

# 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)

Dans l'univers de R, plusieurs packages tels que `{flextable}` (Gohel and Skintzos 2024), `{gt}` (Iannone et al. 2024) ou encore `{stargazer}` (Hlavac 2022) offrent des fonctionnalités de formatage de tableaux pour répondre aux normes de qualité des revues scientifiques. Bien que ces packages nécessitent de nombreuses lignes de code pour atteindre ces normes, ils permettent une personnalisation complète des tableaux. Cette situation est comparable à celle des packages dédiés à la génération de graphiques. Les graphiques sont réalisables avec les packages `{graphics}` (R Core Team 2023), `{lattice}` (Sarkar 2008) ou encore `{ggplot2}` (Wickham 2016). Les possibilités de personnalisation sont également nombreuses.

Au sein de la suite de packages SciViews (Grosjean 2023) qui contient par exemple `{data.io}`, `{svFlow}`, ou encore `{learnitdown}`, nous présentons `{tabularise}` pour les tableaux et `{chart}` pour les graphiques. Ces outils s'appuient respectivement sur `{flextable}` et `{ggplot2}` pour offrir des tableaux et des graphiques représentant différents objets R, tels que `lm` ou `glm`, par exemple.

Ces tableaux et graphiques supportent l'anglais et le français et prennent également en compte les labels des variables pour fournir du contenu prêt à être publié avec un minimum d'instructions R. Ces outils visent à simplifier et à accélérer le processus de préparation des contenus à la publication.

**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, 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 adaptés à la publication, 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 le cadre de SciViews, le package `{tabularise}`. Ce package permet d'obtenir des tableaux prêts à être publiés en une seule instruction pour une série d'objets dans R. Ces premières versions peuvent ensuite être remaniées à l'aide des instructions de `{flextable}`. `{tabularise}` gère l'anglais et le français et prend en compte automatiquement les labels des variables si ils sont présents dans l'objet. De nombreuses méthodes sont disponibles. Certaines offrent des fonctionnalités supplémentaires au-delà de réalisation de tableaux. Par exemple, pour les objets `lm`, `{tabularise}` intègre également des équations dans son rendu en s'appuyant sur le package `{equationomatic}` (Anderson, Heiss, and Sumners 2024). Ces caractéristiques en font un package intéressant pour la recherche et l'enseignement.

---

\*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

```
knitr::opts_chunk$set(echo = TRUE, warning=FALSE, echo=TRUE, message=FALSE)
# Configuration de l'environnement de travail avec SciViews
SciViews::R("model", lang = "fr")
# Charge plusieurs packages dont chart et tabularise
# Charge des packages supplémentaires pour la modélisation
# Spécifie que l'on souhaite les sorties au maximum en français
```

Prenons un cas pratique avec l'utilisation du tableau de données bien connu `penguins` du package `{palmerpenguins}`. Le tableau suivant présente les premières et dernières lignes des données.

```
penguins <- read("penguins", package = "palmerpenguins")
tabularise(penguins[1:5,1:4])
```

Espèce	Île	Longueur du bec [mm]	Largeur du bec [mm]
Adelie	Torgersen	39.1	18.7
Adelie	Torgersen	39.5	17.4
Adelie	Torgersen	40.3	18.0
Adelie	Torgersen		
Adelie	Torgersen	36.7	19.3

Une régression linéaire peut être réalisée afin d'étudier la variation de la longueur du bec en fonction de la largeur du bec et de l'espèce étudiée. Ce modèle peut être résumé par la fonction `summary()` ou encore mis sous forme de tableau avec la fonction `tidy()` du package `{broom}`. Ces deux sorties, bien que riches en informations, nécessitent d'être formatées pour figurer dans un rapport d'analyse. La solution proposée avec la fonction `tabularise()` permet d'obtenir un rendu plus abouti, comme le montrent les deux tableaux suivants, tout en respectant la classe des objets.

```
peng_model <- lm(data = penguins, bill_length ~ bill_depth + species)
```

