

A Test Bookdown Book

Philippe Grosjean

2018-11-28

Contents

Preface	5
1 Introduction	7
1.1 Book-wide references	7
1.2 Citations	9
1.3 Cache long computations	9
1.4 Shiny app	9
1.5 Learnr with one exercise and one quizz	10
1.6 H5P content	10
1.7 Try to add Moodle content	10
1.8 Add a Gist	10
1.9 Embed a youtube video	10
1.10 Special blocks	10
2 Diving In	13
3 Technical Details	15
4 Visualisation I	17
4.1 Nuage de points	18
4.2 Graphiques dans R Markdown	24

Preface

This is just a sandbox to test various bookdown features. The online version of this book is licensed under the Creative Commons Public-Domain.

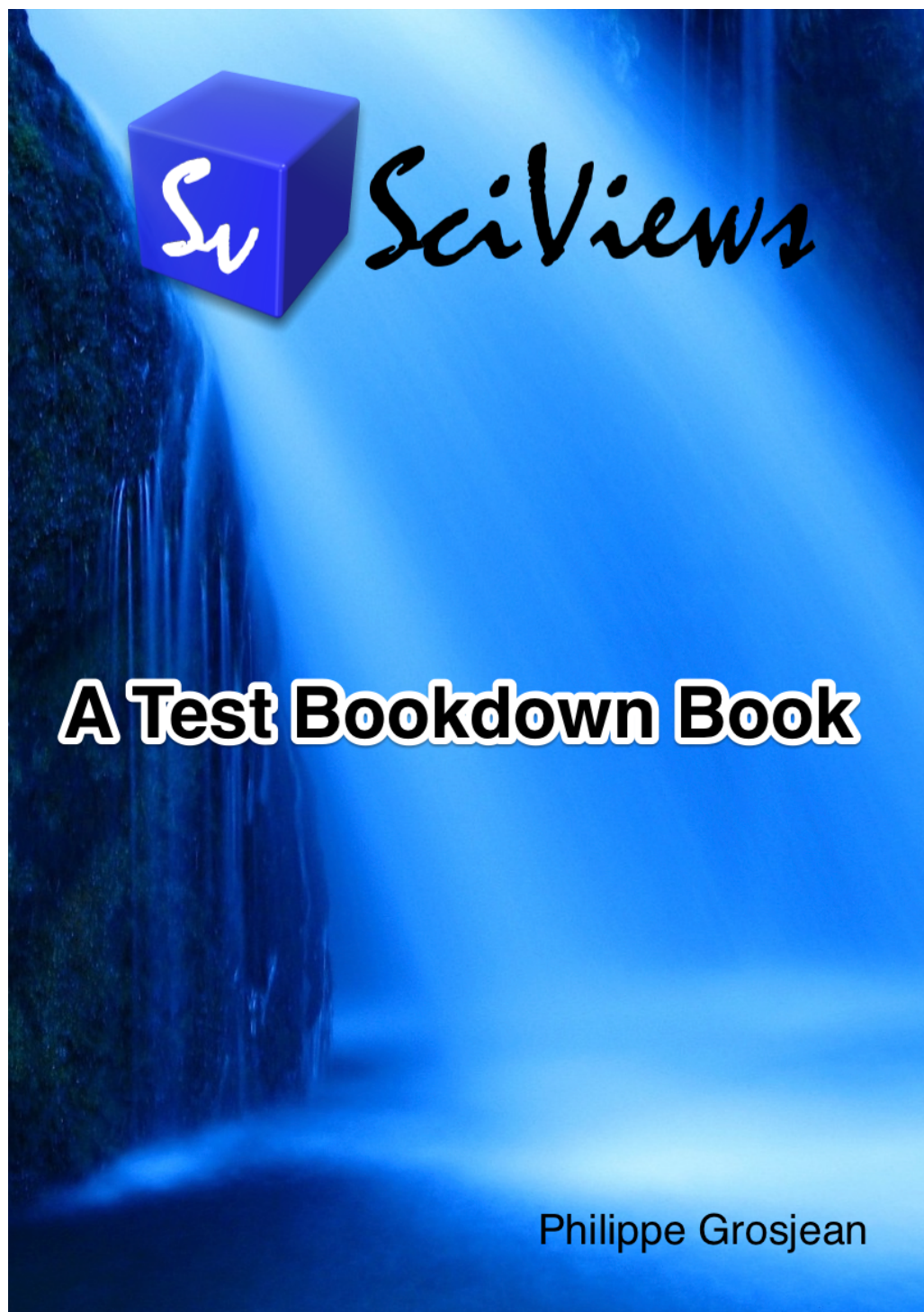


Figure 1:

Chapter 1

Introduction

This is a bookdown chapter mostly for demonstration and testing ... I started from here, then got some pieces from here, and finally skimmed the bookdown book and took some notes hereunder. I had also a look at this blog.

