Bookdown: Flexible Document Creation in RStudio

Guide for Students, Researchers, and Professionals

Joseph Thiers

# 1 Introduction to Bookdown

In today’s fast-paced academic and professional environments, the ability to create dynamic, reproducible documents is crucial. Bookdown empowers users to combine text, code, and visualizations in a single, streamlined workflow. It is ideal for creating the type of documentation that best fits your needs, whether that be single-page assignments, reports, academic papers, or even full-length books.

## 1.1 Why RStudio and Bookdown?

Bookdown offers a range of benefits: - Seamlessly integrates text, code, and figures. - Supports multiple output formats (HTML, PDF, EPUB). - Simplifies the creation of reproducible and professional documents. - Ideal for mathematics, statistics, and data science professionals.

## 1.2 What You’ll Learn in This Tutorial

This tutorial will guide you through the essential aspects of using Bookdown:

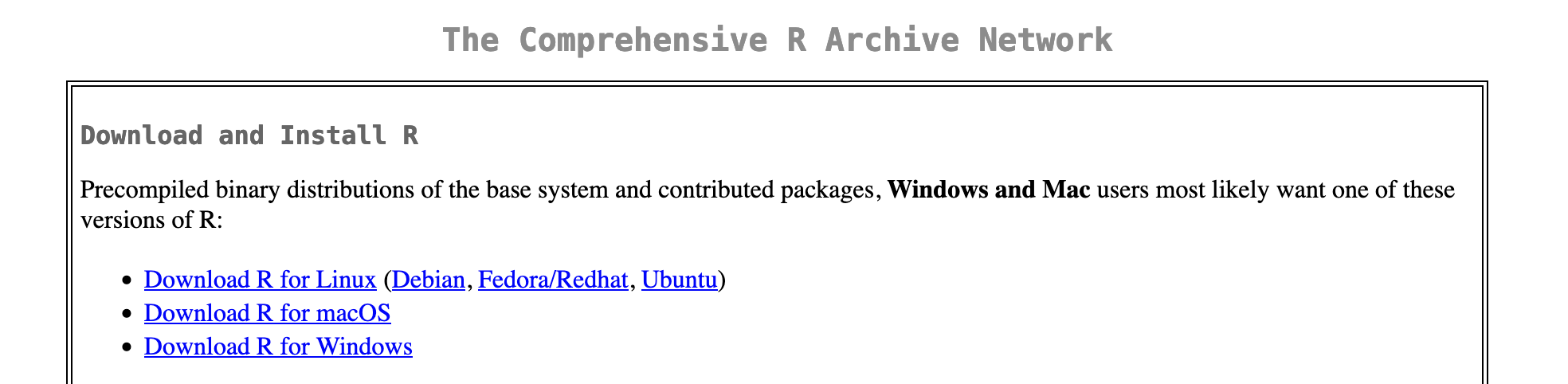
* **Chapter 1**: Introduction to Bookdown – Learn about its purpose and benefits for structured documentation.
* **Chapter 2**: Getting Started – Install R, RStudio, and Bookdown, create a project, and render your first book.
* **Chapter 3**: Writing Content – Organize chapters, use Markdown, format text, add code chunks, and images.
* **Chapter 4**: Cross-Referencing – Reference sections, figures, tables, and equations effectively.
* **Chapter 5**: LaTeX – Add equations, theorems, lemmas, and proofs.
* **Chapter 6**: Advanced Features – Manage citations, use LaTeX packages, and more!
* **Chapter 7**: Customizing Output – Configure formats like HTML, PDF, and EPUB, and style your book with CSS or LaTeX.
* **Chapter 8**: LaTeX Distributions - Different distributions available.
* **Chapter 9**: Advanced Text Formatting Options - Advanced Markdown and Pandoc code to stylize the book to your needs.
* **Chapter 10**: Example Document: Union Earnings Analysis

By the end of this tutorial, you’ll have the knowledge to create, customize, and publish professional-grade documents.

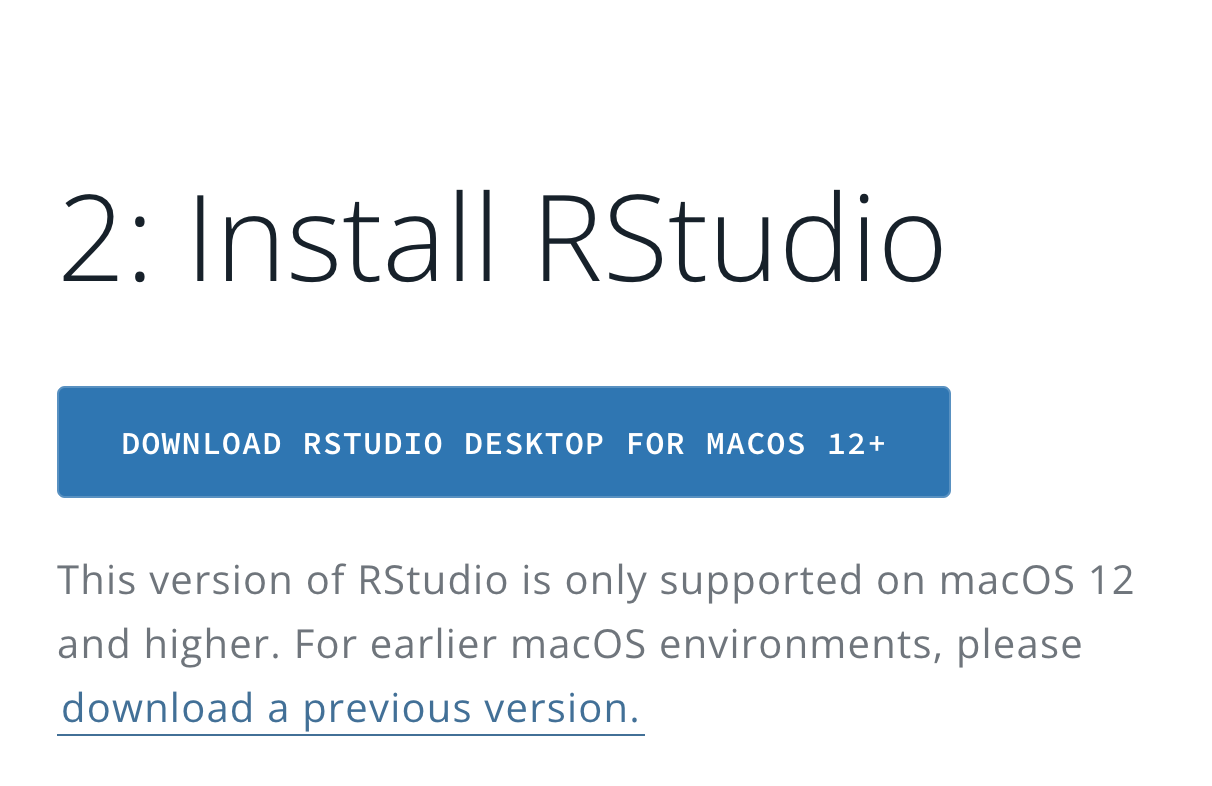
# 2 Getting Started

To get started with Bookdown you need to install R, RStudio, Bookdown, and the LaTeX distrobution of your choice if you wish to outpute as a PDF. This can be accomplished by following these steps::

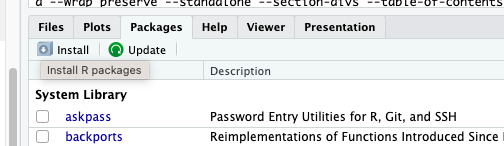
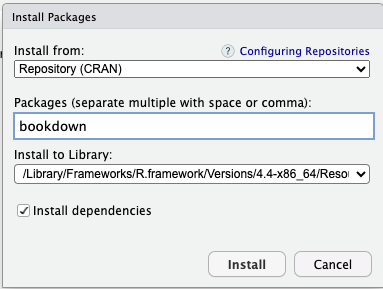
1. **Install R** Go to the [R Project download page](https://cran.r-project.org/) and download the latest version of R for your operating system (Windows, macOS, or Linux). Follow the installation instructions provided.



1. **Install RStudio**:  
   Go to the [RStudio download page](https://posit.co/download/rstudio-desktop/) and select the appropriate version for your operating system. Download and follow the installation instructions.

* 

1. **Install Bookdown**:  
   Once RStudio is installed, install the Bookdown package by typing the following command in the RStudio console:

* install.packages("bookdown")
* Alternatively, you can install the Bookdown package via the RStudio **Packages** pane:
  + Select **Packages** in the bottom right-hand corner of RStudio.
  + 
  + Click **Install**, type bookdown in the **Packages** box, and click **Install**.
  + 

1. **Install LaTeX distribution of your choice**: The distribution you choose is entirely up to you and your needs. For a list of recommended LaTeX distributions please see [LaTeX Distributions Chapter 8](#latexdistributions)

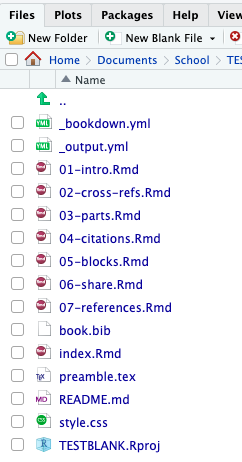
To get started quickly TinyTeX is recommended as it can be installed from within RStudio by running the following code:

install.packages("tinytex")  
tinytex::install\_tinytex()

Once this is complete Bookdown is now installed and you are ready to create your first Bookdown project.

1. **Create a New Bookdown Project in RStudio**:
   * In RStudio, go to **File > New Project**.
   * Select **New Directory** and then **Book Project using Bookdown**.
   * Name your project and choose a location to save it to.

Now you have a newly created Bookdown project ready to be edited to fit your needs. Bookdown comes with a base set of files so that users can get started quickly with minimal fuss. Spend a few moments to explore the files in the bottom right corner of RStudio, these are the default files Bookdown creates to get you started. It even includes sample chapters for you!



At the end of each section is a quick exercise for you to practice on your own files.

Then when you’re ready to complete your project just follow the last step below to render your book.