Les solutions proposées avec la fonction `tabularise()` permettent d'obtenir un rendu plus abouti, comme le montrent les deux tableaux suivants, tout en respectant la classe des objets.

```
summary(peng_model)
```

```
##
## Call:
## lm(formula = bill_length ~ bill_depth + species, data = penguins)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.0300 -1.5828  0.0733  1.6925 10.0313
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    13.2164     2.2475   5.88 9.83e-09 ***
## bill_depth       1.3940     0.1220  11.43 < 2e-16 ***
## speciesChinstrap  9.9390     0.3678  27.02 < 2e-16 ***
## speciesGentoo    13.4033     0.5118  26.19 < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.518 on 338 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.7892, Adjusted R-squared:  0.7874
## F-statistic: 421.9 on 3 and 338 DF,  p-value: < 2.2e-16
```

```
summary(peng_model) |> tabularise()
```

Modèle linéaire				
$\alpha + \beta_1(\text{Largeur du bec [mm]}) + \beta_2(\text{Espèce}_{\text{Chinstrap}}) + \beta_3(\text{Espèce}_{\text{Gentoo}}) + \epsilon$				
Terme	Valeur estimée	Ecart type	Valeur de $t$	Valeur de $p$
$\alpha$	13.22	2.248	5.88	$9.83 \cdot 10^{-09}***$
$\beta_1$	1.39	0.122	11.43	$< 2 \cdot 10^{-16}***$
$\beta_2$	9.94	0.368	27.02	$< 2 \cdot 10^{-16}***$
$\beta_3$	13.40	0.512	26.19	$< 2 \cdot 10^{-16}***$

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

Etendue des résidus : [-8.03, 10.03]

Ecart type des résidus : 2.518 pour 338 degrés de liberté

$R^2$  multiple : 0.7892 -  $R^2$  ajusté : 0.7874

Statistique  $F$  : 421.9 pour 3 et 338 ddl - valeur de  $p$  : < 2.22e-16

```
tidy(peng_model)
```

```
## # A tibble: 4 x 5
##   term          estimate std.error statistic  p.value
##   <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)    13.2      2.25      5.88 9.83e- 9
## 2 bill_depth      1.39     0.122     11.4 8.66e-26
## 3 speciesChinstrap  9.94     0.368     27.0 1.81e-86
## 4 speciesGentoo   13.4      0.512     26.2 2.42e-83
```

```
tabularise(peng_model, type = "tidy")
```

Modèle linéaire				
Longueur du bec [mm] = $\alpha + \beta_1(\text{Largeur du bec [mm]}) + \beta_2(\text{Espèce}_{\text{Chinstrap}}) + \beta_3(\text{Espèce}_{\text{Gentoo}}) + \epsilon$				
Terme	Valeur estimée	Ecart type	Valeur de $t$	Valeur de $p$
$\alpha$	13.22	2.248	5.88	$9.83 \cdot 10^{-09***}$
$\beta_1$	1.39	0.122	11.43	$< 2 \cdot 10^{-16***}$
$\beta_2$	9.94	0.368	27.02	$< 2 \cdot 10^{-16***}$
$\beta_3$	13.40	0.512	26.19	$< 2 \cdot 10^{-16***}$
0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05				