SciViews

~~To convince *someone* to switch to Markdown, show him this. A short discussion of `html/active documents` *versus* `pdf/static printable pages` should be nice here.~~

Now, let's look at special formattings provided by bookdown.

1.1 Book-wide references

With R Markdown, it is only possible to reference items inside the same document, and there is no numbered figures, tables or equations (at least if LaTeX code is not used, but then, you can only generate a PDF). Bookdown fills the gap with new formatting instructions that work across all documents of the book. It also provides updated versions of R Markdown output formats that support numbered figures, tables and equations, e.g., `html_document2` to replace `html_document`, for instance.

1.1.1 Reference to chapter and titles

You can reference **chapters** and **titles** using 1. The book is organized in chapters and all numbers are by chapters. However, there are special level 1 title you can use:

- **Parts:** use `# (PART) Part I {-}` just before the first title of that part.
- **Appendix:** use `# (APPENDIX) Appendix {-}` just before appendices titles. They will be numbered separately.

1.1.2 Numbered figures and tables

Figures and **tables** are in their own environments:

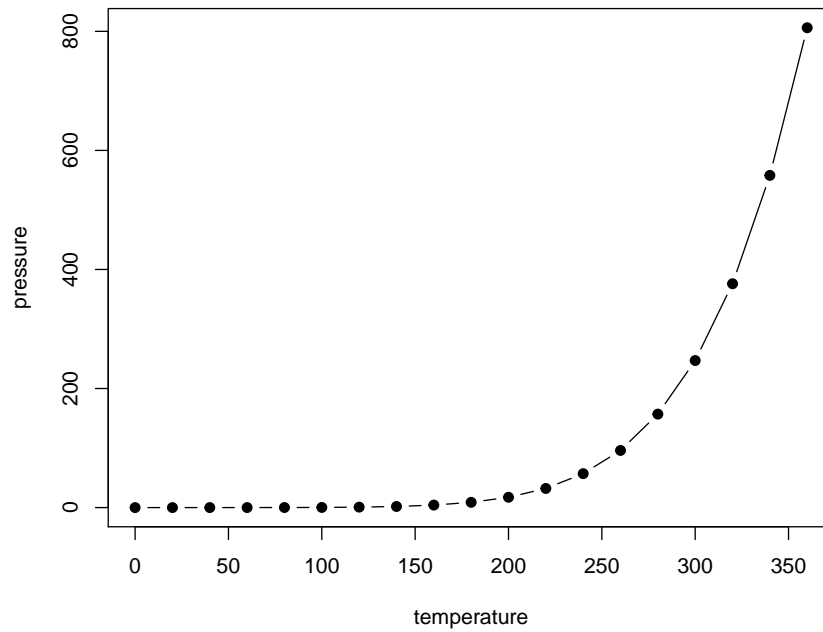


Figure 1.1: Here is a nice figure!

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 1.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 1.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

1.1.3 Numbered equations

To number equations and allow to refer to them, use an `equation` environment and label them with the syntax `(\#eq:label)`:

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (1.1)$$

... and here, I refer to eq. (1.1). In the vase equations are not labelled, use the `equation*` environment instead.

Table 1.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

1.2 Citations

You can reference citations, too. For example, we are using the **bookdown** package (Xie, 2018) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

1.3 Cache long computations

If some computation is time-consuming, we could consider to cache it:

```
# A verrrry long computation!
1 + 1
```

```
## [1] 2
```

1.4 Shiny app

A shiny application directly inside the book using an iframe. This application demonstrates how an histogram changes when the number of classes is modified.

1.5 Learnr with one exercise and one quizz

A simple learnr document from shinyapps.io:

1.6 H5P content

H5P content can be displayed in iframes. Here is an example of a multiple choice widget:

Here is a “drag words” widget:

Here is a video widget:

1.7 Try to add Moodle content

1.8 Add a Gist

and for an R markdown document:

1.9 Embed a youtube video

1.10 Special blocks



This is a **note**.



This is an **information**.



This is a **warning**.



This block can be used in case of **error**.



This is related to **Windows**.



This is related to **MacOS**.



This is related to **Linux**.



This is related to the **BioDataScience** package.



This is a `block2` construct related to `SciViews` or `SciViews::R`:

- item 1
- item 2
- item 3



This is a section related to the **SciViews Box...**

TODO: browse the bookdown book from 2.4 Figures on...

