TP OUMOBIO 8: Modéliser une relation linéaire par une régression

Mathieu Brevet

2025-01-14

Bienvenue dans ce huitième TP sur R. Après avoir vu comment décrire graphiquement et statistiquement la relation entre deux variables quantitatives, nous allons maintenant apprendre à modéliser une relation linéaire, à l'aide d'une régression linéaire.

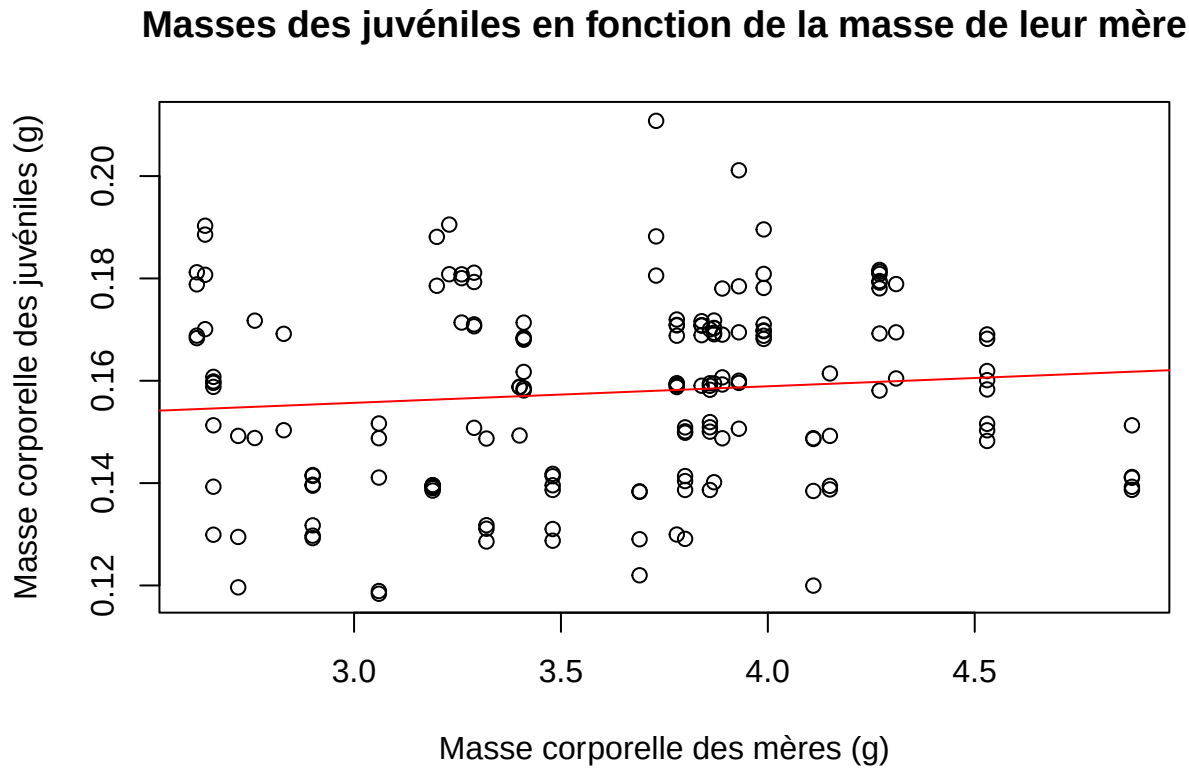
```
setwd("~/ATER PAU 2024/Cours modifiés/OUMOBIOS")


data_lezard = read.table("Suivi_lezard_vivipare.csv", header = T, sep = "\t", dec = ",")
```

Régression linéaire par la méthode des moindres carrés

La **régression linéaire** permet de modéliser une variable quantitative continue par une **relation affine** avec une autre variable (de la forme Y = aX + b, avec Y la **variable réponse** et X la **variable explicative**). Il faut donc vérifier en amont que la relation entre variable semble bien linéaire. Les paramètres de la régression linéaire (coefficient directeur "a" et ordonnée à l'origine "b") sont déterminés à l'aide de la méthode des moindres carrés: on cherche à minimiser la somme des carrés des écarts entre les points observés et la droite de régression (voir cours OUMOBIO pour détails), c'est à dire le carré des **résidus** de la régression.
Dans R l'estimation des paramètres de régression et les métriques statistiques associées peuvent être obtenues avec la fonction **lm(Var1 ~ Var2)**, voici quelques exemples d'applications:

```
# on modélise la régression linéaire entre la masse des juvéniles et des mères:


reg_poids_juv_meres = lm(data_lezard$M_IND ~ data_lezard$M_MOTHERS)
reg_poids_juv_meres = lm(M_IND ~ M_MOTHERS, data = data_lezard)  # alternative



reg_poids_juv_meres$coefficients
coefficients(reg_poids_juv_meres)
# coefficients de la régression ('intercept': ordonnée à l'origine / seconde
# mesure: coefficient directeur)

reg_poids_juv_meres$residuals
residuals(reg_poids_juv_meres)
# résidus de la régression (écarts entre les points observés et la droite de
# régression)

reg_poids_juv_meres$fitted.values
fitted.values(reg_poids_juv_meres)
```

```r
# valeurs des poids des juvéniles (variable réponse) estimés à partir de
# l'équation de régression pour chaque valeurs observés de poids des mères
# (variable explicative) => valeurs prédites


predict(reg_poids_juv_meres, newdata = data.frame(M_MOTHERS = seq(2, 5, 0.1)))
# valeurs prédites de la variable réponse, en fonction d'un ensemble de valeur
# donnée pour la variable explicative



# représentation graphique:

plot(
    jitter(data_lezard$M_IND, 1) ~
        data_lezard$M_MOTHERS, main = "Masses des juvéniles en fonction de la masse de leur mère",
    xlab = "Masse corporelle des mères (g)", ylab = "Masse corporelle des juvéniles (g)"
)

abline(reg_poids_juv_meres, col = "red")
```




```r
# ajout de la droite de régression sur le graphique précédent (fonction
# 'abline()')
```

```r
### Estimation empirique des coefficients de la courbe de régression:


# On met en place un code permettant de chercher les valeurs a et b qui vont
# minimiser les ecarts entre la droite et les points.

SumEc <- function(Y, Ypred)
    {

    ec <- sum((Y - Ypred)^2)

    return(ec)
}



# On cherche a minimiser cette valeur:

# Generation d'un vecteur de valeurs possibles dans lequel on va venir piocher
# des valeurs de a et b.
Valpos <- seq(-0.5, 0.5, 0.001)

# On déclare nos variables (pour l'instant non attribuées (NA), qui vont ?tre
# rempli plus tard)

Va <- NA  #Valeur de a
Vb <- NA  #Valeur de b
Vec <- NA  #Ecart moyen obtenu avec les a et b selectionnés

# Creation d'une boucle FOR
for (i in 1:1e+05)
{

    a <- sample(Valpos, 1)
    b <- sample(Valpos, 1)

    Ypred <- a * data_lezard$M_MOTHERS + b

    Va[i] <- a
    Vb[i] <- b

    Vec[i] <- SumEc(data_lezard$M_IND, Ypred)

}

# Matrice de résultats
mat <- cbind(Va, Vb, Vec)
```

```r
# Quelle résultat parmi les 100 000 générés à le Vec le plus faible ?

which.min(Vec)

mat[which.min(Vec),
    c(1, 2)]  #Affichage de la ligne




# Comparaison avec l'estimation théorique vue en cours:

a <- cov(data_lezard$M_IND, data_lezard$M_MOTHERS)/var(data_lezard$M_MOTHERS)
b <- mean(data_lezard$M_IND) -
    a * mean(data_lezard$M_MOTHERS)
# assez proche des estimations empirique et identique aux sorties de 'lm()'
```

EXERCICE

- Modélisez la régression linéaire entre la taille des juvéniles et leur poids. En faire une représentation graphique.
- Retrouvez les résidus de cette régression sans utiliser les fonctions permettant de les obtenir directement.

```r
 plot(
  jitter(data_lezard$M_IND, 1) ~ jitter(data_lezard$SVL_IND, 1),
  main = "Masses des juvéniles en fonction de leur taille",
  xlab = "Taille des juvéniles (mm)",
  ylab = "Masse corporelle des juvéniles (g)")

# la relation semble (grossièrement) linéaire: on peut donc essayer de la modéliser
# par une régression linéaire


reg_poids_taille_juv = lm(M_IND ~ SVL_IND, data = data_lezard)


 abline(reg_poids_taille_juv, col = "red")
```

```r
 data_lezard$M_IND - reg_poids_taille_juv$fitted.values
```

