PRISM Introduction to $\LaTeX$

Matt Hitt &
Ben Jones

PRISM
Department of Political Science
hitt.23@osu.edu
jones.2781@osu.edu

January 20, 2011

LaTeX: What is it and why should you use it?

- LaTeX (pronounced “lah-tech”) is a tool for preparing documents and presentations.
- It gives the user extensive control over the format and appearance of documents and presentations.
- It has extensive functions for presenting scientific material such as equations, tables and figures.
- It allows you to easily store and display bibliographical information, and to convert citations formats easily.
- It makes your stuff look good.
- It’s free and works across operating systems.

LaTeX: Getting and Installing

- PC
  1. Download and Install MikTek.
  2. Download and Install an editor (WinEdt, Lyx, e.g.)
- MAC
  1. Download and Install TekShop

LaTeX: Getting and Installing

- PC
  1. Download and Install MikTek.
  2. Download and Install an editor (WinEdt, Lyx, e.g.)
- MAC
  1. Download and Install TekShop

LaTeX: Getting and Installing

- PC
  1. Download and Install MikTek.
  2. Download and Install an editor (WinEdt, Lyx, e.g.)
- MAC
  1. Download and Install TekShop

Basic Set-up: File Types (.tex, .bib, .pdf)

- Draft your document in a Tex Editor (e.g. TeXShop, WinEdt).
  - Drafting occurs in a Tex-file which has a `.tex` ending.
  - This is where you place all your content and formatting information.
  - The Tex environment gives you full control but does not attempt to be clever and correct your mistakes or make formatting decisions for you, unlike Microsoft Word.
- To produce your final document, you compile, or typeset your tex file to produce a PDF file with all of the content and formatting.
  - When you typeset, several auxiliary files are produced which are not important to you.
- Bibliographical information is stored in a bib file which has a `.bib` ending.
  - When you typeset a document with bibliographical information, the bib file is called.

Basic Set-up: File Types (.tex, .bib, .pdf)

- Draft your document in a Tex Editor (e.g. TeXShop, WinEdt).
  - Drafting occurs in a Tex-file which has a `.tex` ending.
  - This is where you place all your content and formatting information.
  - The Tex environment gives you full control but does not attempt to be clever and correct your mistakes or make formatting decisions for you, unlike Microsoft Word.
- To produce your final document, you compile, or typeset your tex file to produce a PDF file with all of the content and formatting.
  - When you typeset, several auxiliary files are produced which are not important to you.
- Bibliographical information is stored in a bib file which has a `.bib` ending.
  - When you typeset a document with bibliographical information, the bib file is called.

Basic Set-up: File Types (.tex, .bib, .pdf)

- Draft your document in a Tex Editor (e.g. TeXShop, WinEdt).
  - Drafting occurs in a Tex-file which has a `.tex` ending.
  - This is where you place all your content and formatting information.
  - The Tex environment gives you full control but does not attempt to be clever and correct your mistakes or make formatting decisions for you, unlike Microsoft Word.
- To produce your final document, you compile, or typeset your tex file to produce a PDF file with all of the content and formatting.
  - When you typeset, several auxiliary files are produced which are not important to you.
- Bibliographical information is stored in a bib file which has a `.bib` ending.
  - When you typeset a document with bibliographical information, the bib file is called.

Document Structure

- There are 3 basic parts of a typical LaTex document.

1. The preamble
2. The front matter
3. The body of the text

Document Structure

- There are 3 basic parts of a typical LaTex document.

1. The preamble
2. The front matter
3. The body of the text

Document Structure

- There are 3 basic parts of a typical LaTex document.

1. The preamble
2. The front matter
3. The body of the text

Document Structure

- There are 3 basic parts of a typical LaTex document.

1. The preamble
2. The front matter
3. The body of the text

The Preamble

- The preamble appears at the top of your document, before you actually start writing.
- This is where you specify what type of document you are producing and which packages you want to use in the document, much as you do in R.
  - `\documentclass{article}` `\usepackage{graphics}` etc.
- There are hundreds of packages you can use to control the look and feel of your document, including the margins, the appearance of the header and footer, specific citation styles, and many more.