Chapter 2

Diving In

Now let's talk details.

Chapter 3

Technical Details

Now I'll teach you some crazy math, but I need to work it out first...

Chapter 4

Visualisation I

4.0.0.0.1 Objectifs

- Découvrir –et vous émerveiller de– ce que l’on peut faire avec le logiciel R (R Core Team, 2018)
- Savoir réaliser différentes variantes d’un graphique en nuage de points dans R avec la fonction `chart()`
- Découvrir le format R Markdown (Allaire et al., 2018) et la recherche reproductible
- Intégrer ensuite des graphiques dans un rapport et y décrire ce que vous observez
- Comparer de manière critique un flux de travail “classique” en biologie utilisant Microsoft Excel et Word avec une approche utilisant R et R Markdown ; Prendre conscience de l’énorme potentiel de R

4.0.0.0.2 Prérequis

Si ce n’est déjà fait, vous devez installer et vous familiariser avec la ‘SciViews Box’, RStudio, Markdown. Vous devez aussi maîtriser les bases de Git et de GitHub (avoir un compte GitHub, savoir cloner un dépôt localement, travailler avec GitHub Desktop pour faire ses “commits”, “push” et “pull”). L’ensemble de ces outils a été abordé lors de la création de votre site personnel professionnel du module 1.

Avant de poursuivre, vous allez devoir découvrir les premiers rudiments de R afin de pouvoir réaliser par la suite vos premiers graphiques. Pour cela, vous aurez à lire attentivement et effectuer tous les exercices de deux tutoriels¹.



Démarrez la SciViews Box et RStudio. Dans la fenêtre **Console** de RStudio, entrez l’instruction suivante suivie de la touche **Entrée** pour ouvrir le tutoriel concernant les bases de R :

```
BioDataScience::run("02a_base")
```

Ensuite, vous pouvez également parcourir le tutoriel qui vous permettra de découvrir R sur base d’une analyse concrète (cliquez dans la fenêtre **Console** de RStudio et appuyez sur la touche **ESC** pour reprendre la main dans R à la fin d’un tutoriel) :

```
BioDataScience::run("02b_decouverte")
```

¹Reportez-vous à l’Appendice ?? pour apprendre à utiliser ces tutoriels.

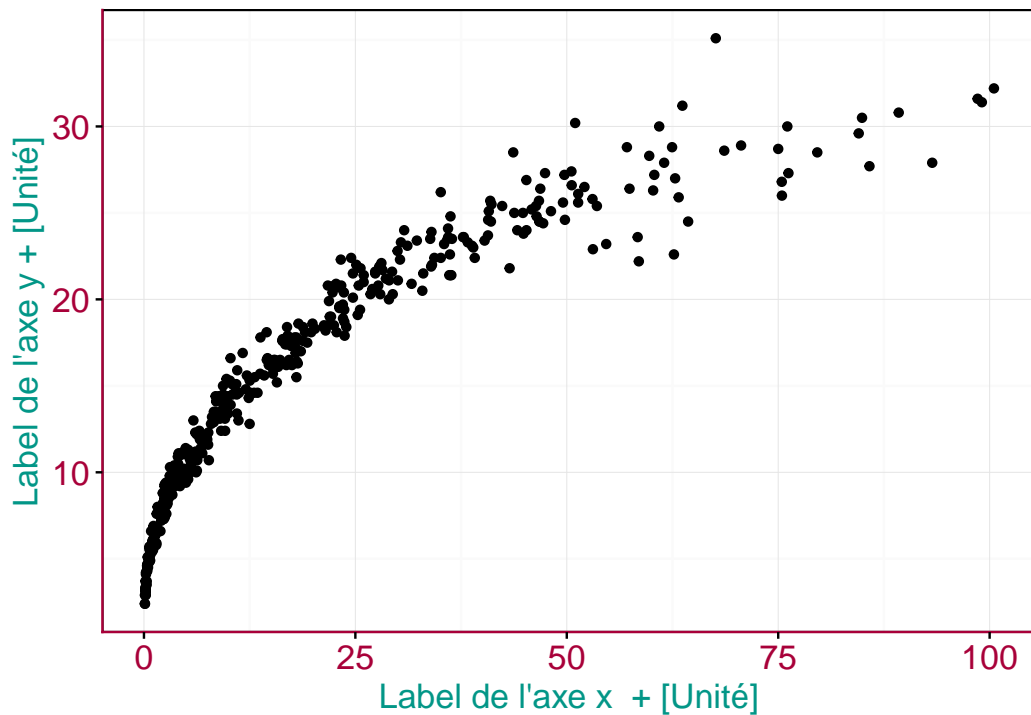


Figure 4.1: Exemple de graphique en nuage de points. Des éléments essentiels sont ici mis en évidence en couleurs (voir texte).