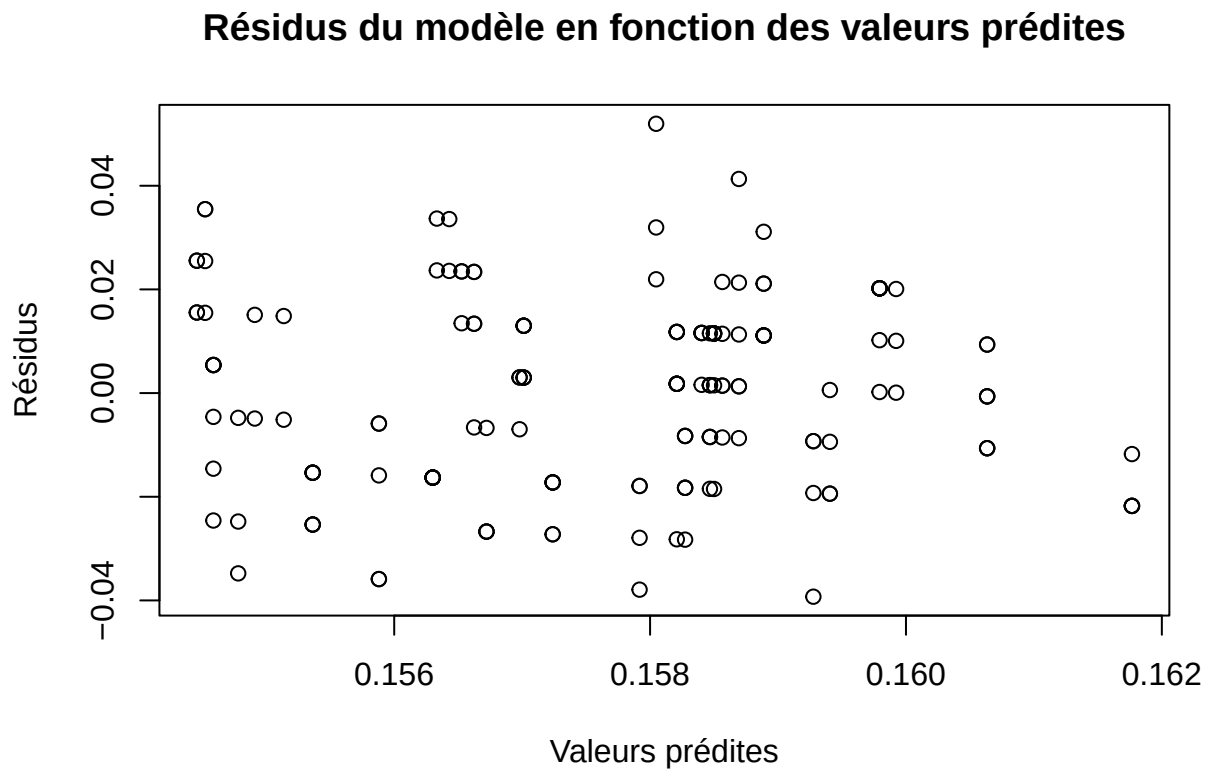
POUR ALLER PLUS LOIN

Il est possible de normaliser la variable réponse en fonction d'une autre variable dans la fonction "lm()". Il suffit d'indiquer la variable permettant la mise à l'échelle en utilisant l'argument "offset". Par exemple, "lm(data_lezardSVL_IND $data_lezard$M_IND, offset = data_lezardM_IND)"$est équivalent à"lm(data_lezard$Corp_IND ~ data_lezard$M_IND)", puisqu'on normalise la taille par le poids dans les deux cas. Cet option est notamment très utile lorsque la variable réponse est associée à différent contexte donnat un poids plus ou moins important aux mesures (par exemple différents temps d'échantillonnage, d'exposition...).

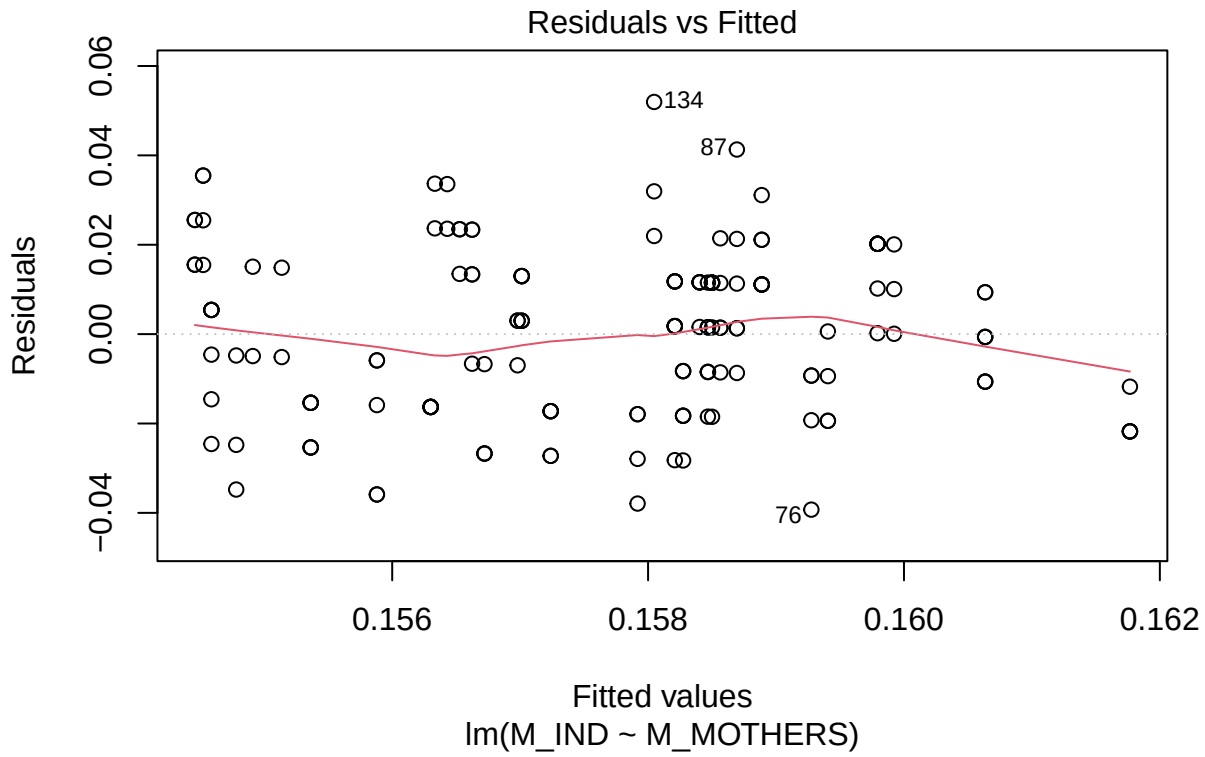
Validité du modèle

Lorsqu'on produit un modèle dans R, il convient toujours de vérifier si le modèle est bien en adéquation avec les données, et donc de vérifier si le modèle produit respecte bien les **prérequis d'une régression linéaire**: en particulier les termes d'erreurs (**résidus**) doivent conserver une **variance constante** (homoscedasticité) et ne doivent **pas varier** en fonction de la variable réponse (marqueur d'une relation non linéaire). On peut aussi vérifier que la distribution des résidus est **normale** (cette hypothèse n'est pas indispensable pour une régression linéaire univariée mais assure une bonne qualité de modèle toutefois) et identifier la présence de **points d'influence** forts, qui pourraient fausser les résultats de la régression (comme par exemple avec d'éventuelles mesures aberrantes).
Le meilleur moyen de vérifier les deux premières hypothèses est d'utiliser un graphique représentant les résidus en fonction des valeurs prédites. Pour vérifier la normalité des résidus on peut utiliser un diagramme quantile-quantile. Pour identifier les points d'influence on utilise un graphique représentant les résidus en fonction des effet de leviers.

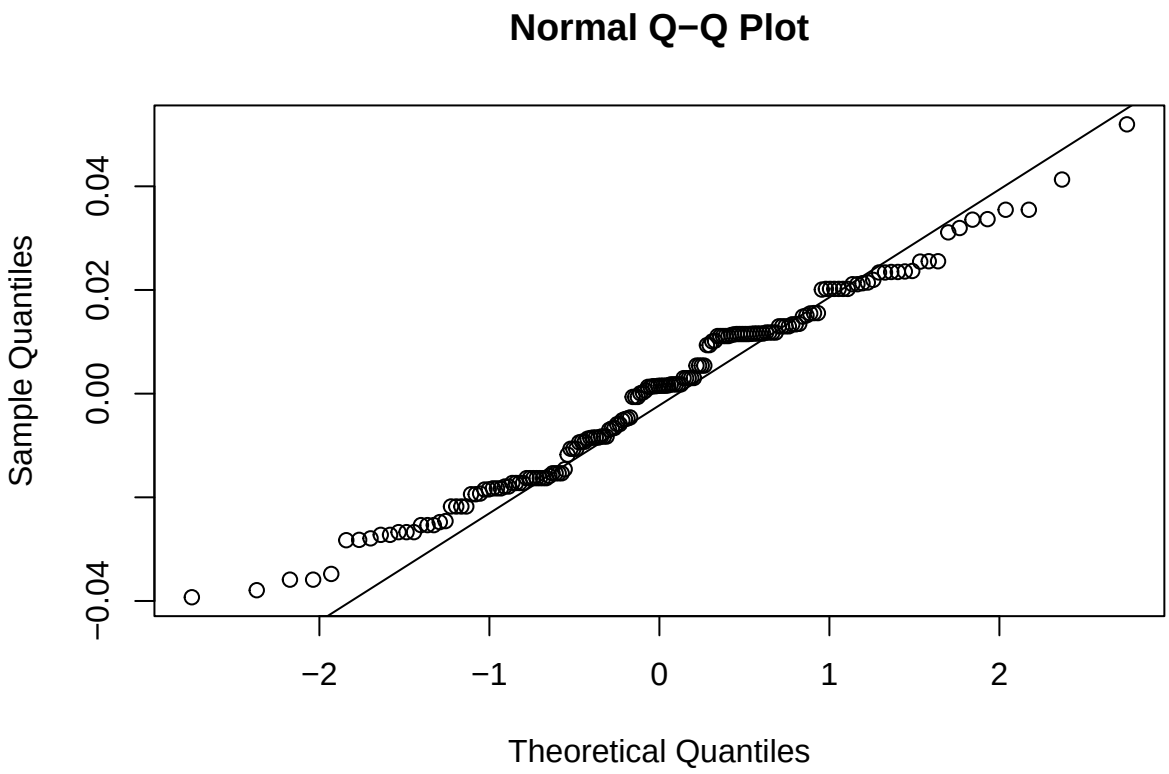
```
plot(
    residuals(reg_poids_juv_meres) ~
        fitted.values(reg_poids_juv_meres),
    main = "Résidus du modèle en fonction des valeurs prédites", xlab = "Valeurs prédites",
    ylab = "Résidus"
)
```