1. **Render Your Newly Created Book**:  
   In the **Build** pane:
   * Select **Build Book** and choose your output format, or select *All formats* to render your files as HTML, PDF, and EPUB.
   * You can also render the book directly from the R console with the following command:

* bookdown::render\_book("index.Rmd")

# 3 Writing Content

In this chapter, we will explore how to write and structure content in Bookdown using R Markdown syntax. Bookdown allows you to create well-organized documents by combining text, code, and references. Here, we’ll cover how to organize chapters, use Markdown and LaTeX for formatting, and format text, code chunks, and images.

## 3.1 Creating Chapters and Sections

Each chapter in Bookdown is represented by a separate .Rmd file, and each .Rmd file should begin with a first-level heading, marked by a single # symbol. For example, this chapter file is 02-writing-content.Rmd and the file starts with:

# Writing Content

### 3.1.1 Organizing Chapters

Chapters are automatically numbered based on their order in the project directory. Ensure that each file name reflects its chapter number (e.g., 02-writing-content.Rmd for Chapter 2).

### 3.1.2 Adding Sections and Subsections

You can add sections and subsections within a chapter using second-level and higher-level headings:

## Section Title  
### Subsection Title

This hierarchy organizes the document according to your needs, and these sections will automatically appear in the table of contents.

## 3.2 Formatting Text in Bookdown

Bookdown supports a wide range of Markdown formatting. Here are a few basics:

* **Bold**: \*\*bold text\*\* → **bold text**
* *Italics*: \*italicized text\* → *italicized text*
* **Bullet Points**:
  + First item
  + Second item
* **Numbered Lists**:
  1. First item
     + Even sublists
     + Like this
  2. Second item

For additional text formatting options refer to [Chapter 9 Advanced Text Formatting](#markdown-advanced).

Use these formatting options to style text and create lists within your chapters.

## 3.3 Adding Code Chunks

One of the strengths of Bookdown is the ability to incorporate code into your document, whether it’s R code, Markdown, LaTeX, Python, or other languages. Below are examples of how to include different types of code in your Bookdown project.

### 3.3.1 1. R Code

R code chunks are written between three backticks (``) with{r}` specifying R as the language:

\```{r eval=FALSE}  
summary(cars)  
\```

This code will display the summary function output of the cars dataset when eval is set to TRUE.

Say you are doing data-anaylsis on the Stars dataset in the dslabs package.

library(dslabs)  
summary(stars)

## star magnitude temp type   
## Altair : 2 Min. :-8.000 Min. : 2500 Length:96   
## \*40EridaniA: 1 1st Qu.:-1.800 1st Qu.: 3168 Class :character   
## \*40EridaniB: 1 Median : 2.400 Median : 5050 Mode :character   
## \*40EridaniC: 1 Mean : 4.257 Mean : 8752   
## \*61CygniA : 1 3rd Qu.:11.325 3rd Qu.: 9900   
## \*61CygniB : 1 Max. :17.000 Max. :33600   
## (Other) :89

Here we get a summary of the stars data.

### 3.3.2 2. Python Code

You can add Python code using {python} in your code chunks, and Bookdown will run the Python script and display the output.

\```{python}  
import numpy as np  
print(np.arange(10))  
\```

This example prints an array of numbers from 0 to 9 using Python.

### 3.3.3 3. Bash/Shell Script

For including shell commands, you can use {bash} as the language identifier.

\```{bash}  
echo "Hello, World!"  
\```

This chunk will run a simple echo command to print “Hello, World!”.

### 3.3.4 4. SQL Code

To add SQL queries, you can use {sql} for the language identifier.

\```{sql, connection="your-db-connection"}  
SELECT \* FROM employees LIMIT 10;  
\```

This SQL code retrieves the first ten rows from the employees table. You need to set up a database connection for this chunk to execute.

### 3.3.5 Customizing Code Chunk Options

You can customize how code chunks appear using chunk options. Here are a few common options:

* echo=FALSE: Hides the code but displays the output.
* eval=FALSE: Shows the code but does not execute it.
* fig.cap="Caption Text": Adds a caption to figures generated from the code chunk.
* out.width="50%": Sets the output width for images generated in the chunk.

Example:

\```{r echo=TRUE, fig.cap="A summary of the cars dataset"}  
summary(cars)  
\```

Experiment with these options to control how your code and output appear.

## 3.4 Adding Images to Your Bookdown Project

Images can enhance your document by providing visual context. In Bookdown, you can add images using Markdown syntax or by embedding images using R code chunks.

### 3.4.1 Adding Images with RStudio GGPLOT

The easiest way to add images is with the built in plotting functions of RStudio, or the ggplot package. Simply add an R code chunk and enter the code you wish to run. For example, say you you want to show a plot showing the wage increases for Union Members compared to non-Union members.

ggplot(wages, aes(x = year)) +  
 geom\_smooth(aes(y = union\_wage, color = "Union Wage"),   
 se = FALSE, method = "loess") +  
 geom\_smooth(aes(y = nonunion\_wage, color = "Nonunion Wage"),   
 se = FALSE, method = "loess") +  
 labs(x = "Year", y = "Wage",  
 color = "Type of Wage") +  
 theme\_minimal(base\_size = 14) +  
 theme(legend.position = "bottom") +  
 scale\_color\_manual(values = c("Union Wage" = "blue", "Nonunion Wage" = "green")) +  
 guides(color = guide\_legend(title = "Wage Type"))

## `geom\_smooth()` using formula = 'y ~ x'  
## `geom\_smooth()` using formula = 'y ~ x'

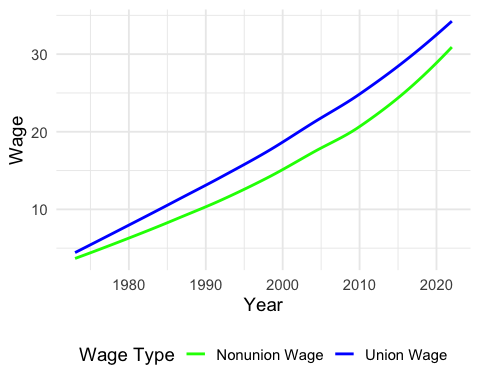


Figure 3.1: Union Vs Non-Union Wages

As you can see the above code then creates a plot and places it in your document for you. This way you can fully customize the plot from within RStudio.

### 3.4.2 Adding Images Using Markdown

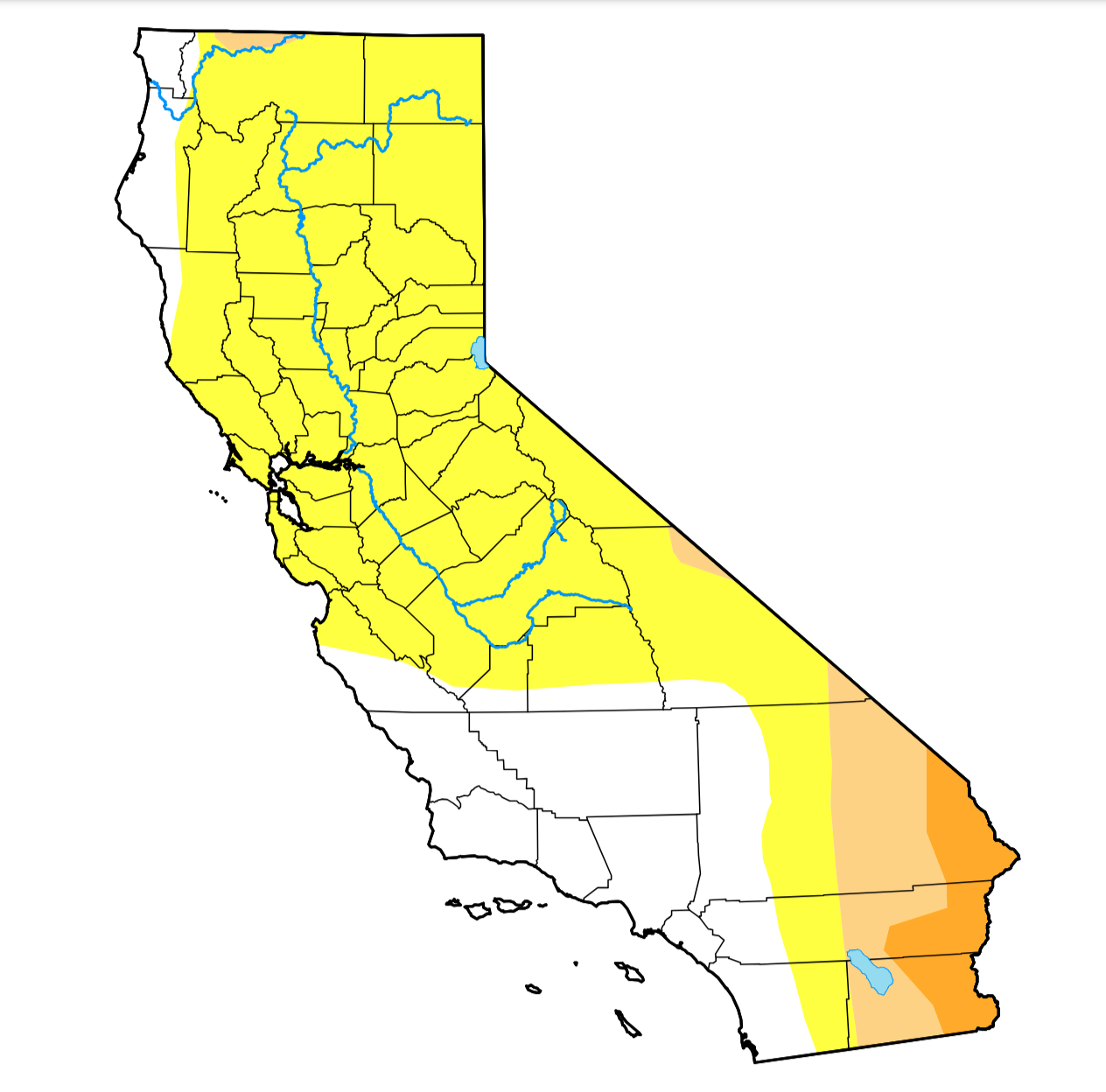
To add an image using Markdown, use the following syntax:

![Alt text for the image](images/drought-map.png)

* **Alt text**: A description of the image that is useful for accessibility.
* **Path to the image**: This can be a relative path (e.g., ``) or a URL.