(BioDataScience est un package R spécialement développé pour ce cours et que vous avez dû installer lors de la configuration de votre SciViews Box, voir Appendice ??).

4.1 Nuage de points

Dès que vous vous sentez familiarisé avec les principes de base de R, vous allez pouvoir réaliser assez rapidement des beaux graphiques. Par exemple, si vous souhaitez représenter une variable numérique en fonction d’une autre variable numérique, vous pouvez exprimer cela sous la forme d’une **formule**²

$$y \sim x$$

que l’on peut lire “y en fonction de x”. Pour les deux variables numériques x et y , la représentation graphique la plus classique est le **nuage de points** (voir Fig. 4.1 pour un exemple).

Les éléments indispensables à la compréhension d’un graphique en nuage de points sont mis en évidence à la Fig. 4.1 :

- Les axes avec les graduations (en rouge),

²Dans R, une **formule** permet de spécifier les variables avec lesquelles on souhaite travailler, et leur rôle. Par exemple ici, la variable x sur l’axe des abscisses et la variable y sur l’axe des ordonnées.

- les labels et les unités des axes (en bleu).

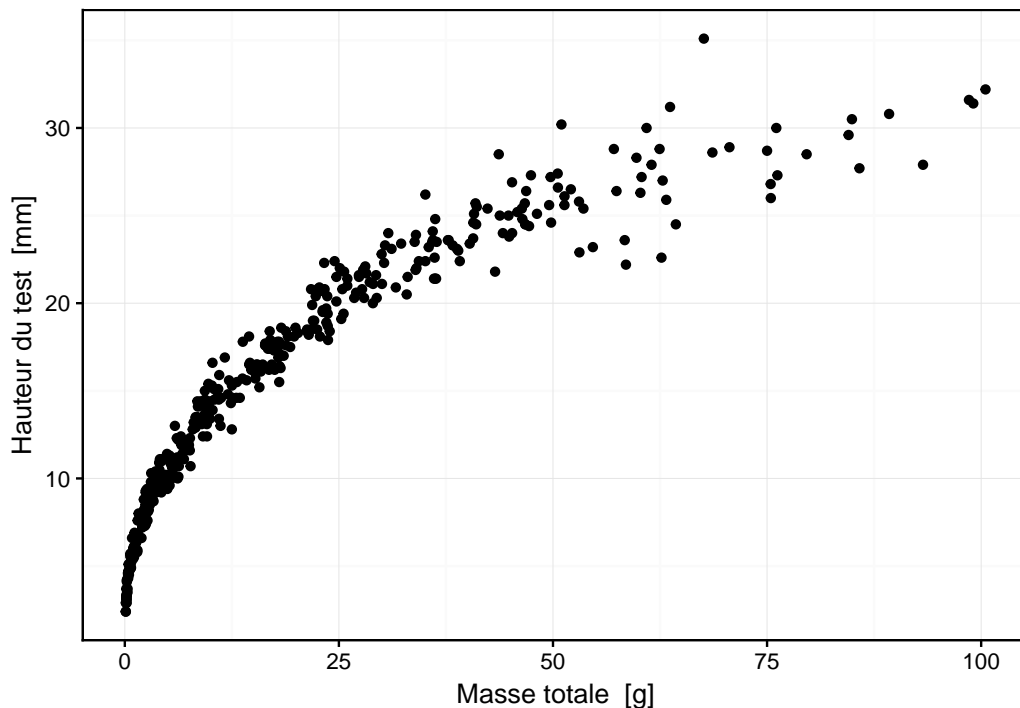
Les instructions dans R pour produire un tel nuage de point sont :

```
# Chargement de SciViews::R
SciViews::R
# Importation du jeu de données
urchin <- read("urchin_bio", package = "data.io", lang = "fr")
```

```
DT::datatable(urchin , filter = 'top', class = c('compact', 'cell-border'), rownames = FALSE)
```

```
# # A tibble: 421 x 19
#   origin diameter1 diameter2 height buoyant_weight weight solid_parts
#   <fct>      <dbl>      <dbl> <dbl>      <dbl>    <dbl>      <dbl>
# 1 Fishe~      9.9       10.2    5          NA    0.522      0.478
# 2 Fishe~     10.5       10.6    5.7        NA    0.642      0.589
# 3 Fishe~     10.8       10.8    5.2        NA    0.734      0.677
# 4 Fishe~      9.6        9.3    4.6        NA    0.370      0.344
# 5 Fishe~     10.4       10.7    4.8        NA    0.610      0.559
# 6 Fishe~     10.5       11.1    5          NA    0.610      0.551
# 7 Fishe~     11        11     5.2        NA    0.672      0.605
# 8 Fishe~     11.1       11.2    5.7        NA    0.703      0.628
# 9 Fishe~      9.4        9.2    4.6        NA    0.413      0.375
# 10 Fishe~    10.1        9.5    4.7        NA    0.449      0.398
# # ... with 411 more rows, and 12 more variables: integuments <dbl>,
# #   dry_integuments <dbl>, digestive_tract <dbl>,
# #   dry_digestive_tract <dbl>, gonads <dbl>, dry_gonads <dbl>,
# #   skeleton <dbl>, lantern <dbl>, test <dbl>, spines <dbl>,
# #   maturity <int>, sex <fct>
```

```
# Réalisation du graphique
chart(data = urchin, height ~ weight) +
  geom_point()
```



