

Présentations efficaces dans le cadre scolaire et universitaire

Julian Steinhart

2022-08-04

Table des matières

1	Introduction	2
1.1	Compétences visées	2
2	Plan du cours	2
3	Concevoir la présentation	2
3.1	Les diapos	2
3.2	Vos notes	4
3.3	Le polycopié	4
4	Get your GitBook	5
5	Editing the book	10
5.1	Creating new chapters	10
5.2	Linking across chapters	11
5.3	Advanced editing	11
6	Figures and tables	11
7	Examples	13
7.1	Statistics with R (H. Quene)	13
7.2	Theory Construction and Statistical Modeling (C. J. van Lissa)	13
7.3	Doing Meta-Analysis in R (C. J. van Lissa)	13
7.4	Métodos quantitativos em Psicologia com R (L. Anunciação)	13
8	Open Educational Resources	13
9	Compatibility with existing systems	14
9.1	Add a hyperlink	14
9.2	Embed the whole book	14
10	License your GitBook	14

1 Introduction

1.1 Compétences visées

Les étudiantes savent:

- Planifier une présentation de manière autonome.
- Créer des diaporamas simples, attractifs et efficaces.
- Délivrer une présentation captivante et convaincante.
- Justifier les choix faites dans une présentation. orale

2 Plan du cours

Table 1: Dates et thèmes du cours

10/09/2022	Introduction Planifier la présentation Créer la présentation Créer le diaporama Le polycopié Comment choisir un grpahe Donner la présentation Répondre À des questions Espace pour vous Espace pour vous Espace pour vous
------------	---

3 Concevoir la présentation

Nous avons déjà vu qu'on ne démarre pas la préparation des sa présentation en cliquant sur PowerPoint. Cependant les diaporamas jouent un rôle important pour la présentation. Les diaporamas servent avant tout à souligner ce que vous dites. Elles ne sont ni notes, ni polycopié pour le public. Dans cette partie nous allons voir comment concevoir ces trois éléments essentielles pour la présentation.

Dans certains cas il sera même possible que vous n'avez pas besoin de diapos. Quand vous voulez discuter en détail de certains diagrammes, graphiques ou faits, il est souvent envisageable de fournir seulement le polycopié et de baser son discours sur lui.

3.1 Les diapos

Dans cette section nous allons voir d'après quels principes on peut concevoir sa présentation, afin de souligner de manière efficace son discours.

- 3.1.1 La grille de base
- 3.1.2 La relation “signal-noise”
- 3.1.3 Picture superiority effect
- 3.1.4 Contraster, aligner, rapprocher et répéter



Figure 1: Est-ce que le titre de la présentation est bien lisible ?

Contraster :

3.1.4.1 Que dire sur les points clés ?

3.1.5 Les éléments du design

3.1.5.1 Répartition de l'espace

3.1.5.2 Typographie Concevez votre présentation pour les gens assis au fond de la salle : Bigger si better! Pour vérifier si votre police de caractères est assez grande vous pouvez imprimer trois diapos sur une page ([Tutorial | YouTube](#)). Quand tout est bien lisible, il le sera aussi quand vous délivrez la présentation.

Quelle police de caractères ? On distingue des polices avec serifs ou sans serifs.

Video provides a powerful way to help you prove your point.

Figure 2: Une police avec serifs.

Video provides a powerful way to help you prove your point.

Figure 3: Une police sans serifs.

