Writing GitBooks Tutorial

Ville Langén

2025-07-09

Contents

Preface

This is a tutorial about making tutorials in GitBook format using RStudio and bookdown.

Whether you're a data scientist, researcher, or educator, creating beautiful, interactive tutorials and documentation is essential for sharing your knowledge effectively. This guide will walk you through the entire process of creating professional-looking GitBook-style tutorials using R, RStudio, and the powerful bookdown package.

The method we'll be covering is so powerful that it not only creates a web page but also generates an EPUB e-book and a PDF document automatically! This very e-book is also released in EPUB and PDF formats:

EPUB format - Perfect for e-readers and mobile devices
PDF format - Print-ready professional document

All three formats are generated from the same source files with a single command!

What You'll Learn

- How to set up your development environment
- Understanding the GitBook format and bookdown
- Writing effective tutorial content with R Markdown
- Publishing your tutorials online

Prerequisites

Basic familiarity with R and R Markdown is helpful but not required. We'll guide you through everything step by step.

Chapter 1

Introduction to GitBook Format

1.1 What is GitBook Format?

GitBook is a modern documentation format that creates beautiful, interactive books and tutorials from Markdown files. Originally developed as a platform for writing and publishing books, the GitBook format has become popular for:

- Technical documentation
- API guides
- Educational tutorials
- Research papers
- Software manuals

The format features:

- **Clean, responsive design** that works on desktop and mobile
- **Interactive navigation** with a collapsible table of contents
- **Search functionality** built-in
- **Code syntax highlighting**
- **Mathematical equations** support
- **Cross-references** and internal linking

1.2 What is bookdown::gitbook?

`bookdown::gitbook` is an R package output format that creates GitBook-style HTML books from R Markdown files. It's part of the `bookdown` package ecosystem created by Yihui Xie.

Key features of `bookdown::gitbook`:

- **Multi-chapter books** from multiple .Rmd files
- **Automatic cross-referencing** of figures, tables, and sections
- **Citation management** with BibTeX
- **Code chunk execution** with live R output
- **Mathematical notation** with LaTeX syntax
- **Customizable themes** and styling

1.3 Real-World Examples

Here are some excellent examples of GitBook-style tutorials and documentation:

1.3.1 Data Science

- R for Data Science by Hadley Wickham
- Advanced R by Hadley Wickham
- Tidy Modeling with R by Max Kuhn and Julia Silge

1.3.2 Statistics and Research

- Statistical Inference via Data Science by Chester Ismay
- Geocomputation with R by Robin Lovelace

1.3.3 Web Development

- Engineering Production-Grade Shiny Apps
- Mastering Shiny by Hadley Wickham

These examples demonstrate the power and versatility of the GitBook format for creating comprehensive, professional documentation that's both beautiful and functional.

Chapter 2

Installation and Setup

Before we can start creating GitBook tutorials, we need to set up our development environment. This chapter will guide you through installing all the necessary tools.

2.1 Installing R

R is the foundation of our toolkit. It's a free, open-source programming language specifically designed for statistical computing and graphics.

2.1.1 Download and Install R

1. Visit the official R website: https://cran.r-project.org
2. Click on "Download R for your operating system"
3. Choose a mirror close to your location
4. Download the latest version of R
5. Run the installer and follow the installation wizard

2.1.2 Verify R Installation

Open your terminal or command prompt and type:

```
R --version
```

You should see output showing the R version number.

2.2 Installing RStudio

RStudio is an integrated development environment (IDE) that makes working with R much more pleasant and productive.

2.2.1 Download and Install RStudio

1. Visit: https://posit.co/download/rstudio-desktop/
2. Download RStudio Desktop (free version)
3. Run the installer appropriate for your operating system
4. Launch RStudio to verify it works

2.2.2 First Launch

When you first open RStudio, you'll see four panes: - **Console** (bottom left): Where R commands are executed - **Environment** (top right): Shows your variables and data - **Files/Plots/Help** (bottom right): File browser and output - **Editor** (top left): Where you'll write your .Rmd files

2.3 Installing Git

Git is essential for version control and publishing your tutorials online.

2.3.1 Download and Install Git

Visit https://git-scm.com/ and download Git for your operating system.

2.3.1.1 Windows

- Download Git for Windows
- Run the installer with default settings

2.3.1.2 macOS

- Git comes pre-installed on most Mac systems
- Alternatively, install via Homebrew: `brew install git`

2.3.1.3 Linux

- Ubuntu/Debian: `sudo apt-get install git`
- CentOS/RHEL: `sudo yum install git`

2.3.2 Configure Git

Set up your identity (replace with your information):

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

2.4 Installing R Packages

Now we'll install the R packages needed for creating GitBooks.