The Preamble

- The preamble appears at the top of your document, before you actually start writing.
- This is where you specify what type of document you are producing and which packages you want to use in the document, much as you do in R.
  - `\documentclass{article}` `\usepackage{graphics}` etc.
- There are hundreds of packages you can use to control the look and feel of your document, including the margins, the appearance of the header and footer, specific citation styles, and many more.

The Preamble

- The preamble appears at the top of your document, before you actually start writing.
- This is where you specify what type of document you are producing and which packages you want to use in the document, much as you do in R.
  - `\documentclass{article}` `\usepackage{graphics}` etc.
- There are hundreds of packages you can use to control the look and feel of your document, including the margins, the appearance of the header and footer, specific citation styles, and many more.

The Front Matter

- The front matter is where you actually begin the document.
  - This is done using the `\begin{document}` command, discussed below.
- This is also where you can set up a title page with your name and the date, as well as an abstract.
  - `\title{}, \author{}, \maketitle`
    - Where `\maketitle` tells LaTex to use the previously specified author and title to construct a title page or heading.
  - `\begin{abstract}, \end{abstract}`

The Front Matter

- The front matter is where you actually begin the document.
  - This is done using the `\begin{document}` command, discussed below.
- This is also where you can set up a title page with your name and the date, as well as an abstract.
  - `\title{}, \author{}, \maketitle`
    - Where `\maketitle` tells LaTex to use the previously specified author and title to construct a title page or heading.
  - `\begin{abstract}, \end{abstract}`

The Front Matter

- The front matter is where you actually begin the document.
  - This is done using the `\begin{document}` command, discussed below.
- This is also where you can set up a title page with your name and the date, as well as an abstract.
  - `\title{}`, `\author{}`, `\maketitle`
    - Where `\maketitle` tells LaTex to use the previously specified author and title to construct a title page or heading.
  - `\begin{abstract}`, `\end{abstract}`

The Front Matter

- The front matter is where you actually begin the document.
  - This is done using the `\begin{document}` command, discussed below.
- This is also where you can set up a title page with your name and the date, as well as an abstract.
  - `\title{}, \author{}, \maketitle`
    - Where `\maketitle` tells LaTex to use the previously specified author and title to construct a title page or heading.
  - `\begin{abstract}, \end{abstract}`

The Body

- The body of the document is where the main text of your document will appear.
- You can divide a document into sections and subsections using the `\section{name}` and `\subsection{name}` commands.
- Formatting the text is done differently than in Word.
  - To indent a paragraph, simply leave a space between two blocks of text.
  - To force text in the final document to appear on the next line, you can use \\ to force subsequent text onto the next line.

The Body

- The body of the document is where the main text of your document will appear.
- You can divide a document into sections and subsections using the `\section{name}` and `\subsection{name}` commands.
- Formatting the text is done differently than in Word.
    - To indent a paragraph, simply leave a space between two blocks of text.
    - To force text in the final document to appear on the next line, you can use \\ to force subsequent text onto the next line.

The Body

- The body of the document is where the main text of your document will appear.
- You can divide a document into sections and subsections using the `\section{name}` and `\subsection{name}` commands.
- Formatting the text is done differently than in Word.
  - To indent a paragraph, simply leave a space between two blocks of text.
  - To force text in the final document to appear on the next line, you can use `\\` to force subsequent text onto the next line.

Commands and Text Options

- To tell LaTeX you are entering a command, preface the command with a backslash: \
  - Thus, whenever LaTex sees a \, that is an indication that what follows is a command, not text.
- Because LaTex does not have a GUI, you will need to use the backslash and commands to control how the text appears.
  - Italicized: `\textit{}`
  - Bold: `\textbf{}`
  - Font size: `\large`, `\tiny` etc.
- Similarly, text spacing can be controlled with:
  - `\singlespacing`, `\doublespacing`, or `\begin{singlespacing}`, `\end{singlespacing}`

Commands and Text Options

- To tell LaTeX you are entering a command, preface the command with a backslash: \
  - Thus, whenever LaTex sees a \, that is an indication that what follows is a command, not text.