Avec la qualité des vidéoprojecteurs d'aujourd'hui il est possible de choisir aussi une police sans serifs pour ses diapos. Pour les textes très longs il convient mieux d'utiliser une police avec serifs. Mais pour des textes plus courts – le cas d'une diaporama – des polices sans serifs sont aussi possibles. Ils ont un caractère plus léger et moderne (cf. BÜHLER, SCHLAICH et SINNER 2017)

3.1.5.3 Couleurs Avec les couleurs du diaporama il est possible de s'exprimer voire d'influencer le public d'une certaine manière. Cf. [Color Meanings](#) pour explications plus précises sur significations des couleurs différents. Que faire si on a trouvé sa couleur principale? Comment obtenir une palette des couleurs adéquates? Un conseil: [Adobe Colors](#). Ici vous pouvez à partir des schémas de combinaison des couleurs définir une palette harmonisée. Il est aussi possible de faire créer une palette à partir d'une photo clé qu'on va utiliser dans sa présentation. En plus il y a un tas de palettes déjà faites à découvrir – donc pas besoin de concevoir une toute nouvelle palette.¹

Lumière éteinte ou allumée ? Quand le vidéoprojecteur est assez fort, pourquoi éteindre la lumière ?

If you want your presentation to be more effective, then don't touch that light switch. Even when you are using slides, the more lights you can keep on, the better off you will be. Remember, you're trying to connect, to tell a story, to sell an idea to the board or other decision makers. It is very difficult to make a connection if the audience can't see you. The audience is not there to witness the narration of slides; they are there to listen to you and become engaged with you and your topic. If the audience can't see you, they will find it difficult to listen, and they are certainly more likely to tune you out. The audience must experience both your "verbal speech" and your "visual speech." A relatively small part of your message is actually verbal. The rest of your message is expressed visually and vocally. Influencing people verbally becomes far more difficult when they can't see you (REYNOLDS 2008, p. 208).

3.1.6 Une foire aux présentations modèles

Tache: Créez un design avec une palette des couleurs, décidez vous aussi pour une police de caractères. Ajoutez ensuite une diapo avec la grille de vos diapos et deux diapos exemplaires. Un modèle est mis à votre disposition.²

3.2 Vos notes

3.3 Le polycopié

Avant de commencer la section : trois diapos par page A4 ne sont pas un polycopié. PowerPoint ou Keynote ne sont pas d'outils pour créer un text. Il y a des outils beaucoup plus appropriés : LaTeX, Microsoft Word, Microsoft Publisher, LibreOffice Writer, Adobe InDesign...

Pour créer des formats différents d'un document. Il peut être approprié de travailler en *Markdown*. Il y a des éditeurs diverses qui permettent d'écrire en *Markdown* (voir Tab.

¹ Pour une explication comment créer une palette de couleurs en PowerPoint, voir cette vidéo: [YouTube](#).

² Vous trouverez des modèles à remplir ici: [Drive](#)

2). *Pandoc* permet ensuite d'émettre des documents dans différents formats.

Table 2: Editeurs Markdown

RStudio	gratuit	possibilité de créer des documents html, pdf, epub...
Typora	15\$	possibilité de créer des documents html, pdf, epub...
Microsoft Visual Studio	gratuit	possibilité de créer des documents html, pdf, epub...

Pour obtenir des textes bien placés, je recommande fortement d'utiliser LaTeX. Si on se connaît bien en conception de documents, Microsoft Publisher, Adobe InDesign ou Scribus (gratuit) peuvent être une très bonne alternative. Microsoft Word est dans presque tous les cas le plus mauvais choix et la création du document prendra plus de temps.

3.3.1 Conception du polycopié

- Répartition sur la page : Vous êtes plutôt contraint à utiliser un format A4. Laissez donc des marges assez larges afin de garder une bonne lisibilité. Une bonne règle est de composer des lignes de 50-70 caractères. Une alternative peut être de faire deux colonnes ou opter pour une mise en page asymétrique. [Edward Tufte](#) conçoit ses [polycopiés](#) par exemple de cette façon.
- Choisissez une police de caractères sérieuse et bien lisible. Pour plus de règles de typographie voir ce post : [Typography Basics | Hubspot](#).
- Sur le polycopié mettez toutes les informations importantes pour comprendre vos propos. Contrairement à votre PowerPoint le polycopié devrait être compréhensible sans avoir entendu votre discours. Selon le type de la présentation ajoutez votre bibliographie.
- Pour les question à la suite de votre discours, il est envisageable d'ajouter vos coordonnées.

