

Des tableaux et des graphiques prêts à publication avec les packages R `{tabularise}` et `{chart}` de la suite SciViews

Guyliann Engels 1*

Philippe Grosjean 2†

Résumé (max 300 mots)

Au sein de la suite de packages SciViews (Grosjean 2023), voir aussi <https://sciviews.r-universe.dev/>, qui vise à simplifier R pour les utilisateurs occasionnels, débutants ou désireux de gagner du temps, nous présentons `{tabularise}` pour les tableaux et `{chart}` pour les graphiques. Ces packages s'appuient respectivement sur `{flextable}` et `{ggplot2}` pour offrir des tableaux et des graphiques représentant différents objets R dans une forme aussi proche que possible de versions prêtes à publication, dans la lignée de `{cowplot}` par exemple.

Pour accélérer et simplifier la création de ces tableaux et graphiques, `{tabularise}` et `{chart}` supportent l'anglais et le français et prennent également en compte les labels des variables pour générer les noms de colonnes des tableaux ou les libelles des axes et des encarts de légende des graphiques. Les équations, au format LaTeX, sont également automatiquement générées pour une large gamme de modèles statistiques grâce à `{equatomatic}` et intégrés dans certains tableaux (par exemple pour les objets `lm` ou `glm`).

Ces tableaux et graphiques étant compatibles avec `{flextable}` et `{ggplot2}`, ils restent entièrement personnalisables à l'aide des fonctions offertes par ces derniers packages ou leurs extensions. Pour l'utilisateur avancé ou plus créatif, ils seront plutôt considérés comme des premières étapes de production de contenu plus spécifique.

Mots-clefs : Prêt à publication – Tableau – Graphique – `tabularise` – `chart` – SciViews

Développement

Lors de la rédaction de documents scientifiques, d'articles de blog ou encore de manuscrits de thèse, des outils tels que Rmarkdown (Allaire et al. 2023) ou Quarto (Allaire 2023) sont couramment utilisés. Ces outils permettent d'intégrer du code R dans des chunks et du texte en Markdown, ce qui constitue la première étape pour produire des documents prêts à être publiés. Cependant, une seconde étape est nécessaire pour créer des tableaux et des graphiques conformes aux normes de publication. Les sorties de base de R ne sont pas adaptées à cette fin, surtout pour les tableaux, mais la communauté R propose de nombreuses solutions parmi les plus de 20 000 packages disponibles sur CRAN, sans compter ceux disponibles depuis BioConductor, GitHub ou Gitlab.

Pour la création de tableaux propres et publiables, on peut citer le package R `{flextable}` (Gohel and Skintzos 2024) ou encore le package `{gt}` (Iannone et al. 2024). Ces outils puissants permettent de formater un tableau à partir d'un data frame. Ils offrent une grande flexibilité pour répondre à tous les besoins, mais nécessitent de nombreuses lignes de code pour obtenir le tableau dans le formatage souhaité. C'est pourquoi nous proposons, dans l'univers SciViews (<https://sciviews.r-universe.dev/>), le package `{tabularise}`. Il permet d'obtenir des tableaux quasiment prêts à être publiés en une seule instruction pour une série d'objets R. Ces premières versions peuvent ensuite être remaniées à l'aide de `{flextable}`. `{tabularise}` gère l'anglais et le français et prend en compte automatiquement les labels des variables s'ils sont présents dans l'objet via un attribut "label". Il prend également en compte les unités si un attribut "units" est défini. Le package `{chart}` fait de même pour les graphiques. Il s'appuie sur `{ggplot2}` ou `{lattice}`. Les labels et unités peuvent être

*Service d'écologie numérique, Institut Complexys & Infotech, Université de Mons, Belgique, guyliann.engels@umons.ac.be

†Service d'écologie numérique, Institut Complexys & Infotech, Université de Mons, Belgique, philippe.grosjean@umons.ac.be

rajoutés directement, ou en utilisant `data.io::labellise()` (`{data.io}` est un autre package de l'univers SciViews). Voici un exemple :

```
# Installation des packages nécessaires depuis l'univers SciViews
# {data.io} pour les labels, {equatags} pour les équations et {modelit} pour des
# graphiques et tableaux supplémentaires pour les modèles statistiques
# (d'autres dépendances sont susceptibles d'être également installées)
#install.packages(c('tabularise', 'chart', 'data.io', 'equatags', 'modelit'),
# repos = c('https://sciviews.r-universe.dev', 'https://cloud.r-project.org'))
library(tabularise)
library(chart)
library(modelit)
```