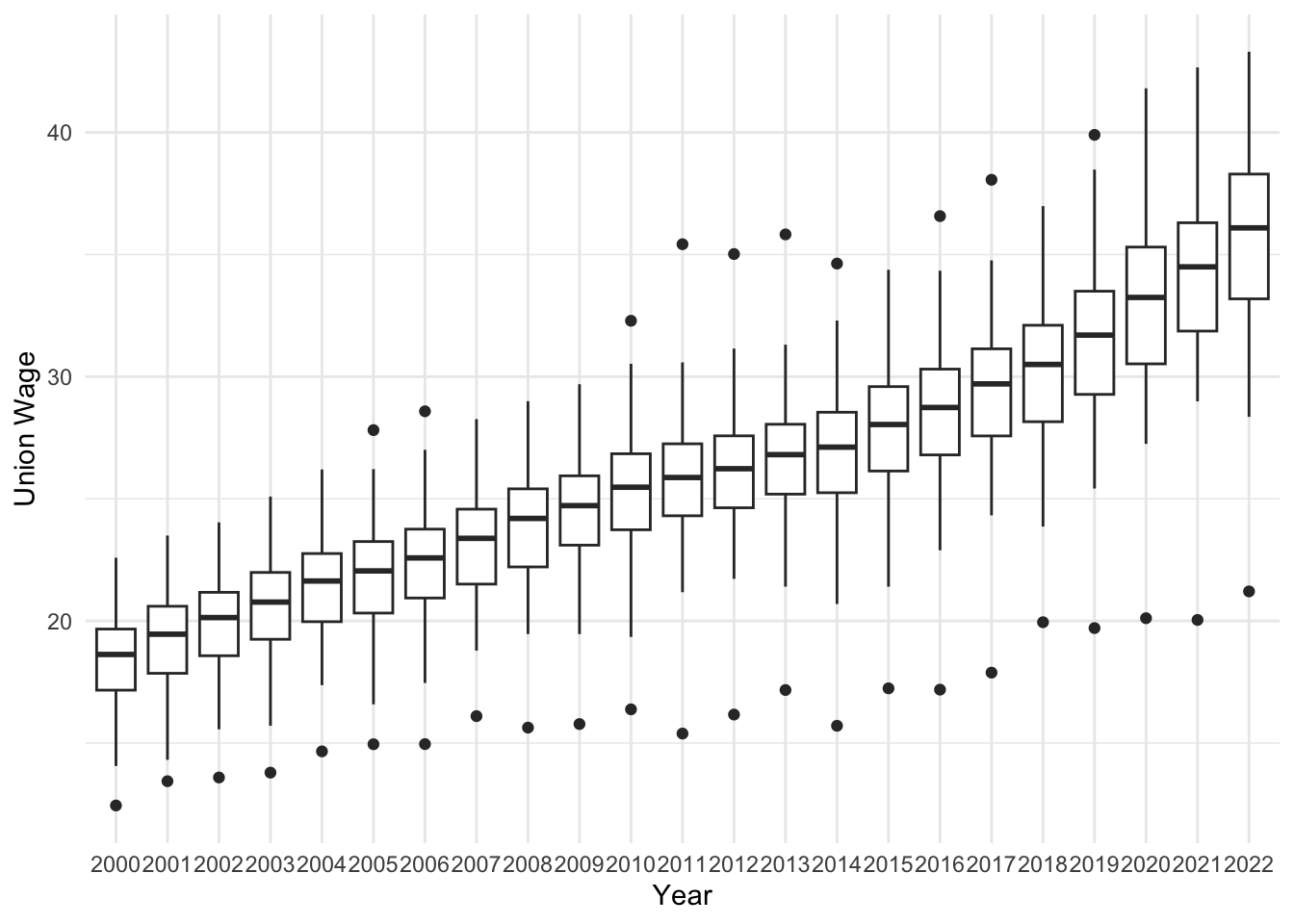
Example:

![A sample image](images/drought-map.png)

 ### Adding Images Using Code Chunks

You can also add images using an R code chunk, which can be useful when the image is generated programmatically. Use the knitr::include\_graphics() function:

knitr::include\_graphics("images/wageyear-1.png")

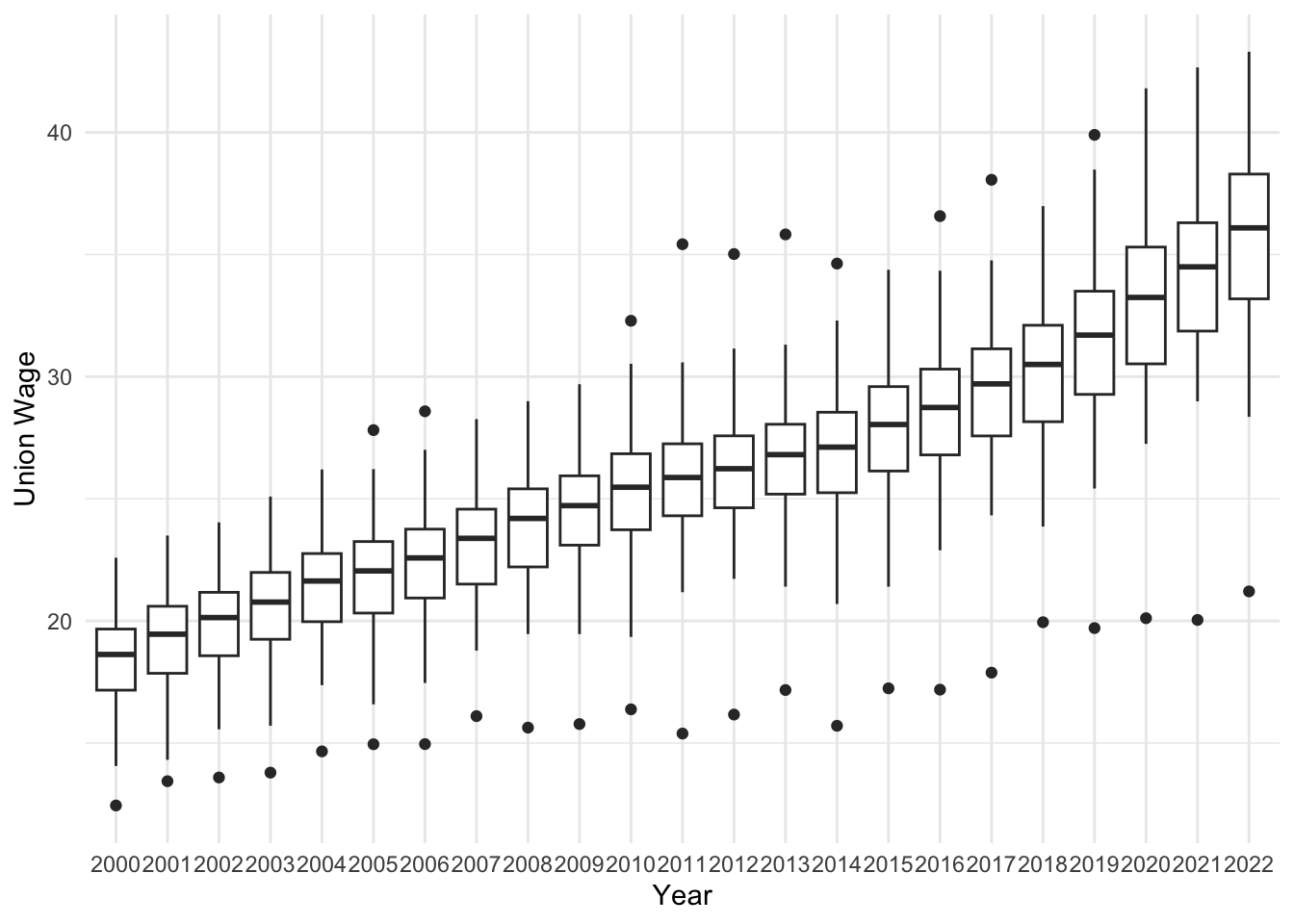


### 3.4.3 Chunk Options for Images

When adding images through code chunks, you can customize their appearance using chunk options such as fig.cap for captions and out.width for sizing.

For example the following has the caption fig.cap="An example image" and has an output width of 25% with out.width="25%. As you can see the image is much smaller than allowing the output width to not be adjusted.

knitr::include\_graphics("images/wageyear-1.png")

 :::: {.blackbox data-latex=““} ::: {.center data-latex=”“} **EXERCISE TIME!** :::

Create a new .Rmd file and write a short chapter that includes different types of Markdown elements (e.g., headers, lists, and images). Experiment with adding code chunks, andtext formatting. ::::

# 4 Cross Referencing and Citatians

Cross-references make it easier for readers to find and link to elements in your book. In Bookdown, you can create cross-references for sections, figures, tables, and equations. This chapter explains how to use cross-references effectively.

## 4.1 Cross-Referencing Sections

To reference a section, first add an ID or tag to the heading by including {#your-id} at the end of the section header. As an example here is the last chapter of the book that is an analysis I created within bookdown set as a single chapter:

# Example Document: Union Earnings Analysis (#examplepaper)

You can then refer to it later, with a clickable link that will take your reader directly to it, in your document as follows:

See Section \@ref(examplepaper) for more information.

## 4.2 See Section 10 for more information.

## 4.3 Cross-Referencing Text

You can assign some text to a label and reference the text using the label elsewhere in your document. This can be particularly useful for long figure/table captions or when you need to reuse text fragments in multiple places.

The syntax for a text reference is (ref:label) text, where label is a unique label throughout the document. It must be in a separate paragraph with empty lines above and below it. The paragraph must not be wrapped into multiple lines, and should not end with whitespace.