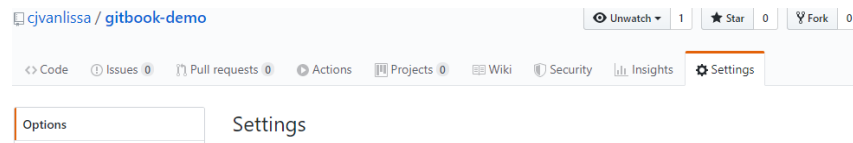
3.3.2 Une foire aux polycopiés modèles

Tâche: Créez des polycopiés modèles en A4 ! Ils devront être fournis en format PDF. De préférence, licenciez-les sous une licence CC0 ou CC BY ([plus d'informations](#)). Téléchargez les modèles sur la plateforme.

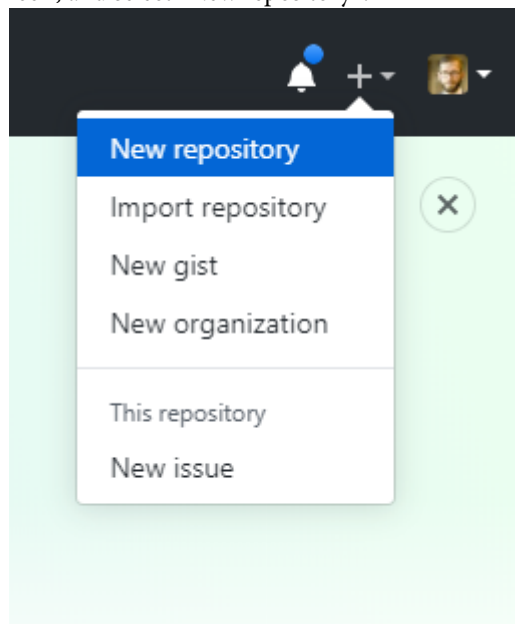
4 Get your GitBook

To get your GitBook, you should follow these steps:

1. Go to <https://github.com/cjvanlissa/gitbook-demo>
2. In the top right of the page, click Fork.
This will copy my `gitbook-demo` repository to your GitHub account.



3. My repository is now copied to your account. It is a template repository, which means that you can create a *new repository* based on this one.
4. Create a new repository for your own GitBook. Create one for a course you've been wanting to update. In the top-right corner of the GitHub website, click the + icon, and select "New repository":



5. In the dialog, select the `gitbook-demo` as "Repository template", and give the repository an appropriate name for your course. Then, press `Create repository`:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Repository template

Start your repository with a template repository's contents.

 cjvanlissa/gitbook-demo ▾

Owner

Repository name *

 cjvanlissa ▾

/ your_course_name ✓

Great repository names are short and memorable. Need inspiration? How about **jubilant-memory**?

Description (optional)

☒

Public

Anyone can see this repository. You choose who can commit.

☐

Private


You choose who can see and commit to this repository.

Create repository


6. Now, go back to Rstudio on your computer. In Rstudio, click **File > New Project**. A dialog will open. If you set up Rstudio with Git correctly, the dialog should have an option to create a new project from Version control. Click it:

New Project


Create Project



New Directory
Start a project in a brand new working directory >



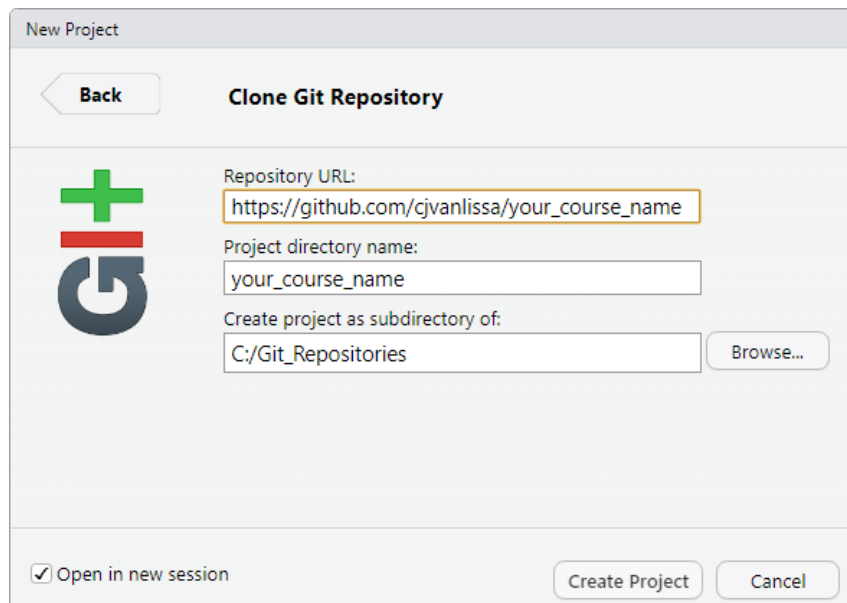
Existing Directory
Associate a project with an existing working directory >



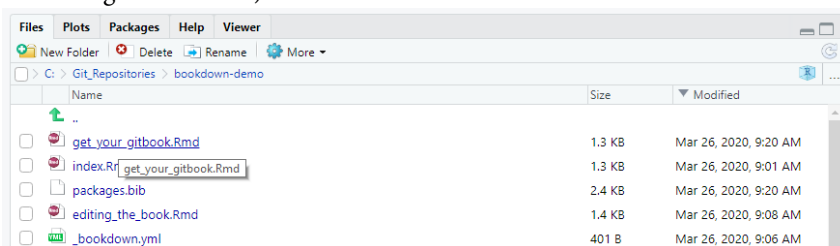
Version Control
Checkout a project from a version control repository >

Cancel

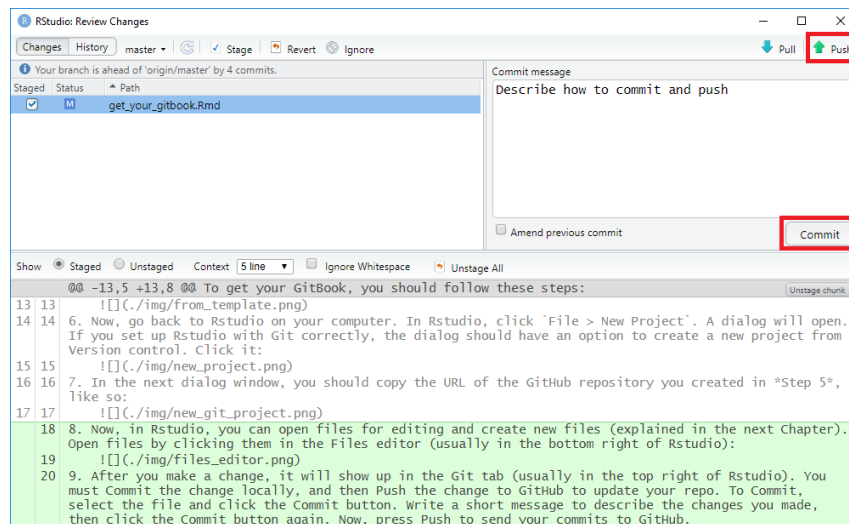
7. In the next dialog window, you should copy the URL of the GitHub repository you created in *Step 5*, like so:



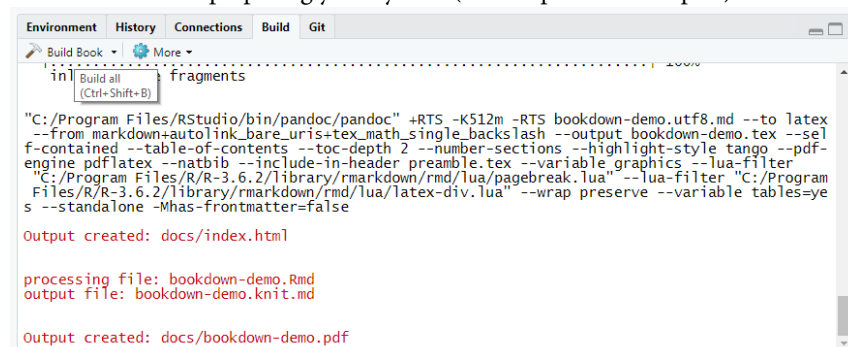
8. Now, in Rstudio, you can open files for editing and create new files (explained in the next Chapter). Open files by clicking them in the Files editor (usually in the bottom right of Rstudio):