2.4.1 Essential Packages

Open RStudio and run these commands in the Console:

```
# Install the main package
install.packages("bookdown")

# Install supporting packages
install.packages(c("rmarkdown", "knitr"))
```

2.4.2 Optional but Recommended Packages

These packages will enhance your GitBook creation experience:

```
# For better plots and data manipulation
install.packages(c("ggplot2", "dplyr", "tidyr"))

# For web publishing
install.packages("servr")

# For advanced formatting
install.packages(c("DT", "htmlwidgets"))
```

2.4.3 Verify Package Installation

Test that bookdown is working:

```
library(bookdown)
packageVersion("bookdown")
```

2.5 Create Your First Project

Let's verify everything is working by creating a test project:

1. In RStudio, go to **File > New Project > New Directory**
2. Choose **Book Project using bookdown**
3. Give it a name like "test-gitbook"
4. Click **Create Project**

RStudio will create a sample bookdown project with example files. You can render it by running:

```
bookdown::render_book("index.Rmd")
```

If everything is installed correctly, this will generate a GitBook in the `_book/` folder!

2.6 Troubleshooting

2.6.1 Common Issues

Issue: "Package 'bookdown' not found" - **Solution**: Make sure you've installed the package: `install.packages("bookdown")`

Issue: Git not recognized - **Solution**: Restart RStudio after installing Git, or add Git to your system PATH

Issue: LaTeX errors when rendering - **Solution**: Install TinyTeX: `tinytex::install_tinytex()`

2.6.2 Getting Help

- **R Documentation**: Use `?function_name` in R console
- **Bookdown Guide**: https://bookdown.org/yihui/bookdown/
- **RStudio Community**: https://community.rstudio.com/

Congratulations! You now have a complete development environment for creating GitBook tutorials.

Chapter 3

Writing Tutorial Content

Now that you have your environment set up, let's learn how to create compelling tutorial content using R Markdown. This chapter covers the essential techniques for writing effective GitBook tutorials.

3.1 Project Structure with .Rmd Files

Each `.Rmd` file in your project becomes a chapter in your GitBook. The structure is simple:

```
my-tutorial/
|-- index.Rmd          # Main page (Chapter 0)
|-- 01-chapter1.Rmd    # Chapter 1
|-- 02-chapter2.Rmd    # Chapter 2
|-- 03-chapter3.Rmd    # Chapter 3
`-- ...
```

3.1.1 File Naming Convention

- **index.Rmd**: Always the main/first page
- **01-intro.Rmd**: Use numbers for ordering
- **XY-descriptive-names.Rmd**: Use descriptive names after numbers

3.2 Headings Create Sections

Headings automatically create your table of contents and navigation structure:

```
# Chapter Title {#chapter-id}

## Main Section
```

```
### Subsection

#### Sub-subsection
```

3.2.1 Important Notes:

- `# Chapter Title` creates a new chapter
- `{#chapter-id}` creates an ID for cross-referencing
- Only use `#` for chapter titles in chapter files
- Use `##`, `###`, `####` for sections within chapters

3.3 Code Chunks: The Heart of R Tutorials

Code chunks are what make R Markdown special. They execute R code and display the results:

3.3.1 Basic Code Chunk

````
```{r}
# This is R code
x <- c(1, 2, 3, 4, 5)
mean(x)
```
````

3.3.2 Code Chunk Options

Control how your code appears and executes:

````
``` r
# echo=TRUE: Show the code
# eval=TRUE: Run the code
# fig.cap: Caption for figures
plot(1:10, 1:10)
```

![(\#fig:chunk-name)My Plot](writing-gitbooks-tutorial_files/figure-latex/chunk-name-1
````

Common options: - `echo=FALSE`: Hide code, show output - `eval=FALSE`: Show code, don't run it - `include=FALSE`: Run code, don't show anything - `message=FALSE`: Hide messages - `warning=FALSE`: Hide warnings

3.4 Creating Plots and Visualizations

R excels at creating beautiful plots:

```r
library(ggplot2)
data(mtcars)

ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_point(aes(color = factor(cyl))) +
  labs(title = "Horsepower vs MPG",
       x = "Miles per Gallon",
       y = "Horsepower",
       color = "Cylinders") +
  theme_minimal()
```