- Because LaTex does not have a GUI, you will need to use the backslash and commands to control how the text appears.
  - Italicized: `\textit{}`
  - Bold: `\textbf{}`
  - Font size: `\large`, `\tiny` etc.

- Similarly, text spacing can be controlled with:
  - `\singlespacing`, `\doublespacing`, or `\begin{singlespacing}`, `\end{singlespacing}`

Commands and Text Options

- To tell LaTeX you are entering a command, preface the command with a backslash: \
  - Thus, whenever LaTex sees a \, that is an indication that what follows is a command, not text.
- Because LaTex does not have a GUI, you will need to use the backslash and commands to control how the text appears.
  - Italicized: `\textit{}`
  - Bold: `\textbf{}`
  - Font size: `\large`, `\tiny` etc.
- Similarly, text spacing can be controlled with:
  - `\singlespacing`, `\doublespacing`, or `\begin{singlespacing}`, `\end{singlespacing}`

Commands and Text Options

- To tell LaTeX you are entering a command, preface the command with a backslash: \
  - Thus, whenever LaTex sees a \, that is an indication that what follows is a command, not text.

- Because LaTex does not have a GUI, you will need to use the backslash and commands to control how the text appears.
  - Italicized: `\textit{}`
  - Bold: `\textbf{}`
  - Font size: `\large, \tiny` etc.

- Similarly, text spacing can be controlled with:
  - `\singlespacing, \doublespacing`, or `\begin{singlespacing}, \end{singlespacing}`

Beginning and Ending Environments

- Many of the special features of LaTeX are controlled within unique environments which begin with `\begin{environment}` and end with `\end{environment}`.
    - The content is written within `\begin{document} \end{document}`
    - **Tables** are written within `\begin{table} \end{table}`, and **figures** within `\begin{figure} \end{figure}`.
    - **Bullet points** take place within `\begin{itemize} \end{itemize}`, with each bullet point denoted by `\item` in the environment.
    - Sections of the document can be **single** or **double spaced** with `\begin{singlespacing} \end{singlespacing}`.
    - **Block quotes** can be set with `\begin{quote} \end{quote}`.
    - **Equations** can be denoted with `\begin{equation} \end{equation}`.

Beginning and Ending Environments

- Many of the special features of LaTeX are controlled within unique environments which begin with `\begin{environment}` and end with `\end{environment}`.
  - The content is written within `\begin{document} \end{document}`
  - **Tables** are written within `\begin{table} \end{table}`, and **figures** within `\begin{figure} \end{figure}`.
  - **Bullet points** take place within `\begin{itemize} \end{itemize}`, with each bullet point denoted by `\item` in the environment.
  - Sections of the document can be **single** or **double spaced** with `\begin{singlespacing} \end{singlespacing}`.
  - **Block quotes** can be set with `\begin{quote} \end{quote}`.
  - **Equations** can be denoted with `\begin{equation} \end{equation}`.

Math Mode (also known as the big payoff from this)

- LaTex is *extremely* useful for producing greek letters and math equations using "math mode," which is denoted as beginning and ending by $.
  - To produce any greek letter, you simply need to use math mode and the backslash: `$\beta$` produces β, `$\omega$` produces ω
  - To capitalize the greek letter, simply capitalize the word, so `$\Omega$` produces Ω.
- Full equations work in much the same fashion:
  - So, for example, the regression equation $y = x_1\beta_1 + x_2\beta_2 + x_2^2\beta_3$ is produced with `$y = x_1\beta_1 + x_2\beta_2 + x_2^2\beta_3$`
  - Or, $y_t = \gamma^t + \frac{2-1^{t-1}}{1+1}$ is produced with `$ y_t = \gamma^t + \frac{2-1^{t-1}}{1+1}$`.