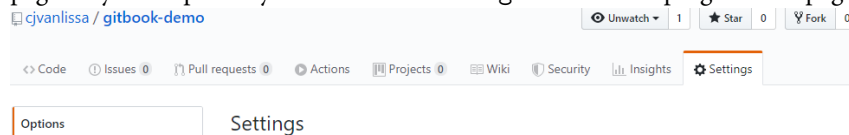
9. After you make a change, it will show up in the Git tab (usually in the top right of Rstudio). You must Commit the change locally, and then Push the change to GitHub to update your repo. To Commit, select the file and click the Commit button. Write a short message to describe the changes you made, then click the Commit button again. Now, press Push to send your commits to GitHub.



10. To render your book as a GitBook, you must “Build” it. In the top-right panel of Rstudio, you see a “Build” tab. In this tab, simply click the “Build Book” button to build your book. You should see a lot of rendering messages, until a window pops up with your brand new GitBook. If you get errors at this stage, you probably made a mistake in preparing your system (see the previous Chapter).



11. Building the book generated a lot of new files in the `./docs` directory. This directory contains the website files for your GitBook. Open the Git tab again, verify that the `./docs` directory is listed, and Commit and Push all of these new files as described in *Step 9*.
12. There is only one last remaining task: To publish your GitBook on GitHub pages. Once you do this, any change to the `./docs` folder that you push to GitHub will lead to an immediate update of your GitBook website. Go back to the GitHub page for your Repository. Click on the Settings tab on the top right of the page:



13. On the Settings page, scroll all the way down until you reach a section called GitHub Pages. There, under the “Source” heading, click the word None, and select master branch /docs folder. When you select it, the page will update, and if you scroll back down to the GitHub Pages section, you will

see the URL where your GitBook is published. The first time, it will take a few minutes for your GitBook to come online. When you publish updates to the GitBook however (simply by following *Step 11* again), the update will be near-instantaneous. The Pages section should now look like this (and that is hopefully the link where you found this book):

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at <https://civanlissa.github.io/gitbook-demo/>.

Source

Your GitHub Pages site is currently being built from the `/docs` folder in the `master` branch. [Learn more.](#)

master branch /docs folder ▾

5 Editing the book

The contents of the book are written in **RMarkdown**. You can use any formatting code that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$. Moreover, you can include chunks of R-code, like this:

The results of these chunks can be rendered to the GitBook:

```
## [1] "This is an R-command!"
```

To edit the book, you can change the text in the `.Rmd` files. Each Rmd file should contain **one and only one** chapter. A chapter is defined by the first-level heading `#`, e.g., `# Editing the book`.

Any sub-headings within the chapter are indicated with several `#` signs, e.g., `##` (level 2) and `###` (level 3).

5.1 Creating new chapters

To create a new chapter, you must follow two steps: 1) Create the file, and 2) Include it in the list of chapters.

First, to create the file for a new chapter in Rstudio, click `File > New File > Text file`. At the top of the file, write your chapter heading, as explained above. Then, click `File > Save`. Save the file as `.Rmd`, without spaces in the file name, e.g.: `editing_the_book.Rmd`.

