Introduction to LaTeX for Papers

Caitlin Steiner
Library Data Services: StatLab
University of Virginia

October 22, 2014

Contents

1 Introduction

1.1 What is LaTeX

LaTeX is not a word processor but is a formatter that is used mainly for typesetting proffessional research papers, theses, etc; it is especially useful for integrating mathematical formulas, theorems, and statistical ouptut. To prepare a LaTeX document one needs to create a text file (".txt") containing commands for LaTeX using some editor and then use a LaTeX program (such as TeXWorks) to convert it into a printable document.

1.2 Advantages and Disadvantages of Uisng LaTeX

The advanatages of LaTeX over a word processor are:

- It's fine control over document structure and formatting makes it a powerful - and formidable - tool.
- You don't need to manually adjust fonts, text sizes, line heights, etc.
- Mathematical formulas are easy to typeset
- The layout, fonts, tables, etc. are consistent throughout the document
- Adding footnotes, citations and references is easy
- It's free!

The disadvanatages of LaTeX over a word processor are:

- Time consumming to learn.
- You don't see the final version of the document while editing it
- It can sometimes be difficult to obtain a certain design or layout for a document
  - Building your own macros or using macros developed by others can change the default layout

1.3 Distribution of LaTeX

- For Windows, you could use:
  - MiKTeX or TeXLive
- For Mac, you could use:
  - MacTeX or XeTeX

2 The LaTeX Document Setup

LaTeX files begin by stating which kind of document will be created with the command:

`\documentclass[options]{document_type}`

The default document type is `article` with a default font size of 10. Other choices of `document_type` are `report`, `book`, `letter`.

There are several `options` choices one can choose from to set the overall format of the document. Multiple can be implemented as long as they are seperated by a comma. These are just a few of the most common

- font size
  - 10pt 11pt 12 pt etc...
- paper size
  - `letterpaper`
  - `legalpaper`
  - `a4paper`
- page formats
  - `onecolumn` (default)
  - `twocolumn`
  - `notitlepage`
  - `titlepage`

A LaTeX document consists of two parts - the Preamble and the Text Body.

2.1 The Preamble

The preamble sets up the document by including any packages that one might use, allows one to define new commands, and sets the properties of the document (e.g., author, title, margins, etc.). Anything written in the preamble won't show up in the printed document, with a few exceptions (e.g., title commands). The preamble is everything between

`\documentclass[options]{document_type}`

and

`\begin{document}`

2.1.1 Packages

Although the class of the document determines its overall general properties, more specific packages can be invoked in the preamble to change the way commands behave or to add extra features. One can load packages using the command:

`\usepackage[options]{package}`

Some of the packages may have options associated with it. What options are available depend on the package itself. Note that just because a package is loaded does not mean one has to use its associated commands within the document.

2.1.2 Setting Document Margins

There are multiple different ways one can set the margins of the document. The simpliest line of code to change the margins uses the `geometry` package.

One can set all the margins at once by

`\usepackage[margin=1in]{geometry}`

or one can set each margin to a specific length using

`\usepackage[top=2in, bottom=1.5in, left=1in, right=1in]{geometry}`

2.1.3 Basic Preamble for Paper

```
\documentclass[12pt]{article}

% Page Formatting
\usepackage[margin=1in]{geometry} % sets overall page margins
\usepackage{setspace} % allows toggling of double/single-spacing
\singlespace % sets document spacing to single
% \doublespace  % sets document to double
\usepackage{indentfirst, parskip} % paragraph customization
\setlength{\parindent}{0.5in} % sets indent length

%Other Useful Packages
\usepackage{caption}
%customizes captions in floating environments like figures and tables
\usepackage{graphicx, float} % graphics package
\usepackage{amsfonts, amsmath, amssymb, amsthm,bm} % math packages
\usepackage{url} % allows including urls
\usepackage{verbatim} % defines environment for un-evaluated code
```

2.2 The Text Body

The text body is where all the text and code that will be compiled and outputted by LaTeX is located. The body is anything between

`\begin{document}`

and

`\end{document}`

2.2.1 Typing in LaTeX

A few helpful hints to help reduce some inital frustration with working with LaTeX:

- `\newline`, `\linebreak`, and `\\` all move whatever follows them to a new line. However, one can not use these commands by themselves in a line without causing an error
- LaTeX does not automatically indent or seperate paragraphs thus
  - If the previous paragraph is terminated with `\\`, then a new not-indented paragraph will be created.
  - If one or more empty lines are left between text, then a new indented paragraph will be created.
  - If you want an indentation but it does not do it automatically use the command, `\indent`, otherwise use `\noindent`.

```
Here is a paragraph and as you can see it is automatically indented but\\
if in our LaTeX code we continue on a new line it does not automatically
force the pdf document to start a new line.\\
But if one ends a paragraph with the double backslah then a not indented
new paragraph is started.

Or one can simiply have an empty line in their LaTeX code to force
a new paragraph.
```

results in:

Here is a paragraph and as you can see it is automatically indented
but if in our LaTeX code we continue on a new line it does not automatically force the pdf document to start a new line.
But if one ends a paragraph with the double backslah then a not indented new paragraph is started.

Or one can simiply have an empty line in their LaTeX code to force a new paragraph.

- `\newpage` or `\pagebreak` put everything below it on a new page.
- `\vspace{}` and `\hspace{}` will manually insert white space vertically and horizontally the inputted distance in the `{}`, respectively.
- To include comments within the document for oneself or for others who may view the code without having it outputted in the document pdf use the percent character `%`. This indicates to LaTeX that it and the rest of the line should be ignored.
  - If multiple lines need to be ignored then each line must be prefixed with `%`. The percent character is also helpful in deactivating commands.
- `‘` (key above the tab) is the open single-quote symbol and `’` is the closed single-quote symbol.
- `‘‘` is the open double-quote symbol and `"` is the closed double-quote symbol.
- The characters `# % ^ & _ { } ~ \` are special characters reserved for different purposes. To use these symbols in your document, apply a prefix backslach, e.g., `\%` to produce %.
- If you don't know the code for a particular character or symbol you can use Detexify, which allows you to draw what you want to type and will return different options
- `verbatim` is an enviroment that allows you to print commands that would otherwise be interpreted. It has both the regular environment segment `\begin{verbatim}` and `\end{verbatim}` as well as a short hand for in line code, `\verb!text!`.

```
To fit a linear model in R, we used \verb!lm(y~x, data=dat)!.
```

results in:
To fit a linear model in R, we used `lm(y~x, data=dat)`.

2.2.2 Title

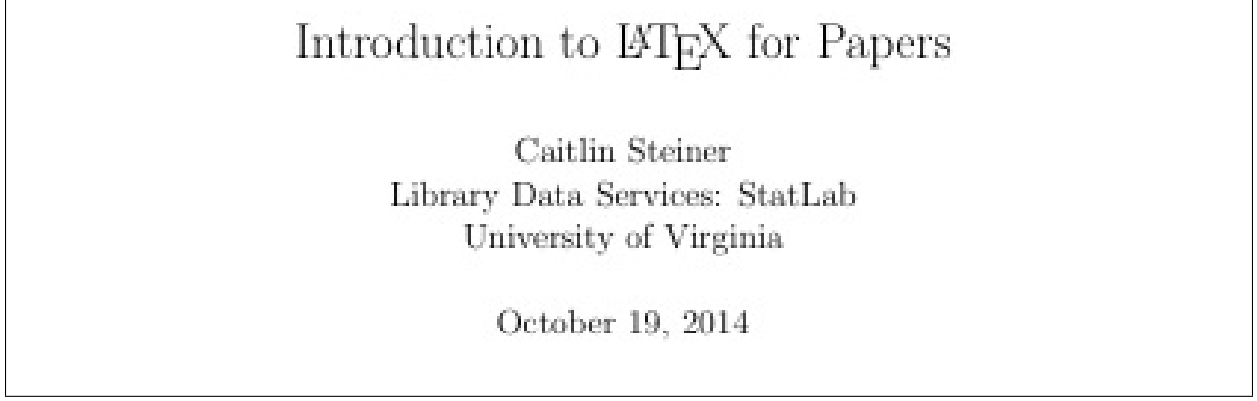
Often a title page is required when writing a paper. To add a title page one can either declare in the preamble or the text body the title, author, and/or date using:

```
\title{title_text}
\author{Author1\\Institute\\Address \and
          Author2\\Institute\\Address \and ...}
\date{\today}
```

And then can create the title page information in the Text Body by using:

```
\maketitle      %Displays the Title
\cleardoublepage      %makes body text start on next page
```

```
\title{Introduction to Latex for Papers}
\author{Caitlin Steiner\\ Research Data Services: StatLab\\
University of Virginia}
\date{\today}
```

