LaTeX Crash Course

Javier Osorio

School of Government and Public Policy

University of Arizona

Spring, 2019

Contents

1 Getting started with LaTeX

LaTeX is a document preparation system for high-quality typesetting. It is the standard system for developing and publishing scientific or technical documents across disciplines. This typesetting system, pronounced as "Lah-tech" or "Lay-tech," allows users to concentrate in the content of their documents without worrying too much about the format or appearance of their documents. The flexibility and stability of LaTeX makes it easy to write journal articles, technical reports, books, and presentations. It allows users to clearly write complex mathematical formulas, structure long documents in sections, embed tables and figures, and manage citations and bibliographies in different styles.

To develop and compile LaTeX documents in your computer, you need to download LaTeX online for free at `https://www.latex-project.org/get/`. LaTeX is not a stand-alone document development software in itself, but a system that bundles different utilities and packages that enable writing and compiling high-quality scientific documents. LaTeX is available from different distributions that already include the most common packages to develop documents. These distributions are available for different operating systems:

- Windows: ProTeXt `http://www.tug.org/protext/`
- Mac OS X: MacTeX `http://www.tug.org/mactex/`
- Linux: LeXLive `https://www.tug.org/texlive/`

LaTeX distributions usually come in large installation files that might take considerable space in your computer. Successful local compilation also generates multiple files in your computer. In addition, you might need to download additional packages for advance users. Also, distributions of local LaTeX operate in a console that often requires running several times to effectively compile a document. These issues are not a big hurdle to run LaTeX in a local environment but, in my experience, developing and compiling LaTeX online is much easier.

To use LaTeX online, I recommend using Overleaf, which is available at `https://www.overleaf.com/`. There are other online LaTeX compilers such as Papieera or Datazar that

also work very well. Feel free to use the one that you like the most. In this workshop, we will be using Overleaf.

To get started in Overleaf, the first step is to log-in to your account. To create a new document, click on the green button "New Project," located at the top left of the interface. Then, click on "Blank Project" and assign a name to your file.

2 Basic Elements

This section discusses the main structure of LaTeX documents; the configuration of the preamble; the cover page including the title, author, and date; sections and subsections; different types of fonts, sizes, and text colors; itemized and enumerated lists; footnotes; hyperlinks; and comments.

2.1 Document Structure

The first step to start a LaTeXdocument is to create a structure. You simply write the following elements:

```
\documentclass{article}
\begin{document}
This is where you write your wonderful ideas in \LaTeX!!!
\end{document}
```

2.2 Preamble

The Preamble is the section where you define the document environment and its characteristics. Here is also where you call a variety of packages to execute specific functions using the command `\usepackage{}`.

The first step is to define the main type of document that you want to create. There are different types of document classes. The most common are:

- `article`: for academic articles.

- `beamer`: for presentation.
- `book`: for books.
- `report`: for long documents, probably your dissertation.
- `letter`: for writing letters.

Define the size of the page and the font in the document class. Let's use a letter size page by setting the `letterpaper` option and choosing a font 12 for the document. There are other font sizes to choose from (`9pt, 11pt, 12pt`). It is also necessary to use the `inputenc` to define the encoding of the document as UTF-8. Also, the `geometry` package allows to specify the page margins of the document.

```
\documentclass[12pt,letterpaper]{article}
\usepackage[utf8]{inputenc}
\usepackage[margin=1in]{geometry}
```

2.3 Title, author, and date

As part of the preamble, Write the title of your paper and enter your name. To specify the date, you can enter a fixed date (e.g. February, 2019) or you can use the `\date{\today}` option to get the current date every time you recompile the file.

```
\title{Really Cool Title}
\author{Your name}
\date{February, 2019}
```

To have the title, author, and date displayed in the paper, it is necessary to enter the command `\maketitle` below the `\begin{\document}` element. At this point, the script of your LaTeXdocument should be looking like this:

```
\documentclass[12pt,letterpaper]{article}
\usepackage[utf8]{inputenc}
\usepackage[margin=1in]{geometry}
\title{Really Cool Title}
\author{Your name}
\date{February, 2019}
\begin{document}
\maketitle
This is where you write your wonderful ideas in \LaTeX!!!
\end{document}
```

You might want to thank all your fans and sponsors. To do so, the `\thanks{}` option generates a footnote where you can add a thank you note or any other information.

```
\author{Your name\thanks{Thanks for being awesome!}}
```

To include your coauthors and indicate their institutions, you will need to include the `authblk` package in the preamble and indicate the authors and institutions:

```
\documentclass[12pt,letterpaper]{article}
\usepackage[utf8]{inputenc}
\usepackage[margin=1in]{geometry}
\usepackage{authblk}
\title{Really Cool Title}
\author[1]{This is you \thanks{Thank your dog}}
\author[2]{This is your friend \thanks{Thank your cat}}
\affil[1]{Department A}
\affil[2]{Department D}
\date{February, 2019}
\begin{document}
\maketitle
This is where you write your wonderful ideas in \LaTeX!!!
\end{document}
```

2.4 Sections and subsections

LaTeX allows dividing the document into separate segments according to a hierarchical structure of sections, subsections, and subsubsections. LaTeX will automatically take care of the size of the font and the numbering section so you just focus on developing the content. If you are writing a presentation, a book, or a report, you might want to create a Table of Contents by including the `\tableofcontents{}` element after the document title. However, that might not a good idea for a journal article.

```
\begin{document}
\maketitle
\tableofcontents{}
\section{Main section}
This is where you write your wonderful ideas in \LaTeX!!!
\subsection{Secondary section}
Keep the ideas flowing.
\subsubsection{You got the idea}
OK, that is good.
\section{To hide the section number}
Simply include an asterisk before the opening bracket
\end{document}
```

2.5 Font type, size, and color

One of the nice things about LaTeXis the possibility of using different types of fonts in a seamless manner. Here is a list of the different font types:

`\textbf{}`	**Text sample in Bold font**
`\texttt{}`	`Text sample in Typewriter`
`\textit{}`	*Text sample in Italics*
`\emph{}`	*Another sample of Italics using emphasis*
`\textrm{}`	Text sample in Roman family
`\textsf{}`	Text sample in Sans serif family
`\textsl{}`	*Text sample in Slanted*
`\textsc{}`	Text sample in Small Capitals
`\uppercase{}`	TEXT SAMPLE IN UPPERCASE

In this way, `you can use a variety of font` *families and styles* to put different emphasis on the content of **your document**. You can even Combine Different Fonts **By Nesting The Styles.**

Sometimes you might find useful to use different font sizes. There are two ways to specify the font size in the text:

1. Using the `\size` command will assign that font size to all the text followed after the command

2. Using the `\begin{size} Your text here \end{size}` command only assigns the specified font size to the text between the begin and end functions.