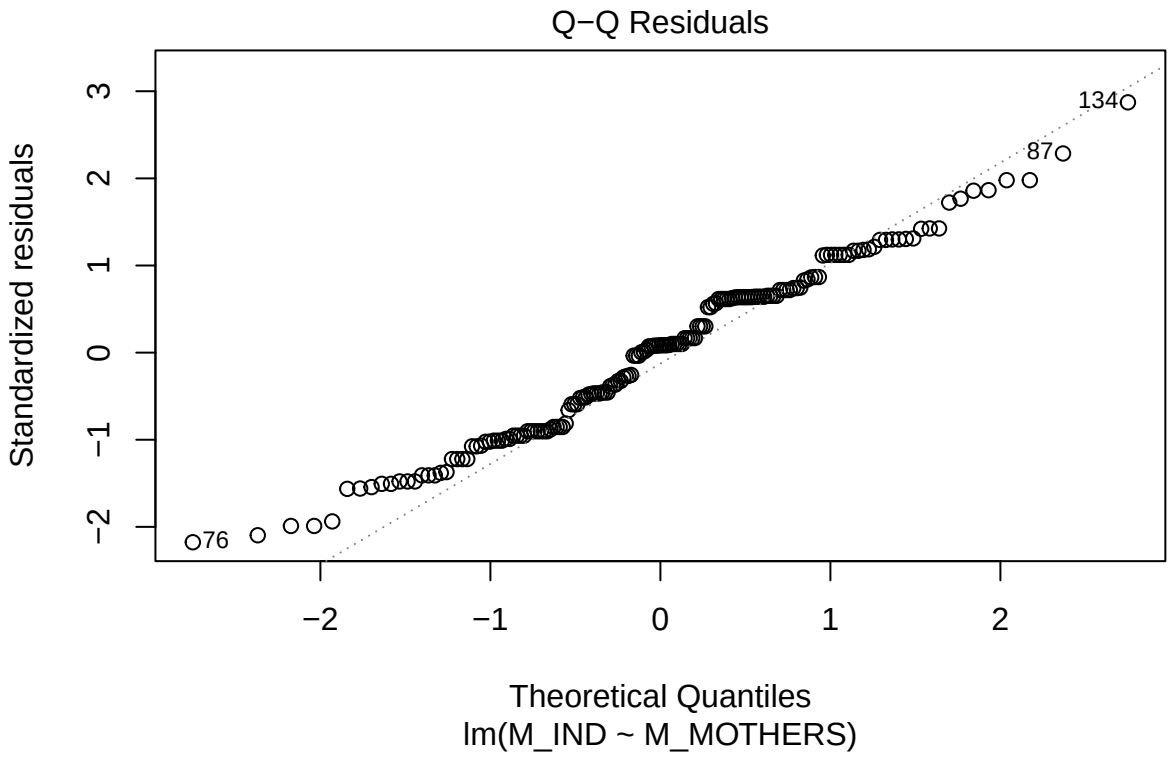

```
plot(reg_poids_juv_meres, 1)
```



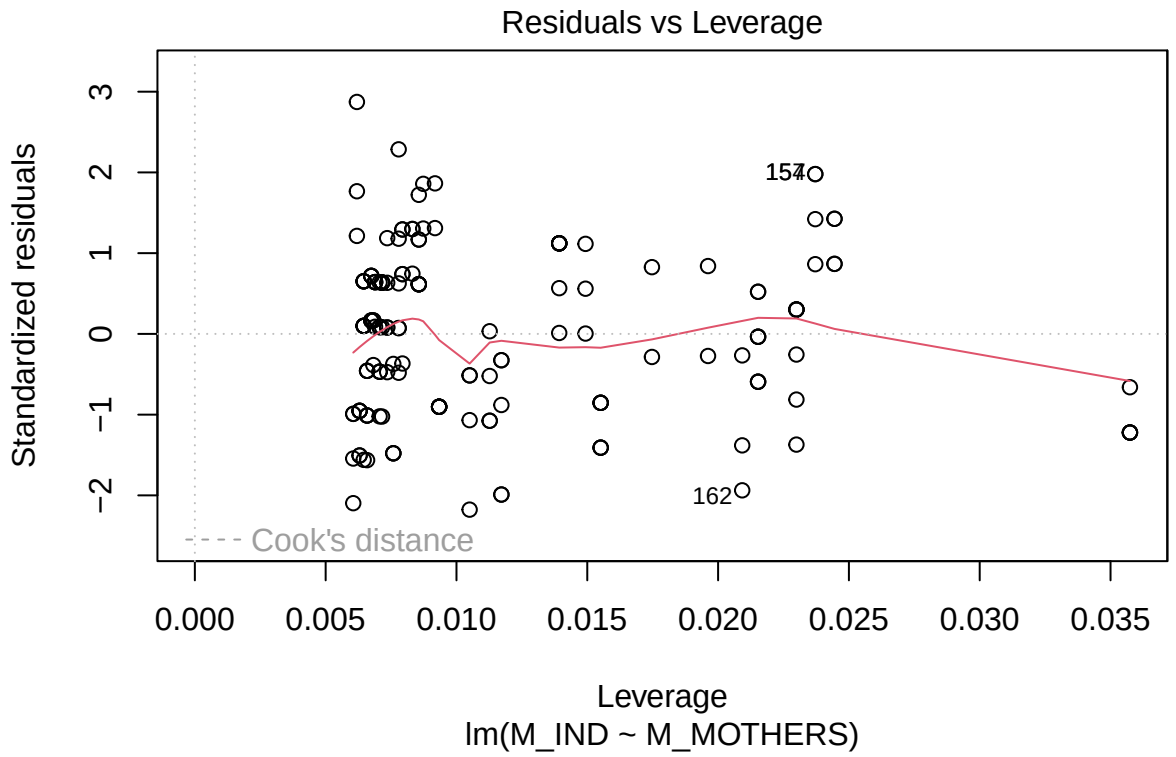

```
# alternative: la ligne rouge indique l'évolution moyenne des résidus en
# fonction des valeurs prédites on vérifie grâce à ce graphique
# l'homoscedasticité et la linéarité des résidus: les résidus doivent être
# distribués de manière homogène de part et d'autre de 0, selon un nuage de
# point homogène (ligne rouge horizontale en 0 sur lme second graphe)


# pas de problème de non linéarité ou de variabilité changeante des résidus ici


qqnorm(resid(reg_poids_juv_meres))
qqline(resid(reg_poids_juv_meres))
```



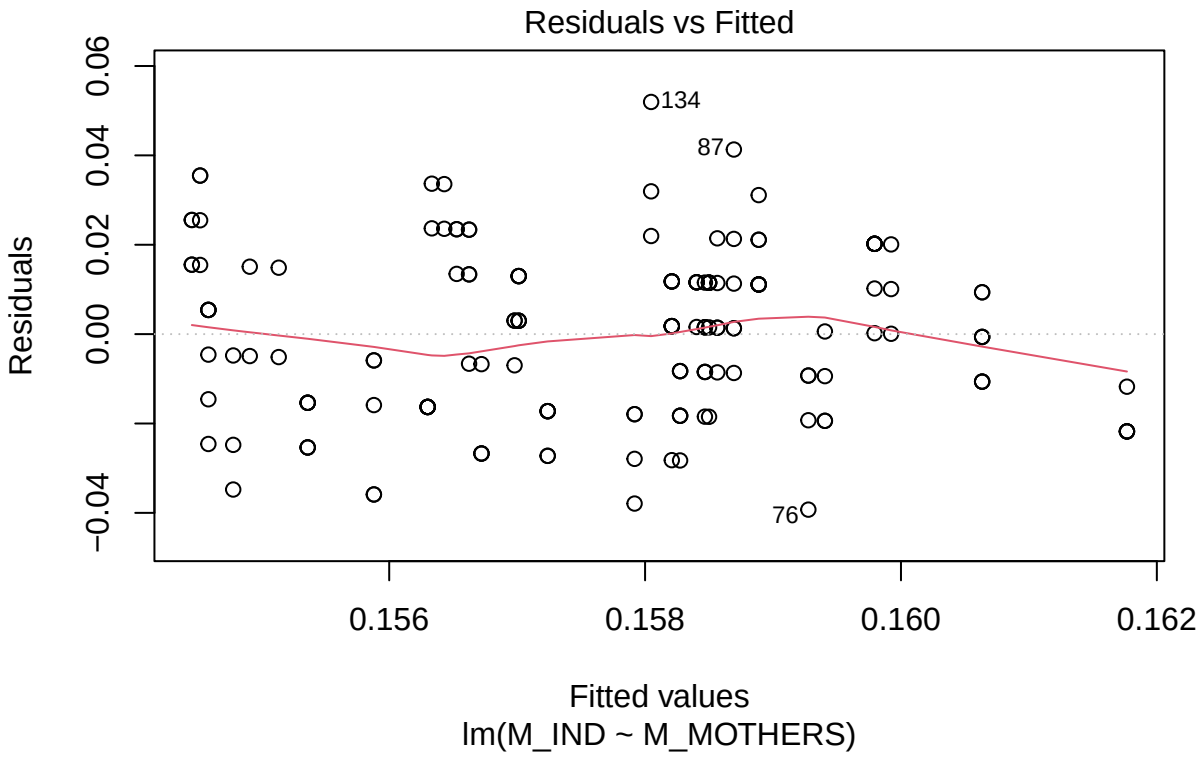

```
plot(reg_poids_juv_meres, 2)  # alternative
```

