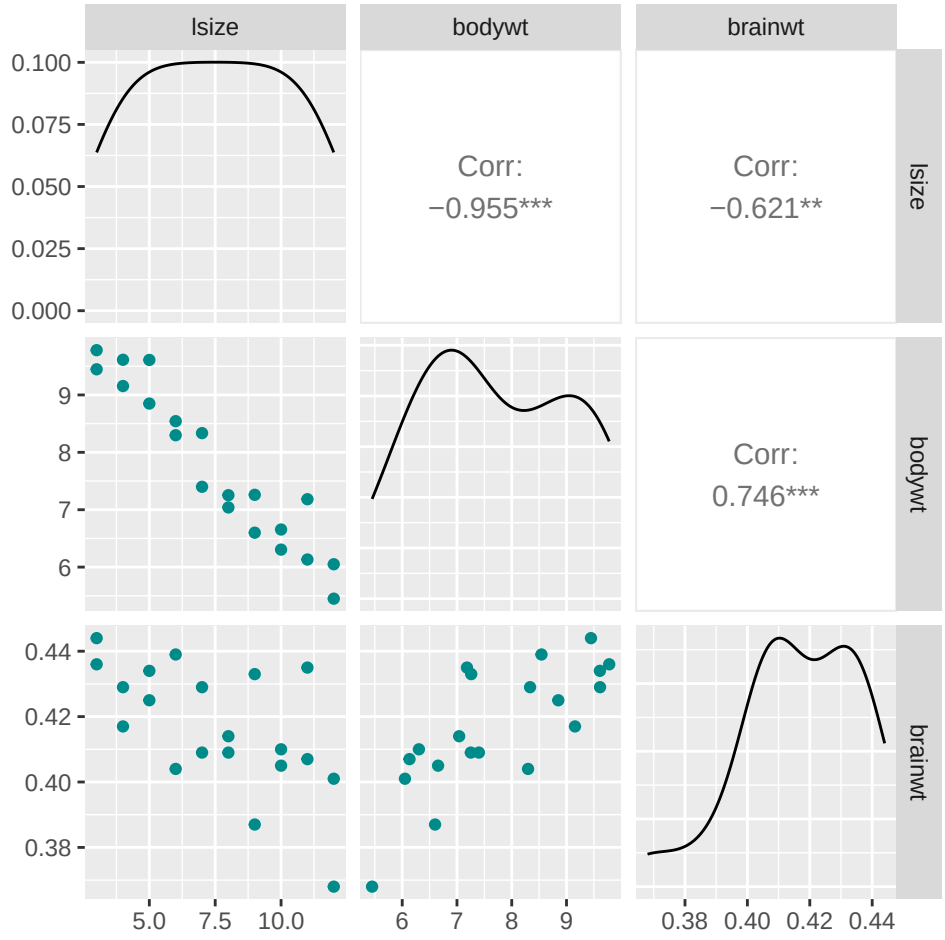
```
litters<-readRDS("data/litters.rds")
litters
```

```
##    lsize bodywt brainwt
## 1      3  9.447   0.444
## 2      3  9.780   0.436
## 3      4  9.155   0.417
## 4      4  9.613   0.429
## 5      5  8.850   0.425
## 6      5  9.610   0.434
## 7      6  8.298   0.404
## 8      6  8.543   0.439
## 9      7  7.400   0.409
## 10     7  8.335   0.429
## 11     8  7.040   0.414
## 12     8  7.253   0.409
## 13     9  6.600   0.387
## 14     9  7.260   0.433
## 15    10  6.305   0.410
## 16    10  6.655   0.405
## 17    11  7.183   0.435
## 18    11  6.133   0.407
## 19    12  5.450   0.368
## 20    12  6.050   0.401
```

3 - a)

Affichage du graphique de corrélations et des nuages de points:

```
ggpairs(litters, lower=list(continuous=wrap("points", colour="cyan4")))
```



3 - b)

À l'aide du nuage de points et de la valeur du coefficient de corrélation d'environ -0.95, on constate un relation négative entre les variables `lsize` et `bodywt`.

3 - c)

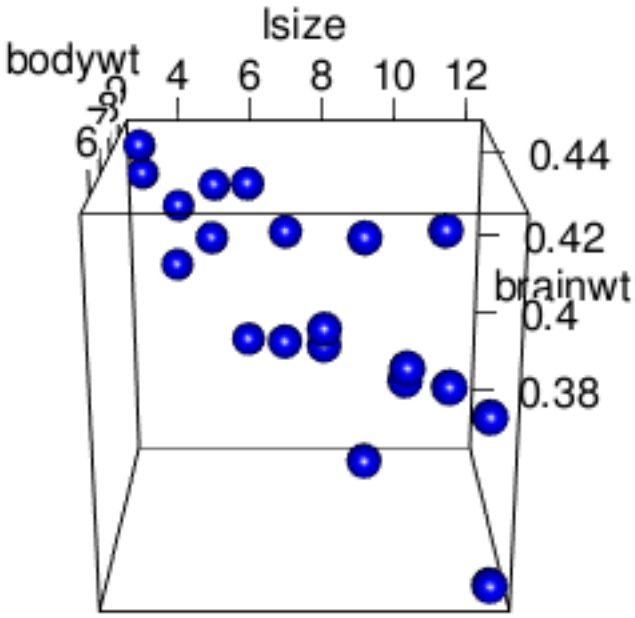
Entre `brainwt` et `lsize`, la relation négative est aussi évidente mais plus mesurée. Le coefficient de corrélation a une valeur d'environ -0.62.

Entre `brainwt` et `bodywt`, on constate une relation positive à l'aide du graphique et du coefficient d'une valeur d'environ 0.74.

3 - d)

Affichage d'un graphique en 3D interactif représentant les 3 variables `lsize`, `bodywt`, `brainwt`:

```
plotids <- with(litters, plot3d(lsize, bodywt, brainwt,
                                type="s", col="blue"))
rglwidget(elementId = "plot3drgl")
```



En manipulant le cube contenant le graphique, on constate un alignement des points dans l'espace.

3 - e)

Bien qu'intéressant à manipuler, il n'est pas très commode de distinguer les deux effets des variables explicatives sur la variable de réponse. Le graphique du point **a)** était déjà suffisant à cet égard.

3 - f)

En utilisant un modèle linéaire multiple avec `brainwt` comme variable de réponse ainsi que `lsize` et `bodywt` comme variables explicatives:

```
litters.fit<-lm(brainwt~lsize+bodywt, data=litters)
litters.fit
```

```
##
## Call:
## lm(formula = brainwt ~ lsize + bodywt, data = litters)
##
## Coefficients:
## (Intercept)        lsize       bodywt
##     0.17825      0.00669      0.02431
```

On a donc l'équation $y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + e$ où e est l'erreur, β_0 vaut environ 0.17, β_1 vaut 0.0067 et β_2 vaut 0.024. De plus, x_1 correspond à la variable `lsize` et x_2 à `bodywt`.

3 - g)

Affichage de la table de résumé pour `litters.fit`:

```
summary(litters.fit)
```

```
##
## Call:
## lm(formula = brainwt ~ lsize + bodywt, data = litters)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0230005 -0.0098821  0.0004512  0.0092036  0.0180760
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.178247   0.075323   2.366  0.03010 *
## lsize       0.006690   0.003132   2.136  0.04751 *
## bodywt      0.024306   0.006779   3.586  0.00228 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01195 on 17 degrees of freedom
## Multiple R-squared:  0.6505, Adjusted R-squared:  0.6094
## F-statistic: 15.82 on 2 and 17 DF,  p-value: 0.0001315
```