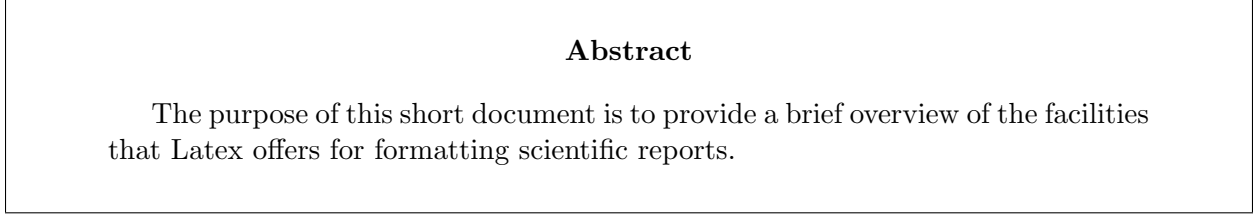


2.2.3 Abstract

One can add an abstract using the `abstract` environment that will automatically add a centered bolded title "Abstract" above the text desired for the abstract.

```
\begin{abstract}
 The purpose of this short document is to provide a brief overview of
 the facilities that Latex offers for formatting scientific reports.
\end{abstract}
```

results in



2.2.4 Headers

One can add a header with a label of their choosing using `\section`, `\subsection`, `\subsubsection`. These generate sequentially nested, bolded header with numbered sections. To remove the automatic numbering use the *-form.

```
\section{Introduction}
\subsection{Background}
\subsubsection{More Specific}
\section*{Methods}
```

results in:

1 Introduction

1.1 Background

1.1.1 More Specific

Methods

2.2.5 Footnotes

One can add a footnote to a paper using `\footnote{}` within the text. This places any text within the curly braces at the bottom of the page underneath a small horizontal line with a corresponding automatically generated numbered footnote notation in the document. Do not leave a space between the command and the word where you wish the footnote marker to appear otherwise the footnote will not look as intended.[1]

2.2.6 Bibliography

One of the easiliest ways to make a Bibiliography is to use the reference management software BibTeX. BibTeX uses a style-independent text-based file format (.bib) for lists of bibliography items, making it easy to cite sources in a consistent manner.

Each entry in BibTeX consists of the type (word after @), a unique citation-key and a number of tags which define various characteristics of a specific bibliography entry. There are several BibTeX editors online, one of which is BibTeX Editor. Once a .bib file is created, it does need to be compiled before compiling the .tex file to get the final document. This needs to occur everytime the .bib file is updated.

[1] Footnote 1

```
@BOOK {LatexBook,
    author    = "Kopka, Helmut and Patrick W. Daly",
    title     = "A Guide to LATEX: Document Preparation for Beginners and Advanced User
    publisher = "Addison-Wesley",
    year      = "1999",
    address   = "Harlow, England"
}
```

Then to add the references to the .tex document one must use the code

```
\bibliographystyle{plainnat}
\bibliography{bib_file}
```

as well as load the package and punctuation style of the references by

```
%% Natbib is a popular style for formatting references.
\usepackage{natbib}
%% bibpunct sets the punctuation used for formatting citations.
\bibpunct{(}{)}{;}{a}{,}{,}
```

To cite a reference in the document use the `\cite` command.

```
For this presentation I used \cite{LatexBook} as a reference book.
```

For this presentation I used Kopka and Daly (1999) as a reference book.

3 Adding Objects in Text Body

3.1 Math Environment

As stated earlier, LaTeX makes inputting equations into a document very simple. There are several different mathematical environments, but these are the most common:

- `$formula$` inputs the `formula` within the text

```
The following is the equation for a line $y=ax+b$
```

results in:
The following is the equation for a line $y = ax + b$

- `$$formula$$` or `\[ formula \]` centers the given `formula` on the following line

```
The following is the equation for a line $$y=ax+b$$
```

results in:
The following is the equation for a line

$$y = ax + b$$

- `\begin{eqnarray} formulas \end{eqnarray}` allows multiple lines to be inputted into the environment with each equation line labeled with a reference number (use the *-form to stop LaTeX from automatically labeling a reference number).
  - One needs to mark the end of an equation line by `\\` and can use & before and after any symbol in all the formulas to align all of them vertically by those symbols.

```
The following is the equation for a line
\begin{eqnarray}
y&=&ax+b \label{eq:am}\\
&=&3x+1
\end{eqnarray}
By using labels on certain equations, we can refer to equations by
number, such as equation ~(\ref{eq:am}).
```

results in:
The following is the equation for a line

$$\begin{aligned} y &= ax + b && (1) \\ &= 3x + 1 && (2) \end{aligned}$$

By using labels on certain equations, we can refer to equations by number, such as equation (1).