Math Mode (also known as the big payoff from this)

- LaTex is *extremely* useful for producing greek letters and math equations using "math mode," which is denoted as beginning and ending by $.
    - To produce any greek letter, you simply need to use math mode and the backslash: `$\beta$` produces β, `$\omega$` produces ω
    - To capitalize the greek letter, simply capitalize the word, so `$\Omega$` produces Ω.

- Full equations work in much the same fashion:
    - So, for example, the regression equation $y = x_1\beta_1 + x_2\beta_2 + x_2^2\beta_3$ is produced with `$y = x_1\beta_1 + x_2\beta_2 + x_2^2\beta_3$`
    - Or, $y_t = \gamma^t + \frac{2-1^{t-1}}{1+1}$ is produced with `$ y_t = \gamma^t + \frac{2-1^{t-1}}{1+1}$`.

Mysterious Errors

- When compiling, latex sometimes gives cryptic error messages. Common ones include:
  - `! Undefined control sequence` Means you entered a command LaTex doesn't know. Check for typos, and make sure you loaded all the packages your commands require.
  - `Runaway argument?` You probably are missing a } somewhere. Or perhaps you forgot a \\ at the end of a row in a table.

- Errors prevent your document from compiling. Hit `e` then enter to stop the compilation and go to the line of code that's causing problems. Then try again!

Mysterious Errors

- When compiling, latex sometimes gives cryptic error messages. Common ones include:
    - `! Undefined control sequence` Means you entered a command LaTex doesn't know. Check for typos, and make sure you loaded all the packages your commands require.
    - `Runaway argument?` You probably are missing a } somewhere. Or perhaps you forgot a \\ at the end of a row in a table.
- Errors prevent your document from compiling. Hit e then enter to stop the compilation and go to the line of code that's causing problems. Then try again!

Mysterious Errors

- When compiling, latex sometimes gives cryptic error messages. Common ones include:
  - `! Undefined control sequence` Means you entered a command LaTex doesn't know. Check for typos, and make sure you loaded all the packages your commands require.
  - `Runaway argument?` You probably are missing a } somewhere. Or perhaps you forgot a \\ at the end of a row in a table.
- Errors prevent your document from compiling. Hit `e` then enter to stop the compilation and go to the line of code that's causing problems. Then try again!

Miscellaneous Quirks and Tips

- Many symbols you would typically use in a paper actually double as commands within LaTeX.
  - To avoid this, use the backspace, e.g. \% to produce %, or \$ to produce $.
- A frequent mistake involves quotation marks. In order to produce “, you actually need to type ‘ ‘ rather than “, which will result in ”.
- To place a footnote, just use `\footnote{}` and place the text of the footnote within the brackets.
- Another useful command is `\pagebreak`, which will start a new page with whatever follows the command.

Miscellaneous Quirks and Tips

- Many symbols you would typically use in a paper actually double as commands within LaTeX.
  - To avoid this, use the backspace, e.g. \% to produce %, or \$ to produce $.
- A frequent mistake involves quotation marks. In order to produce “, you actually need to type ‘ ‘ rather than “, which will result in ”.
- To place a footnote, just use `\footnote{}` and place the text of the footnote within the brackets.
- Another useful command is `\pagebreak`, which will start a new page with whatever follows the command.

Miscellaneous Quirks and Tips

- Many symbols you would typically use in a paper actually double as commands within LATEX.
    - To avoid this, use the backspace, e.g. \% to produce %, or \$ to produce $.
- A frequent mistake involves quotation marks. In order to produce “, you actually need to type ‘ ‘ rather than “, which will result in ”.
- To place a footnote, just use `\footnote{}` and place the text of the footnote within the brackets.
- Another useful command is `\pagebreak`, which will start a new page with whatever follows the command.

Miscellaneous Quirks and Tips

- Many symbols you would typically use in a paper actually double as commands within LaTeX.
  - To avoid this, use the backspace, e.g. \% to produce %, or \$ to produce $.

- A frequent mistake involves quotation marks. In order to produce “, you actually need to type ‘ ‘ rather than “, which will result in ” .

