



Caption, Citation, and booktabs (Table) in $\text{\LaTeX}$

Prof. Pranay Kumar Saha

March 26, 2025





What are Floating Environments?

- Objects like figures and tables are often treated as "floats."
- LaTeX automatically positions floats on the page to optimize layout.
- This prevents large figures/tables from creating awkward page breaks.
- Common floating environments:
 - `figure`
 - `table`



The figure Environment

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=0.8\textwidth]{example-image}
  \caption{A descriptive caption.}
  \label{fig:myfigure}
\end{figure}
```



The figure Environment

- `\begin{figure}` and `\end{figure}`: Enclose the figure content.
- `[htbp]`: Placement specifiers (optional, but recommended).
 - h: Place the figure "here" (approximately).
 - t: Place the figure at the "top" of a page.
 - b: Place the figure at the "bottom" of a page.
 - p: Place the figure on a separate "page" of floats.
 - !: Override internal parameters (use with caution). Example: `[!h]`.
- LaTeX tries these in order; you can use a subset (e.g., `'[hb]'`).
- `\centering`: Centers the figure horizontally.
- `\includegraphics`: Includes the image file (requires the `graphicx` package). Common options: `width`, `height`, `scale`.
- `\caption`: Provides a caption for the figure.
- `\label`: Assigns a label for cross-referencing.



What are Captions?

- Captions provide brief descriptions for figures, tables, and other floating objects.
- They typically appear below the object (for figures) or above the object (for tables).
- They help readers quickly understand the content of the object without having to read lengthy explanations in the main text.
- They are essential for accessibility and clarity, especially in academic writing.



Creating Captions

- Use the `\caption{}` command within the environment of the floating object (e.g., `figure`, `table`).
- The text inside the curly braces becomes the caption text.



Examples

Example (Figure):

```
\begin{figure}
  \centering
  \includegraphics[width=0.5\textwidth]{example-image}
  \caption{This is a sample image.}
\end{figure}
```

Example (Table):

```
\begin{table}
  \centering
  \caption{This is a sample table.}
  \begin{tabular}{c c}
    Header 1 & Header 2 \\
    Data 1    & Data 2
  \end{tabular}
\end{table}
```



Customizing Captions

- The `caption` package provides many options:
 - `\captionsetup`: Control font size, style, label format, and more.
 - Example: `\captionsetup{font=small,labelfont=bf}` (sets caption font to small and label to bold).
- Even without the `caption` package, you can manually format captions:
 - Use font commands (e.g. `\textbf{}`, `\textit{}`, `\small`) inside the `\caption{}` command.
 - Example: `\caption{\textbf{Figure 1:} \textit{A description.}}`.



What are Labels?

- Labels are unique identifiers assigned to objects within your document (figures, tables, equations, sections, etc.).
- They allow you to refer to these objects later using cross-references.
- Labels are **invisible** in the compiled document.



Creating Labels

- Use the `\label{}` command.
- Place the `\label{}` command *immediately after* the `\caption{}` command (for figures and tables), or *after* the command that defines the object (for sections, equations, etc.).
- Choose meaningful label names. A common convention is to use prefixes:
 - `fig:` for figures
 - `tab:` for tables
 - `eq:` for equations
 - `sec:` for sections



Cross-referencing with `\ref`

- Use the `\ref{}` command to create a cross-reference to a labeled object.
- The argument of `\ref{}` is the label name.
- LaTeX automatically replaces `\ref{label_name}` with the corresponding object's number (e.g., "Figure 1", "Table 2", "Section 3").

Example:

As shown in Figure `\ref{fig:sample_image}`, the results...



`\ref` and `\pageref`

- `\ref` produces the number of the section, figure, table, etc.
- `\pageref` produces the *page number* where the label is located.

Example:

The details are provided in Section `\ref{sec:methods}`
on page `\pageref{sec:methods}`.



Using `hyperref` for Clickable References

- The `hyperref` package (already included at the beginning) makes references (and citations) clickable links in the PDF.
- Clicking on a reference will take you to the corresponding labeled object.
- Customize link colors with options to `hyperref`:

```
\usepackage[colorlinks=true,linkcolor=blue,citecolor=green]{hyperref}
```



Why Use Citations?

- Give credit to the original sources of information.
- Support your claims and arguments.
- Allow readers to find and verify your sources.
- Avoid plagiarism.



Basic Citation with `biblatex` (Recommended)

- `biblatex` is a modern and powerful package for managing bibliographies. It offers more flexibility and features than the older `natbib` package.

- **Steps:**

1. Create a `.bib` file (e.g., `references.bib`) containing your bibliographic entries in BibTeX format.

2. Include the `biblatex` package in your preamble:

```
\usepackage[backend=biber,style=authoryear]{biblatex}  
\addbibresource{references.bib}
```

(This uses the 'authoryear' style; many other styles are available). 'biber' is a more modern backend than 'bibtex'.

3. Use the `\cite{}` command in your document to cite entries.
4. Place `\printbibliography` where you want the bibliography to appear (usually at the end).



Example .bib File (references.bib)

```
@article{einstein1905,  
  author = {Einstein, Albert},  
  title = {On the Electrodynamics of Moving Bodies},  
  journal = {Annalen der Physik},  
  volume = {322},  
  number = {10},  
  pages = {891--921},  
  year = {1905}  
}
```

```
@book{knuth1986,  
  author = {Knuth, Donald E.},  
  title = {The TeXbook},  
  publisher = {Addison-Wesley},  
  year = {1986}  
}
```



Citing in Your Document

- Use `\cite{key}` to cite a bibliographic entry, where `key` is the unique identifier from your `.bib` file.
- `biblatex` provides many citation commands:
 - `\cite{key}`: Standard citation (e.g., (Einstein, 1905)).
 - `\parencite{key}`: Citation in parentheses.
 - `\textcite{key}`: Integrates the citation into the text (e.g., Einstein (1905) showed that...).
 - `\citeyear{key}`: Just the year (e.g., 1905).
 - `\citeauthor{key}`: Just the author(s).
 - Many more (see the `biblatex` documentation).

Example:

According to `\textcite{einstein1905}`, the speed of light...
This concept was later expanded upon by `\cite{knuth1986}`.



Compiling with biblatex

1. Run LaTeX (e.g., pdf_latex) on your .tex file.
2. Run Biber (or BibTeX if you are using it as the backend) on your .tex file **without the extension**. Example: `biber mydocument`
3. Run LaTeX again (once or twice) to resolve all references and citations.

Important: If you are using an online editor like Overleaf, it usually handles this process automatically.



Generating the Bibliography

`\printbibliography`

Place this command where you want the bibliography to appear. Usually this is at the end, before `\end{document}`.



Why Different Citation Styles?

- Different academic disciplines and publications have varying conventions for citing sources.
- Styles dictate the format of in-text citations and the bibliography/references list.
- Common elements: author(s), title, publication date, journal/publisher, page numbers, etc.
- `biblatex` offers immense flexibility in choosing and customizing citation styles.
- The `style=` option in `\usepackage[...]{biblatex}` is key.



Author-Year Style (e.g., APA, Chicago)

- Most common in the social sciences and humanities.
- In-text citations: (Author, Year).
- Bibliography: Alphabetical by author's last name.

biblatex Example:

```
\usepackage[backend=biber,style=authoryear]{biblatex}  
\addbibresource{references.bib}
```

`style=authoryear` is a good starting point. Other author-year styles include:

- `authoryear-comp` (compressed author-year citations)
- `authoryear-icomp` (ibid. for repeated citations)
- `chicago-authordate` (Chicago Manual of Style, author-date)
- Many others...



Author-Year Citation Commands

- `\cite{einstein1905}`: (Einstein, 1905)
- `\parencite{einstein1905}`: (Einstein, 1905) - same as `\cite` in many author-year styles.
- `\textcite{einstein1905}`: Einstein (1905)
- `\citeyear{einstein1905}`: 1905
- `\citeauthor{einstein1905}`: Einstein
- `\citetitle{einstein1905}` : On the Electrodynamics of Moving Bodies.



Numeric Style (e.g., IEEE, Vancouver)

- Common in engineering, computer science, and some natural sciences.
- In-text citations: Numbers in brackets (e.g., [1], [2, 5], [7-9]).
- Bibliography: Numbered list, usually in order of appearance.

biblatex Example:

```
\usepackage[backend=biber,style=numeric]{biblatex}  
\addbibresource{references.bib}
```

Other numeric styles:

- `numeric-comp` (compressed numeric citations: [1-3] instead of [1, 2, 3])



Numeric Style Citation Commands

- With `style=numeric`, `\cite{einstein1905}` produces a bracketed number (e.g., [1]).
- The number corresponds to the entry's position in the bibliography.
- Multiple citations: `\cite{einstein1905, knuth1986}` might produce [1, 2].
- With `numeric-comp`, `\cite{key1,key2,key3}` where keys are consecutive would produce something like [1-3].

Example:

- The theory of relativity [1] revolutionized our understanding of space and time. Later developments in computer science [2] built upon these fundamental concepts.



Alphabetic Style

- Less common, but used in some fields.
- In-text citations: Abbreviated author name and year (e.g., [Ein05], [Knu86]).
- Bibliography: Alphabetical by author's last name, then chronologically.

biblatex Example:

```
\usepackage[backend=biber, style=alphabetic]{biblatex}  
\addbibresource{references.bib}
```



Alphabetic Style Citation Commands

- `\cite{einstein1905}` would produce something like [Ein05].
- The exact abbreviation format can be customized.

Example:

- The foundational work [Ein05] laid the groundwork for future research. The principles of typesetting were meticulously described in [Knu86].



Footnote/Endnote Styles (e.g., Chicago Notes and Bibliography)

- Citations appear as footnotes at the bottom of the page or as endnotes at the end of the document (or chapter).
- Often used in humanities disciplines.
- `biblatex` handles this beautifully with styles like `verbose` and `chicago-notes`.

`biblatex` Example (verbose style):

```
\usepackage[backend=biber,style=verbose]{biblatex}  
\addbibresource{references.bib}
```

- `verbose`: A basic footnote style.
- `verbose-ibid`: Uses "ibid." for repeated citations.
- `chicago-notes`: Chicago Manual of Style (notes and bibliography).



Footnote Style Example

- Use `\footnote{\cite{einstein1905}}` or the shorthand `\footcite{einstein1905}`.
- LaTeX automatically numbers the footnotes and places the full citation at the bottom of the page.



Citations using `\bibitem` (No `.bib` file)

- An older method for creating bibliographies *directly within* your LaTeX document.
- Does *not* use an external `.bib` file.
- Less flexible and less recommended than `biblatex`, but useful for small bibliographies or when you need complete control over the formatting.
- Uses the `thebibliography` environment and `\bibitem` commands.



Structure:

`\begin{thebibliography}{99}` % '99' is a placeholder for the widest label

```
\bibitem[Einstein(1905)]{einstein1905}
```

Einstein, A. (1905).

On the electrodynamics of moving bodies.

`\textit{Annalen der Physik}`, `\textit{322}`(10), 891--921.

```
\bibitem[Knuth(1986)]{knuth1986}
```

Knuth, D. E. (1986).

`\textit{The TeXbook}`.

Addison-Wesley.



Summary

- **Captions:** Describe figures, tables, etc., using `\caption{}`.
- **Labels:** Create unique identifiers using `\label{}`.
- **References:** Cross-reference labeled objects using `\ref{}` and `\pageref{}`.
- **Citations:** Cite sources using `\cite{}` (and related commands) and a bibliography managed with `biblatex`. Use a `.bib` file.



Document Structure: Sections and Subsections

- Organize your document into logical parts using sections and subsections.
- Improves readability and navigation.
- LaTeX automatically numbers sections and subsections.



Document Structure: Sections and Subsections

Commands:

- `\section{Section Title}`
- `\subsection{Subsection Title}`
- `\subsubsection{Subsubsection Title}` (use sparingly)
- `\paragraph{Paragraph Title}`
- `\subparagraph{Subparagraph Title}`
- `\part{Part Title}`: Higher level
- `\chapter{chapter title}`: used only in book and report



Example of Sections and Subsections

```
\documentclass{article} % Or book, report, etc.
```

```
\begin{document}
```

```
\section{Introduction}
```

This is the introduction.

```
\subsection{Background}
```

Some background information...

```
\section{Methods}
```

```
\subsection{Experimental Setup}
```

Details of the setup...



Example of Sections and Subsections

```
\subsubsection{Materials} % Subsubsection (use sparingly)
```

List of materials...

```
\paragraph{Procedure} % Paragraph-level heading
```

Step-by-step procedure...

```
\section{Results}
```

...