Listed below are the main elements of mathematical formula symbols given in LaTeX. A more extensive list can be found in the additional reference LatexMathSymbols.pdf.

LATEX code	description
`+, -, /, *`	addition, subtraction, division, multiplication
`[, ], (, ),\{, \}`	parentheses
`{, }`	logically combine parts
`^` (carrot)	raise to a power, use `{ }` if more than 1 digit
`_` (dash)	subscript, use `{ }` if more than 1 digit subscript
`\sum`	summation symbol, $\sum$
`\int`	integration symbol, $\int$
`\frac{num}{denom}`	makes a fraction, $\frac{1}{2}$

3.2 Lists

Enumerate, Itemize, and Description are all enviornments for different sequential bullet points. Each new entry should begin it's line with `\item` for `itemize` and `enumerate` and `\item[label]` for `description`.

- `enumerate` uses number and letters

```
\begin{enumerate}
\item First bullet point
\end{enumerate}
```

results in:

1. First bullet point

- `itemize` uses different character symbols

```
\begin{itemize}
\item First bullet point
\end{itemize}
```

results in:

- First bullet point

- `description` uses words to offset different items

```
\begin{description}
\item First bullet point
\end{description}
```

results in:

First bullet point

Any of the above lists can be nested within one another or itself to the depth of four levels.

```
\begin{enumerate}
\item First bullet
\begin{itemize}
\item First sub-bullet
\begin{enumerate}
\item First subsub-bullet
\item Second subsub-bullet
\end{enumerate}
\item Second sub-bullet
\end{itemize}
\item Second bullet
\end{enumerate}
```

results in:

1. First bullet
    - First sub-bullet
        (a) First subsub-bullet
        (b) Second subsub-bullet
    - Second sub-bullet
2. Second bullet

3.3 Figures

Figures, drawings, graphs, etc. can all be imported (as .jpg, .png, and .pdf) into a document using the `graphicx` and `float` packages (no options necessary) by using the `figure` environment:

```
\begin{figure}[pos]
\centering
\includegraphics[attr1=val1, ...]{imagename}
\caption{This is a figure}
\label{fig:myfirstfig}
\end{figure}
```

The `figure` environment allows graphics to be titled with a caption and automatically numbered. One can use the `\includegraphics` command without the `figure` environment but then it will not automatically be numbered.

The graphic being imported must be saved in the same folder that the .txt file of the document you are creating is found, otherwise the path to the document must be included. The file name should not contain any spaces (use underscores instead) and should contain the file extension when called in the `figure` environment.

It is important to note that figures, and tables as we will learn later, are not always placed exactly where you want them to go. Formatting them to the desired location can be frustrating so it is usually easier to just let LaTeX place the object and use a reference label to refer to them in the document. The figure can be referenced in the text by using $\sim$ `\ref{fig:my_figure}`.

The following are the most common declarations used for the `figure` environment:

- `pos` is the vertical positioning argument. It can take the values
  - `t` – top line of the talbe is aligned with the baseline of the current external line of text
  - `b` – bottom line of the table is aligned with the external baseline
  - `h` – placed here
  - no value defined – table is centered on the external baseline
- `attr=val` are possible attributes of a graphic one can change
  - `width=xx` – Specify the preferred width of the image to `xx`
  - `height=xx` – Specify the perferred height of the image to `xx`
  - `angle=xx` – Rotates the image by `xx` degrees
  - `keepaspectratio` – Preserves the aspect ratio of the original figure

```
\begin{figure}[h]
\centering
\includegraphics[width=4in]{StatLabLogo.jpg}
\caption{UVA StatLab}
\label{fig:Logo}
\end{figure}
The StatLab provides plenty of resources and statistical help.
Just to name a few check out Figure~\ref{fig:Logo}.
```