La fonction `chart()` n'est pas accessible dans R de base, mais l'extension chargée via l'instruction `SciViews::R` rend cette fonction disponible. Elle requiert comme argument le jeu de donnée (`data = urchin`, c'est un objet `dataframe` ou `tibble` dans le langage de R), ainsi que la formule à employer dans laquelle vous avez indiqué le nom des variables que vous voulez sur l'axe des ordonnées à gauche et des abscisses à droite de la formule, les deux membres étant séparés par un "tilde" (`~`). Vous voyez que le jeu de données contient beaucoup de variables (les titres des colonnes du tableau en sortie). Parmi toutes ces variables, nous avons choisi ici de représenter `height` en fonction de `weight`, la hauteur en fonction de la masse des oursins. Jusqu'ici, nous avons spécifié *ce que* nous voulons représenter, mais pas encore *comment* (sous quelle apparence), nous voulons les matérialiser sur le graphique. Pour un nuage de points, nous voulons les représenter sous forme de ... points ! Donc, nous devons ajouter la fonction `geom_point()` pour indiquer cela.

4.1.1 Le nuage de points en vidéo

Vous trouverez une vidéo ci-dessous vous expliquant la création du nuage de points dans R sur ce jeu de données, en analysant d'autres variables.

Retrouvez la vidéo via le lien suivant : <https://www.youtube.com/watch?v=-QzG3Xr202w>

Cette vidéo ne vous a montré que les principaux outils disponibles lors de la réalisation de graphiques. Soyez curieux et expérimentez par vous-même !

4.1.1.0.1 A vous de jouer

Retrouvez une séance d'exercice via une application shiny
uniquement disponible sur la version en ligne

