



R Markdown Bootcamp: Master PDF Reports in Under 10 Minutes

@micosapiens711

Published: July 2024

Personal Edition

Acknowledgments

I would like to express my gratitude to everyone who contributed to the success of this project...

vitorma101

Abstract

This document provides a concise guide on how to configure R Markdown for creating professional PDF reports. It covers the setup of custom cover pages, detailed table of contents (TOC), well-structured content sections, and advanced LaTeX customizations. These customizations include page formatting, acknowledgment placement, the use of a `preamble.tex` file for enhanced control over document layout, and the insertion of a custom “About the Author” page with centered content and hyperlinks. Additionally, this guide explains the optional inclusion of a watermark as a branding or security measure. By following this guide, users can produce polished, customized reports tailored to specific needs.

Vitoroma101

Contents

Abstract	iv
1. Introduction	1
2. Cover Page and Initial Setup	1
2.1 Configuring the YAML Header for Cover Pages	2
2.2 Using LaTeX Commands for a Custom Cover Page	3
2.3 Inserting a Blank Page Between Sections	4
2.4 Inserting Acknowledgments Before the TOC	4
2.5 Formatting the Abstract Page and Setting Up Page Numbers	5
3. TOC Setup and Configuration	5
4. Customizing Document Layout with <code>preamble.tex</code>	6
Steps to Create and Use <code>preamble.tex</code>	6
Detailed Breakdown of <code>preamble.tex</code> Commands	8
5. Customizing the Final Page with Author Information and Hyperlinks	11

1. Introduction

R Markdown is a powerful tool that allows you to create dynamic documents that integrate R code with text, enabling you to generate reports, presentations, and more with ease. One of the key features of R Markdown is its ability to produce high-quality PDFs using LaTeX, a typesetting system commonly used for academic and professional documents.

In this guide, we will explore how to configure R Markdown to create professional reports with custom cover pages, detailed tables of contents (TOC), and well-structured content. Whether you are preparing a report for academic purposes or a professional presentation, mastering these aspects of R Markdown will allow you to deliver polished and customized documents that meet your specific needs.

We will begin by setting up a custom cover page, then move on to configuring the table of contents, and discuss how to insert and format acknowledgments and an abstract page with appropriate page numbers. Additionally, the guide covers the setup and configuration of TOC, and customization of document layout using a `preamble.tex` file. We will also detail how to add a watermark to your document and finalize it with a page containing author information and hyperlinks.

By the end of this guide, you will have the knowledge and tools to create visually appealing and highly functional PDF documents in R Markdown, tailored to your exact specifications.

As a side note, this guide assumes that the reader has basic knowledge of R and R Markdown, so the process of creating the document within RStudio is not covered in detail. The focus is instead on providing advanced tips and configurations to enhance the professionalism and customization of your reports.

2. Cover Page and Initial Setup

In this section, I'll guide you through the process of setting up the cover page in your R Markdown document, along with other preliminary elements before the Table of Contents (TOC). We'll use specific options in the YAML header to configure the document's initial layout, as well as additional settings to ensure a professional format from the very beginning.

2.1 Configuring the YAML Header for Cover Pages

The YAML header in R Markdown is essential for setting up the overall document structure and appearance, including the cover page. The configuration here goes beyond the basic options of `title`, `author`, and `date`, which are typically included by default, and instead focuses on custom settings to achieve a more tailored presentation.

```
---
output:
  pdf_document:
    includes:
      in_header: preamble.tex
  word_document: default
geometry: margin=1in
fontsize: 11pt
linestretch: 1.5
---
```

- **output::** Specifies the format(s) in which the document will be rendered. In this case, it includes options for generating both PDF and Word documents.
 - **pdf_document::** Includes the `in_header: preamble.tex` option, which allows for custom LaTeX commands via the `preamble.tex` file, enhancing control over the document's appearance. This will be explored in more detail later in the document.
 - **word_document::** Specifies that the document can also be rendered as a Word document.
- **geometry::** Adjusts the page margins. A 1-inch margin is used here, providing a standard layout suitable for professional documents.
- **fontsize::** Sets the base font size for the document. The size is set to 11pt, which is commonly used for readability in formal documents.
- **linestretch::** Controls the line spacing, set to 1.5 for improved readability, especially in longer texts.