- To place a footnote, just use `\footnote{}` and place the text of the footnote within the brackets.

- Another useful command is `\pagebreak`, which will start a new page with whatever follows the command.

Figures

- The basic syntax to place a figure within your document is:

```
\begin{figure}
\centering
   \includegraphics{Name of File}
        \caption{} % Title that will be displayed
        \label{} %Internal reference (see below)
\end{figure}
```

- You can either keep the figure in the same folder as the tex document, or you can specify the file path to where it is saved.

Figures

- The basic syntax to place a figure within your document is:

```
\begin{figure}
\centering
   \includegraphics{Name of File}
        \caption{} % Title that will be displayed
        \label{} %Internal reference (see below)
\end{figure}
```

- You can either keep the figure in the same folder as the tex document, or you can specify the file path to where it is saved.

Subfigures

- At times it may be useful to display 2 or more images side by side in a single figure. The subfigure package allows for this.

```
\begin{figure}
\centering
\caption{Name of Figure}
   \subfigure[Name of First Image]
        {\includegraphics{file1}
        \label{}}
   \subfigure[Name of Second Image]
        {\includegraphics{file2}
        \label{}}
\end{figure}
```

Subfigures

- At times it may be useful to display 2 or more images side by side in a single figure. The subfigure package allows for this.

```
\begin{figure}
\centering
\caption{Name of Figure}
   \subfigure[Name of First Image]
        {\includegraphics{file1}
        \label{}}
   \subfigure[Name of Second Image]
        {\includegraphics{file2}
        \label{}}
\end{figure}
```

Tables

- Like figures, tables are made within a table environment, which lets you set a position, caption and label.
- In contrast, you also need to use the `tabular` environment to actually construct the table.
    - The number of columns are defined in this environment/ `\begin{tabular}{l c r}`, for example, would create an environment with 3 columns, left, center and right aligned respectively.
- Individual cells are separated by a &.
- \\separates rows.

Tables

- Like figures, tables are made within a table environment, which lets you set a position, caption and label.
- In contrast, you also need to use the `tabular` environment to actually construct the table.
  - The number of columns are defined in this environment/ `\begin{tabular}{l c r}`, for example, would create an environment with 3 columns, left, center and right aligned respectively.
- Individual cells are separated by a &.
- \\separates rows.

Tables

- Like figures, tables are made within a table environment, which lets you set a position, caption and label.
- In contrast, you also need to use the `tabular` environment to actually construct the table.
  - The number of columns are defined in this environment/ `\begin{tabular}{l c r}`, for example, would create an environment with 3 columns, left, center and right aligned respectively.
- Individual cells are separated by a &.
- \\separates rows.

Tables

- Like figures, tables are made within a table environment, which lets you set a position, caption and label.
- In contrast, you also need to use the `tabular` environment to actually construct the table.
  - The number of columns are defined in this environment/ `\begin{tabular}{l c r}`, for example, would create an environment with 3 columns, left, center and right aligned respectively.
- Individual cells are separated by a &.
- \\separates rows.

Table Example

Example: Expected World Cup 2010 Group D Points Table

```
\begin{table}[!p]     %Set up table environment
\centering
\caption{World Cup 2010 Group D Results} %Caption/title
\begin{tabular}{lcccc|cc|c}\hline     %Columns, alignment, vertical lines
Team & P & W & D & L & GF & GA & Points\\ \hline    % Data separated by &
Australia & 3 & 2 & 1 & 0 & 5 & 1 & 7 \\
Serbia & 3 & 1 & 2 & 0 & 4 & 1 & 5 \\
Germany & 3 & 0 & 2 & 1 & 4 & 6 & 2 \\
Ghana & 3 & 0 & 1 & 2 & 4 & 6 & 1 \\ \hline
\multicolumn{8}{l}{Predictions are \emph{not} realistic.}\\
%Merges columns
\end{tabular}
\end{table}
```

Tables

Example: Expected World Cup 2010 Group D Points Table

Table: World Cup 2010 Group D Results

Team	P	W	D	L	GF	GA	Points
Australia	3	2	1	0	5	1	7
Serbia	3	1	2	0	4	1	5
Germany	3	0	2	1	4	6	2
Ghana	3	0	1	2	4	6	1

Predictions are *not* realistic.