LaTeXallows you to do that by entering the following specifications for your text.

`\Huge`	Text sample in Huge font
`\huge`	Text sample in huge font
`\LARGE`	Text sample LARGE font
`\Large`	Text sample Large font
`\large`	Text sample in large font
`\normalsize`	Text sample in normalsize font
`\small`	Text sample in small font
`\footnotesize`	Text sample in footnotesize font
`\scriptsize`	Text sample in scriptsize font
`\tiny`	Text sample in tiny font

2.6 Text Color

To change the font color, users need to include first the `xcolor` package in the document preamble. This is a very flexible package that allows different modes of editing the color of the text.

The basic usage allows changing the color of the font. The script below shows the syntax.

```
The basic usage allows \textcolor{red}{changing the color of the font}.
```

Another use is to highlight the text using a color box.

```
The basic usage allows \textcolor{red}{changing the color of the font}.
```

Of course, users can combine these scripts to create the following phrase.

```
\colorbox{black}{\textcolor{yellow}{to create the following phrase}}.
```

Users can assign gradual transparency in the color of the text or in the color boxes.

```
transparency in the \textcolor{blue!30}{color of} \textcolor{blue!80}{the text}
or in the \colorbox{red!30}{color} \colorbox{red!80}{boxes}
```

2.7 Lists

LaTeXallows users to generate different types of lists in your document.

Itemize

The following code generates an itemized list of three elements.

```
\begin{itemize}
\item This is your first item
\item This is your second  item
\item This is your third item
\end{itemize}
```

Which generates the following itemized list:

- This is your first item
- This is your second item
- This is your third item

Enumerate

The following code generates an numeric list of three elements.

```
\begin{enumerate}
\item This is your first item
\item This is your second item
\item This is your third item
\end{enumerate}
```

Which generates the following numeric list:

1. This is your first item
2. This is your second item
3. This is your third item

Nesting lists

Users can also nest lists in hierarchical structures.

```
\begin{enumerate}
\item This is the first level
    \begin{enumerate}
    \item This is the second level
        \begin{enumerate}
        \item This is the third level
            \begin{enumerate}
            \item This is the fourth level
            \end{enumerate}
        \end{enumerate}
    \end{enumerate}
\end{enumerate}
```

Which generates the following structural numeric list:

1. This is the first level
    (a) This is the second level
        i. This is the third level
            A. This is the fourth level

LaTeX allows a maximum of four levels of nesting in itemized and numeric lists. Beyond such level, the system generates a compiling error indicating that the item is "Too deeply nested." In such cases, LaTeX will compile the document, but will show the deeply nested items at the fourth level.

Combining lists

LaTeX also allows to combine numeric and itemized lists as the following script shows:

```
\begin{enumerate}
\item This is your first item
\item This is your second item
    \begin{itemize}
    \item This is the first item in the second level
    \item This is the second item in the second level
    \end{itemize}
\item  This is your third item
    \begin{itemize}
    \item This is the first item in the second level
        \begin{itemize}
        \item This is the first item in the third level
        \end{itemize}
    \end{itemize}
\end{enumerate}
```

1. This is your first item
2. This is your second item
    - This is the first item in the second level
    - This is the second item in the second level
3. This is your third item
    - This is the first item in the second level
        - This is the first item in the third level

Customizing bullets in itemized lists

Users can also customize the bullet markers in itemized lists. Here are some ways of customizing item markers:

```
\item This is the default bullet
\item[*] This item shows a customized bullet
\item[-] This item shows another customized bullet
\item[] This item has no bullet
```

Which generates the following list:

- This is the default bullet

* This item shows a customized bullet

- This item shows another customized bullet

  This item has no bullet

2.8 Footnotes

To write a footnote, all you have to do is to use the `\footnote{Text}` command. That will give you a footnote.[1] The nice thing about LaTeX is that you do not have to worry about how many footnotes you enter,[2] the system will automatically take care of the footnote numbering.[3]

2.9 Hyperlinks

To generate an active Uniform Resource Locator (URL) or hyperlink in the document, users need to include the `hyperref` package in the document preamble.

Once loaded, the `\url{www}` command allows users to generate a working URL that explicitly shows the web address in the following way: `https://sgpp.arizona.edu/`.

As an alternative, users may want to hide an active link behind the text. To do so, they need to use the `\href{www}{description}` command, where `www` is the URL and `desc` is the description of the text.

[1] Just like this footnote.
[2] Like this other footnote.
[3] Isn't that nice?

For example, this code `\href{https://sgpp.arizona.edu/}{SGPP Home Page}`, generates this link: SGPP Home Page.