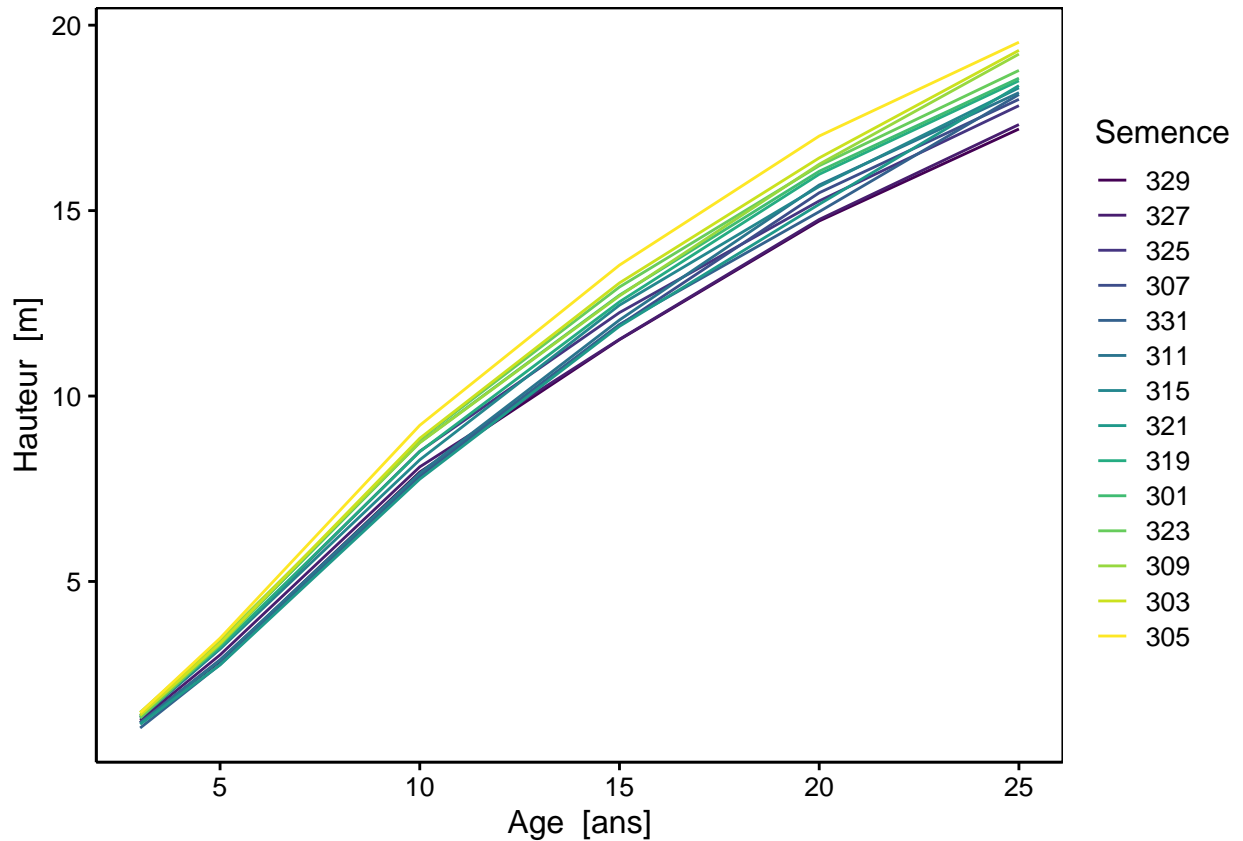
Notez que ces packages ne sont pas encore sur CRAN car ils sont en cours de développement et susceptibles d'évoluer de manière difficilement compatible avec CRAN. Ils seront proposé à CRAN lorsqu'ils seront stabilisés. L'installation depuis R-universe est donc recommandée pour l'instant.

```
# Exemple : données de croissance du pin *Pinus tadea*
data("Loblolly", package = "datasets")
# Conversion de la hauteur de pieds en mètres
Loblolly$height <- round(Loblolly$height * 0.3048, 2)
# La labélisation des variables ne doit se faire qu'une seule fois
Loblolly <- data.io::labellise(Loblolly,
  label = list(height = "Hauteur", age = "Age", Seed = "Semence"),
  units = list(height = "m", age = "ans"))
# Les tableaux et graphiques l'utilisent ensuite automatiquement
# Les variantes éventuelles peuvent être spécifiées avec '$<type>'
tabularise$headtail(Loblolly[, c(3, 1, 2)])
```