```
\end{document}
```



Creating Tables with booktabs

- The `booktabs` package provides commands for creating professional-looking tables.
- Improves table readability and aesthetics compared to standard LaTeX tables.
- Key commands:
 - `\toprule`: Creates a thick horizontal rule at the top of the table.
 - `\midrule`: Creates a medium-thickness horizontal rule, typically used to separate the header from the table body.
 - `\bottomrule`: Creates a thick horizontal rule at the bottom of the table.
 - *Avoid vertical rules!* (a core principle of good table design).



Basic booktabs Table Example

```
\begin{table}
  \centering
  \caption{A simple table using booktabs.}
  \label{tab:simple}
  \begin{tabular}{l c r} % Column alignment: l(left), c(enter), r(right)
    \toprule
    Header 1 & Header 2 & Header 3 \\
    \midrule
    Data 1    & Data 2    & Data 3    \\
    Data 4    & Data 5    & Data 6    \\
    \bottomrule
  \end{tabular}
\end{table}
```



Basic booktabs table example

Table: A simple table using booktabs.

Header 1	Header 2	Header 3
Data 1	Data 2	Data 3
Data 4	Data 5	Data 6



Advanced Table Formatting

- The `array` package (often used with `booktabs`) provides more control over column formatting.
- Useful column specifiers:
 - `p{width}`: Column with a fixed width (e.g., `p{2cm}`). Text will wrap within the column.
 - `m{width}`: Similar to `p`, but vertically centers the content.
 - `b{width}`: Similar to `p`, but aligns the content to the bottom.
 - `>\command`: Applies a command to *every* cell in the column (e.g., `>\centering\arraybackslash p{2cm}` for centered, fixed-width columns. `\arraybackslash` is needed after commands like `\centering` in `tabular`.)



Example: Fixed-Width and Centered Columns

```
\begin{table}
  \centering
  \caption{Table with fixed-width and centered columns.}
  \label{tab:advanced}
  \begin{tabular}{>{\centering\arraybackslash}p{3cm}
                 >{\raggedright\arraybackslash}p{4cm} r}
    \toprule
    Centered Column & Left-Aligned Column & Right-Aligned \\
    \midrule
    This text is centered. & This text wraps and is left-aligned. & 123 \\
    Short & Another long line of text that wraps. & 456 \\
    \bottomrule
  \end{tabular}
\end{table}
```



Fixed-Width and Centered Columns

Explanation:

- `>\centering\arraybackslashp{3cm}`: Creates a 3cm wide column with centered text.
- `>\raggedright\arraybackslashp{4cm}`: Creates a 4cm wide column with left-aligned text (ragged right).
- `'r'`: A standard right-aligned column.



Example: Fixed-Width and Centered Columns

Table: Table with fixed-width and centered columns.

Centered Column	Left-Aligned Column	Right-Aligned
This text is centered.	This text wraps and is left-aligned.	123
Short	Another long line of text that wraps.	456



Best Practices for Tables

- Keep tables concise and focused.
- Use clear and descriptive column headers.
- Align numbers to the decimal point for better readability.
- Use horizontal rules sparingly (as provided by `booktabs`).
- Avoid vertical rules.
- Provide a descriptive caption above the table.
- Label your tables (e.g., `\label{tab:mytable}`) for easy cross-referencing.



What is `multirow`?

- The `multirow` package allows you to create table cells that span multiple rows.
- Useful for grouping related data or creating more complex table layouts.



Basic Usage of `\multirow`

`\multirow{number of rows}{width}{content}`

- `\multirow{number of rows}{width}{content}`:
 - `number of rows`: The number of rows the cell should span. Can be positive (spanning downwards) or negative (spanning upwards).
 - `width`: The width of the cell. Common options:
 - `*`: Use the natural width of the content. This is often the best choice.
 - A fixed length (e.g., `2cm`, `3in`).
 - `\widthof{text}` (from the `calc` package): Sets the width to the width of the given text.
 - `content`: The text or other content to be placed in the cell.
- Important: `\multirow` must be the **first** command in a table cell.



Example of multirow