2.2 Using LaTeX Commands for a Custom Cover Page

In this section, we will explain how to use LaTeX commands within R Markdown to create a custom cover page. This includes centering text and images, adding logos or other images, and formatting titles and author details.

```
\begin{center}

  \includegraphics[width=0.3\textwidth]{logosaurus.JPEG}\par\vspace{1cm}

  \Large \textbf{R Markdown Bootcamp: Master PDF Reports in 5 Minutes}\\

  \vspace{1cm}

  \textbf{micosapiens711}\\

  \vspace{1cm}

  Published: July 2024\\

  \vspace{1cm}

  Personal Edition\\

\end{center}
```

- `\begin{center} ... \end{center}`: This environment is used to center text and images on the cover page. Everything placed within this environment will be centered on the page, making it ideal for titles, author names, and logos.
- `\includegraphics[width=...]{filename}`: This command is used to add logos or images to the cover page. It is part of the LaTeX `graphicx` package and allows you to control the size and placement of images within your document.

Important: The `filename` refers to the name of the image file, which should be stored in the same directory as your R Markdown document. If the image is located in a different directory, you must specify the relative or absolute path to the file.

- `\Large`: Adjusts the font size of the text. This command is typically used to make the title or other important elements stand out on the cover page.
- `\textbf{...}`: Makes the text bold, which is useful for highlighting the title, author names, or other significant details on the cover page.

- `\vspace{...}`: Adds vertical space between elements on the page. The amount of space can be specified using units like `cm`, `mm`, or `in`. This is often used to control the spacing between the title, author name, and other elements on the cover page.

2.3 Inserting a Blank Page Between Sections

Inserting a blank page can help separate major sections in a document. In R Markdown, you can use LaTeX commands to manually insert these pages for better control over your layout.

```
\newpage  
\mbox{ }  
\newpage
```

- `\newpage`: Forces a page break, starting a new page in the document.
- `\mbox{ }`: Creates an empty box (or a blank space) on the page. When used between two `\newpage` commands, it effectively inserts a blank page in the document. The presence of `\mbox{ }` ensures that the page is not completely empty, which prevents some LaTeX processors from removing the blank page due to optimization settings.
- `\newpage`: Forces another page break after the blank page created by `\mbox{ }`, starting a new page for the subsequent content.

2.4 Inserting Acknowledgments Before the TOC

Inserting acknowledgments is a common practice in formal documents, and in R Markdown, this can be done manually using LaTeX commands to ensure it appears exactly where you want it. Below is an explanation of how to insert a section for acknowledgments immediately after the cover page.

```
\newpage  
\section*{Acknowledgments}
```

- `\newpage`: This LaTeX command forces a page break, ensuring that the acknowledgments start on a new page, separate from the cover page.

- **`\section*{Acknowledgments}`**: This command creates a new section titled “Acknowledgments” without numbering. The asterisk (*) ensures that this section is not numbered and doesn’t appear in the table of contents, maintaining a clean and professional look.

2.5 Formatting the Abstract Page and Setting Up Page Numbers

Before starting the abstract, it’s important to configure the page numbering and layout of your document. The following LaTeX commands help you achieve a professional format:

```
\newpage
\pagenumbering{roman}
\setcounter{page}{4}
\pagestyle{mystyle}
```

- **`\newpage`**: Forces a page break, starting a new page for the abstract.
- **`\pagenumbering{roman}`**: Switches the page numbering to Roman numerals, which is often used for sections like the abstract, acknowledgments, and table of contents.
- **`\setcounter{page}{5}`**: Manually sets the current page number to 4. This is useful if you need the abstract to start on a specific page number.
- **`\pagestyle{mystyle}`**: Applies a custom page style, defined earlier in the document, to control the appearance of headers and footers on the abstract page.

3. TOC Setup and Configuration

After setting up the abstract, it’s essential to include a Table of Contents (TOC) to help readers navigate your document. The TOC provides an overview of the structure and allows quick access to different sections.

To add a TOC in R Markdown, use the following LaTeX commands:

```
\newpage
\tableofcontents
\newpage
```

```
\pagenumbering{arabic}  
\setcounter{page}{1}
```

- **\newpage**: Inserts a page break before and after the TOC, ensuring it has its own dedicated page.
- **\tableofcontents**: Generates the Table of Contents (TOC) based on the section headers used throughout the document. It is important to use the appropriate headers (**#**, **##**, **###**, etc.) in your R Markdown document, as the TOC is built from these. Without using these headers, the TOC will not include any sections, resulting in an empty or non-existent TOC.
- **\pagenumbering{arabic}**: Resets the page numbering to Arabic numerals after the TOC.
- **\setcounter{page}{1}**: Resets the page counter to 1, starting fresh after the TOC.