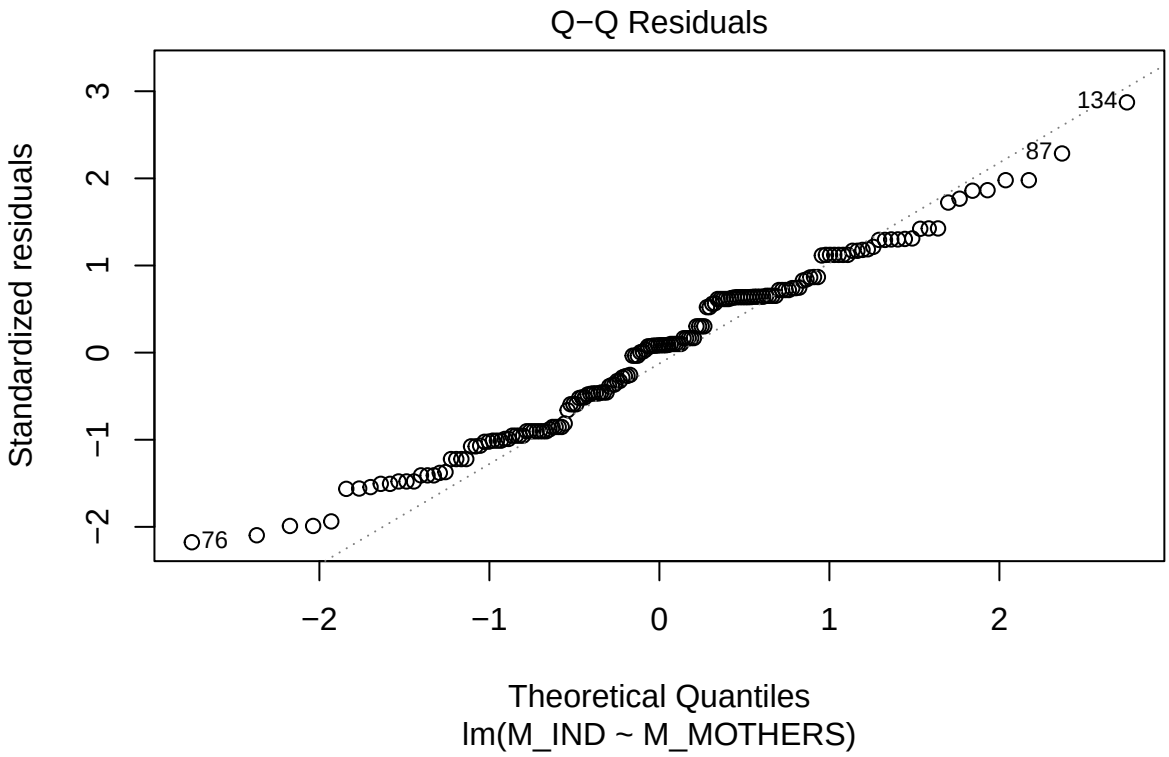


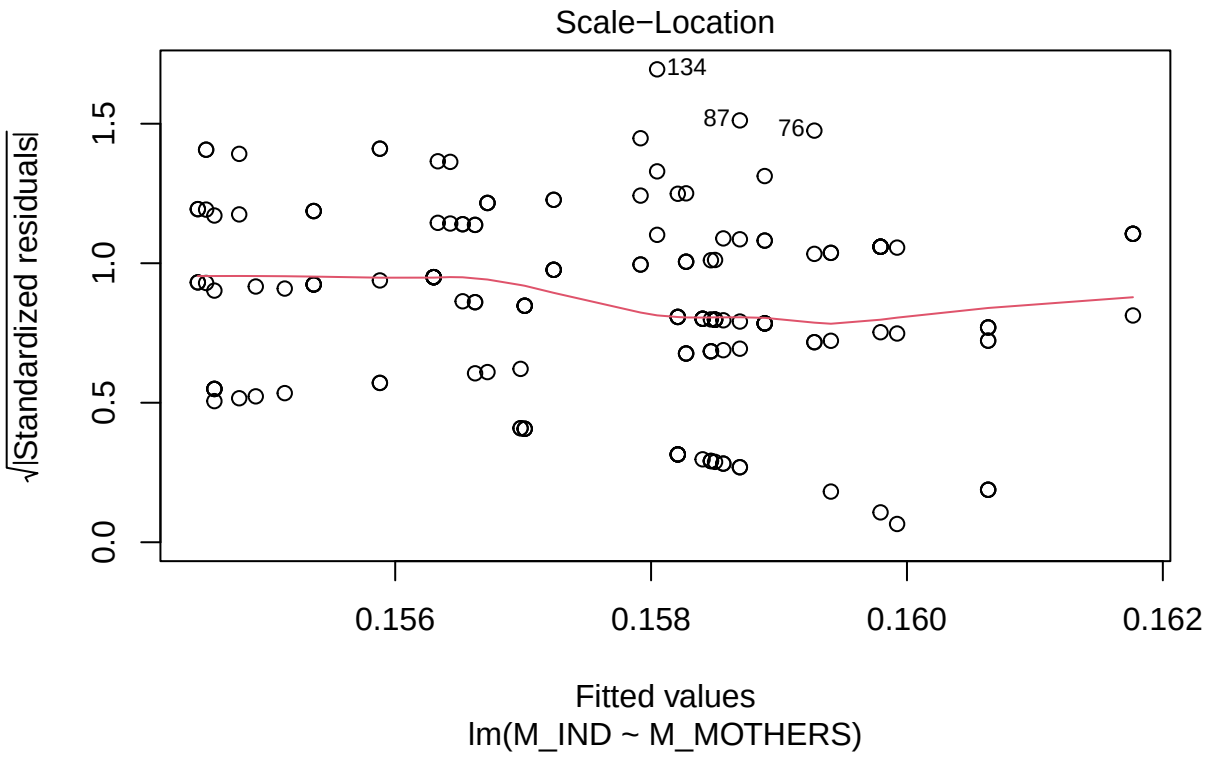

```
# on vérifie ici la normalité des résidus

# normalité discutable, mais cela n'a pas d'importance majeure pour ce type de
# modèle



plot(reg_poids_juv_meres, 5)
```

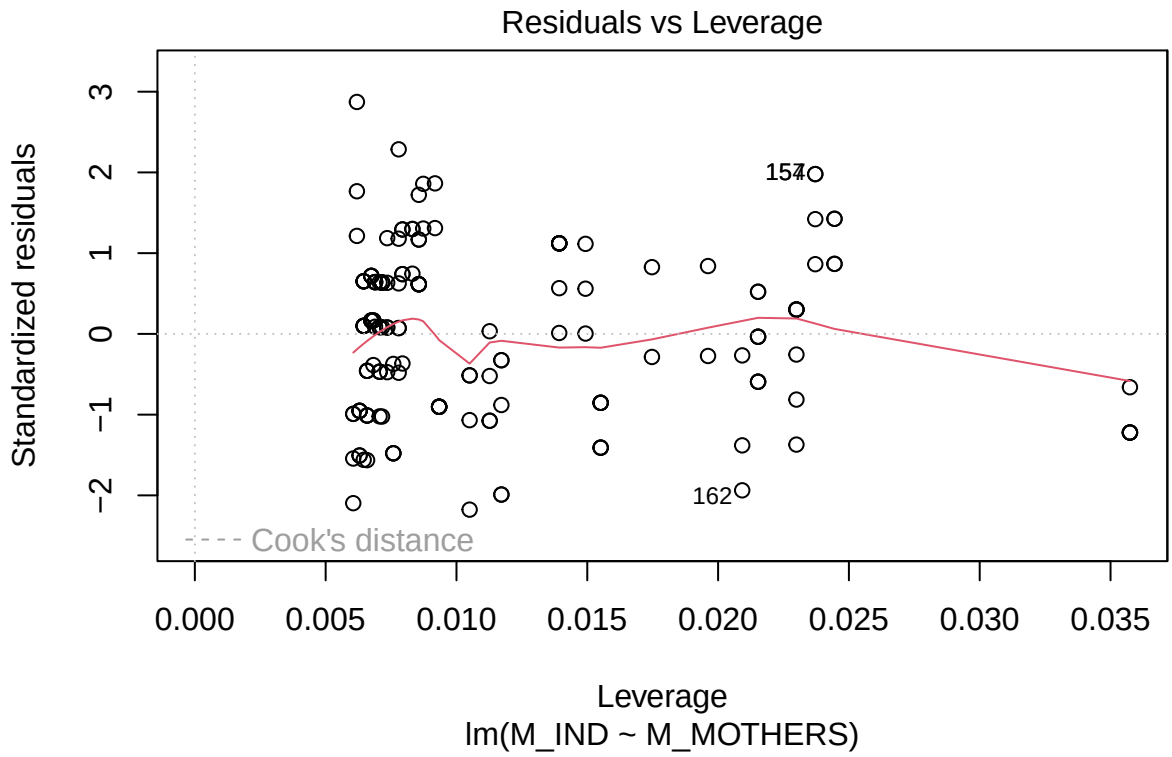



```
# on vérifie ici la présence ou non de point d'influence, la ligne rouge doit
# rester horizontale (sinon: présence de fort point d'influence), les fort
# point d'influence apparaissent au-delà d'une distance de Cook de 0.8 (cette
# limite apparait sur le graphique par une ligne pointillée lorsqu'elle est
# franchise)

# pas de point d'influence important ici




# l'ensemble des grpahiques visant à vérifier la validité du modèle peuvent
# être retrouvés via la commande suivante:

plot(reg_poids_juv_meres)
```