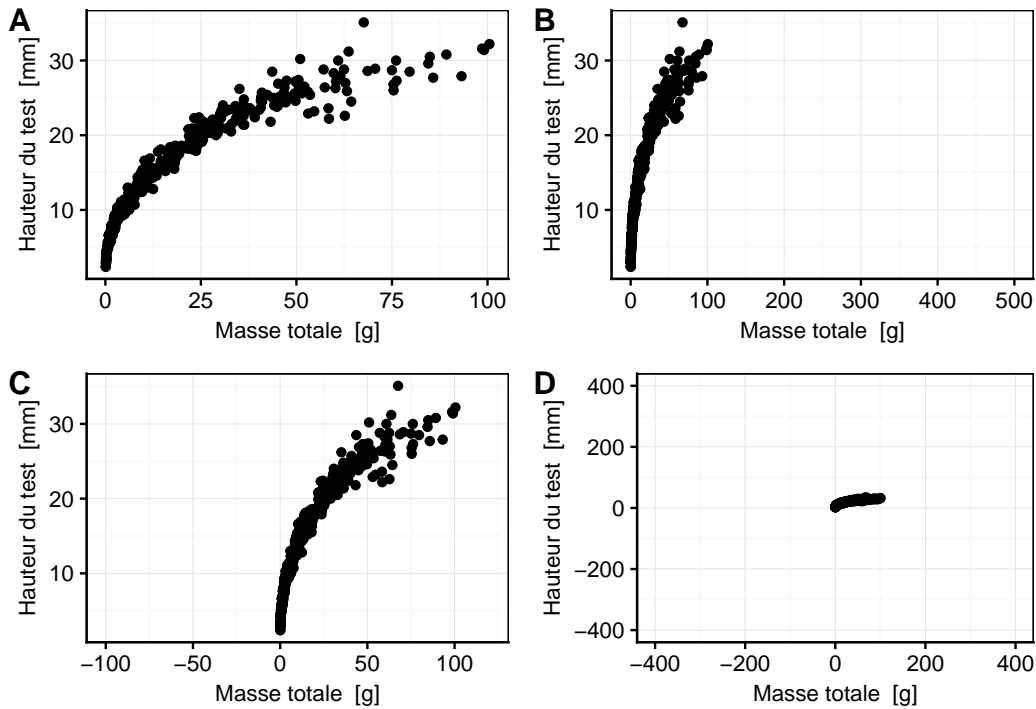


Figure 4.2: Piège du nuage de points. A) graphique initial montrant la variation de la hauteur [mm] en fonction de la masse [g]. B) graphique A avec la modification de l'échelle de l'axe X. C) Graphique A avec une seconde modification de l'axe X. D) Graphique A avec modification simultanée des deux axes.



Dans la fenêtre **Console** de RStudio, entrez l'instruction suivante suivie de la touche **Entrée** pour ouvrir le tutoriel concernant le nuage de points : `BioDataScience::run("02c_nuage_de_points")` N'oubliez pas d'appuyer sur la touche **ESC** pour reprendre la main dans R à la fin d'un tutoriel)

4.1.2 Echelles d'un graphiques

Vous devez être vigilant lors de la réalisation d'un nuage de point particulièrement sur l'étendue des valeurs présentées sur vos axes. Vous devez utiliser votre expertise de biologiste pour vous posez les deux questions suivantes :

- Est ce que l'axe représente des valeurs plausibles de hauteurs et de masses de ces oursins appartenant à l'espèce *Paracentrotus lividus* ?
- Quels est la précision des mesures effectuées ?

Retrouvez une séance d'exercice via une application shiny
uniquement disponible sur la version en ligne

Dans certains cas, la forme du nuage de points peut être distordu par la présence de valeurs aberrantes. Ce n'est pas le cas ici, mais nous pouvons le simuler en distendant artificiellement soit l'axe X, soit l'axe Y, soit les deux :

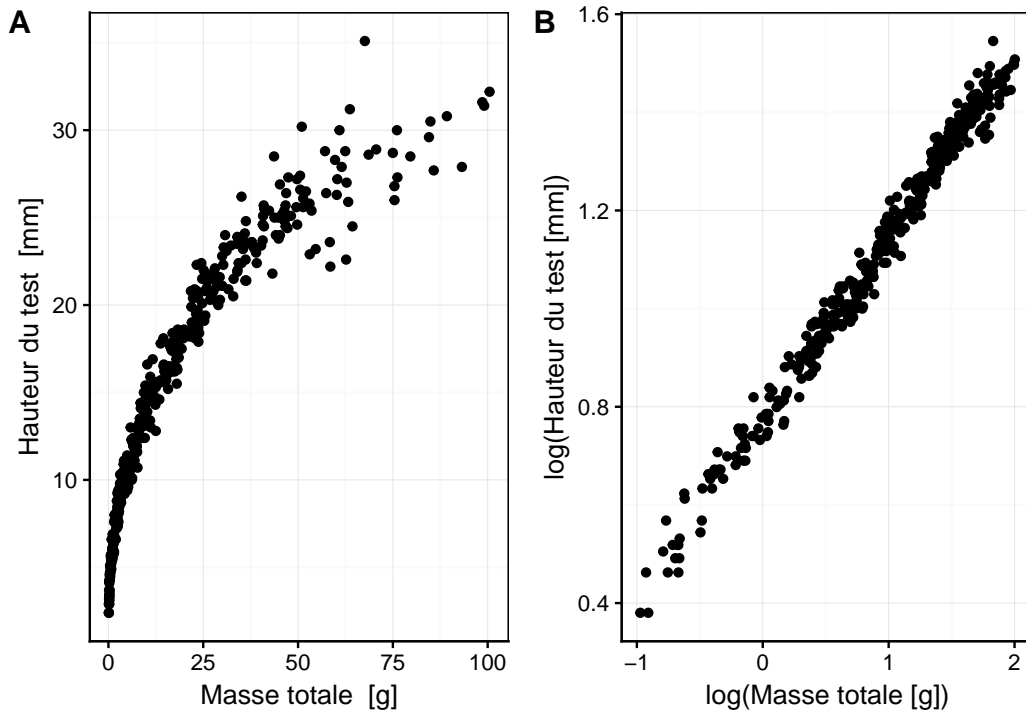


Figure 4.3: A) Hauteur [mm] en fonction de la masse [g] d'oursins violets. B) Logarithme en base 10 de la hauteur [mm] en fonction du logarithme en base 10 de la masse [g] de ces mêmes oursins.