Semence	Hauteur [m]	Age [ans]
301	1.37	3
301	3.32	5
301	8.75	10
301	12.72	15
301	16.06	20
□	□	□
331	2.76	5
331	7.88	10
331	11.93	15
331	14.97	20
331	18.13	25

First and last 5 rows of a total of 84

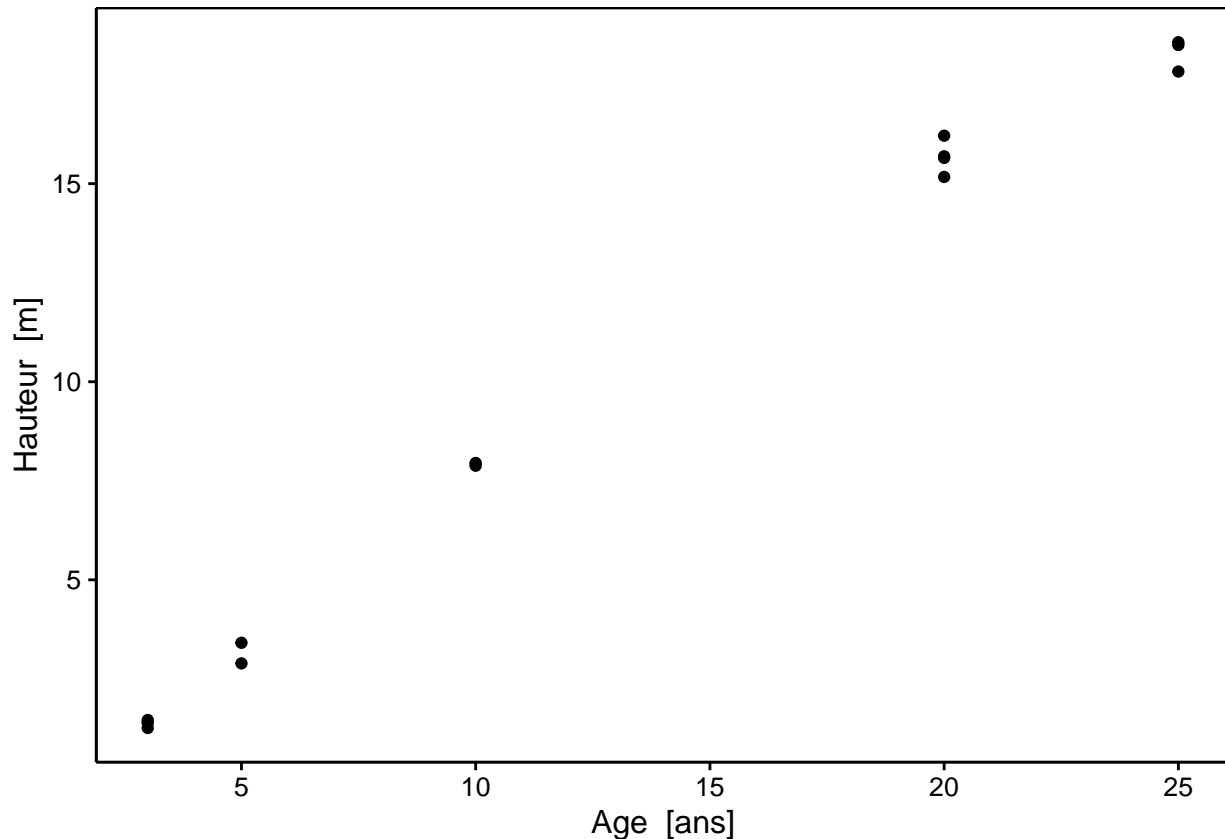
```
chart(Loblolly, height ~ age %col=% Seed) + geom_line()
```



De nombreuses méthodes sont disponibles pour `{tabularise}` et `{chart}` et elles sont encore augmentées par d'autres packages dans l'univers SciViews comme `{modelit}` que nous utilisons ici ou encore `{inferit}`, `{exploireit}`. Pour les objets **lm**, **glm**, **nls**..., `{tabularise}` intègre également des équations dans son rendu en s'appuyant sur les packages `{equationomatic}` (Anderson, Heiss, and Sumners 2024) et `{equatags}`.