results in the figure below

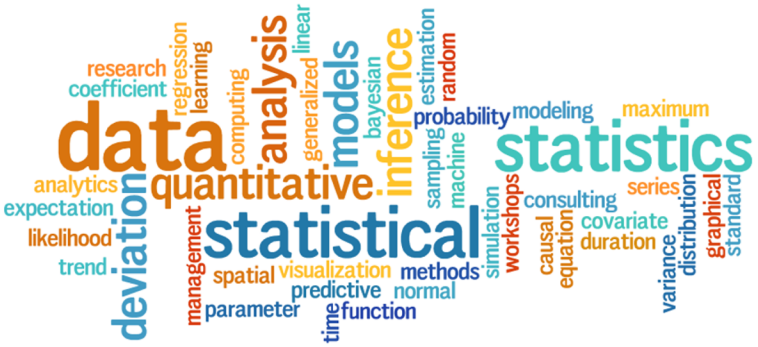


Figure 1: UVA StatLab

The StatLab provides plenty of resources and statistical help. Just to name a few check out Figure 1.

3.4 Tables

Tables are relatively straightforward to generate and go in their own environments, in which the `table` environment envelopes the `tabular` environment (similar to the figure environment above):

```
\begin{table}[pos]
\begin{tabular}{cols}
%rows of table
\end{tabular}
\caption{This table shows some data}
\label{tab:myfirsttable}
\end{table}
```

year	min temp (C)	max temp (C)
1970	−5	35
1975	−7	29
1980	−3	30
1985	−2	32

Table 1: Fictional minimal and maximal temperatures recorded in Cambridge over several years.

As one will see in the example below, the ampersand (&) separates the contents of each cell, every line has to be ended with \\, and one needs to declare a justification for each column. One can also use an additional application LaTable, http://www.ctan.org/pkg/latable.

```
\begin{table}
  \centering
  \begin{tabular}{ccc}
    year & min temp (C) & max temp (C)\\
    \hline
    1970 & $-5$ & 35\\
    1975 & $-7$ & 29\\
    1980 & $-3$ & 30\\
    1985 & $-2$ & 32\\
  \end{tabular}
  \caption{Fictional minimal and maximal temperatures recorded in
    Cambridge over several years.}
  \label{tab:simple}
\end{table}
where one can use a reference label to refer to it,  such as
Table~\ref{tab:simple}.
```

results in the table on the page above

The following are the most common declarations needed for the `tabular` and `table` environment:

- `pos` is the vertical positioning argument. It can take the values
  - `t` – top line of the talbe is aligned with the baseline of the current external line of text
  - `b` – bottom line of the table is aligned with the external baseline
  - `h` – Position here (approximately where it occurs in the text)
  - no value defined – table is centered on the external baseline

- `cols` is the column formatting argument, in which there must be an entry for every column (extra entries can be added for the left and right borders or intercolumn spacing). It can take the following values
  - column formatting
    - `l` – column contents are left justified
    - `r` – column contents are right justified
    - `c` – column contents are centered
    - `*{num}{cols}` – column format in `cols` is reproduced `num` times (e.g. `*{3}{c}` is same as `c c c`)
  - formatting symbols
    - `|` – draws vertical line
    - `||` – draws two vertical lines next to each other
    - `@{text}` – inserts `text` in every line of the table between the two columns where it appears (e.g. `@{\hspace{0.5in}}` adds extra spacing)
    - `>{$}l<{$}` – causes all entries in that column to be math mode (allows you not to have to declare math mode for each entry individually)
- `rows` contain the actual entries in the table, each horizontal row being terminated with `\\`. Each column in each row is sperated by the `&` symbol and each row must contain the same number of entries as the number of columns defined in `cols`. (Entries can be empty)
  - `\hline` – draws a horizontal line the full width of the table below the row that was just ended (`\\`) or at the top of the table, depending on placement. (Two together creates two horizontal lines with a little space between them)
  - `\cline{n-m}` – draws a horizontal line from the left side of column `n` to the right side of column `m`.
  - `\multicolumn{num}{col}{text}` – combines the following `num` columns into a single column with their total width including intercolumn spacing. Only used at the start of a row or right after a column separation symbol `&`
  - `\vline` – draws a vertical line with the height of the row at the location where it appears.

References

Helmut Kopka and Patrick W. Daly. *A Guide to LATEX: Document Preparation for Beginners and Advanced Users*. Addison-Wesley, Harlow, England, 1999.