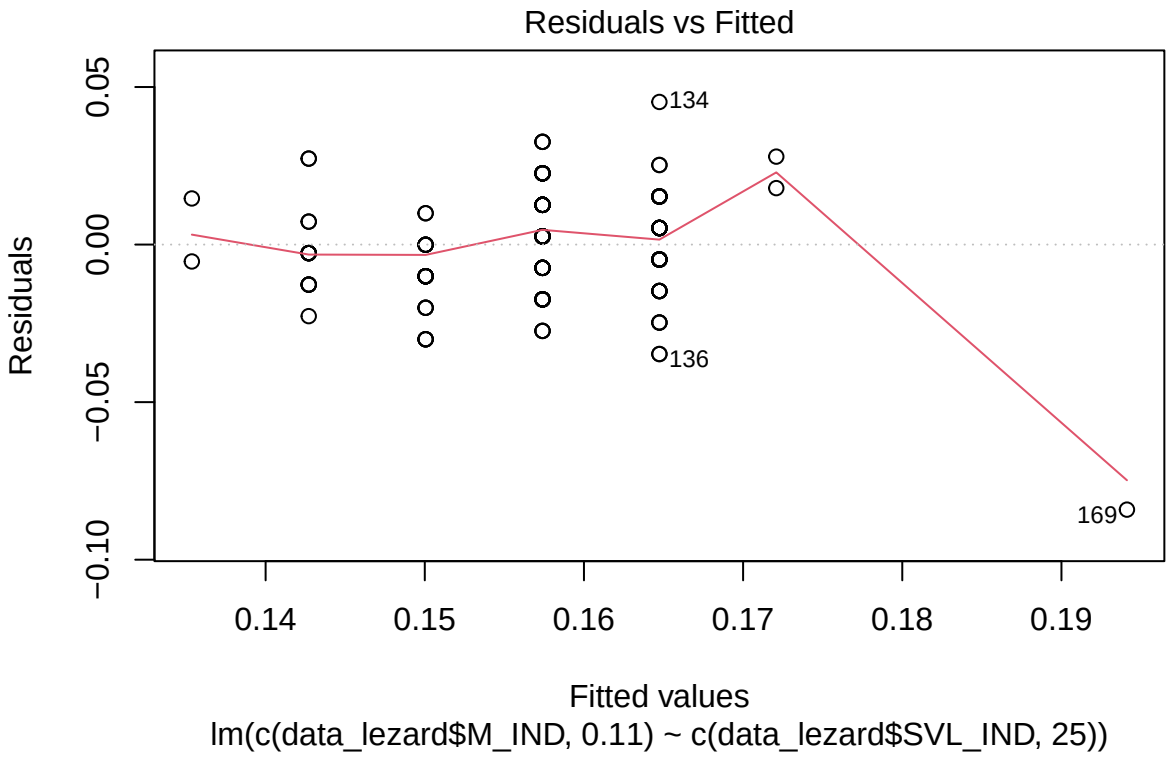

Residuals vs Fitted
Residuals
0.06
0.04
0.02
0.00
−0.04
134
87
76
0.156
0.158
0.160
0.162
Fitted values
lm(M_IND ~ M_MOTHERS)

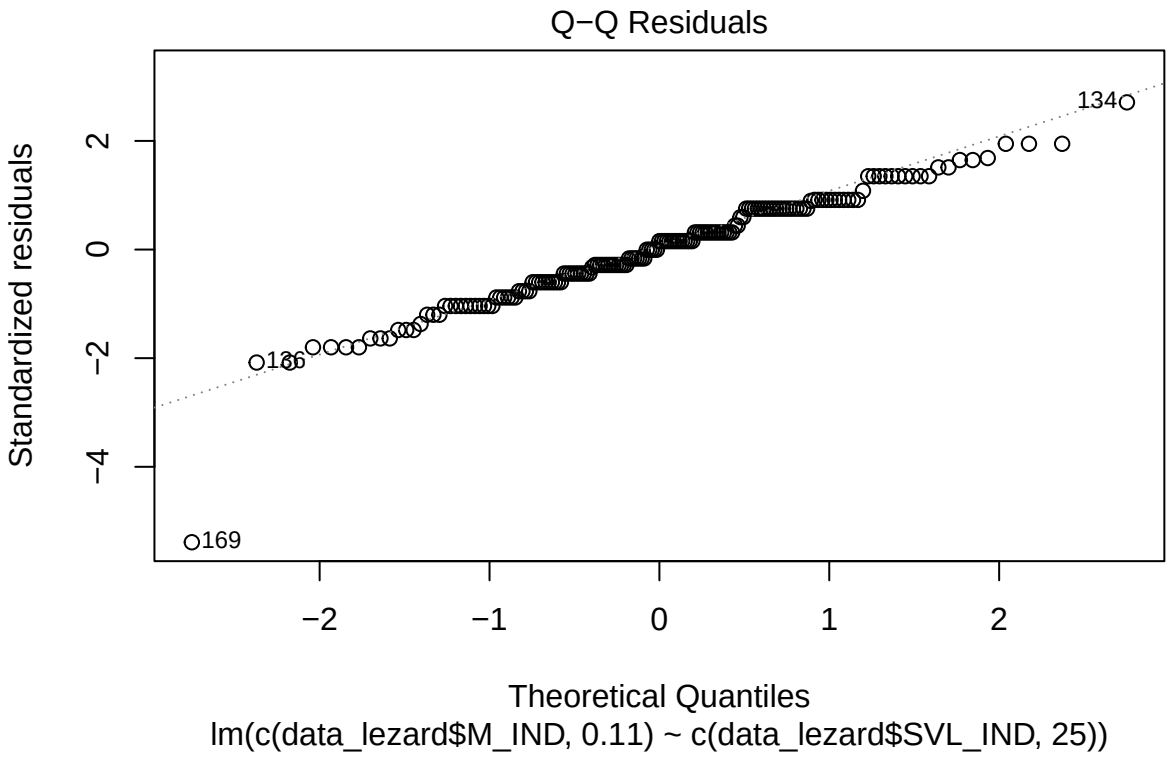

Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(M_IND ~ M_MOTHERS)
134
87
76
−2
−1
0
1
2
3

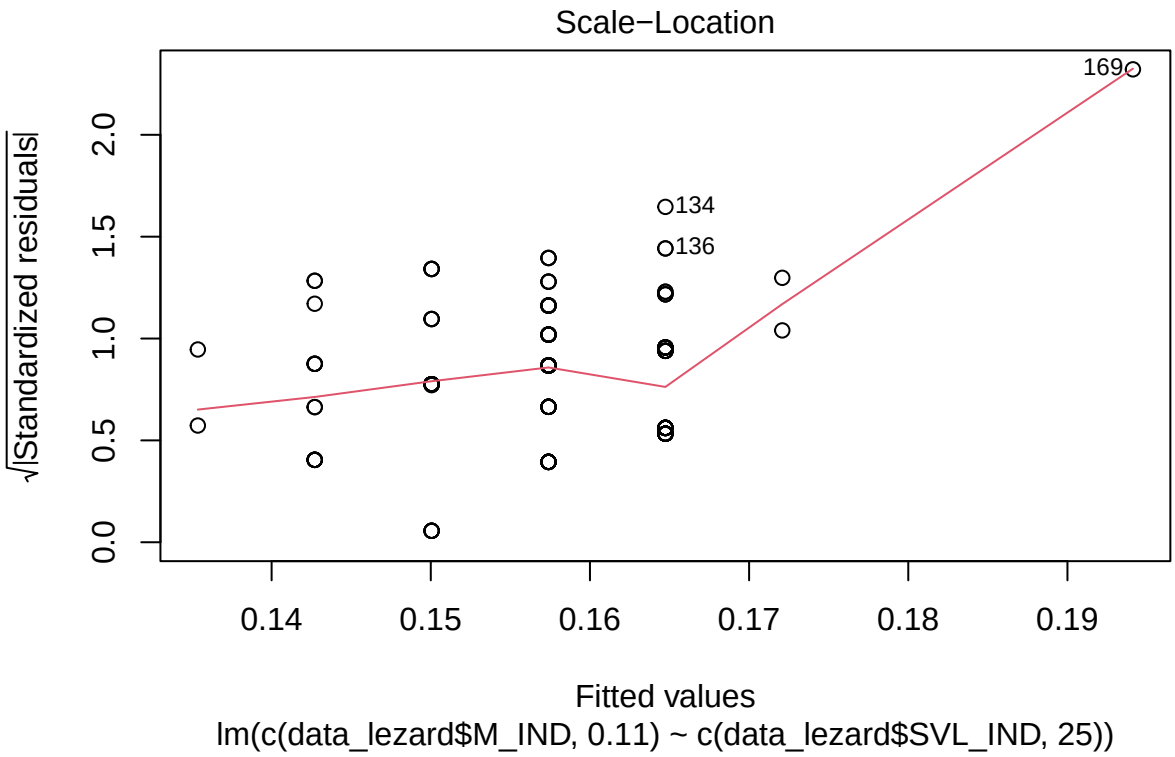

Scale−Location
134
87
76
√|Standardized residuals|
0.0
0.5
1.0
1.5
0.156
0.158
0.160
0.162
Fitted values
lm(M_IND ~ M_MOTHERS)

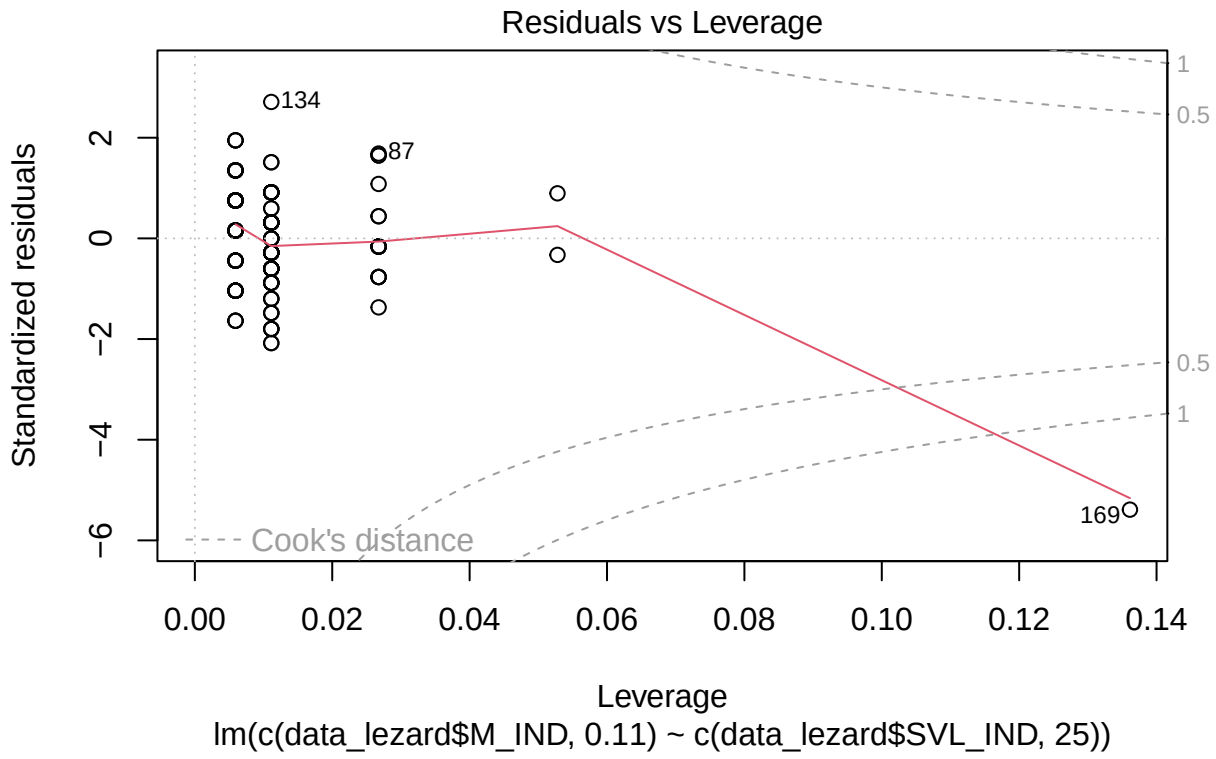



```
# le troisième graphique est souvent utilisé pour tester spécifiquement
# l'hypothèse d'homoscédasticité des résidus: la courbe rouge doit être
# horizontale




## exemple d'un modèle à la validité discutable:

# on réalise une régression mais en ajoutant avec un outlier évident


reg_taille_poids_juv = lm(
    c(data_lezard$M_IND, 0.11) ~
        c(data_lezard$SVL_IND, 25)
)



plot(reg_taille_poids_juv)
```