Voici quelques tableaux et graphiques qu'il est possible de générer à partir d'un objet **lm** (ici, nous ne conservons qu'une seule mesure par arbre (*alias* *Seed*) pour étudier un jeu de données ayant des observations indépendantes les unes des autres).

```
# Ne conserver qu'une mesure par arbre dans `pine`
set.seed(3652)
pine <- dplyr::slice_sample(Loblolly, n = 1, by = Seed)
chart(pine, height ~ age) + geom_point()
```



Voici un modèle linéaire ajusté dans ces données :

```
pine_lm <- lm(data = pine, height ~ age + I(age^2))
summary(pine_lm)
```

```
##
## Call:
## lm(formula = height ~ age + I(age^2), data = pine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.55076 -0.13703  0.08873  0.13593  0.64271
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.039295   0.289531  -7.043 2.14e-05 ***
## age          1.134437   0.058449  19.409 7.37e-10 ***
## I(age^2)     -0.012705   0.002095  -6.064 8.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3345 on 11 degrees of freedom
## Multiple R-squared:  0.9981, Adjusted R-squared:  0.9978
## F-statistic: 2951 on 2 and 11 DF, p-value: 9.617e-16
```

Comparez le résumé “brut” de l’objet **lm** ci-dessus dans R avec la version formatée à l’aide de `tabularise'()` (en français grâce à `lang = "fr"`).

```
summary(pine_lm) |> tabularise(lang = "fr")
```

Modèle linéaire				
Hauteur [m] = $\alpha + \beta_1(\text{Age [ans]}) + \beta_2(\text{age}^2) + \epsilon$				
Terme	Valeur estimée	Ecart type	Valeur de t	Valeur de p
α	-2.0393	0.2895	-7.04	$2.14 \cdot 10^{-05***}$
β_1	1.1344	0.0584	19.41	$7.37 \cdot 10^{-10***}$
β_2	-0.0127	0.0021	-6.06	$8.14 \cdot 10^{-05***}$
0 <= '****' < 0.001 < '***' < 0.01 < '**' < 0.05				

Etendue des résidus : [-0.5508, 0.6427]

Ecart type des résidus : 0.3345 pour 11 degrés de liberté

R^2 multiple : 0.9981 - R^2 ajusté : 0.9978

Statistique F : 2951 pour 2 et 11 ddl - valeur de p : 9.6175e-16

Le tableau de l'ANOVA relatif à ce modèle donne ceci en `tabularise()` (en anglais par défaut) :