Second, to include it in the list of chapters, open the file `_bookdown.yml` (click it in the Files explorer in the bottom right of Rstudio). This file has a list of `.Rmd` files to be included in the book. In this example, the list looks like this:

```
tmp <- readLines("_bookdown.yml")
cat(tmp[grep("^rmd_files", tmp):grep("references\\.Rmd", tmp)], sep = "\n")
```

```
rmd_files: ["index.Rmd", "plan-du-cours.Rmd", "creer-diapos.Rmd", "get_your_gitbook.Rmd",
"editing_the_book.Rmd", "figures_tables.Rmd", "examples.Rmd", "open_educational.Rmd",
"use_in_course.Rmd", "licenses.Rmd", "references.Rmd"]
```

Insert the file name of your new chapter in the desired position in this list.

5.2 Linking across chapters

You can label chapter and section titles using `{#label}` after them. The labels can be used as cross-references. For example, we can link to Chapter 6. If you do not manually label chapters, there will be automatic labels anyway, e.g., Chapter 7.

5.3 Advanced editing

The convenient [Rmarkdown Cheat Sheet](#) by Rstudio covers most of the knowledge required for advanced Rmarkdown editing. You can print it out and stick it to your wall!

6 Figures and tables

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

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

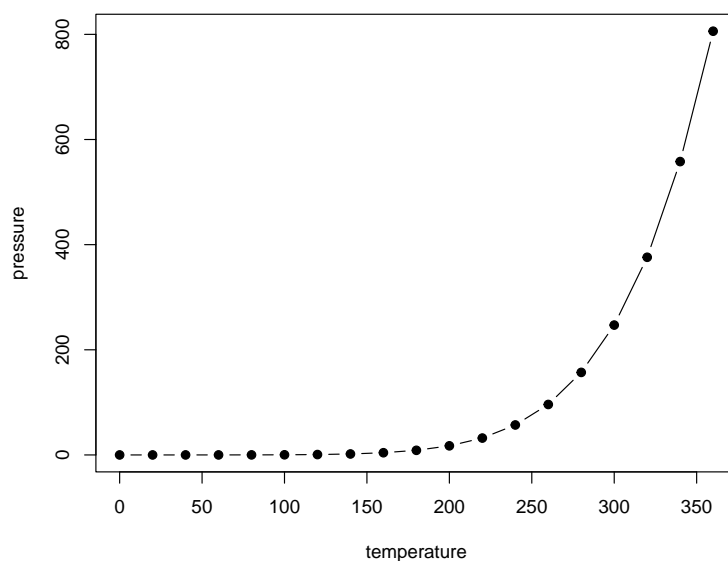


Figure 4: Here is a nice figure!

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 4. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 3.

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

Table 3: 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

You can write citations, too. For example, we are using the **bookdown** package (XIE 2022) in this sample book, which was built on top of R Markdown and **knitr** (XIE 2015).

7 Examples

Here are some examples of other GitBooks for courses; if you want to have your GitBook added to the list, please send a [Pull Request](#) (here's [how to send a pull request](#)).

7.1 Statistics with R (H. Quene)

<https://hugoquene.github.io/emlar2020>

A GitBook for a tutorial on *Statistics with R (Basics)*, held as part of the workshop on Experimental Methods in Language Acquisition Research (EMLAR, <https://emlar.wp.hum.uu.nl/>), Utrecht, on 17 April 2020. This compact introduction helps you with your first steps into R.

7.2 Theory Construction and Statistical Modeling (C. J. van Lissa)

<http://cjvanlissa.github.io/TCSM>

A GitBook for the course “*Theory Construction and Statistical Modeling*”, with some interesting code, for example: Blocks of answers to the tutorial questions that can be collapsed and expanded.

7.3 Doing Meta-Analysis in R (C. J. van Lissa)

<http://cjvanlissa.github.io/Doing-Meta-Analysis-in-R>

A GitBook on doing meta-analysis in R, based on the book ‘Doing Meta-Analysis in R’ by Mathias Harrer, Pim Cuijpers, & David Ebert, and adapted to focus on the [metafor](#) package, and exploring heterogeneity using [metaforest](#). The original can be found here: https://bookdown.org/MathiasHarrer/Doing_Meta_Analysis_in_R/

7.4 Métodos quantitativos em Psicologia com R (L. Anunciação)

<https://anovabr.github.io/mqt/>

This book provides a short and to-the-point exposition on the essentials of statistics, and was written for undergraduate students at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio). To a lesser degree, the mathematical modeling of statistical questions will be addressed. This book might be relevant for Portuguese-speaking students who enroll for laboratory-based statistics and anyone who wants to learn R.