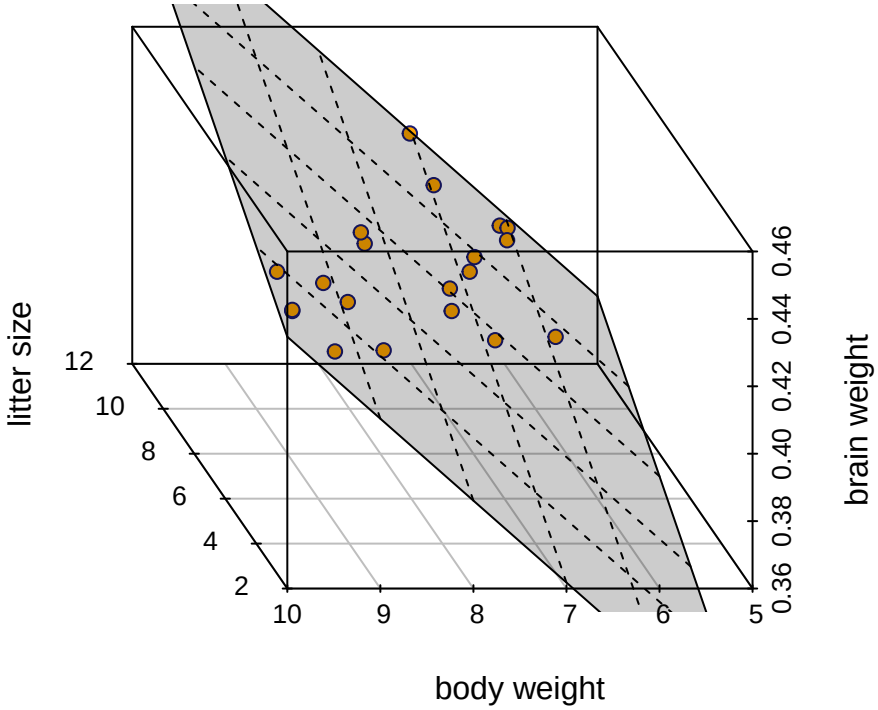
Les deux variables explicatives sont significatives à un niveau de 5. Cependant, `bodywt` est plus significative que `lsize`, étant à un niveau en deçà de 1 contre un niveau d'environ 4.75 pour `lsize`.

Ceci correspond aux coefficients et nuages de points déterminés précédemment.

3 - h)

Affichage d'un nuage de points comprenant les 3 variables:

```
s3d <- scatterplot3d(litters$lsize, litters$bodywt, litters$brainwt, main = "",
color="midnightblue", xlab="litter size", ylab="body weight",
zlab="brain weight", angle = -60, pch=21, bg="orange")
s3d$plane3d(litters.fit, draw_polygon=TRUE, lty.box="solid")
```



3 - i)

Affichage du coefficient de détermination R^2:

```
summary(litters.fit)$r.squared
```

```
## [1] 0.6505341
```

Affichage du coefficient de détermination ajusté R^2_{adj}:

```
summary(litters.fit)$adj.r.squared
```

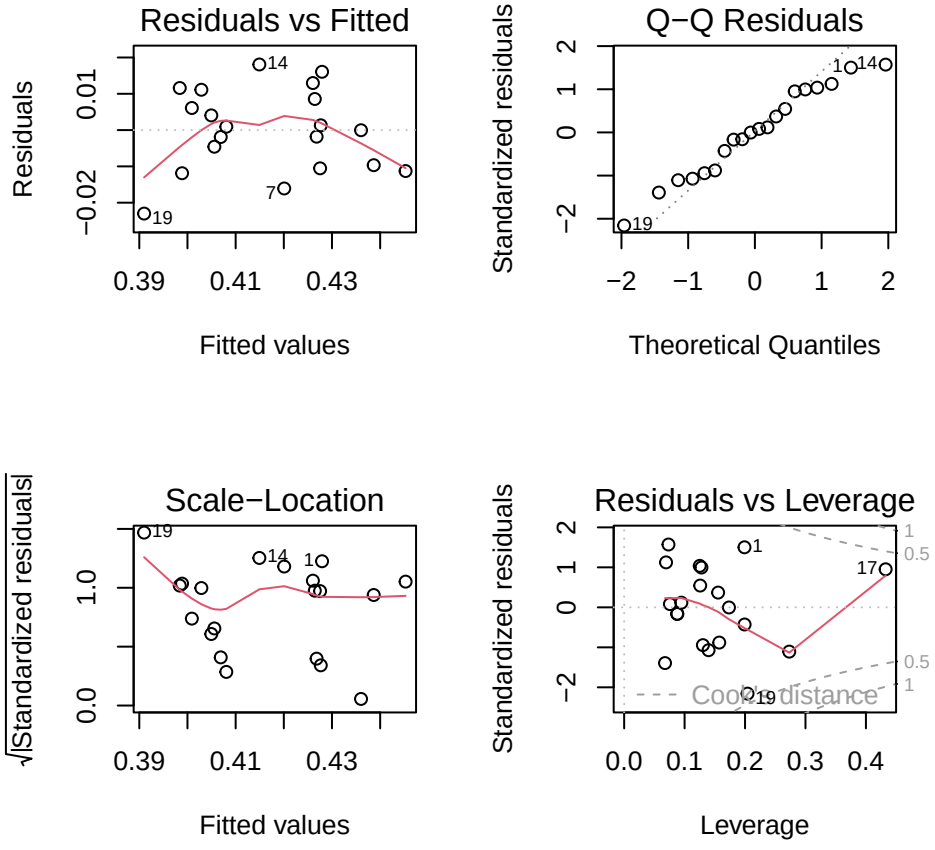
```
## [1] 0.6094205
```

Étant donné que la taille de l'échantillon est petite ($n = 20$), le coefficient ajusté est plus judicieux. On constate ici que sa valeur est nettement plus faible que celle de R^2, ce qui confirme la surestimation de ce dernier.

3 - j)

Affichage des 4 graphiques aidant à la vérification des hypothèses inhérentes au modèle:

```
par(mfrow=c(2, 2))
plot(litters.fit)
```



Nuage de points des résidus (`Residuals vs Fitted`)

Il n'y a pas vraiment de tendance qui se dessine. Les valeurs sont également très proche de l'axe nul, il semble adéquat de valider l'hypothèse de l'espérance des erreurs nulle.

Nuage de points de la racine carrée de la valeur absolue des résidus contre les valeurs ajustées (`Scale-Location`)

La variance des résidus semblent assez constante le long de la tendance. L'hypothèse de variance constante peut être validée.

Graphique des résidus observés standardisés contre les résidus théoriques (`Q-Q Residuals`)

Les points restent en général proche de la droite. On note cependant un léger décrochage au début et à la fin de cette dernière. On peut quand-même accepter l'hypothèse de la normalité des erreurs.

Graphique représentant la distance de Cook (`Residuals vs Leverage`)

Il n'y a pas vraiment points au delà des traitillés de ce graphique, donc de valeurs atypiques notables. On peut noter cependant que **R** a relevé les points **1**, **14** (ici **17**) et **19** pour ces 4 graphiques.

Globalement, les 3 hypothèses ont pu être validés même si le modèle n'est pas idéal. Il serait peut-être judicieux d'ajouter des facteurs où des transformations supplémentaires dans l'équation du modèle.

Exercice 4

Cet exercice a pour but de traiter un échantillon de 38 pinots noirs issu d'une dégustation sensorielle, représenté par 5 variables explicatives: la transparence (*Clarity*), l'arôme (*Aroma*), le corps (*Body*), la saveur (*Flavor*) ainsi que le goût boisé (*Oakiness*). La variable de réponse y est la qualité du vin dégusté.

On charge ces données dans un objet `wine`:

```
wine<-read.csv("data/Wine.csv")
wine
```