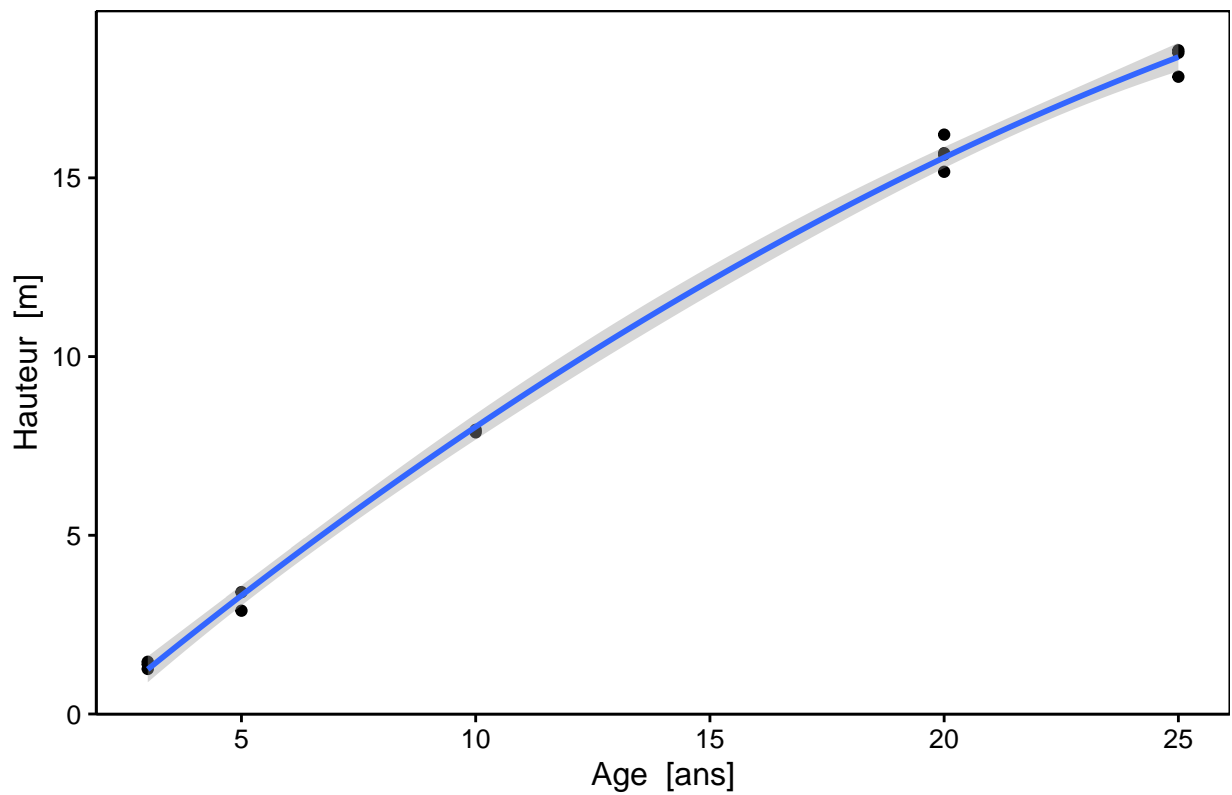
```
# Si un label est suboptimal, il est facilement remplacé avec labs =  
anova(pine_lm) |> tabularise(labs = c(`I(age^2)` = "age^2"))
```

Analysis of variance Response: height					
Term	Df	Sum of squares	Mean squares	$F_{\text{obs.}}$ value	p value
age	1	656.31	656.313	5864.4	$2.34 \cdot 10^{-16***}$
age ²	1	4.12	4.116	36.8	$8.14 \cdot 10^{-05***}$
Residuals	11	1.23	0.112		
0 <= '****' < 0.001 < '***' < 0.01 < '**' < 0.05					

Plusieurs graphiques sont aussi accessibles facilement avec `{chart}`, à commencer par le graphique montrant le modèle ajusté dans les données :