Alternatively, you can enable the TOC directly in the YAML header by setting **toc: true** and **toc_depth: 2**. This will automatically generate a Table of Contents that includes headers up to the specified depth. For example, setting **toc_depth** to 2 will include both **##** and **###** level headers, providing detailed navigation within the document. To avoid duplication, if **toc: true** is used in the YAML, you should either remove the manual TOC insertion or set **toc: false** where appropriate.

4. Customizing Document Layout with `preamble.tex`

As mentioned earlier, the **pdf_document** option in the YAML header allows for the inclusion of a custom LaTeX file via **in_header: preamble.tex**, which enhances control over the document's appearance. To achieve advanced customization in your R Markdown document, you can create a separate LaTeX file called **preamble.tex**. This file allows you to define specific styles and settings that are applied throughout your document, giving you greater control over headers, footers, page numbering, and more.

Steps to Create and Use `preamble.tex`

1. Create the `preamble.tex` file:

- In RStudio, open the “File” menu from the top-left corner.
- Navigate to **New File > Text File**.
- This will open a text editor within RStudio.

2. Write the content of `preamble.tex`:

- Enter your LaTeX customization commands, such as those for headers, footers, and page numbering.

```
\usepackage[none]{hyphenat}
\usepackage{fancyhdr}

\fancypagestyle{plain}{
  \fancyhf{}
  \renewcommand{\headrulewidth}{0pt}
  \renewcommand{\footrulewidth}{0pt}
}
\pagestyle{plain}
\pagenumbering{gobble}
\fancypagestyle{mystyle}{
  \fancyhf{}
  \rfoot{\thepage}
  \renewcommand{\headrulewidth}{0pt}
  \renewcommand{\footrulewidth}{0pt}
}
```

3. Save the file:

- Save the file as `preamble.tex` in the same directory where your R Markdown document is located.

4. Ensure `preamble.tex` is correctly linked:

- Since the `preamble.tex` file is already included via the YAML header as discussed earlier, simply make sure that the path to this file is correct and that it resides in the same directory as your R Markdown document.

Detailed Breakdown of `preamble.tex` Commands

Loading Essential LaTeX Packages

- `\usepackage{fancyhdr}`: This command loads the `fancyhdr` package, which is used for customizing the headers and footers in LaTeX documents. The package provides extensive control over the appearance of headers and footers, allowing for a wide range of customization options.
- `\usepackage[none]{hyphenat}`: This command loads the `hyphenat` package with the `none` option, which disables hyphenation throughout the document. Hyphenation is the splitting of words at the end of lines. Disabling it ensures that words are not broken across lines, leading to a more consistent and visually clean text layout.

First Block: Setting Up the Plain Page Style

- `\fancypagestyle{plain}{ ... }`: This command defines a custom page style named `plain`. The `plain` style is one of the default styles provided by LaTeX and is typically used for chapter beginnings or special pages where minimal formatting is desired. In this `preamble`, we redefine the `plain` style to remove both the header and footer lines.
 - `\fancyhf{}:` This command clears any existing header and footer content. It resets the header and footer fields to be empty, allowing for a fresh start when defining the new style.
 - `\renewcommand{\headrulewidth}{0pt}`: This command removes the header line by setting its width to 0 points. This means that no line will be drawn across the top of the page in the `plain` style.
 - `\renewcommand{\footrulewidth}{0pt}`: Similar to the header line, this command removes the footer line by setting its width to 0 points, ensuring no line is drawn at the bottom of the page in the `plain` style.
- `\pagestyle{plain}`: This command sets the default page style for the document to `plain`. This means that any pages not explicitly set to another style will follow the `plain` style configuration defined earlier. Other default page styles in LaTeX include `empty`, which

removes all headers and footers, and **headings**, which includes chapter and section headings in the header.

- `\pagenumbering{gobble}`: This command disables page numbering by “gobbling” or removing the page numbers. When you use **gobble**, LaTeX automatically stops numbering the pages from the point where this command is issued. It is commonly used in sections where page numbers are not desired, such as title pages or sections with custom formatting. If you want to disable page numbers for specific pages only, you would typically use `\pagenumbering{gobble}` before those pages, and then switch back to a different numbering scheme afterwards (e.g., `\pagenumbering{arabic}`).