2.10 Comments

Invisible comments

To write a comment in the LaTeX script that is not visible in the PDF, simple use the `%` sign. For example, the following annotation:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Important reminder:
Writing in \LaTeX~is lots of fun! Shall you ever get stuck,
% do not despair and
just Google it!!
% There are lots of resources out there to help you.
```

Generates this output:

Writing in LaTeX is lots of fun! Shall you ever get stuck, just Google it!!

Visible comments

To generate notes that are visible in the PDF document, users need to include the `todonotes` package in the preamble of the script. Once loaded, there are different ways of using `todonotes`.

You can use the command `\todo{Your comment}` to create a bubble annotation at the margin of the document.

Do this

This package also allows entering a comment in the body of the document by using the command `\todo[inline]{Your comment}`, which generates the following comment:

Your comment

.

3 Tables

Tables and Figures are *floating* elements that can be positioned and formatted in LaTeXdocuments according to the user's needs. For this reason, these elements are referred as `floats` in the LaTeX jargon.

3.1 Basic Tables

LaTeX has considerable flexibility to allow users to write high-quality tables. In contrast to most text processors that often mess with the format of complex tables, LaTeX maintains a high-degree of stability when producing tables.

To produce your first table, simply create a `tabular` environment:

```
\begin{tabular}{lcr}
Header A & Header B & Header C \\
cell A1  &  cell B1  &  cell C1 \\
cell A2  &  cell B2  &  cell C2 \\
cell A3  &  cell B3  &  cell C3 \\
\end{tabular}
```

Which, produces a basic table:

Header A	Header B	Header C
cell A1	cell B1	cell C1
cell A2	cell B2	cell C2
cell A3	cell B3	cell C3

The number of elements after the `\begin{tabular}` command indicates the number of columns considered in the Table. In this case, the `{lcr}` description corresponds to three columns. The `l`, `c`, and `r` specifications indicate that the fist column is aligned to the left, the second is centered, and the third column is aligned to the right. The `tabular` environment uses the ampersand sign `&` to mark column separations and the `\\` sign to separate rows.

It is easy to improve the looks of this basic table by including additional elements. For example, adding bars to the `{|l|c|r|}` description, will create vertical lines at the extreme of the table and between columns. The command `\hline` draws a horizontal line to separate rows. We can also black font for the column headers. Bringing these elements to the script:

```
\begin{tabular}{|l|c|r|}
\hline
\textbf{Header A} & \textbf{Header B} & \textbf{Header C} \\
\hline\hline
cell A1  &  cell B1  &  cell C1 \\
cell A2  &  cell B2  &  cell C2 \\
cell A3  &  cell B3  &  cell C3 \\ \hline
\end{tabular}
```

This produces a better looking table:

Header A	**Header B**	**Header C**
cell A1	cell B1	cell C1
cell A2	cell B2	cell C2
cell A3	cell B3	cell C3

3.2 Milti-column

To merge cells across columns, you must use the `\multicolumn{}{}{}` command. Notice the three sets of brackets `{}`. The first set of brackets indicates the number of columns to be merged. The second set of brackets indicates the alignment (`l, c, r`) of the cell. Finally, the user writes the content of the cell the third set of brackets. For example, the command `\multicolumn{2}{|c|}{Header B}` will merge two columns, assign a centered alignment with line separators, and enter the text "`Header B`."

When merging cells across columns, it is important to keep in mind the position of the `\multicolumn{}{}{}` and the overall number of columns so it does not alter the general structure of the table as specified in the `tabular` environment.

For example, consider the following script:

```
\begin{tabular}{|l|c|r|}
\hline
\textbf{Header A} & \multicolumn{2}{|c|}{\textbf{Header B}} \\
\hline\hline
cell A1  &  cell B1  &  cell C1 \\
cell A2  &  cell B2  &  cell C2 \\
cell A3  &  cell B3  &  cell C3 \\ \hline
\end{tabular}
```

This generates a table with cells combined across columns:

Header A	**Header B**	
cell A1	cell B1	cell C1
cell A2	cell B2	cell C2
cell A3	cell B3	cell C3

3.3 Multi-row

To merge cells across rows, users must first specify the `\usepackage{multirow}`. To use this package, users must enter the following command `\multirow{}{}{}`. The first set of brackets must indicate the number of rows to be merged. The second set of brackets specifies the width of the merged cell. You can use the `{*}` to keep the natural width of the cell. Users must specify the content of the cell in the third set of brackets. For example, the command `\multirow{2}{*}{rows A2 and A3}` merges a cell across two rows, keeping the natural widht of the column, and including the text "`rows A2 and A3`."

For example, considering the following script:

```
\begin{tabular}{|l|c|r|}
\hline
\textbf{Header A} & \textbf{Header B} & \textbf{Header C} \\
\hline\hline
cell A1  &  cell B1  &  cell C1 \\ \hline
\multirow{2}{*}{cells A2 and A3}  &  cell B2  &  cell C2 \\ \cline{2-3}
       &  cell B3  &  cell C3 \\ \hline
\end{tabular}
```

When using the `multirow` package, keep in mind that the total number of rows must match the overall number of rows in the table. For example, notice that the first cell of the last row is empty in the script above because of the `multirow` specification in the line above. Also, to improve the looks of the Table, the script includes the `\cline` command, that draws a partial horizontal line to separate rows between columns 2 and 3, but not across the entire table as `\hline` would do.

Which produces this table:

Header A	**Header B**	**Header C**
cell A1	cell B1	cell C1
cells A2 and A3	cell B2	cell C2
	cell B3	cell C3

3.4 Combining multi-column and multi-row

Of course, LaTeX allows combining the multi-column and multi-cell Commands to generate more sophisticated tables. For example, consider the following script.