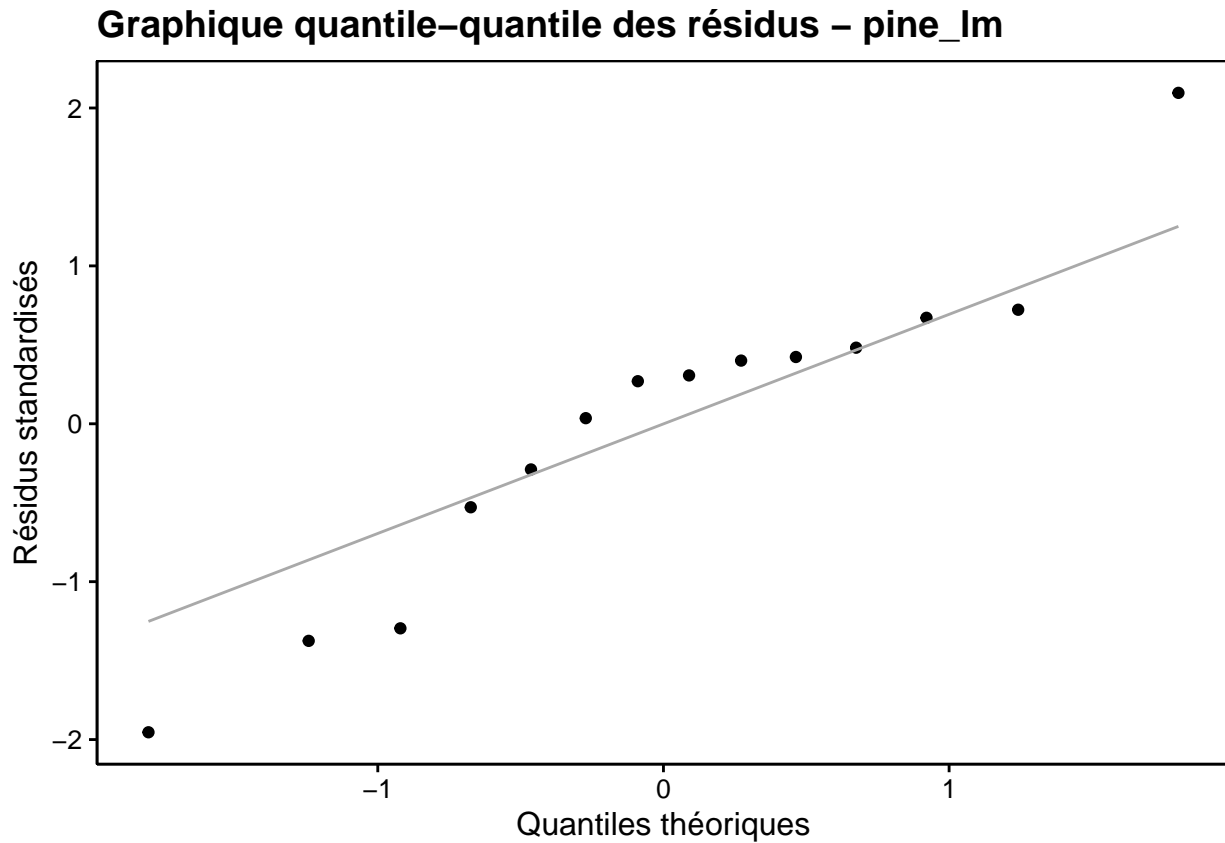
```
chart(pine_lm)
```

Model – pine_lm



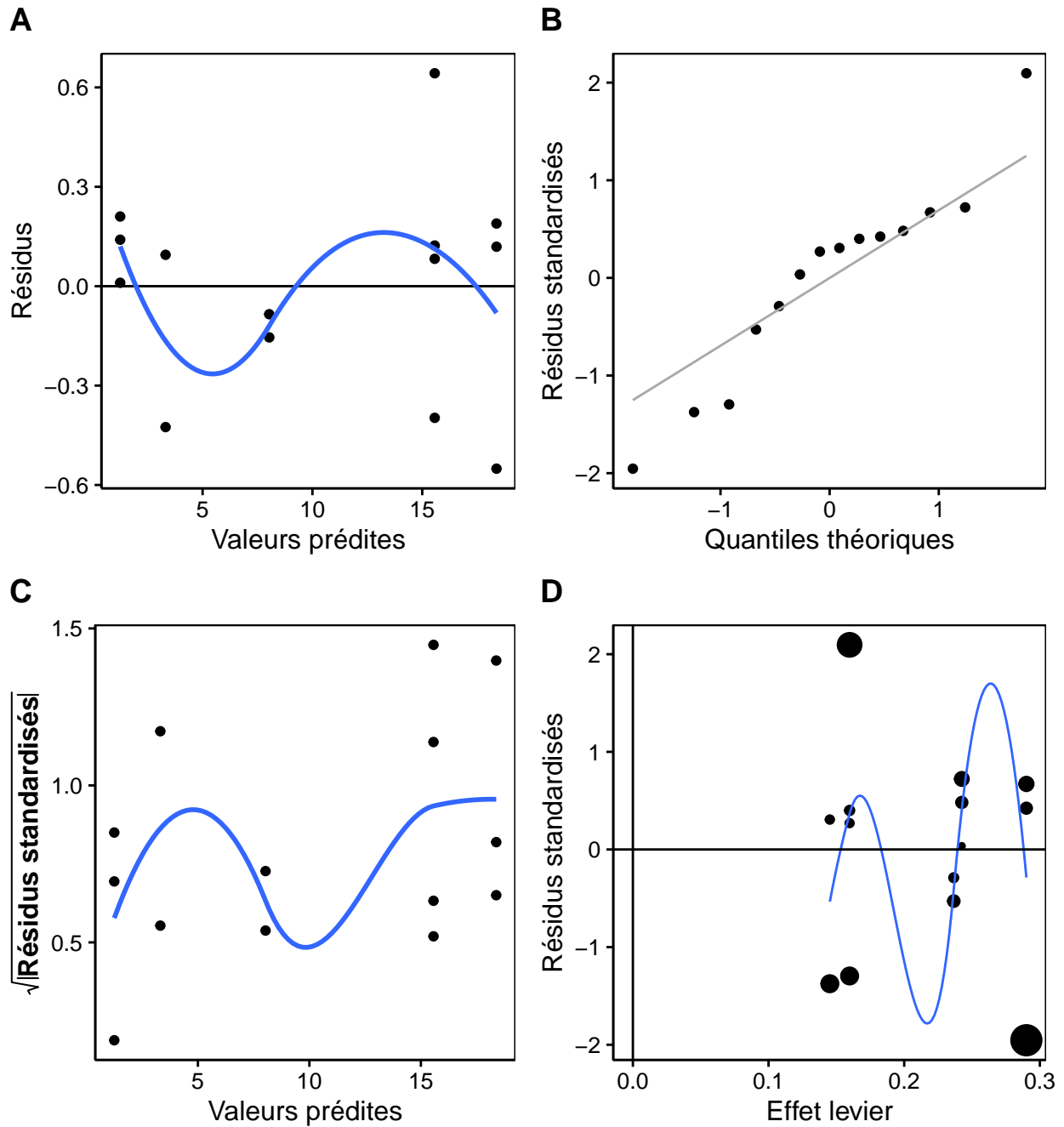
Les graphiques d'analyse des résidus sont accessibles sous forme de variantes (en utilisant `$<type>`). Par exemple, le graphique quantile quantile des résidus :

```
chart$qqplot(pine_lm, lang = "fr")
```