**Additional Guidelines**: - Ensure that text references are in a separate paragraph with empty lines above and below. - Avoid wrapping the paragraph containing the text reference into multiple lines. - Make sure there is no trailing whitespace at the end of the paragraph.

Example:

This is the text we are referencing This is the text we are referencing

Then you can reference this text anywhere in your document using (ref:textreference).

For example:

Then we reference it: This is the text we are referencing.

## 4.4 Cross-Referencing Figures

To cross-reference a figure, set a chunk label and use the fig.cap option to add a caption. Bookdown automatically labels the figure with fig:chunk-label.

For example in the previous chapter we showed a plot comparing Union Wages vs Non-Union Wages. It had the following in the code chunk heading:

{r wagecomparison, fig.cap="Union Vs Non-Union Wages"}

We can easily create a reference back to this plot, that is clickable so users can easily go back to it, with the following:

See Figure \@ref(fig:wagecomparison) for details.

See Figure 3.1 for details.

## 4.5 Cross-Referencing Tables

To cross-reference a table, use knitr::kable() to create a table with a caption. Bookdown automatically labels the table with tab:chunk-label.

Example:

knitr::kable(head(cars), caption = "Table of the first rows of the cars dataset")

Table 4.1: Table of the first rows of the cars dataset

| speed | dist |
| --- | --- |
| 4 | 2 |
| 4 | 10 |
| 7 | 4 |
| 7 | 22 |
| 8 | 16 |
| 9 | 10 |

Reference this table using:

See Table \@ref(tab:cars-table) for details.

See Table 4.1 for details.

## 4.6 Cross-Referencing Equations

For equations, use (\#eq:label) to label the equation and \@ref(eq:label) to reference it. This way when you need to reference the equation at a later point in your document you can.

### 4.6.1 Example: Creating and Labeling an Equation

First you write the equation in LaTeX in the equation environment and include the label as (\#eq:label):

\begin{equation}  
 E = mc^2  
 (\#eq:einstein)  
\end{equation}

Which outputs as such:

Then at a later point when you need to reference it you simply type in \@ref(eq:label). As we know Einsten discovered that (4.1).

## 4.7 Adding Citations and Managing References

Bookdown makes it easy to manage references and add citations by using BibTeX files. Here’s how to set up and include references in your Bookdown project.

### 4.7.1 Step 1: Create a .bib File

First, create a .bib file for your references (e.g., references.bib). You can add references in BibTeX format. Here’s an example entry:

@Book{Bookdown,  
 title = {Bookdown: Authoring Books and Technical Documents with R Markdown},  
 author = {Yihui Xie},  
 publisher = {Chapman; Hall/CRC},  
 year = 2024,  
 edition = {2nd},  
 note = {ISBN 9780367142568},  
 url = {https://bookdown.org/yihui/bookdown/}  
}

### 4.7.2 Step 2: Link the .bib File in index.Rmd

In your index.Rmd file, include the .bib file in the YAML header:

bibliography: [references.bib]  
link-citations: yes

### 4.7.3 Step 3: Cite Sources in Your Text

To cite a source, use [@citation-key] in your text, where citation-key matches the key in your .bib file (e.g., [@Bookdown]). For example:

This tutorial was written thanks to Bookdown [@Bookdown].

This tutorial was written thanks to Bookdown ([Xie 2024](#ref-Bookdown)).

Bookdown will automatically format your citation based on the output style.

### 4.7.4 Step 4: Customize Citation Style (Optional)

If you need a specific citation style, you can add a .csl (Citation Style Language) file in your project and reference it in the YAML header:

csl: "chicago-author-date.csl"

Download .csl files from sources like [Zotero](https://www.zotero.org/styles).

## 4.8 Using Cross-References with Citations

In addition to referencing external sources, Bookdown allows you to cross-reference sections, figures, and tables, as discussed earlier. As a reminder:

As shown in Figure \@ref(fig:example-figure), the trend is evident.

## 4.9 Using cross-references with citations helps keep your document organized and easy to navigate.

**EXERCISE TIME!**

Go ahead and place a few entries in your .bib file. Once you’ve done that go through and reference these in your previously written content.

# 5 LaTeX in Bookdown

Bookdown offers powerful support for LaTeX, allowing you to seamlessly integrate any LaTeX packages you need into your documents. Whether you’re working with mathematical equations, theorems, lemmas, proofs, or other advanced features, this tutorial will guide you through the essentials of using LaTeX in Bookdown, showing you how to effectively incorporate and reference these elements.

## 5.1 Including LaTeX Packages in Bookdown

One of the powerful features of Bookdown is its seamless integration with LaTeX, allowing you to include any LaTeX package that suits your needs. This flexibility is especially useful when working with advanced mathematical notations, custom formatting, or specialized content.

### 5.1.1 Using a preamble.tex File

To include LaTeX packages, you need to create a preamble.tex file and link it in your \_output.yml file. The preamble.tex file is processed before the document is rendered allowing you to load any additional LaTeX packages you may need. The preamble.tex file also allows you to fully cuztomize the output of your PDF using your existing LaTeX knowledge.

1. **Create a preamble.tex File** Create a file on your computer with the name preamble.tex and add it to the directory of your Bookdown project if it does not already exist. If the file already exists simply open the file within RStudi for easy editing. You can include as many packages as you need in the preamble.tex file. For example here is a list of packages and small changes I made to output the final example document in this tutorial as an a6 sized document:

\usepackage{booktabs}  
\usepackage{anyfontsize}  
\usepackage{titlesec}  
\usepackage{fancyhdr}  
\usepackage[paperwidth=4.13in, paperheight=5.83in]{geometry}  
\usepackage[font=small, labelfont=bf]{caption}  
\geometry{left=.5in}  
\geometry{right=.5in}  
\geometry{bottom=.5in}  
\geometry{top=.75in}

1. **Link preamble.tex in \_output.yml**  
   You need to make sure that the preamble is included in your \_output file. Update your \_output.yml file to include the preamble.tex file for PDF output:

bookdown::pdf\_book:  
 includes:  
 in\_header: preamble.tex  
 latex\_engine: xelatex  
 citation\_package: natbib  
 keep\_tex: yes

### 5.1.2 Debugging Errors

If your document fails to compile, check the error message in the console or look at the .log file generated by LaTeX for troubleshooting.

By including custom LaTeX packages, you can extend the functionality of your Bookdown project to suit your unique requirements, whether it’s for academic papers, technical documents, or books.

## 5.2 Mathematical Equations

Bookdown makes it easy to include both inline and display-style equations while also allowing you to reference these equations for easy reference at any point within your project.

### 5.2.1 Inline Equations

Use $...$ to include inline math equations within your text. The following code:

The formula for the area of a circle is $( A = \pi r^2 )$, where $( r )$ is the radius.

Then is displayed as follows:

The formula for the area of a circle is , where is the radius.

### 5.2.2 Display Equations

For equations that you may want to have on their own line you have two options. You can surround your LaTeX code $$...$$ or alternatively use the LaTeX equation environment.

Here we see the usage of $$...$$

$$  
E = mc^2  
$$

Which gives us:

Here we are using the equation environment with a label:

\begin{equation}  
 E = mc^2  
 (\#eq:einstein)  
\end{equation}

Which outputs as such:

Reminder for referencing equations: For easy reference within your document we are also able to add a reference to any equation by using \#eq:your-label as such:

(\#eq:einstein)

So that you can easily reference it later with \@ref(eq:einstein). As an example, we labeled the above equation as (\#eq:einstein). So we can reference it by using \@ref(eq:einstein):

As shown in Equation \@ref(eq:einstein), energy is proportional to mass.

As shown in Equation (4.1), energy is proportional to mass.

## 5.3 Theorems, Lemmas, and Proofs

Bookdown supports theorems, lemmas, and proofs, for structured mathematical writing.

### 5.3.1 Adding a Theorem

Define a theorem using the following syntax:

::: {.theorem #theoremlabel}  
This is a `theorem` environment that can contain \*\*any\*\*  
\_Markdown\_ syntax.  
For any integer $n \geq 1$, the sum of the first $n$ positive integers is given by  
$$  
S = \frac{n(n + 1)}{2}  
$$  
:::

Reference the theorem in your text by add #theoremlabel to your code block as such:

::: {.theorem #theoremlabel}  
This is a `theorem` environment that can contain \*\*any\*\*  
\_Markdown\_ syntax.  
For any integer $n \geq 1$, the sum of the first $n$ positive integers is given by  
$$  
S = \frac{n(n + 1)}{2}  
$$  
:::

Which will cause your theorem to display as follows:

This is a theorem environment that can contain **any** *Markdown* syntax. For any integer , the sum of the first positive integers is given by

If this is a theorem that you need to reference in other parts of your document you can label it as we did above with #theoremlabel. Then when you need to reference it you can do so using \@ref(thm:theoremlabel) wherever needed. As an example, lets refer back to the above Theorem ??.

### 5.3.2 Adding a Lemma

Define a lemma similarly:

::: {.lemma #lemmalabel}  
Let $x\in \mathbb Z$.If $5x-7$ is odd,then $x$ is even.  
:::

Let . If is odd,then is even.

Reference the lemma:

Lemma \@ref(lem:lemmalabel) confirms that the sum of two even integers is even.

Lemma ?? confirms that the sum of two even integers is even.

### 5.3.3 Adding a Proof

Proofs can be added using the proof environment.

::: {.proof}  
\textbf{Proof: }Let $x \in \mathbb Z$. Suppose $7x+5$ is odd. Then $7x+5=2k+1$ for some $k \in \mathbb Z$. Then  
\begin{align\*}  
 7x+5&=2k+1\\  
 7x&=2k-4\\  
 x&=2k-6x-4\\  
 x&=2(k-3x-2).  
\end{align\*}  
Since $k-3x-2 \in \mathbb Z$, $x$ is even.  
:::

Let . Suppose is odd. Then for some . Then

Since , is even.

### 5.3.4 Referencing Various Math Environments

When you need to refer to create a different environment you just need to change what’s between the brackets to the type of environment you are using.