Second Block: Defining a New Page Style for the Rest of the Document

- `\fancypagestyle{mystyle}{ ... }`: This command defines another custom page style named **mystyle**. The contents of this block determine how the headers and footers will appear when this style is applied, allowing for a different configuration from the **plain** style.
 - `\fancyhf{}:` Again, this command clears any existing header and footer content, ensuring a clean slate for defining the **mystyle** page style.
 - `\rfoot{\thepage}:` This command places the page number on the right side of the footer. The `\thepage` command is used to insert the current page number, and placing it within `\rfoot{}` ensures it appears on the right-hand side of the page when the **mystyle** style is applied.
 - `\renewcommand{\headrulewidth}{0pt}:` This command removes the header line in the **mystyle** page style, similar to how it was done in the **plain** style.
 - `\renewcommand{\footrulewidth}{0pt}:` This command also removes the footer line in the **mystyle** page style, keeping the footer area clear except for the page number.

Third Block: Adding a Watermark to Your Document

The third block is optional and focuses on adding a watermark to your document. A watermark can serve as both a security measure and a branding element in professional documents. Below is the LaTeX code used in the `preamble.tex` file to implement a watermark in your R Markdown document:

```

\usepackage{background}
\backgroundsetup{
  scale=3,
  color=black,
  opacity=0.1,
  angle=45,
  position=current page.center,
  vshift=0pt,
  hshift=0pt,
  contents={\textsf{\Huge vitoroma101}}
}

```

- **\usepackage{background}**: Loads the `background` package, which allows you to add background elements such as watermarks to your document.
- **scale=3**: Sets the size of the watermark. A scale of 3 makes the watermark text three times its normal size.
- **color=black** and **opacity=0.1**: Specifies the color and transparency level of the watermark. The watermark is black, and the opacity is set to 0.1, making it faint enough not to interfere with the content.
- **angle=45**: Rotates the watermark text by 45 degrees, placing it diagonally across the page.
- **position=current page.center**: Centers the watermark on each page, ensuring it is consistently placed regardless of the content.
- **vshift=0pt** and **hshift=0pt**: These parameters adjust the vertical and horizontal position of the watermark. With both set to 0pt, the watermark remains centered on the page.
- **contents={\textsf{\Huge vitoroma101}}**: Defines the content of the watermark. In this case, “vitoroma101” is displayed in a large sans-serif font (`\Huge` and `\textsf{}`).

This setup ensures that the watermark “vitoroma101” appears centered and diagonally across each page of the document, with a subtle transparency to maintain readability.

5. Customizing the Final Page with Author Information and Hyperlinks

In this section, we will explain how to create a customized final page in your document that includes author information, centered content, and hyperlinks. This section covers the use of LaTeX commands to ensure that the final page is professionally formatted and includes all necessary details for readers to contact the author or follow their work. Below is an example of how to set up this final page:

```
\newpage
\begin{center}
\Large \textbf{About the Author}
\vspace{1cm}
If you would like to...
\vspace{1cm}
\textbf{Twitter (X):} \href{https://twitter.com/micosapiens711}{@micosapiens711}
\vspace{1cm}
Feel free to...
\end{center}
```

- `\newpage`: Forces a page break, ensuring that the “About the Author” section begins on a new, separate page. This helps maintain a clear structure in the document.
- `\begin{center} ... \end{center}`: This LaTeX environment centers all the content within it horizontally on the page. It’s used here to ensure that the “About the Author” section is centered.
- `\Large`: Adjusts the font size of the text. In this case, it’s used to make the “About the Author” title larger and more prominent.
- `\textbf{}`: Makes the enclosed text bold. This is used to emphasize the title “About the Author” and the names of the platforms (Twitter, GitHub, LinkedIn).
- `\vspace{1cm}`: Adds vertical space of 1 cm between elements on the page. This command is used to create space between the title, the list of platforms, and the closing text. It helps in distributing the content evenly and maintaining a clean layout.

- `\href{URL}{display text}`: This command creates a hyperlink in LaTeX. The `URL` is the link's destination, and the `display text` is what the reader sees in the document. This command is used instead of Markdown's standard link syntax because LaTeX commands are required within the `\begin{center}` environment.

Vitoroma101

About the Author

If you would like to contact me or follow my work, you can find me on the following platforms:

Twitter (X): @micosapiens711

GitHub: vitoroma101

LinkedIn: victoreduardo711

Feel free to reach out for collaborations, discussions, or just to connect!