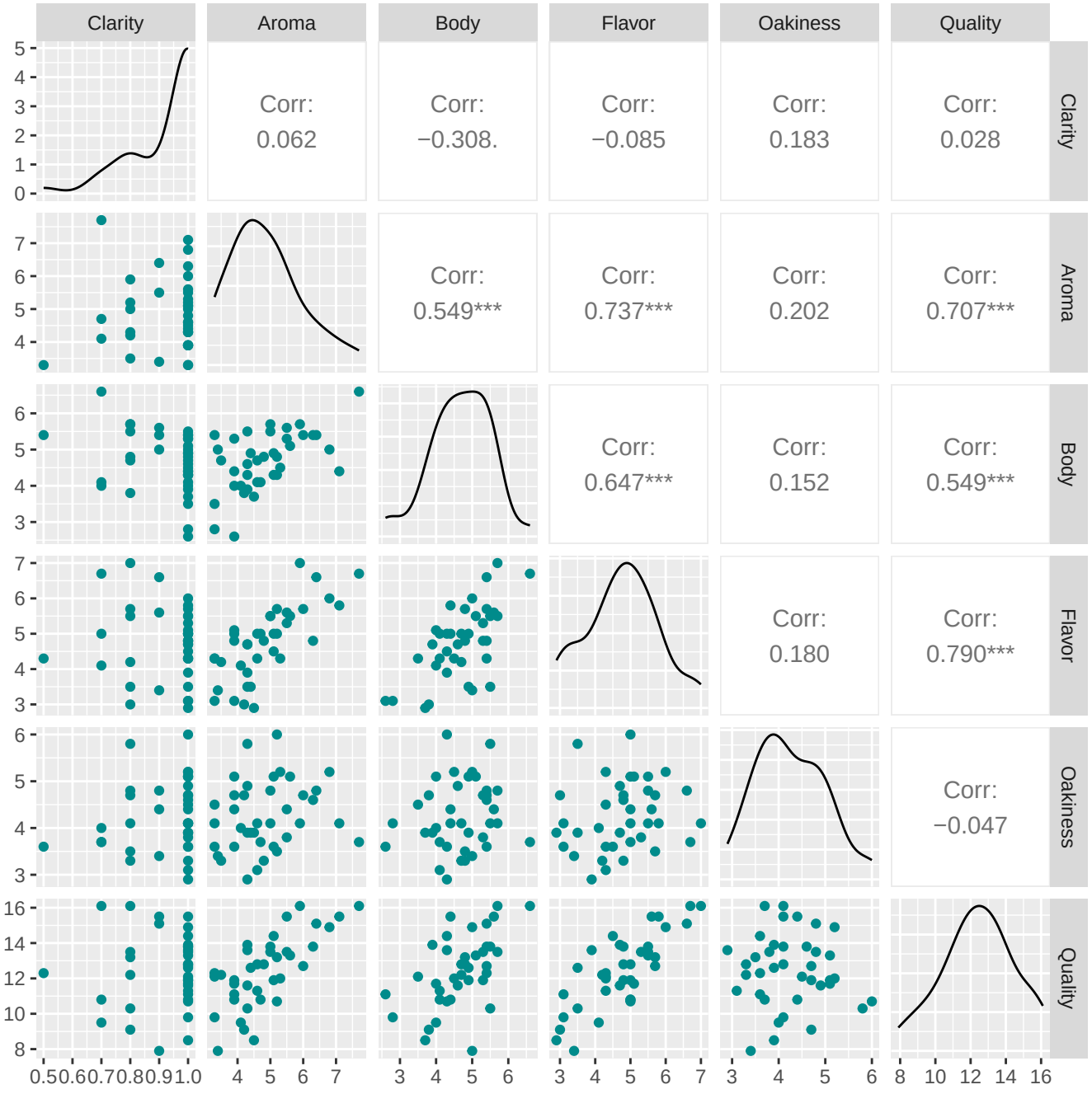
```
##    Clarity Aroma Body Flavor Oakiness Quality
## 1      1.0   3.3  2.8    3.1      4.1     9.8
## 2      1.0   4.4  4.9    3.5      3.9    12.6
## 3      1.0   3.9  5.3    4.8      4.7    11.9
## 4      1.0   3.9  2.6    3.1      3.6    11.1
## 5      1.0   5.6  5.1    5.5      5.1    13.3
## 6      1.0   4.6  4.7    5.0      4.1    12.8
## 7      1.0   4.8  4.8    4.8      3.3    12.8
## 8      1.0   5.3  4.5    4.3      5.2    12.0
## 9      1.0   4.3  4.3    3.9      2.9    13.6
## 10     1.0   4.3  3.9    4.7      3.9    13.9
## 11     1.0   5.1  4.3    4.5      3.6    14.4
## 12     0.5   3.3  5.4    4.3      3.6    12.3
## 13     0.8   5.9  5.7    7.0      4.1    16.1
## 14     0.7   7.7  6.6    6.7      3.7    16.1
## 15     1.0   7.1  4.4    5.8      4.1    15.5
## 16     0.9   5.5  5.6    5.6      4.4    15.5
## 17     1.0   6.3  5.4    4.8      4.6    13.8
## 18     1.0   5.0  5.5    5.5      4.1    13.8
## 19     1.0   4.6  4.1    4.3      3.1    11.3
## 20     0.9   3.4  5.0    3.4      3.4     7.9
## 21     0.9   6.4  5.4    6.6      4.8    15.1
## 22     1.0   5.5  5.3    5.3      3.8    13.5
## 23     0.7   4.7  4.1    5.0      3.7    10.8
## 24     0.7   4.1  4.0    4.1      4.0     9.5
## 25     1.0   6.0  5.4    5.7      4.7    12.7
## 26     1.0   4.3  4.6    4.7      4.9    11.6
## 27     1.0   3.9  4.0    5.1      5.1    11.7
## 28     1.0   5.1  4.9    5.0      5.1    11.9
## 29     1.0   3.9  4.4    5.0      4.4    10.8
## 30     1.0   4.5  3.7    2.9      3.9     8.5
## 31     1.0   5.2  4.3    5.0      6.0    10.7
## 32     0.8   4.2  3.8    3.0      4.7     9.1
## 33     1.0   3.3  3.5    4.3      4.5    12.1
## 34     1.0   6.8  5.0    6.0      5.2    14.9
## 35     0.8   5.0  5.7    5.5      4.8    13.5
```

```
## 36     0.8   3.5  4.7    4.2      3.3    12.2
## 37     0.8   4.3  5.5    3.5      5.8    10.3
## 38     0.8   5.2  4.8    5.7      3.5    13.2
```

4 - a)

Affichage du graphique de corrélations et nuages de points:

```
ggpairs(wine, lower=list(continuous=wrap("points", colour="cyan4")))
```



Les variables `Flavor` et `Aroma` possèdent les deux meilleures coefficients de corrélation par rapport à la variable réponse (0.790 et 0.707 respectivement).

Certaines variables ont des coefficients très mauvais comme `Quality` et `Clarity`. Leurs graphiques respectifs vis-à-vis de `Quality` montre cette tendance fortement non-linéaire.

4 - b)

Création d'un modèle comprenant toutes les variables explicatives:

```
wine.fit1<-lm(Quality~Clarity+Aroma+Body+Flavor+Oakiness, data=wine)
wine.fit1
```

```
##
## Call:
## lm(formula = Quality ~ Clarity + Aroma + Body + Flavor + Oakiness,
##     data = wine)
##
## Coefficients:
## (Intercept)      Clarity        Aroma         Body       Flavor     Oakiness
##      3.9969       2.3395       0.4826       0.2732       1.1683      -0.6840
```

On a l'équation $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + e$ où :

- y est la variable réponse `Quality`
- β_0 vaut environ 3.99 et correspond à `(Intercept)`
- β_1 vaut environ 2.34 et x_1 correspond à Clarity
- β_2 vaut environ 0.48 et x_2 correspond à Aroma
- β_3 vaut environ 0.27 et x_3 correspond à Body
- β_4 vaut environ 1.17 et x_4 correspond à Flavor
- β_5 vaut environ -0.68 et x_5 correspond à Oakiness
- e correspond à l'erreur

4 - c)

Affichage de la table de résumé du modèle:

```
summary(wine.fit1)
```

```
##
## Call:
## lm(formula = Quality ~ Clarity + Aroma + Body + Flavor + Oakiness,
##     data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.85552 -0.57448 -0.07092  0.67275  1.68093
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.9969     2.2318   1.791 0.082775 .
## Clarity       2.3395     1.7348   1.349 0.186958
## Aroma         0.4826     0.2724   1.771 0.086058 .
```

```
## Body          0.2732     0.3326   0.821 0.417503
## Flavor        1.1683     0.3045   3.837 0.000552 ***
## Oakiness     -0.6840     0.2712  -2.522 0.016833 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.163 on 32 degrees of freedom
## Multiple R-squared:  0.7206, Adjusted R-squared:  0.6769
## F-statistic: 16.51 on 5 and 32 DF,  p-value: 4.703e-08
```

On constate que les variable `Flavor` et `Oakiness` sont les deux seules étant significative à un niveau de signification de 5%. `Oakiness` a donc la deuxième variable la plus significative malgré le fait qu'elle est le deuxième pire coefficient de corrélation, comme vu au point **a)**.