::: {.environment}  
Your information goes here  
:::

When you are referencing the environment you would just ensure that you use the label prefix from the below table for the environment you wish to reference, changing the word prefix in:\@ref(prefix:label) to the appropriate environment.

Table 5.1: Theorem environments in Bookdown.

| Environment | Printed Name | Label Prefix |
| --- | --- | --- |
| theorem | Theorem | thm |
| lemma | Lemma | lem |
| corollary | Corollary | cor |
| proposition | Proposition | prp |
| conjecture | Conjecture | cnj |
| definition | Definition | def |
| example | Example | exm |
| exercise | Exercise | exr |
| hypothesis | Hypothesis | hyp |

## 5.4 Tips for LaTeX in Bookdown

* **Use Labels Consistently**: Use meaningful and unique labels for cross-referencing.
* **Use Math Mode**: Always enclose mathematical symbols in $...$ or $$...$$ to render correctly.
* **Add Theorem Styles**: Customize theorem environments in \_bookdown.yml for specific needs: markdown theorem: lab: "Theorem " lem: "Lemma "

**EXERCISE TIME!**

Add an equation and a theorem to your document using LaTeX. Alternatively, copy over some LaTeX code you’ve written before and place it in one of your chapters.

# 6 Advanced Features

In this chapter, we’ll explore some advanced features of Bookdown.

## 6.1 Using LaTeX for Advanced Formatting

Bookdown supports advanced formatting using **LaTeX**. This allows you to create custom environments, add annotations, and format mathematical content beautifully.

### 6.1.1 Example: Adding a Custom Box

You can define a custom LaTeX environment for a “black note box” in the preamble.tex file:

\setlength{\fboxsep}{.8em}  
  
\newenvironment{blackbox}{  
 \definecolor{shadecolor}{rgb}{0, 0, 0} % black  
 \color{white}  
 \begin{shaded}}  
 {\end{shaded}}

Then use the custom box in your content:

:::: {.blackbox data-latex=""}  
::: {.center data-latex=""}  
\*\*EXAMPLE!\*\*  
:::  
Giving you this as an output.  
::::

**EXAMPLE!**

Giving you this as an output.

## 6.2 Customizing Document with LaTeX

You can even customize the entire look and output of your document by editing the content in your preamble. For example, say you wish to create an a6 size booklet and so need to modify the font sizes as I did in the for the Example Document at the end of this tutorial. In thise the preamble would look as follows:

\usepackage{booktabs}  
\usepackage{anyfontsize}  
\usepackage{titlesec}  
\usepackage{fancyhdr}  
\usepackage[paperwidth=4.13in, paperheight=5.83in]{geometry}  
\usepackage[font=small, labelfont=bf]{caption}  
\geometry{left=.5in}  
\geometry{right=.5in}  
\geometry{bottom=.5in}  
\geometry{top=.75in}  
\renewcommand{\normalsize}{\fontsize{6pt}{8pt}\selectfont}  
\titleformat{\section}{\fontsize{8pt}{10pt}\bfseries}{\thesection}{1em}{}  
\titleformat{\subsection}{\fontsize{6pt}{8pt}\bfseries}{\thesubsection}{1em}{}  
\titleformat{\subsubsection}{\fontsize{7pt}{9pt}\bfseries}{\thesubsubsection}{1em}{}  
\usepackage{tocloft}  
\renewcommand{\cftsecfont}{\fontsize{6pt}{8pt}}  
\renewcommand{\cftsubsecfont}{\fontsize{7pt}{9pt}}  
\renewcommand{\cftsubsubsecfont}{\fontsize{6pt}{8pt}}  
\usepackage{listings}  
\lstset{  
 basicstyle=\ttfamily\fontsize{2pt}{4pt},  
 breaklines=false,  
 frame=single,  
}  
\usepackage{titlesec}  
\titleformat{\chapter}[display]  
 {\fontsize{14pt}{18pt}\bfseries}   
 {\chaptername~\thechapter}   
 {1em}   
 {}  
\titleformat{\section}  
 {\fontsize{10pt}{12pt}\bfseries}   
 {\thesection}   
 {1em}   
 {}  
\titleformat{\subsection}  
 {\fontsize{10pt}{12pt}\bfseries}   
 {\thesubsection}   
 {1em}   
 {}  
\setlength{\fboxsep}{.8em}

## 6.3 Adding Custom CSS and JavaScript

Enhance the appearance and functionality of your Bookdown project by adding custom CSS and JavaScript.

### 6.3.1 Step 1: Create a Custom CSS File

Create a styles.css file with your custom styles:

body {  
 font-family: "Arial", sans-serif;  
}  
h1 {  
 color: #4CAF50;  
}

### 6.3.2 Step 2: Include the CSS File in the YAML Header

Add the following to your index.Rmd:

output:  
 bookdown::gitbook:  
 css: styles.css

### 6.3.3 Step 3: Adding JavaScript (Optional)

For dynamic behavior, include JavaScript files similarly:

output:  
 bookdown::gitbook:  
 includes:  
 in\_header: "scripts.js"

**EXERCISE TIME!**

Use custom LaTeX to add a unique visual element to your content (e.g., a shaded box or note). Play around with the settings and see what you like best.

# 7 Customizing Output

In this chapter, we’ll explore how to customize the output of your Bookdown project. Bookdown supports several output formats, such as HTML, PDF, and EPUB, and allows you to customize each format to match your project’s needs. We’ll cover choosing output formats, modifying appearance, and configuring output settings.

## 7.1 Choosing an Output Format

Bookdown provides multiple output formats that allow you to publish your document in various ways:

* **HTML**: Ideal for online documentation or sharing on the web.
* **PDF**: Useful for print-ready documents, especially for academic or professional reports.
* **EPUB**: E-book format, compatible with e-readers for mobile access.

To specify output formats, edit the \_output.yml file in your project directory. Here’s an example configuration:

bookdown::gitbook:  
 css: "style.css"  
 split\_by: "chapter"  
  
bookdown::pdf\_book:  
 latex\_engine: xelatex  
 citation\_package: natbib  
  
bookdown::epub\_book: default

This configuration tells Bookdown to create HTML (using GitBook style), PDF, and EPUB formats. Customize each format’s settings to control the output style.

## 7.2 Available Output Options for Bookdown

### 7.2.1 1. GitBook (bookdown::gitbook)

This is one of the most popular output formats, producing an HTML book with interactive features such as search and navigation. Infact this is how the book has been outputted.

Example:

output:  
 bookdown::gitbook:  
 css: style.css # Optional: Custom CSS for styling  
 config:  
 toc:  
 collapse: section # Controls the collapsing of the table of contents

### 7.2.2 2. PDF Book (bookdown::pdf\_book)

Generates a PDF version of your book using LaTeX. This format is often used for printed versions.

Example:

output:  
 bookdown::pdf\_book:  
 latex\_engine: xelatex # Specify the LaTeX engine to use (e.g., pdflatex, xelatex, lualatex)  
 includes:  
 in\_header: preamble.tex # Include additional LaTeX setup files  
 citation\_package: natbib # Citation package to use

### 7.2.3 3. EPUB Book (bookdown::epub\_book)

Creates an EPUB file, which is a format commonly used for eBooks.

Example:

output:  
 bookdown::epub\_book:  
 toc: yes # Include table of contents  
 css: style.css # Optional: Custom CSS for EPUB styling

### 7.2.4 4. HTML Document (rmarkdown::html\_document)

Produces a single HTML document, rather than a book format.

Example:

output:  
 rmarkdown::html\_document:  
 toc: true # Include table of contents  
 toc\_depth: 3 # Depth of the table of contents  
 number\_sections: true # Number sections in the output

### 7.2.5 5. Word Document (bookdown::word\_document2)

Generates a Word document (.docx) version of your book, useful for sharing with editors.

Example:

output:  
 bookdown::word\_document2:  
 toc: yes # Include table of contents  
 toc\_depth: 2 # Depth of table of contents

### 7.2.6 6. Tufte Handout (bookdown::tufte\_html\_book / bookdown::tufte\_pdf\_book)

Generates a book in Tufte style, known for its distinctive design, which is suitable for visually focused content.

Example:

output:  
 bookdown::tufte\_html\_book: default

For PDF:

output:  
 bookdown::tufte\_pdf\_book: default

### 7.2.7 7. HTML with Bookdown-Specific Features (bookdown::html\_document2)

Similar to rmarkdown::html\_document but includes additional cross-referencing features from Bookdown.

Example:

output:  
 bookdown::html\_document2:  
 toc: true  
 number\_sections: true

### 7.2.8 8. GitHub Document (rmarkdown::github\_document)

Produces a document suitable for rendering on GitHub.

Example:

output:  
 rmarkdown::github\_document:  
 toc: true # Include table of contents

## 7.3 Customizing HTML Output

To customize the HTML format, Bookdown offers the bookdown::gitbook and bookdown::html\_document2 options:

* **GitBook**: The default HTML style, which includes a side navigation bar and a search function. This format is ideal for online documentation.
* **HTML Document**: A simpler format without the sidebar, suitable for single-page reports.

You can adjust HTML settings in \_output.yml:

bookdown::gitbook:  
 css: "style.css"  
 config:  
 toc:  
 collapse: section  
 download: ["pdf", "epub"]

## 7.4 Customizing PDF Output

To generate a high-quality PDF, you’ll need to install a LaTeX distribution like TinyTeX.

In \_output.yml, you can specify options to control PDF formatting:

bookdown::pdf\_book:  
 latex\_engine: xelatex  
 citation\_package: natbib

* **latex\_engine**: Specifies the LaTeX engine (e.g., xelatex, pdflatex). Using xelatex improves font compatibility.
* **citation\_package**: Choose between natbib or biblatex for handling citations.

### 7.4.1 Document Size

To customize the size of your PDF document, you can modify the geometry option in the LaTeX preamble. For example, to set the paper size to A4 and customize the margins:

output:  
 bookdown::pdf\_book:  
 includes:  
 in\_header: preamble.tex

In the preamble.tex file, add:

\usepackage[paperwidth=5.5in, paperheight=8.5in, margin=0.75in]{geometry}

This sets the document size to A4 and the margins to 1 inch.