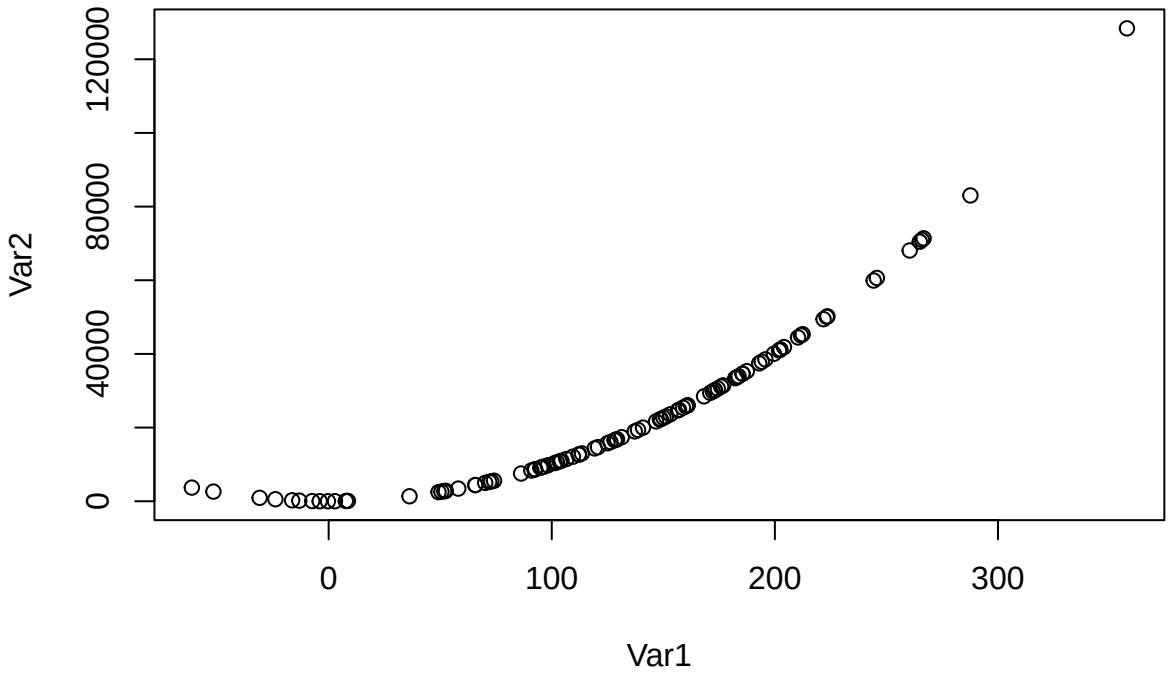

Residuals vs Fitted
Residuals
0.05
0.00
−0.05
−0.10
134
136
169
0.14
0.15
0.16
0.17
0.18
0.19
Fitted values
lm(c(data_lezard$M_IND, 0.11) ~ c(data_lezard$SVL_IND, 25))


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(c(data_lezard$M_IND, 0.11) ~ c(data_lezard$SVL_IND, 25))
134
136
169
2
0
−2
−4
−2
−1
0
1
2

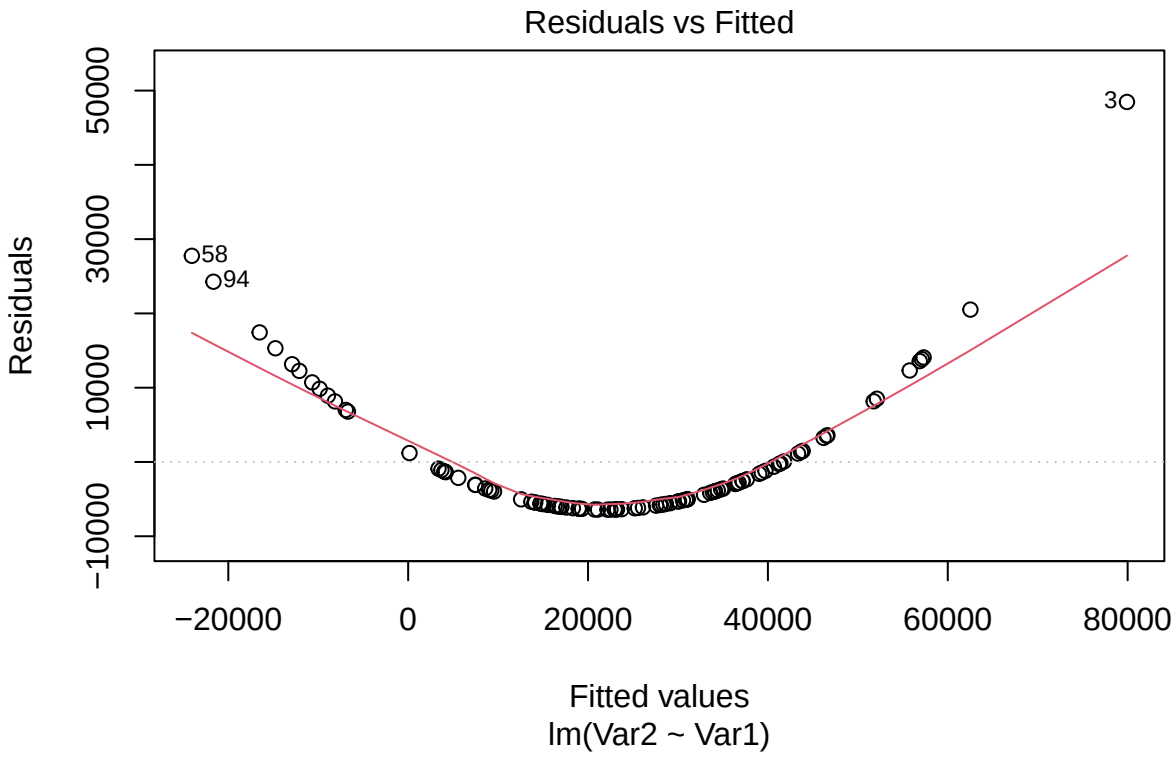

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0.14
0.15
0.16
0.17
0.18
0.19
169
134
136
Fitted values
lm(c(data_lezard$M_IND, 0.11) ~ c(data_lezard$SVL_IND, 25))