Un graphique composite, reprenant les quatre graphiques d'analyse des résidus les plus utilisés, s'obtient comme ceci (notez que les quatre graphiques sont libellés A-D pour en faciliter la description dans la légende du graphique) :

```
chart$residuals(pine_lm, lang = "fr")
```



D'autres objets sont également supportés. Par exemple, un modèle non linéaire peut être ajusté dans ces données, ce qui est plus adéquat pour des mesures de croissance. Nous utilisons alors un objet **nls**. Si nous ajustons un modèle de Gompertz ($\text{height} = a \cdot e^{(-b1 \cdot b2^{\text{age}})}$) nous obtenons ceci :

```
pine_nls <- nls(data = pine, height ~ SSgompertz(age, a, b1, b2))
# Résumé de l'analyse
summary(pine_nls) |> tabularise(lang = "fr")
```


Modèle non linéaire de Gompertz				
$\text{height} = a \cdot e^{(-b1 \cdot b2^{\text{age}})} + \epsilon$				
Terme	Valeur estimée	Ecart type	Valeur de $t_{\text{obs.}}$	Valeur de p
a	20.765	0.72637	28.6	$1.13 \cdot 10^{-11***}$
$b1$	3.755	0.20549	18.3	$1.40 \cdot 10^{-09***}$
$b2$	0.876	0.00827	105.9	$< 2 \cdot 10^{-16***}$

