

# 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. Pour la création de tableaux propres et publiables, on peut citer les packages R `{tinytable}` (Arel-Bundock, n.d.), 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. Dans l'univers SciViews (<https://sciviews.r-universe.dev/>), le package `{tabularise}` 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 rajoutés directement, ou en utilisant `data.io::labellise()` (`{data.io}` est un autre package de l'univers SciViews). Voici un exemple :

---

\*Service d'écologie numérique, Institut Complexys & Infortech, Université de Mons, Belgique, [guyliann.engels@umons.ac.be](mailto:guyliann.engels@umons.ac.be)

†Service d'écologie numérique, Institut Complexys & Infortech, Université de Mons, Belgique, [philippe.grosjean@umons.ac.be](mailto:philippe.grosjean@umons.ac.be)

```
# 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)
```

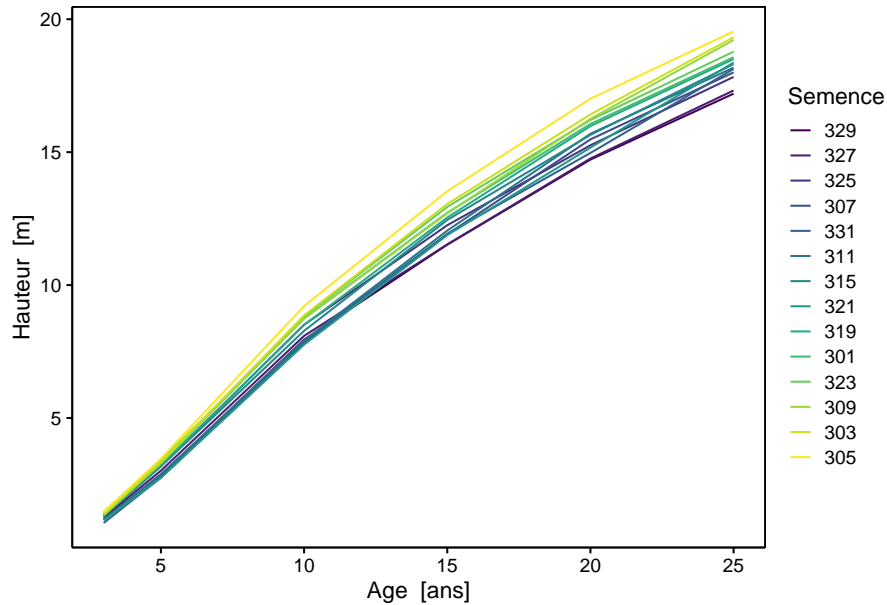
Notez que ces packages ne sont pas encore sur CRAN car ils sont en cours de développement et susceptibles d'évoluer de manière difficilement compatible avec CRAN. Ils seront proposé à CRAN lorsqu'ils seront stabilisés. L'installation depuis R-universe est donc recommandée pour l'instant.

```
# 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 labelisation des variables ne doit se faire qu'une seule fois
Loblolly <- data.io::labelise(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)], sep = "...", lang = "fr")
```

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

Premières et dernières 5 lignes d'un total de 84

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



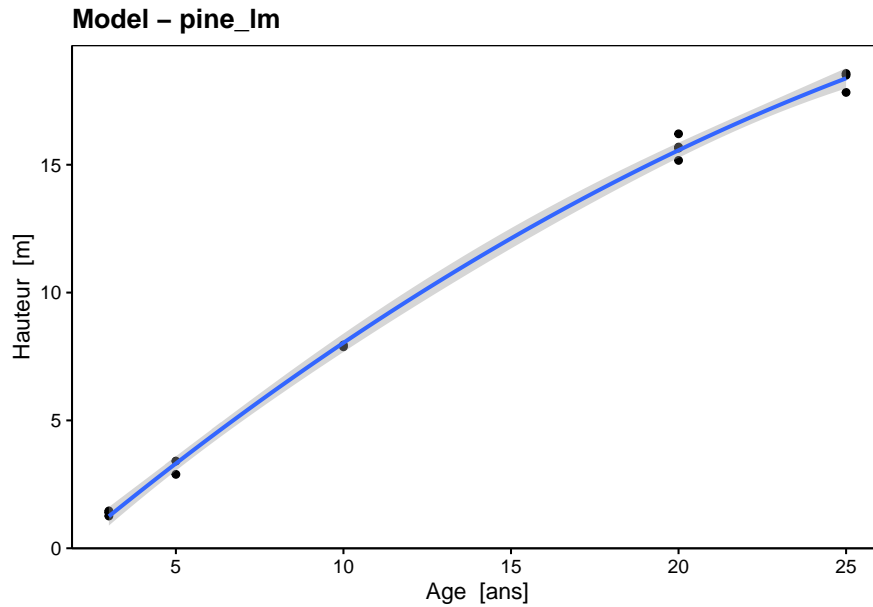
De nombreuses méthodes sont disponibles pour `{tabularise}` et `{chart}` et elles sont encore augmentées par d'autres packages dans l'univers SciViews comme `{modelit}` que nous utilisons ici ou encore `{inferit}`, `{exploireit}`. Pour les objets `lm`, `glm`, `nls`..., `{tabularise}` intègre également des équations dans son rendu en s'appuyant sur les packages `{equationomatic}` (Anderson, Heiss, and Sumners 2024) et `{equatags}`. Voici un exemple d'un tableau et d'un graphique 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)
# Modèle
pine_lm <- lm(data = pine, height ~ age + I(age^2))
pine_lm |> tabularise$tidy(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 $p$
$\alpha$	-2.0393	0.2895	-7.04	$2.14 \cdot 10^{-05}$ ***
$\beta_1$	1.1344	0.0584	19.41	$7.37 \cdot 10^{-10}$ ***
$\beta_2$	-0.0127	0.0021	-6.06	$8.14 \cdot 10^{-05}$ ***
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)
```



Vous retrouvez plusieurs autres exemples dans les vignettes associées à ces packages comme `modelit` et 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”. 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. *Equationomatic: Transform Models into 'LaTeX' Equations*. <https://github.com/datalorax/equationomatic>.
- Arel-Bundock, Vincent. n.d. *Tinytable: Simple and Configurable Tables in 'HTML', 'LaTeX', 'Markdown', 'Word', 'PNG', 'PDF', and 'Typst' Formats*. <https://vincentarelbundock.github.io/tinytable/>.
- 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>.