4.1.3 Transformation des données

Vous avez la possibilité d'appliquer une transformation de vos données (il est même conseillé de le faire) afin qu'elles soient plus facilement analysables. Par exemple, il est possible d'utiliser des fonctions de puissance, racines, logarithmes, exponentielles³ pour modifier l'apparence du nuage de points dans le but de le rendre plus linéaire (car il est plus facile d'analyser statistiquement des données qui s'alignent le long d'une droite). Par exemple, sur nos données de hauteurs et masses d'oursins, la transformation double-logarithmique ($\log(x)$ et $\log(Y)$) fonctionne très bien pour rendre le nuage de points plus linéaire :

```
# Réalisation du graphique de la hauteur en fonction de la masse
a <- chart(urchin, height ~ weight) +
  geom_point()
# Application du logarithme sur les deux variables représentées
b <- chart(urchin, log10(height) ~ log10(weight)) +
  geom_point() +
  labs(x = "log(Masse totale [g])", y = "log(Hauteur du test [mm])")
# Assemblage des graphiques
combine_charts(list(a, b))
```

³Pour les proportions (`prop`) ou les pourcentages (`perc`) (valeurs bornées entre 0 et 1 ou 0 et 100%) la transformation arc-sinus est souvent utilisée : $prop = \arcsin \sqrt{prop}$ ou $perc = \arcsin \sqrt{perc/100}$.

4.1.3.0.1 Pièges et astuces

RStudio permet de récupérer rapidement des instructions à partir d'une banque de solutions toutes prêtes. Cela s'appelle des **snippets**. Vous avez une série de snippets disponibles dans la SciViews Box. Celui qui vous permet de réaliser un graphique en nuage de points s'appelle `.cbxy` (pour `chart` -> `bivariate` -> `xy-plot`). Entrez ce code et appuyez ensuite sur la tabulation dans un script R, et vous verrez le code remplacé par ceci dans la fenêtre d'édition :

```
chart(data = DF, YNUM ~ XNUM) +
  geom_point()
```

Vous avez à votre disposition un ensemble de snippets que vous pouvez retrouver dans l'aide-mémoire consacré à **SciViews**. Vous avez également à votre disposition l'aide-mémoire sur la visualisation des données (**Data Visualization Cheat Sheet**) qui utilise la fonction `ggplot()` plutôt que `chart()` et une interface légèrement différente pour spécifier les variables à utiliser pour réaliser le graphique (`aes(x = ..., y = ...)`).

4.1.3.0.2 A vous de jouer

Une nouvelle tâche va vous être demandée ci-dessous en utilisant GitHub Classroom ???. Cette tâche est un travail **individuel**. Une fois votre assignation réalisée, faites un clone local de votre dépôt et placez-le dans le sous-dossier **projects** de votre dossier partagé avec la SciViews Box **shared**. Vous aurez alors un nouveau projet RStudio ??



Les instructions R que vous expérimentez dans un learnR peuvent être employées également dans un script d'analyse. Sur base du jeu de données `urchin_bio`, explorez différents graphiques en nuages de points. Utilisez l'URL suivante pour accéder à votre tâche : - https://classroom.github.com/a/eYrXLy_u

Inspirez-vous du script dans le dépôt `sdd1_iris`. Vous devez commencer par faire un “fork” du dépôt, puis un clone sur votre ordinateur en local pour pouvoir l'utiliser.

– https://github.com/BioDataScience-Course/sdd1_iris

Prêtez une attention toute particulière à l'organisation d'un script R. En plus des instructions R, il contient aussi sous forme de commentaires, un titre, la date de la dernière mise à jour, le nom de l'auteur, et des sections qui organisent de façon claire le contenu du script. A ce sujet, vous trouverez des explications détaillées concernant l'utilisation des scripts R dans l'annexe ??.

4.1.3.0.3 Pour en savoir plus

- Visualisation des données dans R for Data Science. Chapitre du livre portant sur la visualisation des données, en anglais.
- ggplot2 nuage de point. Tutoriel en français portant sur l'utilisation d'un nuage de points avec le package `ggplot2` et la fonction `geom_point()`.
- Fundamentals of Data Visualization. Un livre en anglais sur les fondamentaux de la visualisation graphique.
- R Graphics Cookbook - Chapter 5: Scatter Plots. Un chapitre d'un livre en anglais sur l'utilisation du nuage de points.
- `geom_point()`. La fiche technique de la fonction (en anglais).