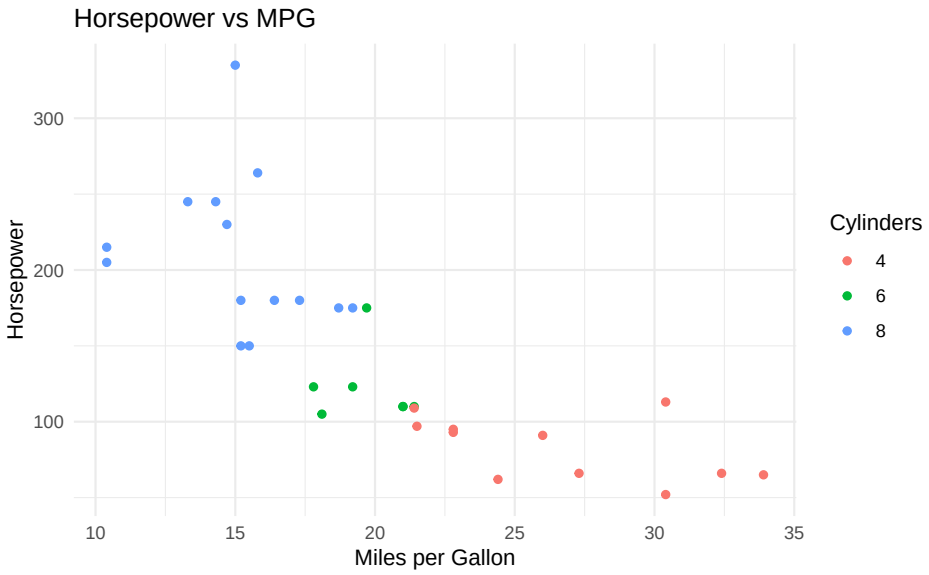

Figure 3.1: Sample Scatter Plot

3.5 Inline R Code

You can embed R results directly in your text using inline code:

```
The mean of mtcars$mpg is 20.09 miles per gallon.
```

Result: The mean of mtcars$mpg is 20.09 miles per gallon.

3.6 Mathematical Notation

LaTeX syntax works beautifully in GitBooks:

3.6.1 Inline Math

```
The formula is $y = mx + b$ where $m$ is the slope.
```

Result: The formula is $y = mx + b$ where m is the slope.

3.6.2 Block Math

```
$$
\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i
$$
```

Result:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

3.7 Cross-References and Links

3.7.1 Referencing Figures and Tables

```
See Figure \@ref(fig:plot-example) for the scatter plot.
```

Result: See Figure 3.1 for the scatter plot.

3.7.2 Referencing Sections

```
As we learned in Chapter \@ref(intro), GitBook format is powerful.
```

Result: As we learned in Chapter 1, GitBook format is powerful.

3.7.3 Creating Tables

```r
knitr::kable(
  head(mtcars[, 1:4]),
  caption = "First few rows of mtcars dataset",
  booktabs = TRUE
)
```

Reference it with: `Table \@ref(tab:cars-table)`

3.8 Optional Enhancements

3.8.1 Footnotes

Add footnotes for additional information:

Table 3.1: First few rows of mtcars dataset

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110
Datsun 710	22.8	4	108	93
Hornet 4 Drive	21.4	6	258	110
Hornet Sportabout	18.7	8	360	175
Valiant	18.1	6	225	105

```
This is important text^[This is a footnote explaining the importance.].
```

Result: This is important text[1].

3.8.2 Citations with BibTeX

First, create a `references.bib` file:

```
@book{wickham2016r,
  title={R for data science: import, tidy, transform, visualize, and model data},
  author={Wickham, Hadley and Grolemund, Garrett},
  year={2016},
  publisher={O'Reilly Media}
}
```

Then cite it:

```
According to @wickham2016r, data science is a powerful approach.
```

3.8.3 Adding Images

```
![Caption for image](path/to/image.png)

# Or with more control:

```{r}
knitr::include_graphics("path/to/image.png")
```
```

“‘

3.8.4 Callouts and Special Blocks

Create attention-grabbing callouts:

[1] This is a footnote explaining the importance.

```
> **Note:** This is an important note that readers should pay attention to.

> **Warning:** Be careful with this command as it might delete files.

> **Tip:** Here's a helpful tip to make your work easier.
```

Results:

> **Note:** This is an important note that readers should pay attention to.
>
> **Warning:** Be careful with this command as it might delete files.
>
> **Tip:** Here's a helpful tip to make your work easier.

Chapter 4

Publishing Your GitBook

Once you've written your tutorial content, it's time to render and publish your GitBook. This chapter covers both local rendering for testing and online publishing for sharing with the world.

4.1 Rendering Locally

Local rendering lets you preview your GitBook before publishing. This is essential for testing and debugging.