### 7.4.2 Document Type

You can change the document type by modifying the LaTeX class used in the preamble.tex. For example, use the article class instead of the default book class:

\documentclass{article}

This is useful for a more compact layout, like that of a research paper or report.

### 7.4.3 LaTeX Engine

You can specify the LaTeX engine used to compile your PDF. Bookdown supports different engines, such as pdflatex, xelatex, and lualatex.

Example:

output:  
 bookdown::pdf\_book:  
 latex\_engine: xelatex

Using xelatex or lualatex provides better font support, especially for special characters or non-Latin scripts.

### 7.4.4 Custom LaTeX Packages

Include custom LaTeX packages in the preamble.tex file to extend the functionality of your PDF output.

Example:

\usepackage{amsmath}

This includes the amsmath package for advanced mathematical formatting. You can also use \newcommand to define custom commands used throughout your document.

### 7.4.5 Page Layout Customization

To customize the page layout, use packages like fancyhdr to modify headers and footers.

Example:

\usepackage{fancyhdr}  
\pagestyle{fancy}  
\fancyhead[LE,RO]{Your Book Title}  
\fancyfoot[CE,CO]{\thepage}

This sets custom headers and footers, including the book title and page numbers.

## 7.5 Customizing EPUB Output

To create an EPUB e-book, use bookdown::epub\_book in \_output.yml. Bookdown handles most EPUB formatting automatically, but you can make some modifications:

bookdown::epub\_book:  
 stylesheet: "style.css"  
 cover\_image: "images/cover.jpg"  
 toc: yes

This configuration adds a cover image, applies the CSS stylesheet, and includes a table of contents.

## 7.6 Specifying Global Settings in \_bookdown.yml

The \_bookdown.yml file allows you to set global configurations, such as the order of chapters, the naming convention for output files, and the label format for figures and tables. Here’s an example:

book\_filename: "my-book"  
rmd\_files: ["index.Rmd", "01-introduction.Rmd", "02-writing-structuring-content.Rmd", "03-customizing-output.Rmd"]  
language:  
 label:  
 fig: "Figure "  
 tab: "Table "  
delete\_merged\_file: true

* **book\_filename**: Sets the base filename for output files.
* **rmd\_files**: Specifies the order of chapters.
* **language.label**: Customizes labels for figures and tables.
* **delete\_merged\_file**: Deletes intermediary files after rendering, keeping the directory clean.

## 7.7 Example Output

To render all formats simultaneously, you can use the render\_book() function in the R console:

bookdown::render\_book("index.Rmd", output\_format = "all")

This command generates HTML, PDF, and EPUB files as specified in \_output.yml.

**EXERCISE TIME!**

Modify your Bookdown project’s \_output.yml to customize the output format. Try changing the appearance of the HTML and PDF outputs. Write a paragraph explaining how the customizations improved the document’s look and usability.

When you’re ready go ahead and render your book into HTML, PDF, and EPUB formats. Publish it on GitHub Pages or share it with someone else. Write a paragraph reflecting on what you learned about the publishing process and any challenges you encountered.

# 8 LaTeX Distributions

To render PDF outputs with Bookdown, you need to install a LaTeX distribution. Below is a list of popular options, categorized by operating system and user preferences:

## 8.1 Recommended LaTeX Distribution

### 8.1.1 1. **TinyTeX** (Recommended)

* **Description**: A lightweight, cross-platform LaTeX distribution designed to work seamlessly with R and Bookdown.
* **Installation**: Run the following commands in R:
* install.packages("tinytex")  
  tinytex::install\_tinytex()
* **Advantages**:
  + Minimal installation size.
  + Automatically installs missing packages when rendering.
* **Website**: [TinyTeX Documentation](https://yihui.org/tinytex/)

## 8.2 Additional LaTeX Distributions

### 8.2.1 2. **TeX Live**

* **Description**: A comprehensive LaTeX distribution suitable for Linux and cross-platform users.
* **Installation**:
  + **Linux**:
  + sudo apt-get install texlive-full
  + **macOS and Windows**: Download from [TeX Live](https://www.tug.org/texlive/).
* **Advantages**:
  + Full-featured with a vast collection of LaTeX packages.
  + Stable and widely used.
* **Website**: [TeX Live Documentation](https://www.tug.org/texlive/)

### 8.2.2 3. **MikTeX**

* **Description**: A user-friendly LaTeX distribution popular among Windows users.
* **Installation**: Download and install from [MikTeX](https://miktex.org/).
* **Advantages**:
  + On-demand installation of missing packages.
  + Easy-to-use package manager.
* **Website**: [MikTeX Documentation](https://miktex.org/)

### 8.2.3 4. **MacTeX** (for macOS)

* **Description**: A macOS-specific version of TeX Live with additional tools for macOS users.
* **Installation**: Download and install from [MacTeX](https://www.tug.org/mactex/).
* **Advantages**:
  + Tailored for macOS with GUI tools like TeXShop.
  + Includes a full TeX Live distribution.
* **Website**: [MacTeX Documentation](https://www.tug.org/mactex/)

### 8.2.4 5. **ProTeXt** (for Windows)

* **Description**: A Windows-specific distribution that combines MikTeX with a user-friendly installer.
* **Installation**: Download and install from [ProTeXt](https://www.tug.org/protext/).
* **Advantages**:
  + Streamlined setup for beginners.
  + Integrates LaTeX editors like TeXworks.
* **Website**: [ProTeXt Documentation](https://www.tug.org/protext/)

Choose the distribution that best fits your operating system and needs. For most users, TinyTeX is the easiest to install and manage, especially if you’re using R and Bookdown.

# 9 Advanced Text Formatting Options

Markdown and it’s enchanged version Pandoc included with Bookdown allow for a wide variety of text formatting, making it easy to structure documents and highlight important content. Below, you’ll find a comprehensive guide to formatting options you can use in Markdown and Pandoc.

## 9.1 1. Headers

Headers are used to create section headings, and the number of # symbols represents the level of the header.

* # Header 1
* ## Header 2
* ### Header 3
* #### Header 4
* ##### Header 5
* ###### Header 6

## 9.2 2. Emphasis

* **Bold**: \*\*bold text\*\* or \_\_bold text\_\_
* *Italics*: \*italic text\* or \_italic text\_
* ***Bold and Italic***: \*\*\*bold and italic\*\*\* or \_\_\_bold and italic\_\_\_

## 9.3 3. Strikethrough

* ~~Strikethrough~~: ~~strikethrough text~~

## 9.4 4. Lists

### 9.4.1 4.1 Bullet Lists

* - or \* creates a bullet list.
* - First item  
  - Second item  
   - Subitem

### 9.4.2 4.2 Numbered Lists

* 1. creates a numbered list.
* 1. First item  
  2. Second item  
   1. Subitem

### 9.4.3 4.3 Task Lists

* [ ] creates a task list.
* - [x] Completed task  
  - [ ] Incomplete task

## 9.5 5. Blockquotes

* Blockquote: Use > for blockquotes.
* > This is a blockquote.

## 9.6 6. Code

### 9.6.1 6.1 Inline Code

* Wrap code with backticks: `code`

### 9.6.2 6.2 Code Blocks

* Use triple backticks () for code blocks.markdown ```python print(“Hello, World!”) ``` ```

## 9.7 7. Horizontal Rule

* Use three or more -, \*, or \_ to create a horizontal line.
* ---

## 9.8 8. Links

* Inline link: [link text](URL)
* Reference link:
* [link text][reference]  
    
  [reference]: http://example.com

## 9.9 9. Images

* Inline image: ![alt text](image-url)
* Reference image:
* ![alt text][image-ref]  
    
  [image-ref]: http://example.com/image.png

## 9.10 10. Tables

Tables can be created using pipes (|) and hyphens (-).

| Header 1 | Header 2 |  
|----------|----------|  
| Cell 1 | Cell 2 |  
| Cell 3 | Cell 4 |

## 9.11 11. Footnotes

* Footnote syntax: Here is a footnote reference[^1].
* Define the footnote elsewhere: [^1]: This is the footnote content.

## 9.12 12. Definition Lists (Pandoc)

Pandoc extends Markdown by supporting definition lists.

Term 1  
: Definition 1  
  
Term 2  
: Definition 2

## 9.13 13. Math

### 9.13.1 13.1 Inline Math

* Use single dollar signs: $E = mc^2$

### 9.13.2 13.2 Block Math

* Use double dollar signs:
* $$   
  E = mc^2   
  $$

### 9.13.3 13.3 LaTeX Environment

\[  
E = mc^2  
\]

## 9.14 14. Custom Blocks (Pandoc)

Pandoc supports custom blocks for HTML and LaTeX conversion.

::: {.custom-class}  
Custom content here.  
:::

## 9.15 15. HTML Tags

You can directly use HTML tags for additional formatting.

<span style="color: red;">Red text</span>

## 9.16 16. Metadata Blocks (Pandoc)

You can add metadata at the beginning of the document.

---  
title: "Document Title"  
author: "Author Name"  
date: "2024-11-17"  
---

## 9.17 17. Line Breaks

* Add two spaces at the end of a line to create a line break.
* Alternatively, use <br> for a line break in HTML.

## 9.18 18. Escaping Characters

* Use a backslash (\) before a character to escape it.
* \\*Not italic\\*

## 9.19 19. Highlight (Pandoc)

To highlight text, use ==highlighted text==. This is only available in ==Pandoc.==

This is ==highlighted== text.

## 9.20 20. Superscript and Subscript

### 9.20.1 20.1 Superscript

* X^2^ becomes X².

### 9.20.2 20.2 Subscript

* H~2~O becomes H₂O.

# 10 Example Document: Union Earnings Analysis

### 10.0.1 What This Analysis Is All About

In this document, we’re diving into data on wages, demographics, and state info to see how union membership stacks up against nonunion wages. We want to understand what makes a difference in pay and job security across different states and communities. Spoiler alert: unions have a big impact, and we’re here to show you how!