```
\begin{tabular}{|l|c|r|}
\hline
\textbf{Header A} & \multicolumn{2}{|c|}{\textbf{Header B}} \\
\hline\hline
cell A1  &  cell B1  &  cell C1 \\ \hline
\multirow{2}{*}{cells A2 and A3}  &  cell B2  &  cell C2 \\ \cline{2-3}
       &  cell B3  &  cell C3 \\ \hline
\end{tabular}
```

Which produces the following table:

Header A	**Header B**	
cell A1	cell B1	cell C1
cells A2 and A3	cell B2	cell C2
	cell B3	cell C3

3.5 Cell Color

To change the color of cells in a table, users must call the `colortbl` package in the document preamble. Then, users have to enter the following command `cellcolorcolor` within the `&` delimiters of the cell to be colored. In the following script, the specification `\cellcolor{lightgray}` in the three cells of the top row fills in gray the header of the table. The `colortbl` package also allows indicating the percentage of intensity of a color. To do so, the use has to include an excalmation sign `!` followed by a number ranging from [0-100] to indicate the percentage of color intensity. For example, the specification of `\cellcolor{blue!25}` in A1 fills the cell in blue at 25% of intensity, `\cellcolor{blue!50}` in A2 fills the cell in blue at 50%, and `\cellcolor{blue!75}` in A3 fills the cell in blue at 75%.

The following scrip:

```
\begin{tabular}{|l|c|r|}
\hline
\cellcolor{lightgray}\textbf{Header A} &
\cellcolor{lightgray}\textbf{Header B} &
\cellcolor{lightgray}\textbf{Header C} \\
\hline\hline
\cellcolor{blue!25}cell A1  &  cell B1  &  cell C1 \\
\cellcolor{blue!50}cell A2  &  cell B2  &  cell C2 \\
\cellcolor{blue!75}cell A3  &  cell B3  &  cell C3 \\ \hline
\end{tabular}
```

Will produce this colored table:

Header A	**Header B**	**Header C**
cell A1	cell B1	cell C1
cell A2	cell B2	cell C2
cell A3	cell B3	cell C3

3.6 Table Position, Title, and Numeric Reference

LaTeX allows additional table specifications such as placing the table in a specific position in the document, indicating the table title, as well as automatically tracking the number of tables and making references to tables. To manipulate these features, users need to create a `table` environment. For example, consider the following code:

```
\begin{table}[h!]
  \begin{center}
    \caption{My first table.}
    \label{tab:table1}
    \begin{tabular}{|l|c|r|}
    \hline
    \textbf{Header A} & \multicolumn{2}{|c|}{\textbf{Header B}} \\
    \hline\hline
    cell A1  &  cell B1  &  cell C1 \\
    cell A2  &  cell B2  &  cell C2 \\
    cell A3  &  cell B3  &  cell C3 \\ \hline
    \end{tabular}
  \end{center}
\end{table}
```

Start by opening a `{table}` environment. The `{h!}` specification after `{table}` places the table "*here*", or approximately to where the table is indicated in the script. Also, the exclamation sign `{!}` overrides other LaTeXparameters that might interfere with positioning the table in the designated place. There are different position specifications in LaTeX:

- `{H}`: place the float *exactly* at the location in the script
- `{h}`: place the float approximately "*here*."
- `{t}`: place the float on top of the page.
- `{b}`: place the float at the bottom of the page.
- `{p}`: pace the float on a special page for floats.
- `{!}`: overrides internal parameters to identify the "best" place of the Table

In order to make the table look nicer, the user can create a `{center}` environment within the `{table}` environment to center the entire Table.

To add a Title to the Table, the user has to assign a name within the `\caption{Title}` command. Placing the `\caption{}` above the `{tabular}` environment, as indicated in this example, will place the Title above the table. In contrast, when the `\caption{}` is placed

below the `{tabular}`, the Table Title appears at the bottom of the Table. In this particular example, the Table Title is `\caption{My First Table}`.

To reference a Table (or any other kind of object such as chapters, sections, figures, equations, etc.) across the document, users need to specify first the name of the label using the following command: `\label{name}`, included within the `table` environment. LaTeX will automatically generate a counter for each type of object (e.g. Tables or Figures). After specifying the label, the user has to indicate in the body of the document the reference to that specific Table using the command `\ref{name}`. In this particular example, the label and reference name of this float is `{tab:table1}`.

The script with the elements mentioned above generates Table I.

Table I: My First Table.

Header A	**Header B**	
cell A1	cell B1	cell C1
cell A2	cell B2	cell C2
cell A3	cell B3	cell C3

4 Figures

LaTeX considers Figures as floating objects, just like it does with Tables. Therefore, the functions discussed before for positioning Tables, assigning a Title, and making cross-references in the document also apply to Figures.

The `float` package improves the basic LaTeX functions for figures and tables. It is particularly useful to better manage the position of floating objects. So, make sure to include it in your document preamble.

To insert a Figure in the document, the user needs first to create a Figure environment with the following command: `\begin{figure}`. To enter the Figure, the user then has to use the command `\includegraphics{name.ext}`, where `.ext` is the extension indicating the type of file. LaTeX can process the most popular image formats such as `.png`, `.jpg`,

`.pdf`. When specifying the file name, make sure to type the file name and extension correctly (remember, LaTeX is case sensitive). Otherwise, the system will issue an error message and will not render the image. Overleaf has a nice auto-fill function that prompts the names of image files existing within the project when entering the `\includegraphics{}` command.

The following script generates Figure 1 as an output.

```
\begin{figure}[H]
  \centering
  \includegraphics{image1.png}
  \caption{My First Figure.}
  \label{fig:figure1}
\end{figure}
```

Figure 1: My First Figure.

4.1 Adjusting the Figure Size

LaTeX allows adjusting the size of a Figure by increasing or deducing the dimensions of the image in a proportional manner, by indicating a specific width and/or height, or by considering the proportion of the page.

Resizing proportional to the image

Users can increase or reduce the size of the image proportionally by indicating the `scale`. For example, the command `\includegraphics[scale=0.5]{image1.png}` will reduce the image at half of its original size as presented in Figure 2.

Figure 2: Reduced image.

In contrast, the command `\includegraphics[scale=1.2]{image1.png}` will render the image at 120% of its original size as shown in Figure 3.

Figure 3: Expanded Figure.

Adjusting the width and height

Another way of modifying the Figure size in LaTeX is by specifying the width and height measures of the image. The following command will render the image with a width of 4 cm and a height of 6 cm `\includegraphics[width=4cm,height=6cm]{image1.png}` as shown in Figure 4.

Figure 4: Stretched Figure (4 cm x 6 cm).

Users can also use the imperial metric system to specify the figure dimensions. For example, the command `\includegraphics[width=5in,height=1.5in]{image1.png}` generates an image of 5 inches in width and 1.5 inches in height, as indicated in Figure 5.

Figure 5: Stretched Figure (5 in x 1.5 in).

Resizing proportional to the page

Finally, LaTeX allows managing the size of a Figure with respect to the area available in the page. For example, the command `\includegraphics[width=0.25\linewidth]{image1.png}` will render an image proportional to 25% of the width of the page as shown in Figure 6.

Figure 6: Figure size at 25% of the page.

4.2 Multiple Images

LaTeX allows including multiple images within the same Figure. To do so, the user has to embed a `subfigure` environment within the `figure` environment. All other parameters of the figure environment apply to the subfigure. For example, the following code will generate panels 7a and 7b in Figure7.