4 - d)

Affichage du coefficient de détermination R^2:

```
summary(wine.fit1)$r.squared
```

```
## [1] 0.7205992
```

Affichage du coefficient de détermination ajusté R^2_{adj}:

```
summary(wine.fit1)$adj.r.squared
```

```
## [1] 0.6769428
```

4 - e)

Explication des différentes méthodes:

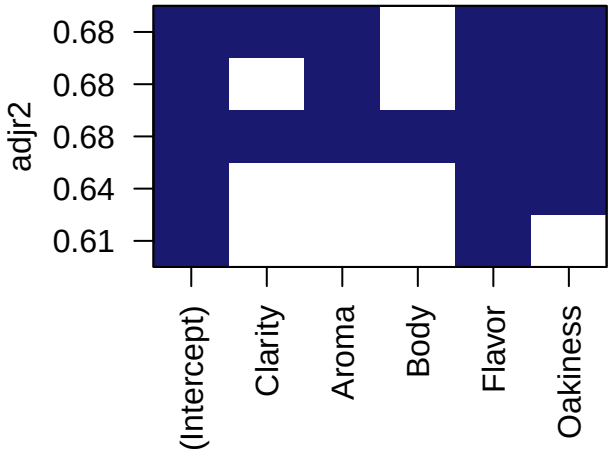
- R^2_{adj} : La valeur R^2_{adj} a pour but de "corriger" le coefficient R^2 (sa tendance croissante au fur et à mesure que l'on rajoute des variables) en prenant en compte le nombre d'observations. Il est tout le temps au plus inférieur ou égal R^2. Sa formule est $R^2_{adj} = 1 - (1 - R^2)(\frac{n-1}{n-2})$ et elle a l'avantage d'être relativement simple.
- AIC : Une méthode qui utilise le nombre de variables indépendantes ainsi qu'un estimateur du maximum de vraisemblance du modèle, et que l'on applique sur plusieurs modèles à tester ou départager. Sa formule est $AIC = 2K - 2\ln(L)$ où K est le nombre de variable indépendantes et L l'estimateur. Plus la valeur d'AIC est basse, meilleur est considéré le modèle.
- BIC : Une méthode assez similaire à AIC, mais où la correction du surnombre d'attributs est un peu plus robuste. Sa formule est $BIC = \ln(n)K - 2\ln(L)$ où n est le nombre d'observations et K et L identiques à ceux de la formule d'AIC. Sa limitation principale est que le nombre n doit être bien supérieur au nombre k.
- C_p : Une méthode encore une fois similaire au critère d'Akaike. $C_p = \frac{1}{n}(RSS + 2p\hat{\sigma}^2)$ où RSS est la somme résiduelles des carrés, p est le nombre de variables et $\hat{\sigma}^2$ une estimation de

la variance des prédictions du modèle complet. Sa principale limitation est le fait que cette méthode est réellement valide sur des échantillons de taille supérieure.

4 - f)

Application des critères R^2_{adj}, BIC et C_p:

```
choix<-regsubsets(Quality~Clarity+Aroma+Body+Flavor+Oakiness, data=wine,
                  nbest=1, nvmax=11)
plot(choix, scale="adjr2", col="midnightblue")
```




```
leaps<-regsubsets(Quality~Clarity+Aroma+Body+Flavor+Oakiness, data=wine,
                  nbest=10)
summary(leaps)
```

```
## Subset selection object
## Call: regsubsets.formula(Quality ~ Clarity + Aroma + Body + Flavor +
##     Oakiness, data = wine, nbest = 10)
## 5 Variables  (and intercept)
##          Forced in Forced out
## Clarity      FALSE      FALSE
## Aroma        FALSE      FALSE
## Body         FALSE      FALSE
## Flavor       FALSE      FALSE
## Oakiness     FALSE      FALSE
## 10 subsets of each size up to 5
## Selection Algorithm: exhaustive
##           Clarity Aroma Body Flavor Oakiness
## 1  ( 1 )  " "     " "   " "  "*"    " "
## 1  ( 2 )  " "     "*"   " "  " "    " "
## 1  ( 3 )  " "     " "   "*"  " "    " "
## 1  ( 4 )  " "     " "   " "  " "    "*"
## 1  ( 5 )  "*"     " "   " "  " "    " "
## 2  ( 1 )  " "     " "   " "  "*"    "*"
## 2  ( 2 )  " "     "*"   " "  "*"    " "
## 2  ( 3 )  "*"     " "   " "  "*"    " "
```

```
## 2  ( 4 )  " "     " "   "*"  "*"    " "
## 2  ( 5 )  " "     "*"   " "  " "    "*"
## 2  ( 6 )  " "     "*"   "*"  " "    " "
## 2  ( 7 )  "*"     "*"   " "  " "    " "
## 2  ( 8 )  "*"     " "   "*"  " "    " "
## 2  ( 9 )  " "     " "   "*"  " "    "*"
## 2  ( 10 ) "*"     " "   " "  " "    "*"
## 3  ( 1 )  " "     "*"   " "  "*"    "*"
## 3  ( 2 )  "*"     " "   " "  "*"    "*"
## 3  ( 3 )  " "     " "   "*"  "*"    "*"
## 3  ( 4 )  "*"     "*"   " "  "*"    " "
## 3  ( 5 )  " "     "*"   "*"  "*"    " "
## 3  ( 6 )  "*"     " "   "*"  "*"    " "
## 3  ( 7 )  " "     "*"   "*"  " "    "*"
## 3  ( 8 )  "*"     "*"   "*"  " "    " "
## 3  ( 9 )  "*"     "*"   " "  " "    "*"
## 3  ( 10 ) "*"     " "   "*"  " "    "*"
## 4  ( 1 )  "*"     "*"   " "  "*"    "*"
## 4  ( 2 )  " "     "*"   "*"  "*"    "*"
## 4  ( 3 )  "*"     " "   "*"  "*"    "*"
## 4  ( 4 )  "*"     "*"   "*"  "*"    " "
## 4  ( 5 )  "*"     "*"   "*"  " "    "*"
## 5  ( 1 )  "*"     "*"   "*"  "*"    "*"
```

On constate que, par exemple, le meilleur modèle avec trois variables explicatives est celui comprenant `Aroma`, `Flavor` et `Oakiness`. Les qualités de ces variables déjà pu être appréciée dans les points précédents.

Ajustement du nouveau modèle

On va donc utiliser ces deux variables pour l'ajustement d'un nouveau modèle plus simple `wine.fit2`:

```
wine.fit2<-lm(Quality~Aroma+Flavor+Oakiness, data=wine)
wine.fit2
```

```
##
## Call:
## lm(formula = Quality ~ Aroma + Flavor + Oakiness, data = wine)
##
## Coefficients:
## (Intercept)        Aroma       Flavor     Oakiness
##      6.4672       0.5801       1.1997      -0.6023
```

Résumé

Table de résumé du nouveau modèle:

```
summary(wine.fit2)
```

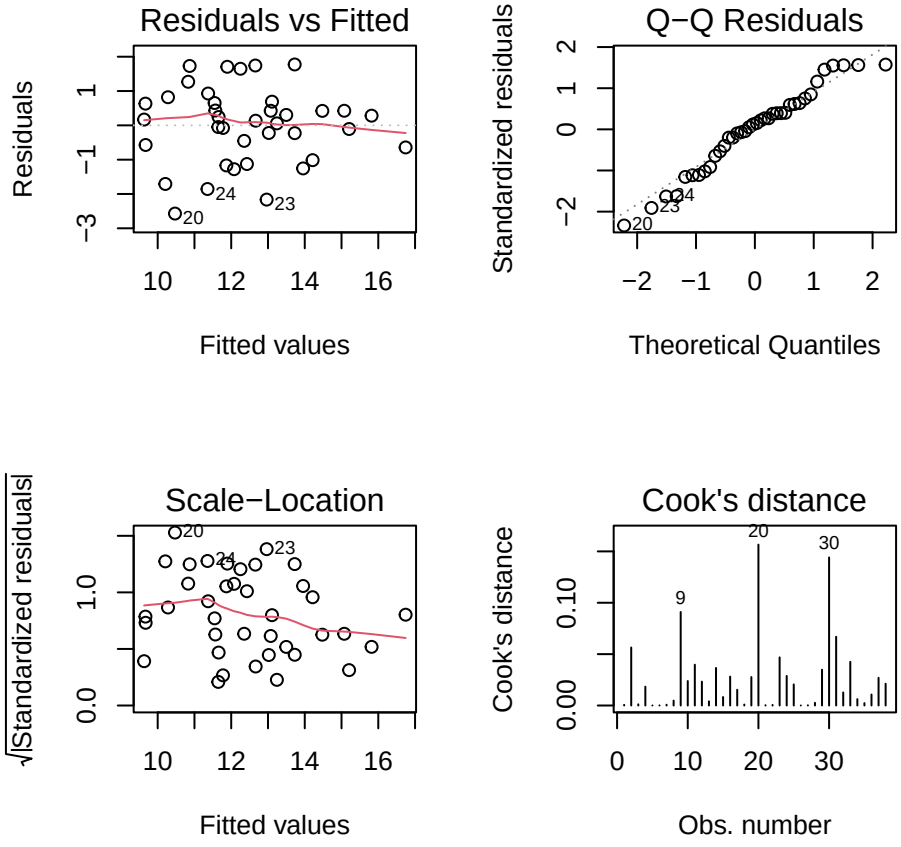
```
##
```

```
## Call:
## lm(formula = Quality ~ Aroma + Flavor + Oakiness, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5707 -0.6256  0.1521  0.6467  1.7741
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.4672     1.3328   4.852 2.67e-05 ***
## Aroma         0.5801     0.2622   2.213 0.033740 *
## Flavor        1.1997     0.2749   4.364 0.000113 ***
## Oakiness     -0.6023     0.2644  -2.278 0.029127 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.161 on 34 degrees of freedom
## Multiple R-squared:  0.7038, Adjusted R-squared:  0.6776
## F-statistic: 26.92 on 3 and 34 DF,  p-value: 4.203e-09
```

Vérification des hypothèses

Affichage des 4 graphiques aidant à la vérification des hypothèses inhérentes au modèle:

```
par(mfrow=c(2, 2))
plot(wine.fit2, which=c(1:4))
```



Nuage de points des résidus (`Residuals vs Fitted`)

Il n'y a pas vraiment de tendance qui se dessine et la variance est constante. Il semble adéquat de valider l'hypothèse de l'espérance des erreurs nulle.