### 10.0.2 Where the Data Came From

We used datasets from R for Data Science Online Learning Community ([2024](#ref-tidytuesday)) that cover wage and union membership trends across the U.S. from 1973 to 2022. These datasets come from government surveys and are packed with info on union membership, wages, and employment stats. We use three main datasets demographics.csv, wages.csv, and states.csv which let us track how unionization has affected wages over time. The data tells a compelling story about how unions have shaped workers’ lives for decades, and we’re here to shine a light on that.

### 10.0.3 Why This Matters

We want to understand how unions impact wages and employment patterns. Are union members earning more? Are they better off than nonunion workers? Which communities benefit the most from union membership? Our goal is to highlight the real power of unions in making workers’ lives better. We hope to give you a better idea of how collective bargaining makes a difference, both for individuals and for entire communities.

## 10.1 Data Preparation

### 10.1.1 Getting the Data Ready

We loaded three datasets—demographics, wages, and states using some simple code in R. These datasets have key info: demographics.csv tells us about employment by different demographic groups, wages.csv gives us union vs. nonunion wages, and states.csv tells us about state-level union activity and employment. Here’s how we got started:

setwd("~/Documents/School/RaceIncomeCalifornia")  
demographics <- readr::read\_csv('demographics.csv')  
wages <- readr::read\_csv('wages.csv')  
states <- readr::read\_csv('states.csv')

### 10.1.2 Quick Data Overview

We took a close look at what’s inside each dataset. The demographics.csv shows union membership and employment data broken down by things like gender, race, and age. The wages.csv lets us compare wages for union and nonunion workers, while states.csv gives a bigger picture at the state level. Understanding what’s in each dataset helps us figure out how to connect it all together to tell a story about unions and wages.

### 10.1.3 Filtering to Focus on the Present

I decided to focus on data from 2000 onwards. Why? Because we want to see what’s happening now, or at least in the recent past. Older data might not be as relevant to the present-day challenges and victories of workers, so we filtered it down to what’s most useful.

filtered\_wages <- wages[wages$year >= 2000, ]  
filtered\_states <- states[states$year >= 2000, ]  
filtered\_demographics <- demographics[demographics$year >= 2000, ]

### 10.1.4 Merging the Data

We merged the wages and states datasets to make it easier to look at everything together. This helps us connect wages (union and nonunion) with state-level data, like union membership rates. By combining this data, we can get a fuller picture of how unions are making a difference for workers in each state.

merged\_data <- merge(states, wages, by = "year", all.x = TRUE)

### 10.1.5 Making Data Manipulation Easier

I converted the filtered datasets into a faster, more efficient format using data.table. This makes it easier to manipulate big datasets without a lot of waiting around, which is especially helpful when plotting graphs.

if (!requireNamespace("data.table", quietly = TRUE)) {  
 install.packages("data.table")  
}  
library(data.table)  
  
setDT(filtered\_wages)  
setDT(filtered\_states)  
setDT(filtered\_demographics)  
  
setDT(filtered\_wages)  
setDT(filtered\_states)  
setDT(filtered\_demographics)

## 10.2 Analysis and Aggregations

### 10.2.1 Calculating Key Metrics

In this chapter, I calculated some key metrics to understand wage trends and union impact. I wanted to get a clear sense of how union and nonunion wages compare and how union membership has changed over time.

* **Mean Wages**: First, we calculated the average wages for union and nonunion workers over the years. This helped us see the difference in pay between the two groups.

wage\_trends <- filtered\_wages[,   
 list(mean\_union\_wage = mean(union\_wage,   
 na.rm = TRUE),  
 mean\_nonunion\_wage = mean(nonunion\_wage,   
 na.rm = TRUE)), by = list(year)]

This code calculates the average union and nonunion wages for each year from 2000 onwards, allowing us to visualize and compare wage trends between unionized and nonunionized workers. These calculations provide an overview of how wages for both categories have evolved over the years.

* **Union Wage Premium**: I also calculated the “union wage premium,” which is basically the extra amount union workers make compared to nonunion workers.

union\_wage\_premium <- filtered\_wages[,   
 list(raw\_premium = mean(union\_wage\_premium\_raw, na.rm = TRUE),  
 adjusted\_premium = mean(union\_wage\_premium\_adjusted,   
 na.rm = TRUE)), by = year]

The union wage premium calculation helps us quantify the economic benefits of union membership. The adjusted premium, in particular, sheds light on whether the observed wage differential is purely due to union membership or influenced by other worker characteristics.

* **Union Membership Trends**: I looked at how union membership has changed over time. Understanding how union density (the percentage of workers in a union) has shifted helps us connect the dots between membership levels and wage benefits.

membership\_trends <- filtered\_demographics[,   
 list(average\_p\_members = mean(p\_members, na.rm = TRUE)), by = year]

By calculating the average percentage of union members for each year, we can identify any notable increases or decreases in union membership. These trends help inform our understanding of how unions have gained or lost influence over the past few decades.

### 10.2.2 Rescaling Data for Better Visualization

In addition to the core aggregations described above, further calculations were performed to deepen our understanding of the dataset and prepare for visualizations.

* **Average Wage per Year**: I also calculated an overall average wage (for both union and nonunion workers) for each year to provide a broader view of wage growth over time.

average\_wage\_by\_year <- filtered\_wages[,  
 list(average\_wage = (mean(union\_wage, na.rm = TRUE) +   
 mean(nonunion\_wage, na.rm = TRUE)) / 2),   
 by = list(year)]

This calculation helps in understanding whether wages have increased consistently over the years and if union membership has had an impact on overall wage trends. By averaging the wages of union and nonunion workers, we provide a generalized view of wage progression in the broader labor market.

* **Members Rescaling**: I rescaled the number of union members so we could better visualize membership levels alongside other employment metrics. This helps us create clearer, more impactful graphs.

merged\_data$members\_scaled <- merged\_data$members \* 100

The rescaling of union membership data helps align the values with other metrics, making visualizations more intuitive and interpretable. It ensures that union membership figures can be easily compared alongside other variables such as employment or wages in different graphical plots.

With these metrics calculated, we’re ready to dive into visualizing the power of unions and how they impact workers’ lives.

## 10.3 Visualization of Results

### 10.3.1 Average Wage Trends Over Time

I used ggplot to create a graph that shows the average wages from 2000 to 2023. It’s easy to see that wages have generally gone up—but union wages are leading the way.

library(ggplot2)  
ggplot(average\_wage\_by\_year, aes(x = year, y = average\_wage)) +  
 geom\_line() +   
 labs(x = "Year", y = "Average Wage") +  
 theme\_minimal() +  
 scale\_color\_brewer(palette = "Set1")

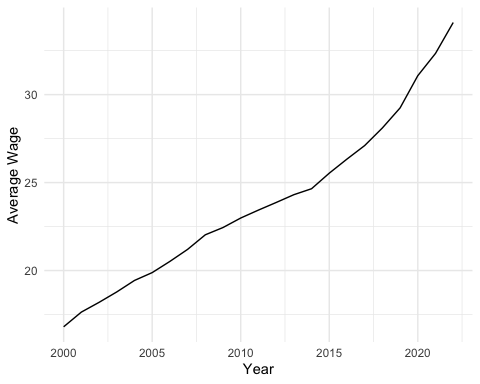


Figure 10.1: Average Wage

The 10.1 visualization displays the trend of average wages over the two-decade span starting from the year 2000 up to 2023. This plot shows a steady increase in wages, highlighting an upward trajectory that suggests a positive economic trend. The x-axis represents the years, while the y-axis denotes the average wage figures. The line graph is particularly useful for tracking changes over time, making it easier to visualize the rate of growth or any periods of stagnation.

### 10.3.2 Union Wages Comparison by Year

Next, we used a boxplot to compare union wages over different years. This gives us a look at how wages have varied for union workers.

ggplot(filtered\_wages, aes(x = factor(year), y = union\_wage)) +  
 geom\_boxplot() +  
 labs(x = "Year", y = "Union Wage") +  
 theme\_minimal()

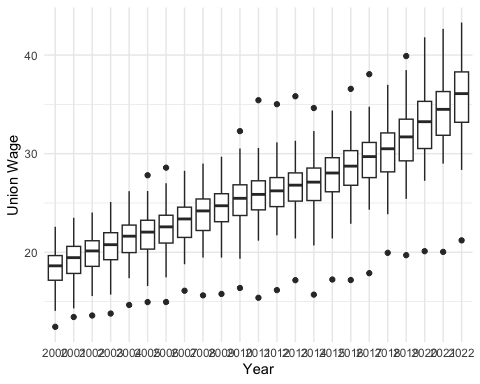


Figure 10.2: Union Wages by Year

This boxplot 10.2 illustrates union wages over different years, capturing the range of wages as well as median values. By comparing the median line in each box, we can see if union wages have experienced upward or downward trends. The presence of outliers in some years indicates that there were wage values that significantly deviated from the rest of the dataset, providing insights into wage inequality within union jobs. Additionally, the increasing height of the boxes over time may suggest growing wage variability among union workers.

### 10.3.3 Employment Trends by State, Including Union Membership

I wanted to show how employment and union membership vary across different states. For this, we used a combination of bar charts and line graphs. The idea here is to illustrate not just the number of employed people in each state, but also how many of them are union members. This is important because higher union membership often means better wages and working conditions for more people.

ggplot() +  
 geom\_bar(data = states, aes(x = state, y = employment),   
 stat = "identity", fill = "steelblue", alpha = 0.7) +  
 geom\_line(data = merged\_data,   
 aes(x = state, y = members\_scaled,   
 group = state), color = "red") +  
 labs(x = "State",   
 y = "Employment/union members", y.sec = "Members (scaled)") +  
 theme\_minimal() +  
 scale\_fill\_brewer(palette = "Pastel1") +  
 coord\_flip() +  
 scale\_y\_continuous(sec.axis = sec\_axis(~ . / 100, name = "Members"))

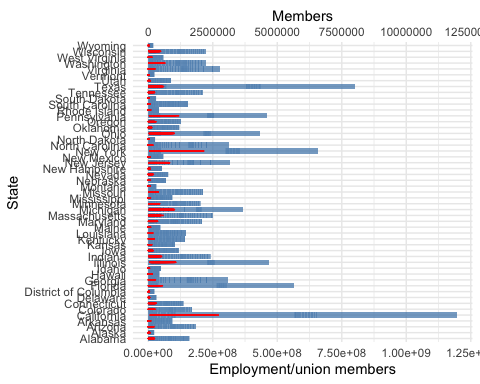


Figure 10.3: Employement Trends by State

In this graph 10.3, the blue bars show how many people are working in each state, while the red line shows how many of them are in unions. The states with the highest union membership—like California, New York, and Pennsylvania—are leading the way in worker rights and fair treatment. It’s pretty simple: more unions = better lives for workers.