4.1.3.0.4 Testez vos acquis



Dans la fenêtre **Console** de RStudio, entrez l'instruction suivante et puis appuyez sur la touche **Entrée** pour ouvrir le tutoriel de challenge concernant le nuage de points :

```
BioDataScience::run("02d_np_challenge")
```

N'oubliez pas de vous enregistrer (login GitHub et email UMONS) au début, et d'appuyer sur la touche **ESC** pour reprendre la main dans R à la fin d'un tutoriel.

4.2 Graphiques dans R Markdown

Un fichier R Markdown est un fichier avec une extension `.Rmd`. Il permet de combiner le langage Markdown que vous avez déjà abordé au premier module avec du code R, tel que celui utilisé dans la première partie de ce module 2.

4.2.1 R Markdown en vidéo

La vidéo ci-dessous vous montre ce qu'est R Markdown, un format hybride entre Markdown et R bien pratique pour inclure vos graphiques directement dans un rapport. Elle vous montre aussi comment transformer un script R en document R Markdown (ou R Notebook, qui en est une variante).

```
# Retrouvez une séance d'exercice lié à une vidéo
# disponible via l'url suivant : https://h5p.org/h5p/embed/369438
```

Les balises spéciales R Markdown à retenir sont les suivantes :

- en entrée de chunk R : ````${r}```` seul sur une ligne. Il est aussi possible de rajouter un nom, par exemple, ````${r} graphique1```` et/ou des options, par exemple, ````${r}, echo=FALSE, results='hide'```` pour cacher et le code et le résultat dans le rapport),
- en sortie de chunk R : ````${r}```` seul sur une ligne.

Vous devez bien entendu avoir autant de balises d'entrée que de balises de sortie. Des explications plus détaillées se trouvent dans l'annexe ?? dédiée au R Markdown. De plus, l'écriture d'un rapport d'analyse scientifique doit respecter certaines conventions. Vous trouverez des explications à ce sujet dans l'annexe ??.



Vous ne devez bien évidemment pas commencer avec un script R. Vous pouvez commencer d'emblée avec un R Markdown/R Notebook et écrire vos instructions R directement dedans. Il vous est toujours possible d'exécuter ces instructions ligne après ligne dans la fenêtre **Console** pour les tester tout comme à partir d'un script R.

4.2.1.0.1 Pour en savoir plus

- Communicating results with R Markdown explique la même chose que dans la vidéo, avec plus de détails et des liens vers d'autres documents utiles (en anglais).

- What is R Markdown?. Vidéo en anglais + site présentant les différentes possibilités, par les concepteurs de R Markdown (RStudio).
- Introduction to R Markdown. Tutoriel en anglais, par RStudio.
- R Markdown : the definitive guide est **le** manuel par excellence pour R Markdown (en anglais uniquement, malheureusement).
- Aide-mémoire R Markdown: dans les menus de RStudio : **Help -> Cheatsheets -> R Markdown Cheat Sheet**
- Référence rapide à Markdown : dans les menus RStudio **Help -> Markdown Quick Reference**
- Introduction à R Markdown. Présentation en français par Agrocampus Ouest - Rennes.
- Le langage R Markdown. Introduction en français concise, mais relativement complète.
- Reproducible reports with R Markdown. Une explication en anglais de la raison d'être de R Markdown.
- Why I love R Notebooks explique (en anglais) pourquoi le format R Notebook est particulièrement bien adapté à la science des données.

4.2.1.0.2 A vous de jouer

- Vous allez maintenant manipuler un **R Notebook** pour construire de manière interactive une analyse en même temps que le rapport associé.



Partez du projet `sdd1_urchin_bio` que vous avez obtenu via le lien GitHub Classroom dans la première partie de ce module. Votre objectif est de comprendre les données proposées, en utilisant des visualisations graphiques appropriées et en documentant le fruit de votre étude dans un rapport R Notebook. Utilisez le graphique en nuage de points que vous venez d'étudier, bien sûr, mais vous êtes aussi encouragés à expérimenter d'autres formes de visualisation graphique.

- Flux de travail “classique” en biologie (Microsoft Excel et Word) comparé à R et R Markdown.

Une nouvelle tâche va vous être demandée ci-dessous en utilisant GitHub Classroom **??**. Cette tâche est un travail **en équipe**. Une fois votre assignation réalisée, faites un clone de votre dépôt et placez-le dans le dossier `shared/projects`.



Comparez le workflow classique en biologie via Microsoft Office avec l'utilisation de R - R Markdown en suivant les explications dans le fichier `README.mddu` dépôt accessible depuis : - <https://classroom.github.com/g/2Cii2dws>

Bibliography

- Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., and Chang, W. (2018). *rmarkdown: Dynamic Documents for R*. R package version 1.10.
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.
- Xie, Y. (2018). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.7.