Nuage de points de la racine carrée de la valeur absolue des résidus contre les valeurs ajustées (`Scale-Location`)

La variance des résidus semblent également assez constante le long de la tendance, bien qu'un peu moins aux extrémités. L'hypothèse de variance constante peut être validée.

Graphique des résidus observés standardisés contre les résidus théoriques (`Q-Q Residuals`)

Les points restent en général proche de la droite, un peu moins aux extrémités. On peut tout de même accepter l'hypothèse de la normalité des erreurs.

Graphique représentant la distance de Cook (`Cook's distance`)

Les points **9**, **20** et **30** ont la plus grande distance de Cook.

Globalement, les 3 hypothèses ont pu être validés même si le modèle n'est pas idéal.

Performances

Comparaison des performances, avec affichage du score de performance:

```
compare_performance(wine.fit1, wine.fit2, rank=T)
```

```
## # Comparison of Model Performance Indices
##
## Name      | Model |    R2 | R2 (adj.) |  RMSE | Sigma | AIC weights | AICc weights | BI
## ----------------------------------------------------------------------------------------
## wine.fit2 |    lm | 0.704 |     0.678 | 1.098 | 1.161 |       0.709 |        0.860 |
## wine.fit1 |    lm | 0.721 |     0.677 | 1.067 | 1.163 |       0.291 |        0.140 |
```

On constate que le premier modèle possède des scores R^2 et R^2_{adj} supérieures au second, ce qui est attendu étant donné la nature de cette méthode à augmenter sa valeur lorsque le nombre de variables explicatives augmente lui aussi. Cependant, le score de performance global fournit par la fonction montre que le second modèle est meilleur en ce sens.

4 - g)

Prédiction de la qualité d'un nouveau vin avec le second modèle:

```
xnew <- matrix(c(7.7, 6.7, 3.7), nrow=1)
colnames(xnew) <- c("Aroma", "Flavor", "Oakiness")
xnew <- as.data.frame(xnew)
predict(wine.fit2, xnew, interval="pred")
```

```
##        fit      lwr      upr
## 1 16.74346 14.09382 19.39311
```

La qualité vaut environ 16.74

Exercice 5

Cet exercice à pour but d'expliquer la méthode des arbres de régression.

Les arbres de régression sont une méthode d'apprentissage supervisé issue des arbres de décision ayant pour but d'effectuer de la prédiction de valeurs continues. La méthode consiste à utiliser un arbre où chaque sommet correspond à un test logique sur une certaine variable explicative, et où le résultat permet de descendre vers d'autres sommet de test puis jusqu'à un sommet final comprenant la valeur prédite par le modèle.

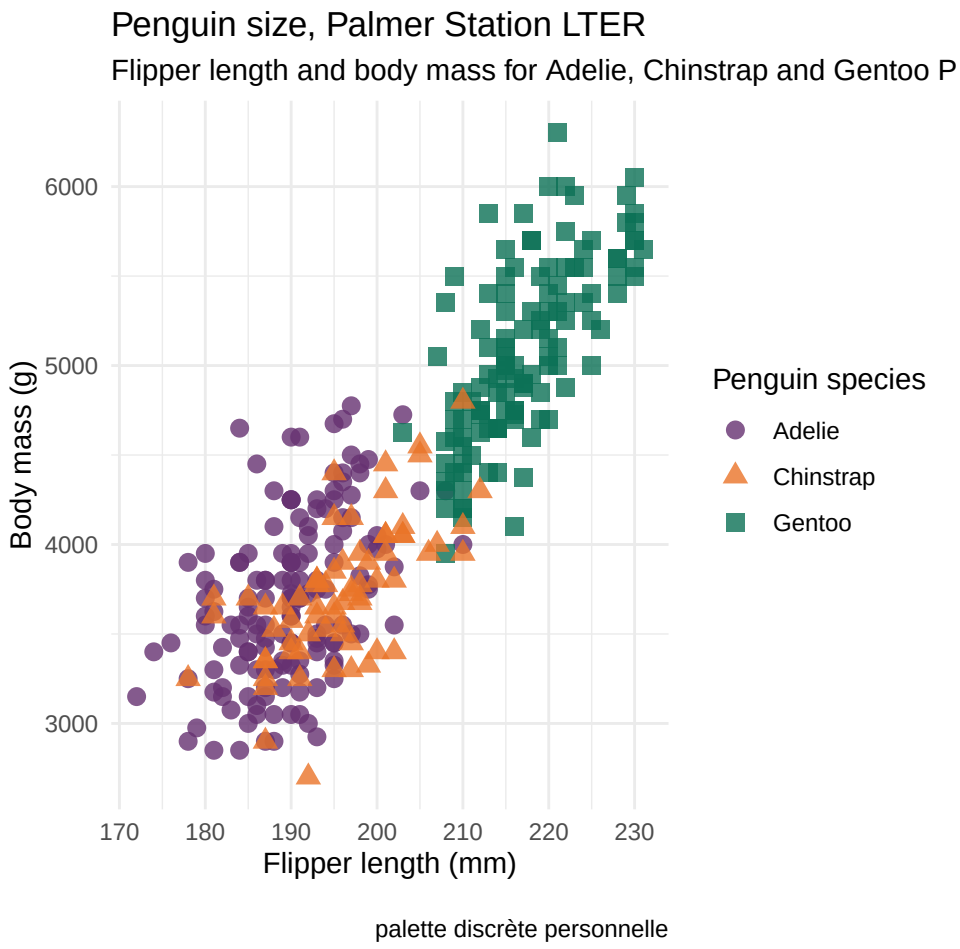
Exercice 6

Cet exercice à pour but de se familiariser avec différentes palettes et polices applicable à un graphique en **R**.

6 - a)

Affichage du premier graphique:

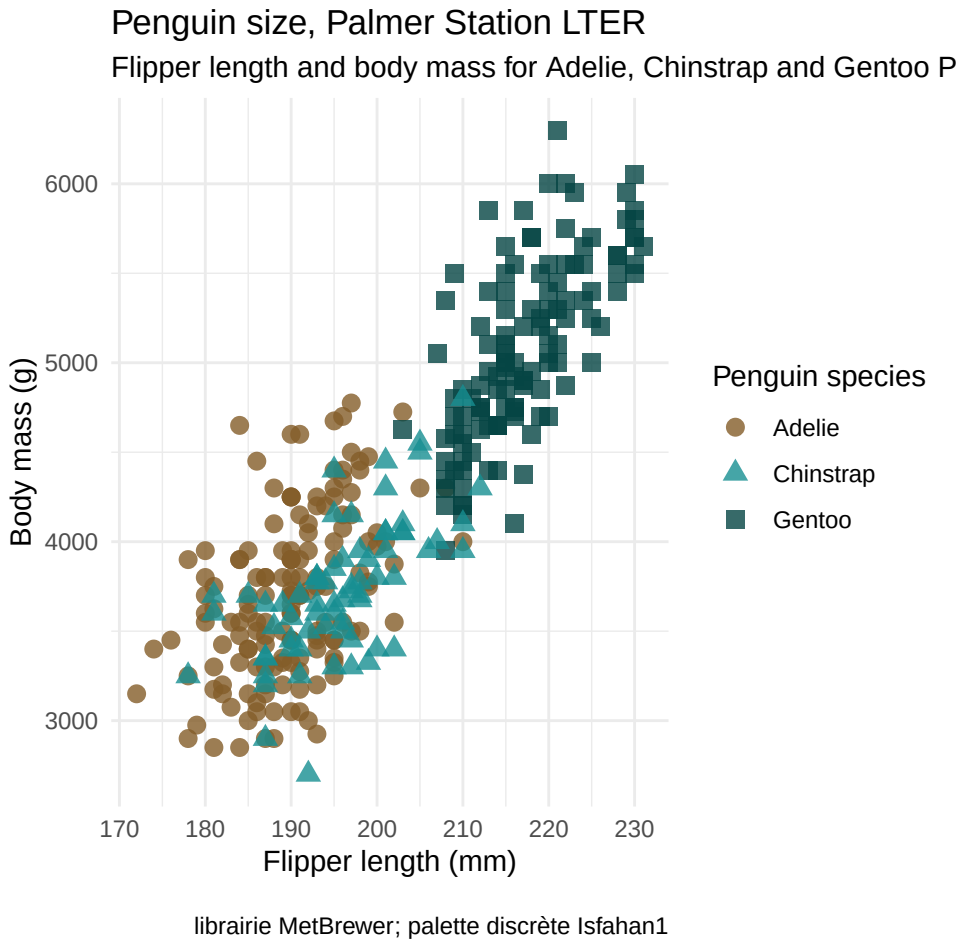
```
library(tidyverse)
library(palmerpenguins)
library(palettes)
discrete_pal <- pal_colour(c("#663171", "#EA7428", "#0C7156"))
ggplot(data=penguins, aes(x=flipper_length_mm, y=body_mass_g)) +
geom_point(aes(color=species,
shape=species),
size=3,
alpha=0.8) +
scale_colour_palette_d(discrete_pal) +
labs(title = "Penguin size, Palmer Station LTER",
subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins",
x = "Flipper length (mm)",
y = "Body mass (g)",
color = "Penguin species",
shape = "Penguin species",
caption = "\n palette discrète personnelle") +
theme_minimal()
```


Penguin size, Palmer Station LTER
Flipper length and body mass for Adelie, Chinstrap and Gentoo P
Body mass (g)
6000
5000
4000
3000
170
180
190
200
210
220
230
Flipper length (mm)
Penguin species
Adelie
Chinstrap
Gentoo
palette discrète personnelle

6 - b)

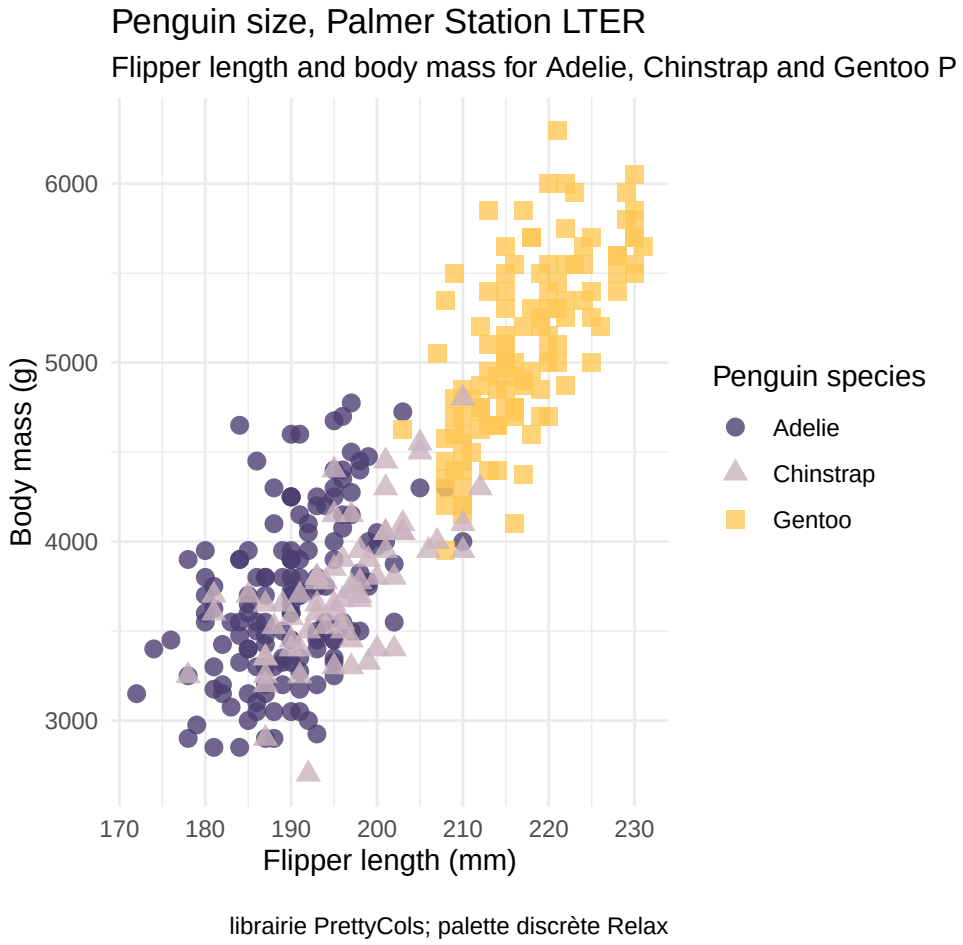
Affichage du second graphique, avec la palette `Isfahan1` de `MetBrewer`:

```
ggplot(data=penguins, aes(x=flipper_length_mm, y=body_mass_g)) +
geom_point(aes(color=species,
shape=species),
size=3,
alpha=0.8) +
scale_colour_met_d(name="Isfahan1") +
labs(title = "Penguin size, Palmer Station LTER",
subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins",
x = "Flipper length (mm)",
y = "Body mass (g)",
color = "Penguin species",
shape = "Penguin species",
caption = "\n librairie MetBrewer; palette discrète Isfahan1") +
theme_minimal()
```



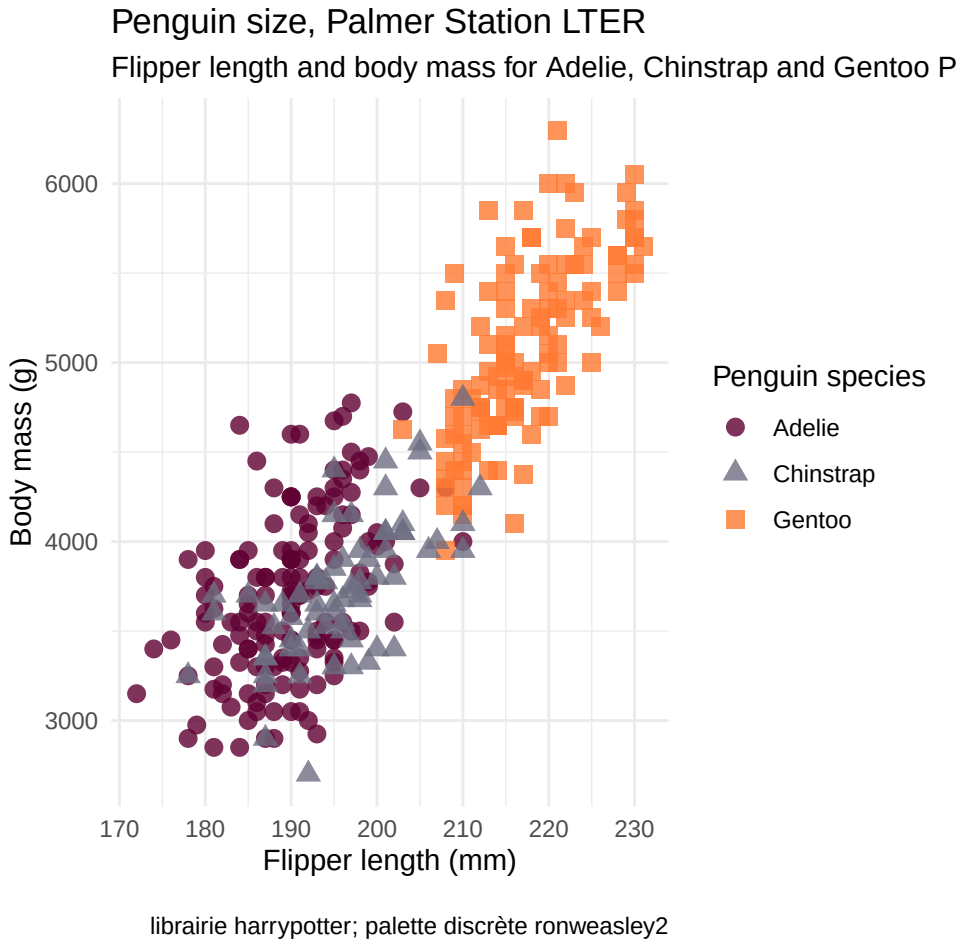
Affichage du troisième graphique, avec la palette `Relax` par `PrettyCols`:

```
ggplot(data=penguins, aes(x=flipper_length_mm, y=body_mass_g)) +
geom_point(aes(color=species,
shape=species),
size=3,
alpha=0.8) +
scale_colour_pretty_d(name="Relax") +
labs(title = "Penguin size, Palmer Station LTER",
subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins",
x = "Flipper length (mm)",
y = "Body mass (g)",
color = "Penguin species",
shape = "Penguin species",
caption = "\n librairie PrettyCols; palette discrète Relax") +
theme_minimal()
```