4.1.1 Basic Rendering Command

The fundamental command to render your GitBook:

```
bookdown::render_book("index.Rmd", "bookdown::gitbook")
```

This command: 1. Processes all your .Rmd files in order 2. Executes all R code chunks 3. Generates HTML files in the `_book/` directory 4. Creates navigation and cross-references

4.1.2 Alternative Rendering Methods

4.1.2.1 Using RStudio Build Tab

1. Open your project in RStudio
2. Go to the **Build** tab (usually in the top-right pane)
3. Click **Build Book**
4. Select **bookdown::gitbook**

4.1.2.2 Using the Knit Button

- Open `index.Rmd`

- Click the **Knit** dropdown
- Select **Knit Book**

4.1.3 Previewing Your Book

After rendering, you can preview your book:

```
# Open the book in your browser
servr::httw("_book")
```

Or simply open `_book/index.html` in your web browser — my preferred method, every time.

4.1.4 Multiple Output Formats

You can render to multiple formats simultaneously:

```
# Render GitBook, PDF, and EPUB
bookdown::render_book("index.Rmd", "all")
```

Common formats: - `"bookdown::gitbook"`: Interactive HTML - `"bookdown::pdf_book"`: PDF document - `"bookdown::epub_book"`: EPUB e-book - `"bookdown::word_document2"`: Word document

4.2 Publishing Online with GitHub + Netlify

This is the most popular and free way to publish GitBooks online.

4.2.1 Step 1: Prepare Your Project

4.2.1.1 Clean Up Files

Make sure your project only contains necessary files:

```
my-gitbook/
|-- index.Rmd
|-- 01-intro.Rmd
|-- 02-install.Rmd
|-- 03-content.Rmd
|-- 04-publish.Rmd
|-- _bookdown.yml
|-- _output.yml
|-- book.bib (if using citations)
`-- README.md
```

4.2.1.2 Create .gitignore

Create a `.gitignore` file to exclude build artifacts:

```
# R and RStudio files
.Rproj.user
.Rhistory
.RData
.Ruserdata
*.Rproj

# Bookdown files
_book/
_bookdown_files/
*.rds

# OS generated files
.DS_Store
.DS_Store?
._*
.Spotlight-V100
.Trashes
ehthumbs.db
Thumbs.db
Desktop.ini
$RECYCLE.BIN/
*.lnk

# Temporary files
*~
*.tmp
*.temp
```

4.2.2 Step 2: Create GitHub Repository

1. **Create a new repository on GitHub**
   - Go to github.com
   - Click **New Repository**
   - Give it a descriptive name (e.g., "my-r-tutorial")
   - Make it **Public** (required for free Netlify hosting)
2. **Initialize Git in your project**

```
git init
git add .
git commit -m "Initial commit"
```

3. **Connect to GitHub**

```
git remote add origin https://github.com/yourusername/your-repo.git
git branch -M main
```

```
git push -u origin main
```

4.2.3 Step 3: Set Up Netlify

1. **Sign up for Netlify**
   - Go to netlify.com
   - Sign up with your GitHub account
2. **Create a new site**
   - Click **New site from Git**
   - Choose **GitHub**
   - Select your repository
3. **Configure build settings**
   - **Build command**: `Rscript -e "bookdown::render_book('index.Rmd')"`
   - **Publish directory**: `_book/`
   - Click **Deploy site**

4.2.4 Step 4: Customize Your Netlify Site

4.2.4.1 Custom Domain (Optional)

- In your Netlify dashboard, go to **Domain settings**
- Add your custom domain
- Follow DNS configuration instructions

4.2.4.2 Build Environment

Add these environment variables in Netlify: - `R_VERSION`: `4.3.0` (or your preferred R version)

4.3 Updating Your Published Book

Simply push changes to your GitHub repository:

```
git add .
git commit -m "Update content"
git push origin main
```

Netlify will automatically rebuild and deploy your site.

4.4 Troubleshooting Common Issues

4.4.1 Build Failures

Problem: R packages not found

```
Error: Package 'ggplot2' not found
```

Solution: Update your build command to install packages:

```
Rscript -e "install.packages(c('ggplot2', 'dplyr')); bookdown::render_book('index.Rmd')"
```

Problem: LaTeX errors in PDF rendering

```
Error: LaTeX failed to compile
```

Solution: Install TinyTeX or remove PDF output temporarily:

```
tinytex::install_tinytex()
```

Congratulations! Your GitBook tutorial is now live and accessible to the world. Remember to keep your content updated and respond to reader feedback to make your tutorial even better.