```
\begin{figure}[h]
\centering
    \begin{subfigure}{0.3\textwidth}
        \includegraphics[scale=1]{image1.png}
        \caption{UofA logo}
        \label{fig:figure7a}
    \end{subfigure}
    \begin{subfigure}{0.3\textwidth}
        \includegraphics[scale=1]{image2.jpg}
        \caption{Cat logo}
        \label{fig:figure7b}
    \end{subfigure}
\caption{Caption for this figure with two images}
\label{fig:figure7}
\end{figure}
```

(a) UofA logo

(b) Cat logo

Figure 7: Figure with two images.

When dealing with figures of different sizes, you can use any of the approaches discussed before to redefine the dimensions of the images within the `subfigure` environment. For example, Figure 8 presents the images of Panel 8a and Panel 8b with the same size using the `\includegraphics[width=4cm,height=4cm]{}` specification.

(a) UofA logo (b) Cat logo

Figure 8: Figure with two resized images.

5 Math Symbols and Equations

- Line formulas
- Floating formulas
- Greek and special characters

One of the nicest features of LaTeX is its great capacity for presenting mathematical writing in a clear and flexible manner. This is one of the key features that makes LaTeX be the industry standard for communicating scientific ideas. The `mathtools` enhances the functionality and flexibility for writing mathematical notation. So, make sure to always include it in the preamble of your document.

5.1 In line mathematical expression

One way to write a mathematical expression in a line of text is by using the "`\( \)`" delimiters. In this math mode, the "`\(`" delimiter is used to begin the mathematical expression and the second delimiter "`\)`" must be used to close it. For example, the script:

```
I  love the binomial theorem \((x + y)^{2}\) above all!
```

Generates the following sentence: I love the binomial theorem $(x + y)^2$ above all!

A second way of writing equations in line is using the "`$$`" delimiters. Using this delimiter in the following script:

```
I  love the binomial theorem $(x + y)^{2}$ above all!
```

Generates the following sentence: I love the binomial theorem $(x + y)^2$ above all!

There is a third mode for writing in line mathematical expressions using the `\begin{math} \end{math}` delimiters. For example, the following code:

```
I  love the binomial theorem \begin{math}(x + y)^{2}\end{math} above all!
```

Generates the following sentence: I love the binomial theorem $(x + y)^2$ above all!

5.2 Equation mode

There are also three ways of writing stand-alone equations in LaTeX. One way is using the "`\[  \]`" delimiters. For example, the code:

```
\[ (x + y)^{2} \]
```

Generates the following equation:

$$(x + y)^2$$

The second approach, uses the "`$$  $$`" delimiters. For example, the code:

```
$$(x + y)^{2}$$
```

Generates the following equation:

$$(x + y)^2$$

The third mode to write equations uses the "`\begin{equation} \end{equation}`" delimiters. This environment has the advantage of automatically numbering the equations, and lets the user generate equation labels to reference them in the text. For example:

```
Equation \ref{eq:bin} is the best!
\begin{equation}
\label{eq:bin}
(x + y)^{2}
\end{equation}
```

Equation 1 is the best!

$$(x + y)^2 \tag{1}$$

5.3 Basic mathematical notation

LaTeX has the capacity to write complex mathematical expressions in a clear and flexible way. Consider the following basic notation examples:

- Superscripts and subscritps:

`a^2`	a^2
`a^{2}`	a^2
`a^{2+b}`	a^{2+b}
`a_{1}`	a_1
`a_{1+c}`	a_{1+c}
`a^{b}_{c+3}`	a^b_{c+3}

- Relations:

`>`	$>$
`<`	$<$
`\geq`	$\geq$
`\leq`	$\leq$
`\subset`	$\subset$
`\supset`	$\supset$
`\subseteq`	$\subseteq$
`\supseteq`	$\supseteq$

- Parentheses, brackets, braces

`(x)`	(x)
`[x]`	$[x]$
`\{x\}`	$\{x\}$
`\big(x\big)`	$\big(x\big)$
`\Big(x\Big)`	$\Big(x\Big)$
`\bigg(x\bigg)`	$\bigg(x\bigg)$
`\Bigg(x)(`	$\Bigg(x\Bigg($

- Sums and integrals

`\sum_{i=1}^{10} t_i`	$\sum_{i=1}^{10} t_i$
`\displaystyle\sum_{i=1}^{10} t_i`	$\displaystyle\sum_{i=1}^{10} t_i$
`\int_0^{\infty} e^{-x}`	$\int_0^{\infty} e^{-x}$
`\displaystyle\int_0^{\infty} e^{-x}`	$\displaystyle\int_0^{\infty} e^{-x}$

- Fractions

`\frac{a}{a+b}`	$\frac{a}{a+b}$
`\displaystyle\frac{a}{a+b}`	$\displaystyle\frac{a}{a+b}$
`\frac{a}{a+b} = \frac{c+d}{e}`	$\frac{a}{a+b} = \frac{c+d}{e}$
`\displaystyle\frac{a}{a+b} = \frac{c+d}{e}`	$\displaystyle\frac{a}{a+b} = \frac{c+d}{e}$

- Greek letters

`\alpha`	α
`\beta`	β
`\gamma`	γ
`\delta`	δ
`\theta`	θ
`\pi`	π
`\Gamma`	Γ
`\Delta`	Δ
`\Pi`	Π
`\Sigma`	Σ
`\Omega`	Ω

5.4 Putting everything together

To put everything together, consider the following script for equation 2:

```
\begin{equation}
\label{eq:reg}
y_{i,t} = \alpha +  \beta_{1} y_{i,t-1} + \beta_{2} X_{i,t} +
\beta_{2} X^{2}_{i,t} +  \epsilon _{i,t}
\end{equation}
```

$$y_{i,t} = \alpha + \beta_1 y_{i,t-1} + \beta_2 X_{i,t} + \beta_2 X^2_{i,t} + \epsilon_{i,t} \tag{2}$$

The following script generates a more sophisticated equation:

```
\begin{equation}
\label{eq:formula}
\frac{\delta^{2}}{e_{\lambda}} \geq \sum_{i=1}^{10}(r-\gamma)+\frac{\pi}{e^{n}}
\end{equation}
```

The result is equation 3:

$$\frac{\delta^2}{e_\lambda} \geq \sum_{i=1}^{10}(r-\gamma) + \frac{\pi}{e^n} \tag{3}$$

5.5 Other symbols

Here are some useful lists of math symbols that you might find useful:

- This is a short, but nice list of math symbols:
  `http://milde.users.sourceforge.net/LUCR/Math/mathpackages/mathabx-symbols.pdf`
- This is probably the longest, but most complete list of symbols I have seen:
  `https://www.rpi.edu/dept/arc/training/latex/LaTeX_symbols.pdf`

6 Importing regression results from R

6.1 Using the stargazer package in R

The "stargazer" package allows exporting tables and regression results from R into LaTeX.

First, let's open RStudio and run the following setup script:

```
#Install stargazer. Do this only once
install.packages("stargazer")


#Load the library
library(stargazer)


#Set your own working directory
setwd("C:/Users/javie/Desktop")
```

Then, get the data and run some regression models:

```
# Load the mtcars database
data <- mtcars


# Regression
model1 <- lm(mpg ~ am, data=data)
   summary(model1)
model2 <- lm(mpg ~ am + vs, data=data)
   summary(model2)
model3 <- lm(mpg ~ am + vs + gear, data=data)
   summary(model3)
```

Now, generate a regression table in `.tex` format to import in LaTeX:

```
stargazer(model1, model2, model3, type="latex", out="mytable1.tex")
```

6.2 Paste the table script in LaTeX

After running the R script, you will be able to find the file `mytable1.tex` in your working folder. Open the file, copy its content and paste it into your LaTeX script.

I recommend pasting the table code inside a single space environment to improve the looks of the table (`\begin{singlespace}...\end{singlespace}`). I also recommend eliminating the table extra space (`\\[-1.8ex]`) and commenting out the empty lines to avoid inefficient use of space in the table (`% & & & \\`). These changes should generate the following output:

	Dependent variable:		
	mpg		
	(1)	(2)	(3)
am	7.245***	6.067***	7.050***
	(1.764)	(1.275)	(2.091)
vs		6.929***	7.022***
		(1.262)	(1.286)
gear			−0.851
			(1.424)
Constant	17.147***	14.594***	17.293***
	(1.125)	(0.926)	(4.612)
Observations	32	32	32
R^2	0.360	0.686	0.690
Adjusted R^2	0.338	0.664	0.657
Residual Std. Error	4.902 (df = 30)	3.491 (df = 29)	3.531 (df = 28)
F Statistic	16.860*** (df = 1; 30)	31.690*** (df = 2; 29)	20.778*** (df = 3; 28)
Note:			*p<0.1; **p<0.05; ***p<0.01

You can also edit the labels of the dependent and independent variables:

```
stargazer(model1, model2, model3,
dep.var.labels=c("Miles per gallon"),
covariate.labels=c("Manual transmission","Straight engine","Num. of gears"),
type="latex", out="mytable2.tex")
```

	Dependent variable:		
	Miles per gallon		
	(1)	(2)	(3)
Manual transmission	7.245***	6.067***	7.050***
	(1.764)	(1.275)	(2.091)
Straight engine		6.929***	7.022***
		(1.262)	(1.286)
Num. of gears			−0.851
			(1.424)
Constant	17.147***	14.594***	17.293***
	(1.125)	(0.926)	(4.612)
Observations	32	32	32
R^2	0.360	0.686	0.690
Adjusted R^2	0.338	0.664	0.657
Residual Std. Error	4.902 (df = 30)	3.491 (df = 29)	3.531 (df = 28)
F Statistic	16.860*** (df = 1; 30)	31.690*** (df = 2; 29)	20.778*** (df = 3; 28)
Note:			*p<0.1; **p<0.05; ***p<0.01

7 Bibliography

7.1 Generating a BibTeX file with Mendeley

Mendeley is a citation management program that offers a variety of features to manage bibliographic resources. You can download the basic version of Mendeley for free at `https://www.mendeley.com/`. The features included in this program allow users to:

- Create collections of bibliographic references
- Manage citations
- Read and annotate PDF documents
- Create notes or summaries per document
- Automatically extract metadata from PDF documents
- Cloud-based syncrhonization of collections across devices
- Insert citations and automatically create bibliographies in World and OpenOffice
- Export collections in BibTeX format
- Many other

There are some good alternatives to Mendeley such as Zotero or EndNote. If you already use any citation manager system, please feel free to keep doing it. Just make sure you learn how to export your citations into BibTex format so you can use them in LaTeXdocuments.

There are different ways of adding files to Mendeley:

1. Create a citation manually
    - Click on the black arrow next to the Add button
    - This will display a menu, click on Add Entry Manually
    - This will open a window where you can enter the information of different types of citations (e.g. journal articles, books, etc.)
    - When entering the information, make sure you enter the **Citation Key**
        - I recommend simply using `AuthorYYYY` format.
2. Drag a journal article (in PDF format) into Mendeley

- Use any journal article that you already have in your computer (or download one from the web)
- Simply drag the PDF file into Mendeley
- Mendeley will automatically extract the citation meta-data of the article
- Always double check the accuracy of the meta-data.

3. Upload a citation file

   - Many journals offer the possibility of downloading article citations
   - Just download an article citation in BibTex format (`.bib`).
   - Then upload the citation into Mendeley
     - Click on the black arrow of the Add button
     - This will display a menu, click on Add files
     - Browse to the location of your file and select it
     - Click on Open

4. Use the online plug-in

   - We will not cover how to use the Mendeley Web Importer and the Word Plug-In in this workshop, but you will finds a nice explanation in this Youtube video: `https://www.youtube.com/watch?v=XTfVCiksapk`

To generate a collection of references in BibTex format, follow these steps:

1. Select the citations in Mendeley's main window
2. Click on the File menu, then click on Export
3. This opens a window where you get to specify the file name and its location
4. Make sure you save the file in BibTeX format (`*.bib`)

The following is an example of the standard structure of a BibTeX file containing a journal article and a book reference.