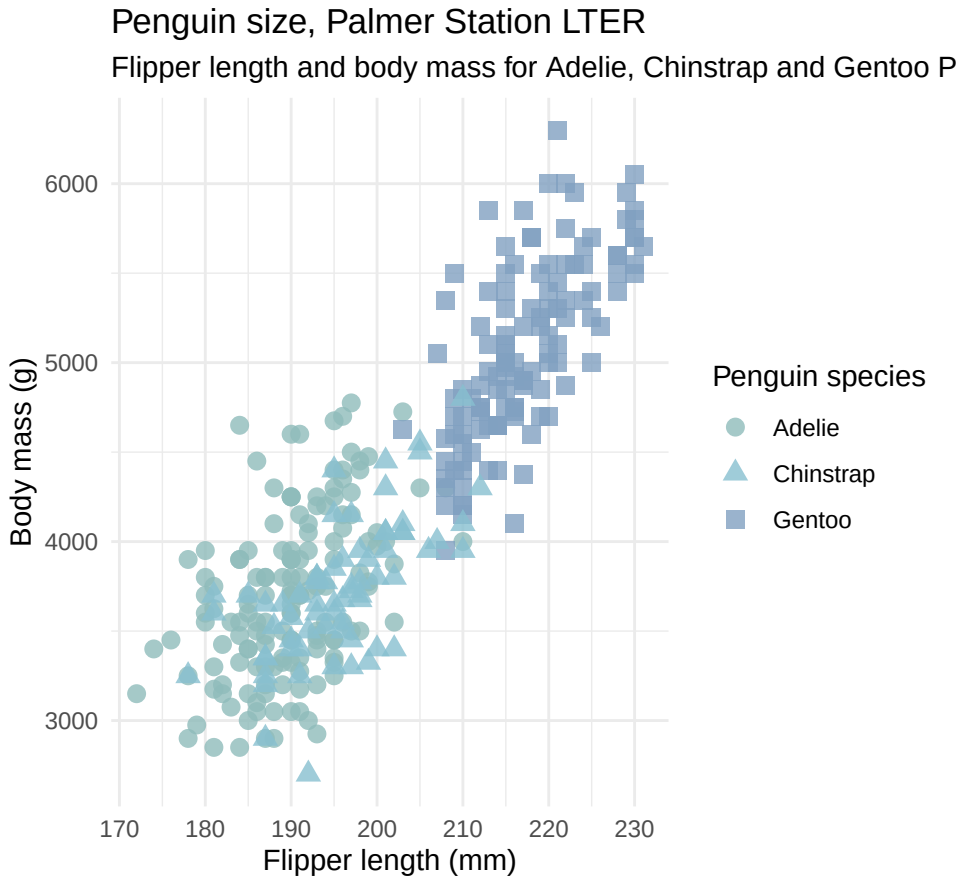
Affichage du quatrième graphique avec la palette `ronweasley2` de `harrypotter`:

```
ggplot(data=penguins, aes(x=flipper_length_mm, y=body_mass_g)) +
geom_point(aes(color=species,
shape=species),
size=3,
alpha=0.8) +
scale_colour_hp_d("ronweasley2") +
labs(title = "Penguin size, Palmer Station LTER",
subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins",
x = "Flipper length (mm)",
y = "Body mass (g)",
color = "Penguin species",
shape = "Penguin species",
caption = "\n librairie harrypotter; palette discrète ronweasley2") +
theme_minimal()
```



Affichage du cinquième graphique avec la palette `nord::frost` de `paletteer`:

```
ggplot(data=penguins, aes(x=flipper_length_mm, y=body_mass_g)) +
geom_point(aes(color=species,
shape=species),
size=3,
alpha=0.8) +
scale_colour_paletteer_d("nord::frost") +
labs(title = "Penguin size, Palmer Station LTER",
subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins",
x = "Flipper length (mm)",
y = "Body mass (g)",
color = "Penguin species",
shape = "Penguin species",
caption = "\n librairie paletteer; palette discrète frost de la librairie nord") +
theme_minimal()
```



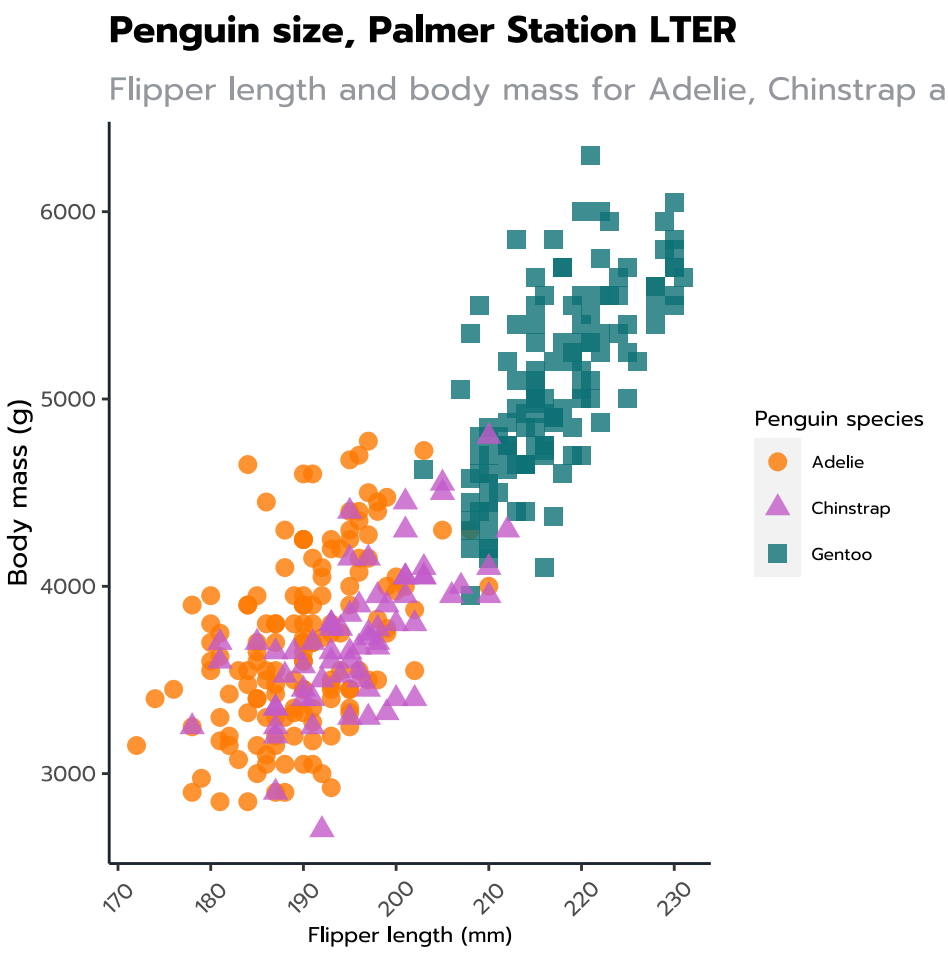
librairie paletteer; palette discrète frost de la librairie nord

6 - c)