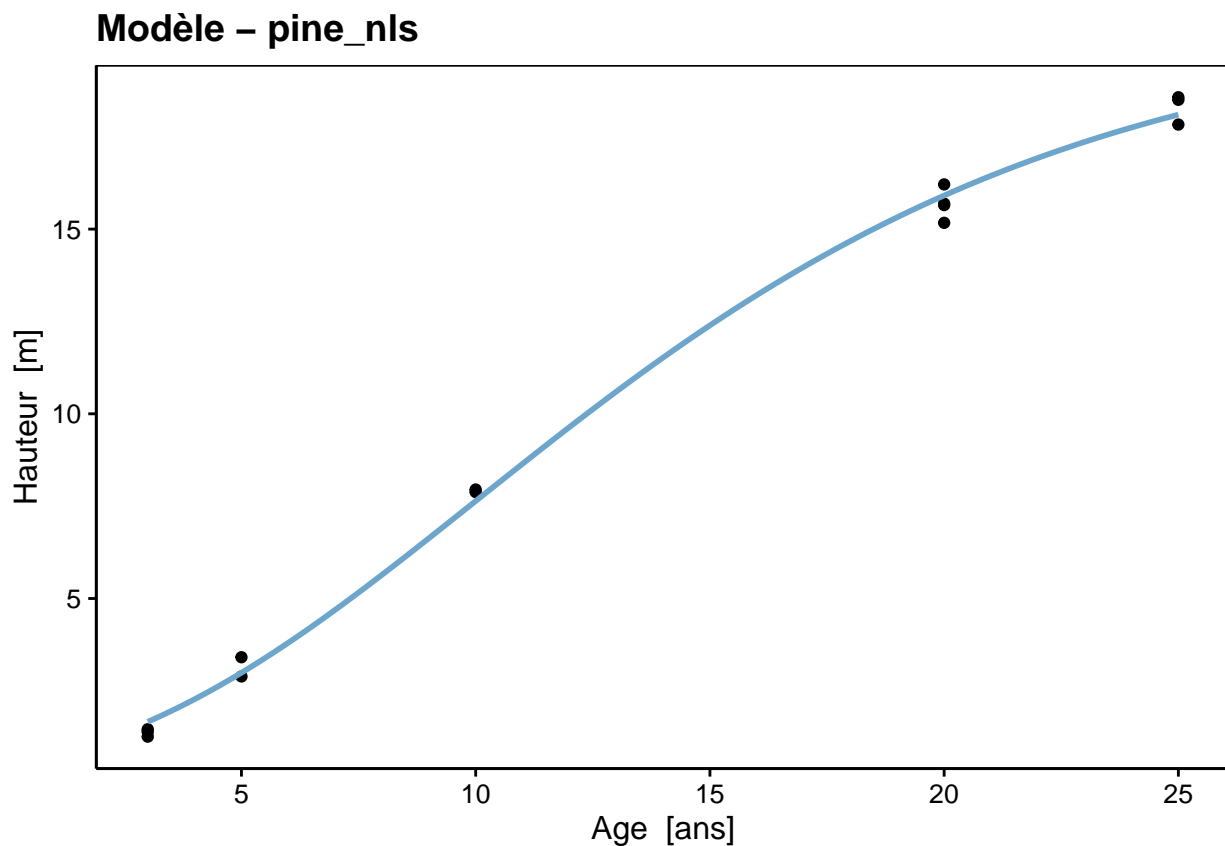
0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05

Ecart type des résidus : 0.4076 pour 11 degrés de liberté

Nombre d'itérations pour converger : 4

Tolérance atteinte à la convergence : 5.673e-06

```
# Graphique du modèle ajusté dans les données
chart(pine_nls, lang = "fr")
```



Vous retrouvez plusieurs autres exemples d'utilisation dans le module relatif à la régression linéaire, ainsi que dans les autres chapitres des cours de sciences des données biologique donnés à l'Université de Mons.

En conclusion, `{tabularise}` et `{chart}` sont deux packages développés pour faciliter les sorties tabulaires et graphiques de R, avec un rendu “presque prêts à être publié”. Grâce à `{flextable}`, les tableaux apparaissent correctement formatés quel que soit le format de sortie (HTML, PDF, Word, PowerPoint). L’usage des labels et unités directement insérés dans les données permet une présentation homogène des libellés de colonnes (tables) et d’axes (graphiques) et les équations des modèles sont automatiquement générées pour les modèles statistiques. Ces sorties tabulaires et graphiques restent, cependant, éditables avec les instructions de `{flextable}` et `{ggplot2}`, respectivement. Il ne s’agit donc pas de forcer l’utilisateur à utiliser des tables stéréotypées, mais de lui fournir un bon point de départ pour des sorties de qualité qu’il pourra ensuite adapter à sa guise.

Références

- Allaire, JJ. 2023. *Quarto: R Interface to ‘Quarto’ Markdown Publishing System*. <https://CRAN.R-project.org/package=quarto>.
- Allaire, JJ, Yihui Xie, Christophe Dervieux, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, et al. 2023. *Rmarkdown: Dynamic Documents for r*. <https://github.com/rstudio/rmarkdown>.
- Anderson, Daniel, Andrew Heiss, and Jay Sumners. 2024. *Equatiomatic: Transform Models into ‘LaTeX’ Equations*. <https://github.com/datalorax/equatiomatic>.
- Gohel, David, and Panagiotis Skintzos. 2024. *Flextable: Functions for Tabular Reporting*. <https://ardata-fr.github.io/flextable-book/>.
- Grosjean, Philippe. 2023. *SciViews::r*. MONS, Belgium: UMONS. <https://sciviews.r-universe.dev/>.
- Iannone, Richard, Joe Cheng, Barret Schloerke, Ellis Hughes, Alexandra Lauer, JooYoung Seo, and Ken Brevoort. 2024. *Gt: Easily Create Presentation-Ready Display Tables*. <https://gt.rstudio.com>.