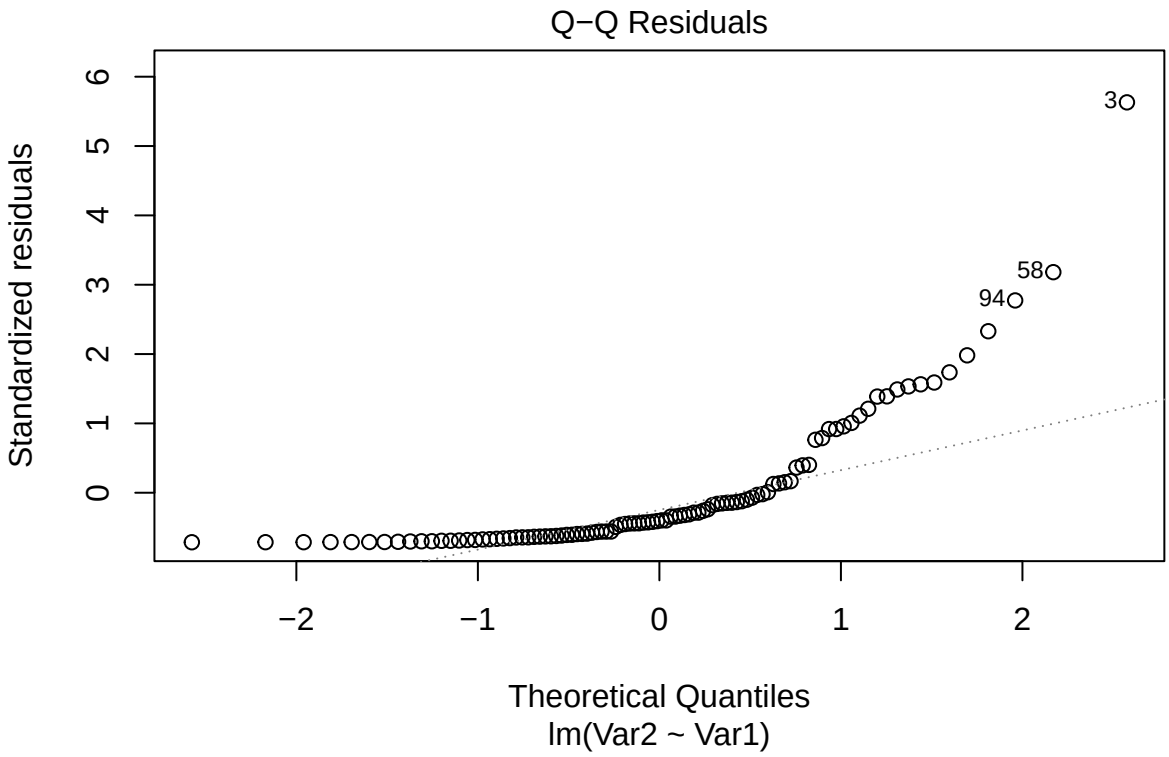


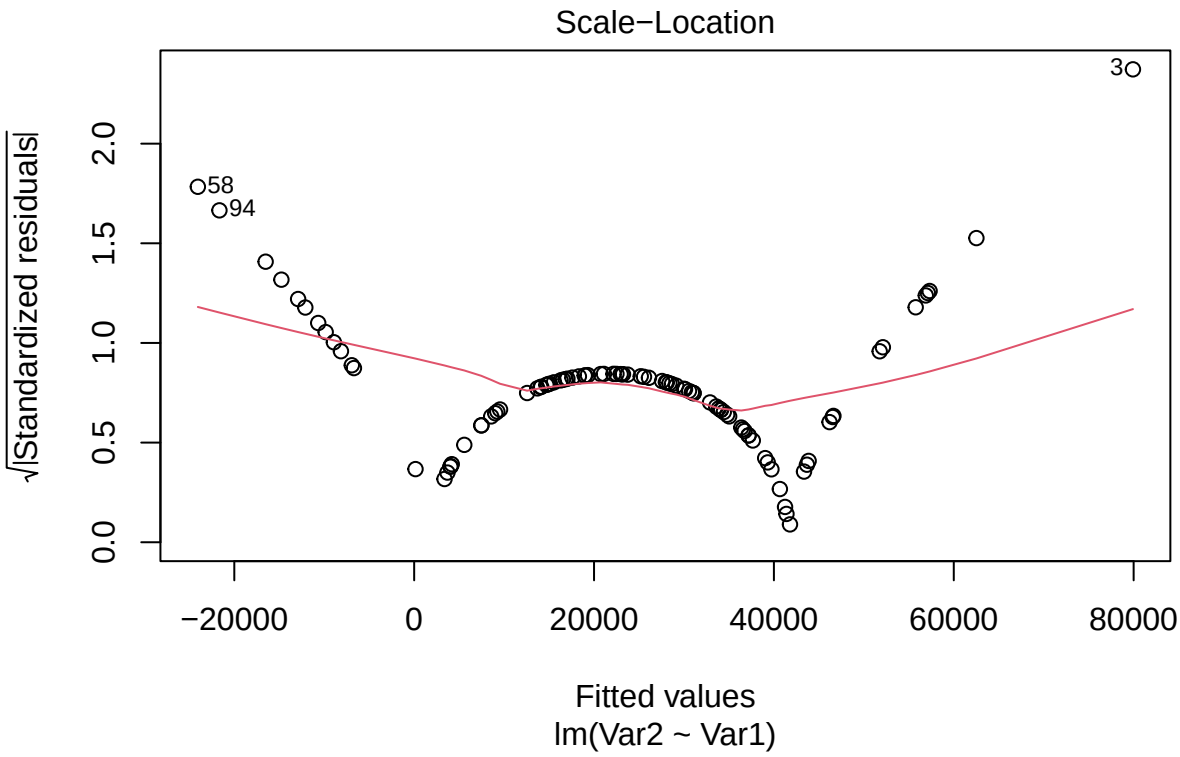

```
# on visualise très bien la présence de l'outlier sur les graphique de
# vérification



# on réalise une régression entre deux variables dont la relation n'est pas
# linéaire:


Var1 = rnorm(100, 120, 80)
Var2 = Var1 + Var1^2

plot(Var2 ~ Var1)
```

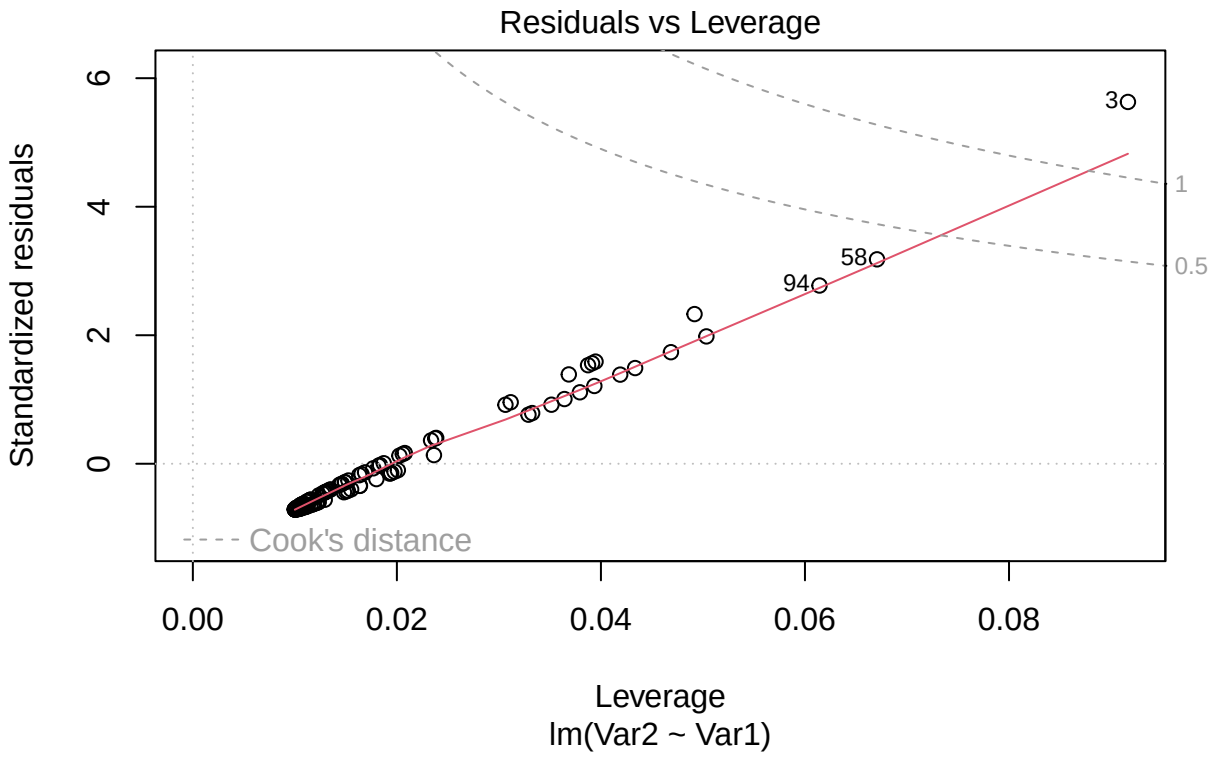



```
reg_problem = lm(Var2 ~ Var1)


plot(reg_problem)
```


Residuals vs Fitted
Residuals
50000
30000
10000
−10000
58
94
3
−20000
0
20000
40000
60000
80000
Fitted values
lm(Var2 ~ Var1)


Q−Q Residuals
Standardized residuals
0
1
2
3
4
5
6
3
58
94
−2
−1
0
1
2
Theoretical Quantiles
lm(Var2 ~ Var1)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
58
94
3
−20000
0
20000
40000
60000
80000
Fitted values
lm(Var2 ~ Var1)




```
# problème très clair de non linéarité des résidus !
```

EXERCICE

- Vérifiez la validité de la régression linéaire entre la taille des juvéniles et leur poids.

```
plot(reg_poids_taille_juv)
```

```
# on n'observe pas de déviations majeures par rapport aux attendus d'une régression
# linéaire (pas de problème de non linéarité ou de variabilité changeante des résidus,
# normalité des résidus approximativement respectée, pas de points de levier majeur):
# le modèle effectué est donc valide
```

Interpréter les sorties du modèle

Le coefficient directeur de la régression permet d'estimer l'intensité de la relation entre les deux variables, plus celui-ci est élevé (en valeur absolue) plus la variable réponse augmente rapidement avec l'augmentation de la variable explicative: en langage statistique on parle de **taille d'effet**. Pour facilement comparer des tailles d'effets (si nécessaire) il est préférable de **centrer-réduire** les variables en amont pour s'affranchir de leur dimensionnalité.
D'autre part, l'adéquation du modèle au donnée, c'est à dire la part de variance observée expliquée par le modèle, peut être obtenue à partir de la métrique du **R^2 (coefficient de détermination)**, qui est égal au

carré du coefficient de corrélation de Pearson (pour info une démonstration mathématique de cette relation peut être trouvée ici: https://statproofbook.github.io/P/slr-rsq.html, https://statproofbook.github.io/P/slr-corr).

```
summary(reg_poids_juv_meres)$r.squared
```

```
## [1] 0.01016273
```

```
# très faible adéquation au données, seulement environ 1% de la variance expliqué



data_lezard$Corp_IND =  data_lezard$SVL_IND/data_lezard$M_IND

reg_corp_poids_juv = lm(data_lezard$M_IND ~ data_lezard$Corp_IND)

summary(reg_corp_poids_juv)$r.squared
```

```
## [1] 0.8009573
```

```
# 80% de la variance expliquée, bien meilleure adéquation au donnée



# Comparaison des coefficients:


reg_poids_juv_meres = lm(scale(M_IND) ~ scale(M_MOTHERS), data = data_lezard)

reg_corp_poids_juv = lm(scale(data_lezard$M_IND) ~ scale(data_lezard$Corp_IND))


reg_poids_juv_meres$coefficients
```