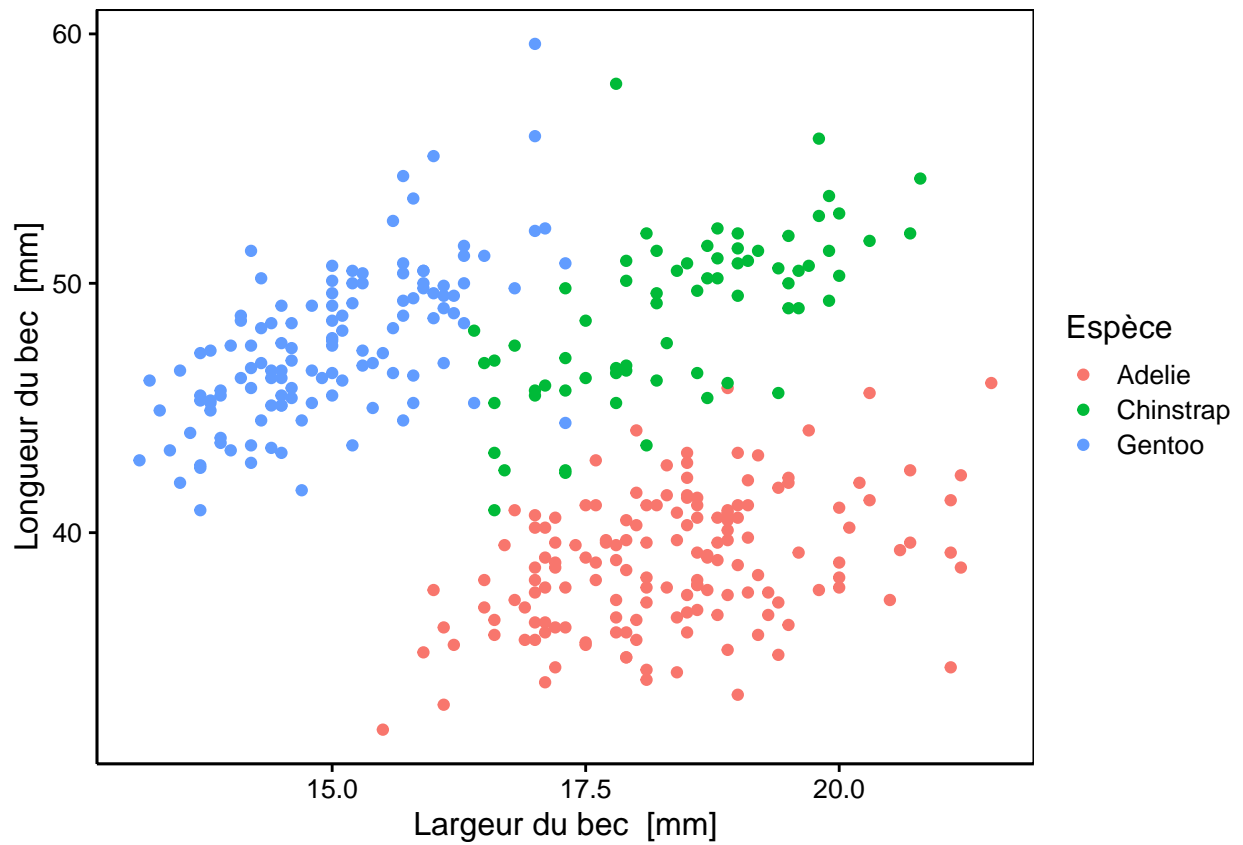
Vous retrouvez plusieurs autres exemples d'utilisation dans l'enseignement dans les modules relatif à la régression linéaire des cours de Sciences des données biologique données à l'Université de Mons.

Vous retrouvez plusieurs autres exemples d'utilisation dans l'enseignement dans les modules relatifs à la régression linéaire des cours de sciences des données biologiques donnés à l'Université de Mons.

En ce qui concerne les graphiques, la communauté R propose principalement trois solutions : les packages R {graphics} (R Core Team 2023), {lattice} (Sarkar 2008) et {ggplot2} (Wickham 2016). Le package {chart} propose une interface unique pour unifier ces trois moteurs graphiques, mais s'appuie principalement sur {ggplot2}. L'interface formule est utilisée, ce qui permet de se rapprocher des instructions que l'on peut utiliser, par exemple, dans la fonction `lm()`. Tout comme {tabularise}, {chart} s'adapte à l'anglais et au français. Il utilise les labels et les unités des variables pour libeller automatiquement les axes du graphique. De nombreuses méthodes ont été développées pour fournir des graphiques adaptés à toute une série d'objets.

En reprenant l'exemple réalisé ci-dessus, on peut le visualiser sous la forme d'un nuage de points avec les instructions suivantes. L'interface formule permet, entre autres, d'obtenir une haute ressemblance avec la formule proposée par la fonction `lm()`.