Affichage d'un graphique en utilisant `magrittr` entre autres.

```
font_add_google(name = "Prompt", family = "prompt")
showtext_auto()
penguin_palette <- list("Adelie" = "#fd7901",
"Chinstrap" = "#c35bca",
"Gentoo" = "#0e7175",
"dark_text" = "#1A242F",
"light_text" = "#94989D")

penguins %>%
filter(!is.na(flipper_length_mm), !is.na(body_mass_g)) %>%
ggplot(aes(x=flipper_length_mm, y=body_mass_g)) +
geom_point(aes(color=species, shape=species), size=3, alpha=0.8) +
scale_colour_manual(values=penguin_palette) +
labs(title = "Penguin size, Palmer Station LTER",
subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins",
x = "Flipper length (mm)",
y = "Body mass (g)",
color = "Penguin species",
shape = "Penguin species") +
theme(
panel.grid = element_blank(),
panel.background = element_rect(fill="white"),
text = element_text(family="prompt", size=8),
plot.title = element_text(size=14, face="bold", margin=margin(b=10), hjust=0),
plot.subtitle = element_text(size=12, colour = penguin_palette$light_text),
axis.title.y = element_text(size = 10),
axis.text = element_text(size = 8),
axis.text.x = element_text(angle = 45, vjust = 0.5),
axis.line.x = element_line(colour = penguin_palette$dark_text, linewidth = 0.5,
linetype = "solid"),
axis.line.y = element_line(colour = penguin_palette$dark_text, linewidth = 0.5,
linetype = "solid"),
panel.border = element_blank()
)
```

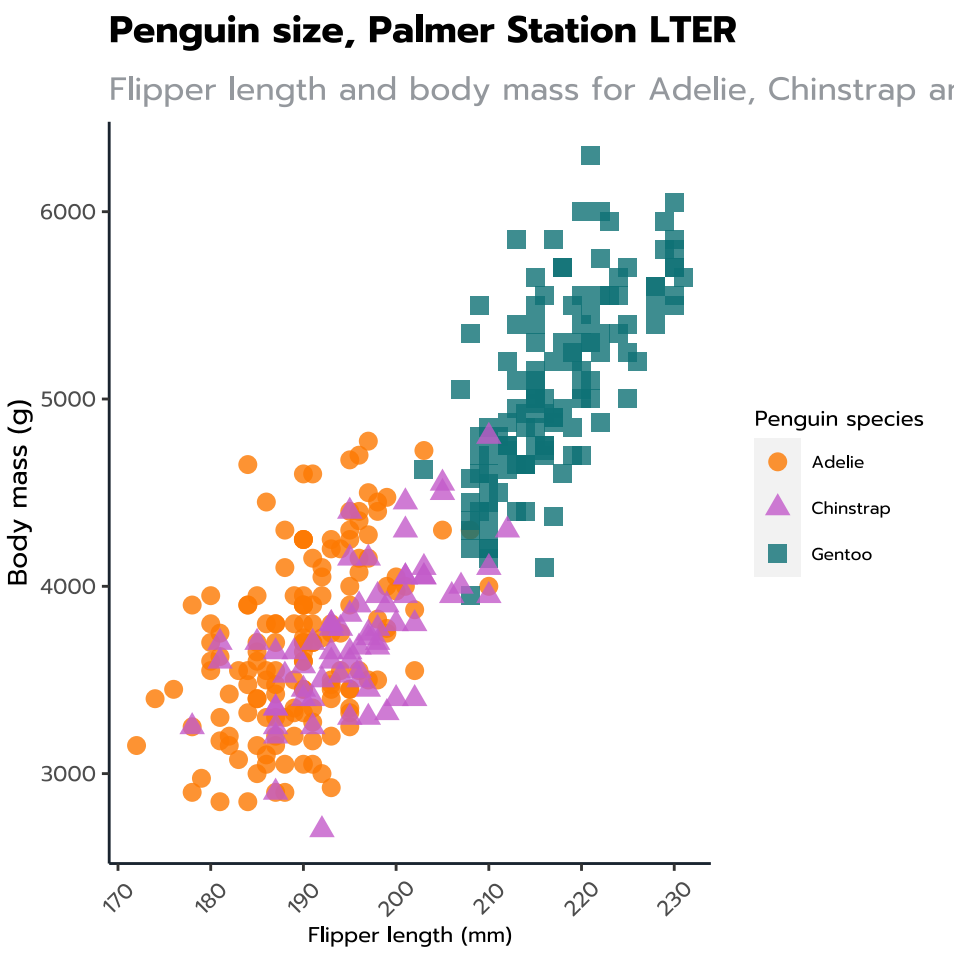

Penguin size, Palmer Station LTER
Flipper length and body mass for Adelie, Chinstrap ar
Body mass (g)
6000
5000
4000
3000
170
180
190
200
210
220
230
Flipper length (mm)
Penguin species
Adelie
Chinstrap
Gentoo

6 - d)

Affichage d'un graphique avec la police `Henny Penny` (*Note*: RStudio ne semble pas générer la bonne police, malgré le code ci-dessous qui semble être correct):

```
font_add_google(name = "Henny Penny", family = "henny")
showtext_auto()
penguin_palette <- list("Adelie" = "#fd7901",
"Chinstrap" = "#c35bca",
"Gentoo" = "#0e7175",
"dark_text" = "#1A242F",
"light_text" = "#94989D")

penguins %>%
filter(!is.na(flipper_length_mm), !is.na(body_mass_g)) %>%
ggplot(aes(x=flipper_length_mm, y=body_mass_g)) +
geom_point(aes(color=species, shape=species), size=3, alpha=0.8) +
scale_colour_manual(values=penguin_palette) +
labs(title = "Penguin size, Palmer Station LTER",
subtitle = "Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins",
x = "Flipper length (mm)",
y = "Body mass (g)",
color = "Penguin species",
shape = "Penguin species") +
theme(
panel.grid = element_blank(),
panel.background = element_rect(fill="white"),
text = element_text(family="prompt", size=8),
plot.title = element_text(size=14, face="bold", margin=margin(b=10), hjust=0),
plot.subtitle = element_text(size=12, colour = penguin_palette$light_text),
axis.title.y = element_text(size = 10),
axis.text = element_text(size = 8),
axis.text.x = element_text(angle = 45, vjust = 0.5),
axis.line.x = element_line(colour = penguin_palette$dark_text, linewidth = 0.5,
linetype = "solid"),
axis.line.y = element_line(colour = penguin_palette$dark_text, linewidth = 0.5,
linetype = "solid"),
panel.border = element_blank()
)
```



6 - e)

La commande suivante:

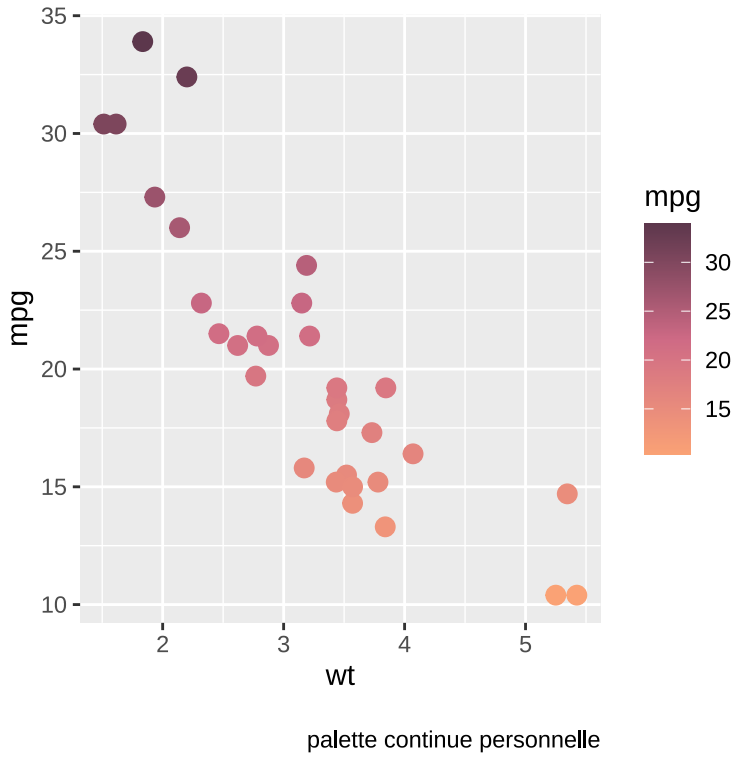
```
... %>%
filter(!is.na(flipper_length_mm), !is.na(body_mass_g)) %>%
...
```

Assure un filtrage des données qui enlève les données manquantes dans `flipper_length_mm` et `body_mass_g`.

6 - f)

Affichage d'un graphique avec palette personnelle continue:

```
continuous_pal <- pal_colour(c("#FAA275", "#CE6A85", "#5C374C"))
ggplot(mtcars, aes(wt, mpg, colour = mpg)) +
geom_point(size = 3) +
scale_colour_palette_c(continuous_pal) +
labs(
caption = "\n palette continue personnelle"
)
```



Conclusion

Ce travail pratique a permis d'approfondir et d'appliquer les connaissances abordées lors du cours, ainsi que de se familiariser encore plus avec le logiciel **R** et ses différentes librairies tierces.