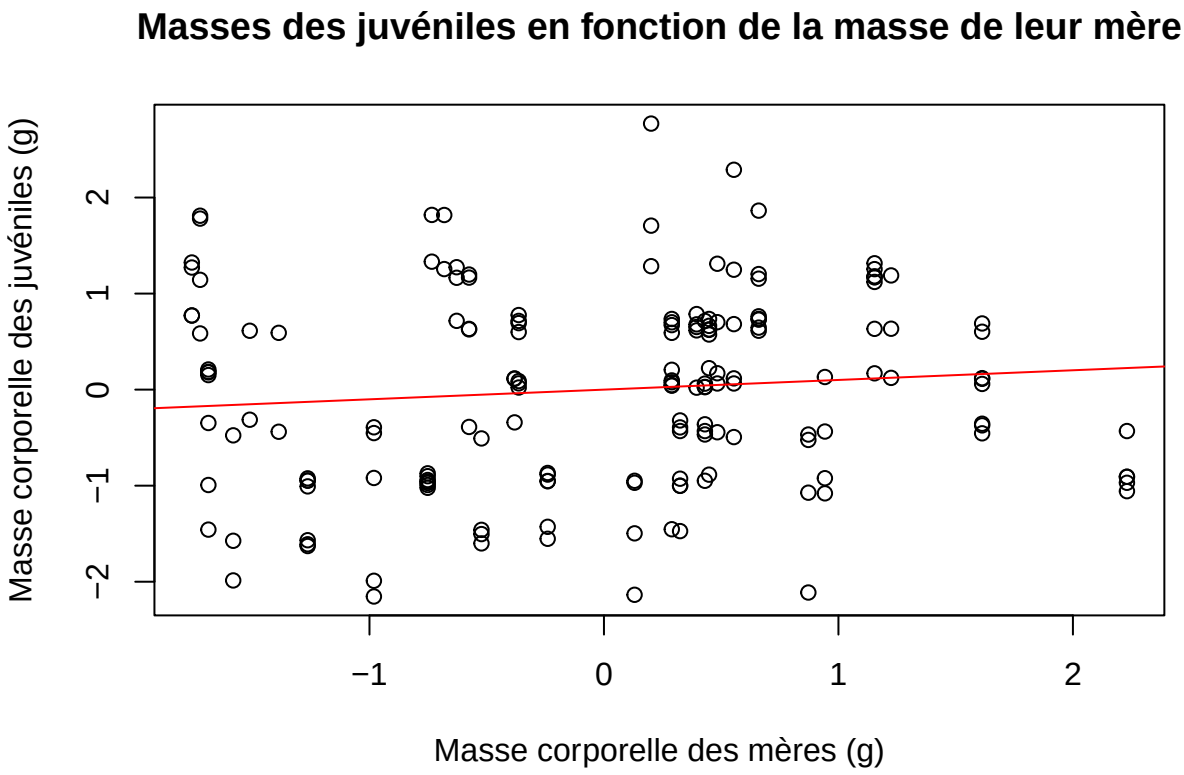
```
##      (Intercept) scale(M_MOTHERS)
##    -7.310598e-16     1.008104e-01
```

```
reg_corp_poids_juv$coefficients
```

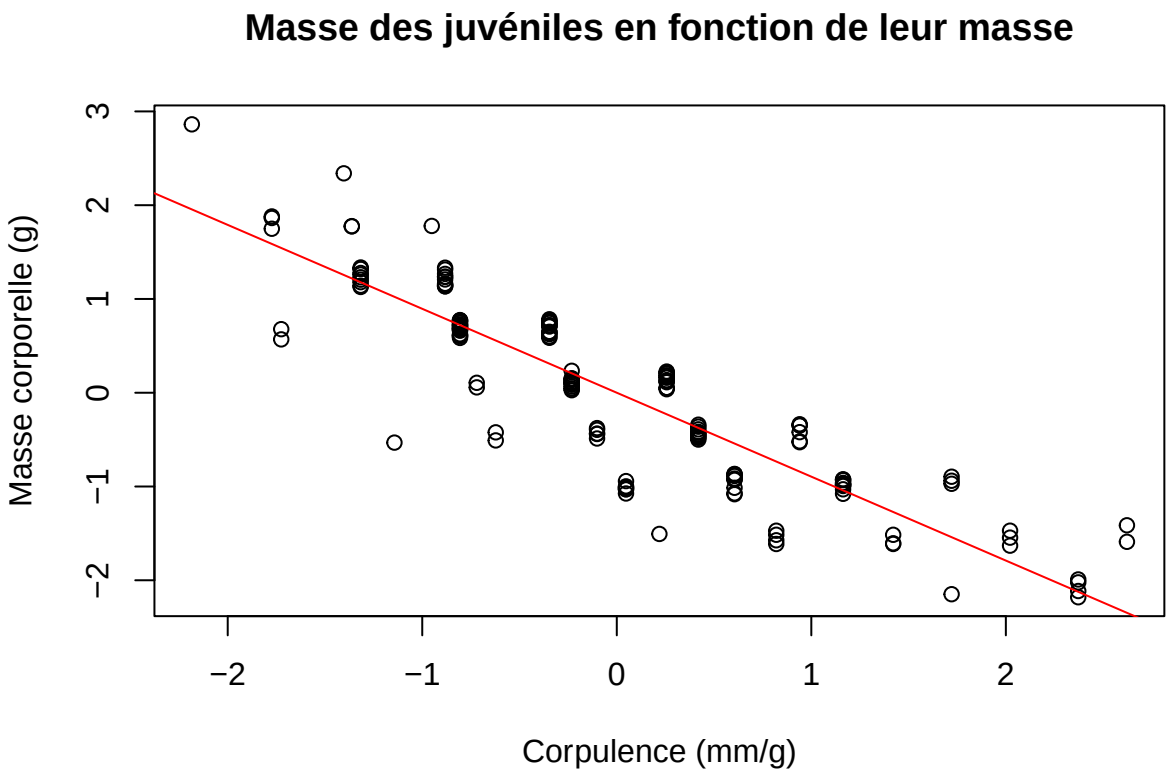
```
##                 (Intercept) scale(data_lezard$Corp_IND)
##               -1.079235e-15               -8.949622e-01
```

```
# après avoir centré réduit les variables, on observe qu'une augmentation de la corpulence
# des individus a un plus grand impact que l'augmentation de la masse des mères sur la
# variation de poids des individus (valeur absolue du coefficient directeur supérieure)



# cela est parfaitement visible sur les graphiques représentant ces régressions:
```

```r
plot(
 jitter(scale(data_lezard$M_IND), 1) ~
   scale(data_lezard$M_MOTHERS),
 main = "Masses des juvéniles en fonction de la masse de leur mère",
 xlab = "Masse corporelle des mères (g)",
 ylab = "Masse corporelle des juvéniles (g)")

abline(reg_poids_juv_meres, col = "red")
```




```r
plot(
 jitter(scale(data_lezard$M_IND), 1) ~
   scale(data_lezard$Corp_IND),
 main = "Masse des juvéniles en fonction de leur masse",
 xlab = "Corpulence (mm/g)",
 ylab = "Masse corporelle (g)")

abline(reg_corp_poids_juv, col = "red")
```



EXERCICE

- Interprétez les résultats de la régression linéaire entre la taille des juvéniles et leur poids.

```
summary(reg_poids_taille_juv)$r.squared
# 29,5 % de la variance observée (pour les variation de poids des juvéniles) est expliquée
# par notre modèle (c'est à dire par les variations de taille des juvéniles):
# l'adéquation du modèle au données est donc de plutôt bonne qualité



reg_poids_taille_juv$coefficients
# le poids des juvéniles augmente d'environ 0.01g (c'est à dire 10mg)
# quand la taille augmente d'1 mm
```

Bilan

Lorsque la relation entre deux variables continues semble être **affine (à toujours vérifier en amont !)** on peut modéliser la relation entre les deux variables par une **régression linéaire** (dans R: "**lm(Var1 ~ Var2)**", avec "Var1" la **variable réponse** et "Var2" la **variable explicative**).

Après avoir réalisé le modèle, il convient toujours de vérifier la **validité** de celui-ci, en visualisant si les résidus de la régression (dans R: "**residuals(nom_modele)**") respectent bien certaines hypothèses requises: ils

doivent conserver une **variance constante** (homoscedasticité) et ne doivent **pas varier** en fonction de la variable réponse (marqueur d'une relation non linéaire). On peut aussi vérifier que la distribution des résidus est **normale** (cette hypothèse n'est pas indispensable pour une régression linéaire univariée mais assure une bonne qualité de modèle toutefois) et identifier la présence de **points d'influence** forts, qui pourraient fausser les résultats de la régression (comme par exemple avec d'éventuelles mesures aberrantes). Cet ensemble d'hypothèse peut être vérifié à l'aide de la fonction "**plot(nom_modele)**" dans R (attention à bien connaître l'interprétation à faire de chaque sortie graphique ainsi obtenue: voir plus haut dans ce TP).

Deux mesures statistiques permettent principalement d'interpréter le modèle: le **coefficient directeur** de la régression (dans R: deuxième valeur obtenue à l'aide de "**coefficients(nom_modele)**") indique la variation modélisée de la variable réponse lorsque la variable explicative augmente d'une unité, cela permet d'obtenir une **taille d'effet** potentiellement comparable entre plusieurs modèles après avoir centré-réduit les variables (il faut toutefois que la comparaison ait un sens); le **coefficient de détermination** (ou R-squared) de la régression (dans R: "**summary(nom_modele)$r.squared**") permet de connaître **l'adéquation du modèle aux données**, c'est à dire à quel point le modèle représente bien les variations existantes de la variable réponse (équivalent au **pourcentage de variation expliqué par le modèle quand on le multiplie par 100**).

On peut illustrer graphiquement la régression linéaire en ajoutant la droite de régression au graphique en nuage de point via la commande **abline(nom_regression)** (il est recommandé d'ajouter une couleur à cette ligne pour qu'elle soit facilement visible et légendable). Plus généralement, les prédictions du modèle peuvent être obtenue grâce à la commande **predict(nom_modele, nouvelles_données)**