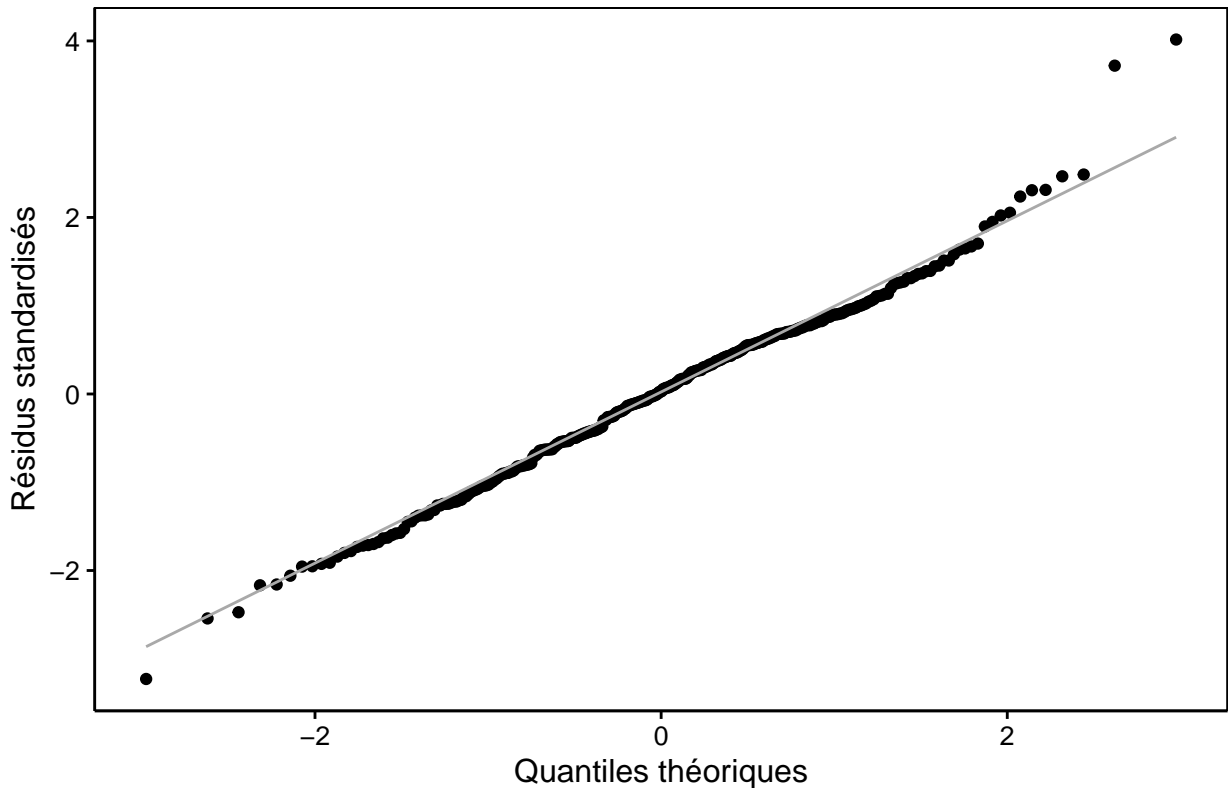
```
chart(data = penguins, bill_length ~ bill_depth %col=% species) +
  geom_point()
```



Tout comme `tabularise`, `chart()` propose des graphiques spécifiques lors de l'utilisation de certains objets dans R.

```
chart(peng_model, type = "qqplot")
```

## Graphique quantile–quantile des résidus – peng\_model



Nous utilisons le package `{chart}` pour la grande majorité des graphiques que nous réalisons dans les 34 modules des cours de sciences des données biologiques de l’Université de Mons.

En conclusion, `{tabularise}` et `{chart}` sont deux packages développés dans le cadre de SciViews pour faciliter et améliorer les sorties tabulaires et graphiques de R, prêts à être publiés.

## 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. *Equationmatic: Transform Models into 'LaTeX' Equations*. <https://github.com/datalorax/equationmatic>.
- 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/>.
- Hlavac, Marek. 2022. *Stargazer: Well-Formatted Regression and Summary Statistics Tables*. Bratislava, Slovakia: Social Policy Institute. <https://CRAN.R-project.org/package=stargazer>.
- 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>.
- R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Sarkar, Deepayan. 2008. *Lattice: Multivariate Data Visualization with r*. New York: Springer. <http://lmdvr.r-forge.r-project.org>.
- Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>.