```
@article{Baggins2019,
author = {Baggins, Frodo},
journal = {The Shire Journal of Metallurgy},
title = {{How to Melt a Ring}},
volume = {10},
number = {1},
pages = {20 -- 30},
year = {2019}
}
@book{WoodlandC2018,
address = {Tucson},
author = {Woodland, Legolas and The Gray, Gandalf},
publisher = {University of Arizona Press},
title = {{Treaty on Orc Control}},
year = {2018}
}
@article{Gamgee2018,
author = {Gamgee, Sam and Baggins, Frodo and Baggins, Bilbo and Took, Peregrin},
journal = {Journal of Long Distance Travel},
title = {{Around the Middle-earth in 80 Days}},
volume = {32},
number = {4},
pages = {9 -- 27},
year = {2018}
}
```

To add the BibTeX collection to Overleaf, follow these steps:

1. In the left panel of Overleaf, click on the Upload icon
2. Drag the BibTeX file into Overleaf's upload window
3. Make sure the uploaded file shows the `.bib` extension

4. Click on the uploaded BibTeX file to make sure the content is correct

7.2 Managing citations in LaTeX

There are different packages for managing citations in LaTeX(e.g. `natbib`, `cite`, or `biblatex`). Here we will be using `natbib`, which is very broadly used as it offers considerable flexibility for managing citations and producing references in a wide range of citation styles. There are two main elements involved in enabling citations in LaTeX using `natbib`:

1. Call the `natbib` package in the document preamble:

```
\usepackage{natbib}
```

2. Specify the bibliography style that you want to use (e.g. APA, Chicago, etc.) and the BibTeX file containing the collection of references:

```
\bibliographystyle{apsr}
\bibliography{mybiblio}
```

   - In this case, the bibliography style follows the format specifications of the American Political Science Association (APSR).

     There are many other citation styles to choose from, such as `chicago`, `apa`, `apalike`, etc.

   - The bibliography file is `mybiblio.bib`, created in Mendeley.

   - Insert these commands in the part of the script that you want the bibliography to appear. LaTeX will automatically create a References page in the specified location.

The two main commands in the `natbib` package are `\citep{}`, used for parenthetical citations, and `\citet{}`, which is used for textual citations. There are some other variations of these commands to customize citations according to the needs of your document. Here are some examples:

`\citep{Baggins2019}`	(Baggins, 2019)
`\citep[p. 17]{Baggins2019}`	(Baggins, 2019, p. 17)
`\citep[e.g.][p. 17]{Baggins2019}`	(e.g. Baggins, 2019, p. 17)
`\citet{WoodlandC2018}`	Woodland and The Gray (2018)
`\citet[chap. 1]{WoodlandC2018}`	Woodland and The Gray (2018, chap. 1)
`\citep{Gamgee2018}`	(Gamgee et al., 2018)
`\citep*{Gamgee2018}`	(Gamgee, Baggins, Baggins and Took, 2018)
`\citet{Gamgee2018}`	Gamgee et al. (2018)
`\citet*{Gamgee2018}`	Gamgee, Baggins, Baggins and Took (2018)
`\citep{Baggins2019,Gamgee2018}`	(Baggins, 2019; Gamgee et al., 2018)
`\citet{Baggins2019,Gamgee2018}`	Baggins (2019); Gamgee et al. (2018)
`\citeauthor{Baggins2019}`	Baggins
`\citeyear{Baggins2019}`	2019
`\citeyearpar{Baggins2019}`	(2019)

As an example, consider the following script:

```
The seminal paper by \citet{Baggins2019} launched a new idea. Early critics
remained skeptical \citep{Gamgee2018} until \citet[chap. 1]{WoodlandC2018}
found new evidence.
```

This would produce the following output:

The seminal paper by Baggins (2019) launched a new idea. Early critics remained skeptical (Gamgee et al., 2018) until Woodland and The Gray (2018, chap. 1) found new evidence.

For more details on the use of the `natbib` package, see: `http://ctan.mirrors.hoobly.com/macros/latex/contrib/natbib/natnotes.pdf`

8 Beamer Presentations

In LaTeX, a document containing presentation slides is called Beamer. All the LaTeX commands for content development and formatting discussed before are also applicable to a

Beamer document. To create presentations in LaTeX, it is necessary to create a new file using the `beamer` document class. Let's create a new document in Overleaf click on the green button "New Project," located at the top left of the interface. Then, click on "Blank Project" and assign a name to your file.

Beamer offers a wide range of customization of the general frame and looks of your presentation by specifying three main components:

- Theme
- Color
- Font

Based on these three elements, Beamer allows an enormous variety of presentation frameworks. Take a look at them in this link: `https://hartwork.org/beamer-theme-matrix/`, Here is a short list of some themes, colors, and fonts for your to consider:

- Short list of Beamer themes:
  - `default`
  - `Antibes`
  - `Berkeley`
  - `Berlin`
  - `Boadilla`
  - `Copenhagen`
  - `Frankfurt`
  - `Madrid`
  - `Singapore`
  - `Warsaw`
- Short list of Beamer colors:
  - `default`
  - `albatross`
  - `beaver`
  - `beetle`
  - `crane`

  - `dolphin`
  - `dove`
  - `fly`
  - `lily`
  - `orchid`

- List of Beamer fonts:

  - `default`
  - `professionalfonts`
  - `serif`
  - `structurebold`
  - `structureitalicserif`
  - `structuresmallcapsserif`

8.1 Beamer Preamble

Just like any LaTeX document, the preamble is the script section where you define the main characteristics of the presentation and call specific packages to perform the functions that you need.

Consider the following default preamble script. Feel free to customize the theme, color, and font any way you want. I am going to be using the `Boadilla` theme, with the `beaver` color, and the `serif` font.

```
\documentclass{beamer}
\usepackage[utf8x]{inputenc}
\mode<presentation>
{
  \usetheme{default}      % Define the theme
  \usecolortheme{default} % Define the color
  \usefonttheme{default}  % Define the font
}
```

8.2 Title, Author, and Date

Beamer allows additional customization of the title, author, institute, and date. These features are enabled in the `[options]` section available after the `\title`, `\author`, `\institute`, and `\date` commands. By populating these options, Beamer will present this information at the bottom of each slide throughout the entire presentation. For example, the following script:

```
\title[Short Title]{Really Cool Presentation Title}
\author[Osorio]{Javier Osorio}
\institute[UofA]{School of Government and Public Policy \\
      University of Arizona}
\date[February, 2019]{Presentation prepared for the \\
     \LaTeX Crash Course \\
     Tucson, AZ, \\
     February, 2019}
```

Will generate this presentation title page:

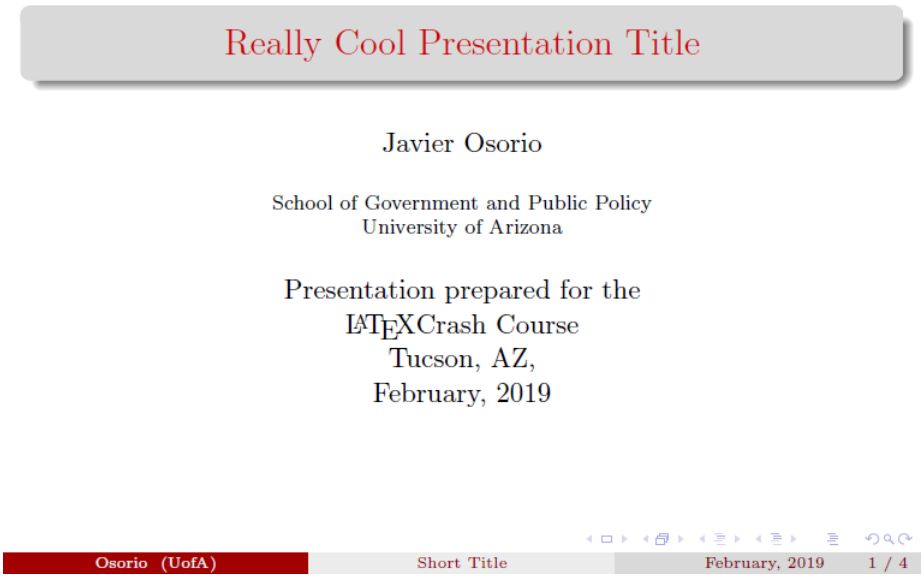

Figure 9: Beamer presentation.

8.3 Presentation slides

After specifying the preamble and the title page, you can now start developing the content of your presentation slides in Beamer. When developing your content, consider the following elements:

1. Just like any other LaTeX document, the entire content of the presentation must be contained within the `\begin{document}...\end{document}` commands.

2. The content of each slide must be contained within the `\begin{frame}...\end{frame}` commands.

3. Insert the `\titlepage` right after the `\begin{document}` command.

4. If you want to create a Table of Contents showing the sections of your presentation, you must enter the `\tableofcontents{}` command.

    - If you do not want a Table of Contents, you can simply delete these lines from the script, or comment them out with a %.

5. You can use ALL the format and content commands discussed before to develop the content of your slides. This includes, but is not limited to, the following basic functions:

    - Sections and subsections
    - Font type, size, and color
    - Text color in line and in table
    - Lists (itemized and numeric)
    - Tables and Figures
    - Mathematical notation

Below, you can finds an example for a set of presentation slides in Beamer. This assumes that you already included the document preamble, necessary packages, and title page information.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{document}
%-----------------------------------------------%
\begin{frame}
   \titlepage
\end{frame}
%-----------------------------------------------%
\begin{frame}{Table of Contents}
   \tableofcontents{}
\end{frame}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\section{Introduction}
%-----------------------------------------------%
\begin{frame}{Introduction}
   Text
   \begin{itemize}
     \item Some more text here
     \item And here
   \end{itemize}
\end{frame}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\section{Content}
%-----------------------------------------------%
\begin{frame}{Keep going}
   It seems that you are inspired today!
\end{frame}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\end{document}
```

References

Baggins, Frodo. 2019. "How to Melt a Ring." *The Shire Journal of Metallurgy* 10(1):20 – 30.

Gamgee, Sam, Frodo Baggins, Bilbo Baggins and Peregrin Took. 2018. "Around the Middle-earth in 80 Days." *Journal of Long Distance Travel* 32(4):9 – 27.

Woodland, Legolas and Gandalf The Gray. 2018. *Treaty on Orc Control*. Tucson: University of Arizona Press.