```
\begin{table}
  \centering
  \caption{Table with multirow cells.}
  \label{tab:multirow}
  \begin{tabular}{l l l}
    \toprule
    Category & Subcategory & Value \\
    \midrule
    \multirow{2}{*}{Category 1} & Subcat A & 10 \\
    & Subcat B & 20 \\
    \midrule
    Category 2 & Subcat C & 30 \\
    \bottomrule
  \end{tabular}
\end{table}
```



Example of `multirow`

Table: Table with multirow cells.

Category	Subcategory	Value
Category 1	Subcat A	10
	Subcat B	20
Category 2	Subcat C	30



Multicolumn Cells in Tables

- The `\multicolumn` command allows you to create table cells that span multiple columns.
- Useful for:
 - Creating headers that span multiple columns.
 - Grouping related data under a common heading.
 - Creating complex table layouts.



`\multicolumn` Syntax

`\multicolumn{number of columns}{alignment}{content}`

- `\multicolumn{number of columns}{alignment}{content}`:
 - number of columns: The number of columns the cell should span.
 - alignment: The alignment of the cell's content. This uses the same column specifiers as the `tabular` environment:
 - l: Left-aligned.
 - c: Centered.
 - r: Right-aligned.
 - p{width}: Fixed-width, top-aligned.
 - m{width}: Fixed-width, vertically centered.
 - b{width}: Fixed-width, bottom-aligned.
 - You can also use | for vertical rules and > to apply commands to the cell (see previous examples with `array`).
 - content: The text or other content to be placed in the cell.
- Important: `\multicolumn` *overrides* the column alignment specified in the `tabular` environment's



Basic \multicolumn Example

```
\begin{table}
  \centering
  \caption{Table with a multicolumn header.}
  \label{tab:multicolumn}
  \begin{tabular}{l c c c}
    \toprule
    \multicolumn{4}{c}{Overall Header} \\
    \midrule
    Header 1 & Header 2 & Header 3 & Header 4 \\
    \midrule
    Data 1 & Data 2 & Data 3 & Data 4 \\
    Data 5 & Data 6 & Data 7 & Data 8 \\
    \bottomrule
  \end{tabular}
\end{table}
```



Basic \multicolumn Example

Table: Table with a multicolumn header.

Overall Header			
Header 1	Header 2	Header 3	Header 4
Data 1	Data 2	Data 3	Data 4
Data 5	Data 6	Data 7	Data 8



Combining `\multicolumn` and Vertical Rules

```
\begin{table}
  \centering
  \caption{Multicolumn with vertical rules.}
  \label{tab:multicolumn_rules}
  \begin{tabular}{|l|c|c|c|} % Vertical rules in the tabular definition
    \hline
    \multicolumn{4}{|c|}{Overall Header} \\ % Vertical rule in multicolumn
    \hline
    Header 1 & Header 2 & Header 3 & Header 4 \\
    \hline
    Data 1    & Data 2    & Data 3    & Data 4    \\
    \hline % Corrected: Added \\ to end the last row
  \end{tabular}
\end{table}
```



Combining `\multicolumn` and Vertical Rules

Note: The vertical rules in the `\multicolumn` alignment specifier (`|`) must match the vertical rules in the `tabular` definition. If they don't, you'll get errors or unexpected results. The `\hline` commands are equivalent to `\toprule`, `\midrule` and `\bottomrule` when not using 'booktabs'.



Combining `\multicolumn` and Vertical Rules Example

Table: Multicolumn with vertical rules.

Overall Header			
Header 1	Header 2	Header 3	Header 4
Data 1	Data 2	Data 3	Data 4



Multirow and Multicolumn Together

You can combine `\multirow` and `\multicolumn` within the same table.

```
\begin{tabular}{c c c c}
\toprule
\multicolumn{2}{c}{Group 1} & \multicolumn{2}{c}{Group 2} \\
\cmidrule(lr){1-2} \cmidrule(lr){3-4}
Header A & Header B & Header C & Header D \\
\midrule
\multirow{2}{*}{Data 1} & Data 2 & Data 3 & Data 4 \\
& Data 5 & Data 6 & Data 7 \\
\bottomrule
\end{tabular}
```



Multrow and Multicolumn Together Example

Group 1		Group 2	
Header A	Header B	Header C	Header D
Data 1	Data 2	Data 3	Data 4
	Data 5	Data 6	Data 7



Further Resources

- The biblatex documentation (extensive!): <http://ctan.org/pkg/biblatex>
- Overleaf documentation: <https://www.overleaf.com/learn>
- LaTeX Wikibook: <https://en.wikibooks.org/wiki/LaTeX>



Caption, Citation, and booktabs (Table) in $\text{\LaTeX}$

Thank You for Listening!