### 10.3.4 Union vs Nonunion Wage Trends Over Time

We wanted to make it crystal clear how much better union wages are compared to nonunion wages. This graph shows just that—union vs. nonunion wages over time.

ggplot(wages, aes(x = year)) +  
 geom\_smooth(aes(y = union\_wage, color = "Union Wage"),   
 se = FALSE, method = "loess") +  
 geom\_smooth(aes(y = nonunion\_wage, color = "Nonunion Wage"),   
 se = FALSE, method = "loess") +  
 labs(x = "Year", y = "Wage",  
 color = "Type of Wage") +  
 theme\_minimal(base\_size = 14) +  
 theme(legend.position = "bottom") +  
 scale\_color\_manual(values = c("Union Wage" = "blue", "Nonunion Wage" = "green")) +  
 guides(color = guide\_legend(title = "Wage Type"))

## `geom\_smooth()` using formula = 'y ~ x'  
## `geom\_smooth()` using formula = 'y ~ x'

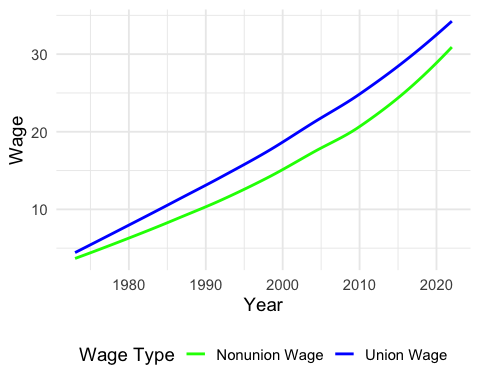


Figure 10.4: Union Vs. Union Wage Over Time

Union wages (in blue) are consistently higher than nonunion wages (in green). It’s proof that collective bargaining works. When workers join together, they get better pay—it’s as simple as that.

### 10.3.5 Union Wage Premium Over Time

I also took a look at the “union wage premium,” which is just a fancy way of saying how much extra money union workers make compared to nonunion workers. This graph shows that premium over time.

ggplot() +  
 geom\_point(data = wages, aes(x = year, y = union\_wage\_premium\_raw),   
 color = "orange") +  
 geom\_smooth(data = wages,   
 aes(x = year,   
 y = union\_wage\_premium\_adjusted), method = "loess", color = "purple") +  
 labs(x = "Year", y = "Wage Premium") +  
 theme\_minimal() +  
 scale\_color\_brewer(palette = "Dark2")

## `geom\_smooth()` using formula = 'y ~ x'

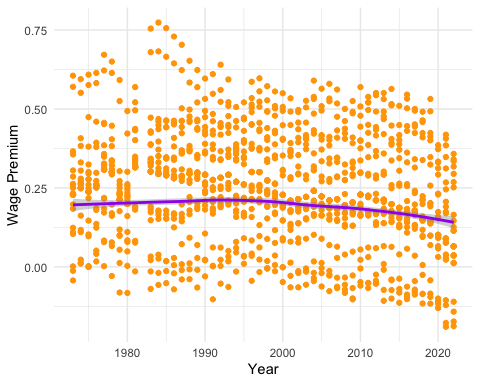


Figure 10.5: Union Wage Premium Over Time

The orange points show the raw premium, while the purple line is the adjusted premium. What this graph tells us is that unions are getting workers paid more, even when you take other factors into account. That’s real value for workers. But we see that over time this is trending down.

### 10.3.6 Union Membership Trends Over Time

We also looked at how union membership has changed over time with this area chart. It’s important because union membership has been on the decline, and we need to change that.

ggplot(demographics, aes(x = year, y = p\_members)) +  
 geom\_area(fill = "lightblue", alpha = 0.5) +  
 labs(x = "Year", y = "Percentage of Union Members") +  
 theme\_minimal() +  
 scale\_fill\_brewer(palette = "Blues")

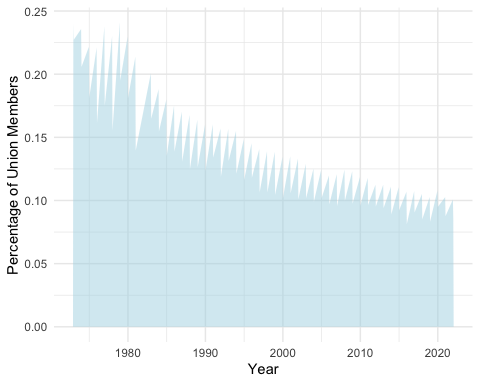


Figure 10.6: Uniont Membership Trends

The drop in union membership is clear. Fewer union members mean less bargaining power, and that’s why wages haven’t risen as fast as they should. We need to support union growth to bring back fair wages and worker protections.

### 10.3.7 Employment in Different Sectors Over Time

Lastly, we looked at employment trends across different sectors. This stacked bar graph shows how employment has changed from manufacturing to more service-focused jobs over time.

ggplot(states, aes(x = year, y = employment, fill = sector)) +  
 geom\_bar(stat = "identity", position = "stack") +  
 labs(x = "Year", y = "Employment") +  
 theme\_minimal() +  
 scale\_fill\_brewer(palette = "Spectral")

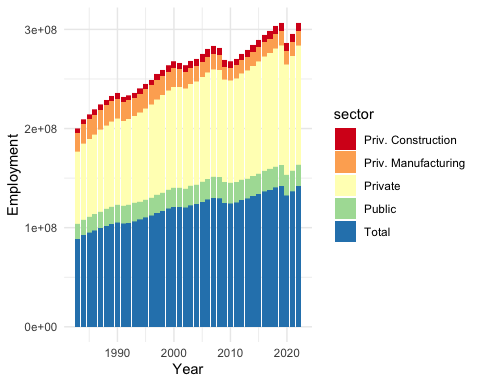


Figure 10.7: Setor Employement

The private sector—especially services—has grown a lot, while manufacturing has shrunk. We need unions in all these sectors to keep fighting for fair pay and worker rights, no matter what kind of work people are doing.

## 10.4 Discussion and Summary

### 10.4.1 Wage Growth and Union Influence

Une thing is crystal clear: union workers make more money. The graphs and numbers show that if you’re in a union, you’re getting paid better compared to folks who aren’t. We’re talking about both the raw numbers (the difference in pay) and the adjusted numbers (where we consider factors like age, job type, etc.). And guess what? Even when you adjust for those things, union workers still come out on top.

When we looked at the trends over time, union wages have consistently been higher. Sure, there’s been some narrowing of the gap lately, but unions are still the best bet for fair wages. If unions start losing power, everyone’s pay could start sliding back. We need to keep unions strong if we want to keep fighting for better wages for all workers.

### 10.4.2 Decline in Union Membership

One of the more worrying things we saw is how union membership has been dropping for the past couple of decades. There are a bunch of reasons for this. The economy has shifted—manufacturing, where unions were strong, has shrunk, and we’ve moved towards more service jobs, which are harder to organize. Plus, with globalization, a lot of jobs went overseas or became automated, and that took away a lot of union power.

And let’s not forget the gig economy. More and more people are working gig jobs or part-time, which makes it tough for them to organize and join unions. Fewer unions mean less power for workers to negotiate fair pay and benefits, and that’s not good for anyone. If we want to turn this around, we need to make it easier for workers in all kinds of jobs to join unions and fight for their rights.

### 10.4.3 Shifts in Employment by Sector

Another big trend we saw is that employment has shifted from industries like manufacturing and construction into the service sector. The service industry has been booming, but it hasn’t been easy for unions to gain a foothold there. That means a lot of workers in service jobs are missing out on the protections and pay boosts that unions can provide.

The public sector is one of the few places where unions are still doing okay, but even there, unions are under pressure from budget cuts and changes to policies that aim to cut spending. If we want to improve job quality for everyone, unions need to find ways to grow in these new industries. The service sector needs unions more than ever to help protect workers’ rights and improve their pay.

### 10.4.4 Implications for the Future

So, what does all this mean for the future? Well, we’ve got some challenges, but also opportunities. Unions are still crucial for securing better wages, but the decline in membership is a problem we need to solve. The answer might be new organizing strategies—unions need to adapt to reach workers in tech, service, and gig industries. If unions can innovate, they can still be a powerful force for worker rights.

If union membership keeps shrinking, we’re going to see more income inequality. That means the gap between the rich and the poor will get wider, and that’s bad news for everyone. Unions help level the playing field, so we need to support them, whether through better labor laws or by organizing in new industries.

The shift to more service-based jobs could be an opportunity if unions can step in and start organizing those workers. The future of the labor market is going to depend a lot on how well unions, workers, and policymakers can adapt to these changes. We need to work together to make sure everyone gets fair wages and decent working conditions, no matter where they work.

Unions have always been about making things fair for workers, and that mission hasn’t changed. We need to keep building union power to make sure everyone gets a fair shake in this rapidly changing economy.

R for Data Science Online Learning Community. 2024. “TidyTuesday: A Weekly Data Project Aimed at Improving Data Wrangling and Visualization Skills.” <https://github.com/rfordatascience/tidytuesday>.

Xie, Yihui. 2024. *Bookdown: Authoring Books and Technical Documents with r Markdown*. 2nd ed. Chapman; Hall/CRC. <https://bookdown.org/yihui/bookdown/>.