8 Open Educational Resources

UNESCO defines Open Educational Resources as *teaching, learning and research materials in any medium – digital or otherwise – that reside in the public domain or have been released under an open license that permits no-cost access, use, adaptation and redistribution by others with no or limited restrictions*.

Open Educational resources can help lighten the workload on individual teachers, who can collaborate with the development of high-quality open access resources, instead of having to develop their own proprietary materials from scratch. Moreover, Open Educational resources are inclusive, lowering the barrier to knowledge acquisition for learners around the world, and enabling lifelong learning for those outside academia.

Many universities support Open Educational Resources. Here are just a few (feel free to [send a pull request](#) with other relevant resources).

- **OER Commons**: A freely accessible online library of open educational resources.
- **Utrecht University Figshare**: Open learning objects from Utrecht University.
- **Johns Hopkins University OCW**: Open public health courses and materials.
- **University of Pittsburgh OER**: Big List of Open Educational Resources.
- **MERLOT**: Online learning and support materials and content creation tools, led by an international community of educators, learners and researchers.

9 Compatibility with existing systems

Many universities offer digital platforms for learning. You might wish to embed your GitBook within these existing systems. Here are two ways in which you might do that. Currently, this section only discusses BlackBoard, but the same principles should apply to other platforms.

9.1 Add a hyperlink

You can add a link to your GitBook in the BlackBoard course menu by following [this tutorial](#).

9.2 Embed the whole book

You can add a Blank Page to your BlackBoard course menu, and fill that page with a full-size “iframe” - a web page within the web page. [This tutorial](#) explains how to do it. It is possible that your university is blocking this feature, however.

10 License your GitBook

In the spirit of Open Science, it is good to think about making your course materials Open Source. That means that other people can use them. In principle, if you publish materials online without license information, you hold the copyright to those materials. If you want them to be Open Source, you must include a license. It is not always obvious what license to choose.

The Creative Commons licenses are typically suitable for course materials. This GitBook, for example, is licensed under CC-BY 4.0. That means you can use and remix it as you like, but you must credit the original source.

If your project is more focused on software or source code, consider using the [GNU GPL v3 license](#) instead.

You can find [more information about the Creative Commons Licenses here](#). Specific licenses that might be useful are:

- [CC0 \(“No Rights Reserved”\)](#), everybody can do what they want with your work.
- [CC-BY 4.0 \(“Attribution”\)](#), everybody can do what they want with your work, but they must credit you. Note that this license may not be suitable for software or source code!

For compatibility between CC and GNU licenses, see [this FAQ](#).

Références

- BÜHLER, Peter, Patrick SCHLAICH et Dominik SINNER (2017). *Typografie: Schrifttechnologie - Typografische Gestaltung - Lesbarkeit*. Bibliothek der Mediengestaltung. DOI: 10.1007/978-3-662-53912-5. Berlin, Heidelberg : Springer Berlin Heidelberg. DOI : 10.1007/978-3-662-53912-5. URL : <http://link.springer.com/10.1007/978-3-662-53912-5>.
- REYNOLDS, Garr (2008). *Presentation zen: simple ideas on presentation design and delivery*. Voices that matter. OCLC: ocn156831268. Berkeley, CA : New Riders.
- XIE, Yihui (2015). *Dynamic Documents with R and knitr*. 2nd. ISBN 978-1498716963. Boca Raton, Florida : Chapman et Hall/CRC. URL : <http://yihui.name/knitr/>.
- (2022). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.26. URL : <https://CRAN.R-project.org/package=bookdown>.