Getting Tables Out of Stata & R

```
eststo: reg y x1 x2 x3
eststo: reg y x1 x2 x3 x4
esttab using filename.tex, se r2 b(3) se(3)
```

- This code (install with `findit estout` in STATA) estimates two models, stores the results and exports them in table format to a Tex file called filename.tex, which is saved in your working directory.
- esttab command specifies the table should include standard errors (se), R-squared values (r2), and report betas and standard errors with 3 decimal points.
- In R, use the package and command `apsrtable(model object)` to produce pastable LaTex code.

Getting Tables Out of Stata & R

```
eststo: reg y x1 x2 x3
eststo: reg y x1 x2 x3 x4
esttab using filename.tex, se r2 b(3) se(3)
```

- This code (install with `findit estout` in STATA) estimates two models, stores the results and exports them in table format to a Tex file called filename.tex, which is saved in your working directory.
- esttab command specifies the table should include standard errors (se), R-squared values (r2), and report betas and standard errors with 3 decimal points.
- In R, use the package and command `apsrtable(model object)` to produce pastable LaTex code.

Placement of Floats

- The location of figures and tables in the final document roughly corresponds to the commands in the tex *document*.

- There are 4 options to better control this: t, b, h, p; which mean top, bottom, here and (separate) page respectively.
  - These options are used in conjunction with begin: `\begin{figure}[h]` or `\begin{table}[t]`.
  - You can also use ! to emphasize to LaTex that you want the float to appear at the specified location: `\begin{}[h!]`.

- Despite these options, at times you may need to experiment with where the float commands appear in the text to get the final location the way you want.

Placement of Floats

- The location of figures and tables in the final document roughly corresponds to the commands in the tex *document*.

- There are 4 options to better control this: t, b, h, p; which mean top, bottom, here and (separate) page respectively.
  - These options are used in conjunction with begin: `\begin{figure}[h]` or `\begin{table}[t]`.
  - You can also use `!` to emphasize to LaTex that you want the float to appear at the specified location: `\begin{}[h!]`.

- Despite these options, at times you may need to experiment with where the float commands appear in the text to get the final location the way you want.

Placement of Floats

- The location of figures and tables in the final document roughly corresponds to the commands in the tex *document*.
- There are 4 options to better control this: t, b, h, p; which mean top, bottom, here and (separate) page respectively.
  - These options are used in conjunction with begin: `\begin{figure}[h]` or `\begin{table}[t]`.
  - You can also use `!` to emphasize to LaTex that you want the float to appear at the specified location: `\begin{}[h!]`.
- Despite these options, at times you may need to experiment with where the float commands appear in the text to get the final location the way you want.

.bib files

- Bibliographic information used in LaTeX documents is stored in .bib files.
- An entry in a .bib file looks like this:

```
@article{GrossmanHelpman94,
Author = {Gene M. Grossman and Elhanan E. Helpman},
Journal = {American Economic Review},
Number = {4},
Pages = {833-850},
Title = {Protection for Sale},
Volume = {84},
Year = {1994},}
```

- Slightly different formats are used for different types of works.
- Mistakes are easy to make when entering information, so we recommend using a front end program such as BibDesk (Mac) or JabRef (Windows).

.bib files

- Bibliographic information used in LaTeX documents is stored in .bib files.
- An entry in a .bib file looks like this:

```
@article{GrossmanHelpman94,
Author = {Gene M. Grossman and Elhanan E. Helpman},
Journal = {American Economic Review},
Number = {4},
Pages = {833-850},
Title = {Protection for Sale},
Volume = {84},
Year = {1994},}
```

- Slightly different formats are used for different types of works.
- Mistakes are easy to make when entering information, so we recommend using a front end program such as BibDesk (Mac) or JabRef (Windows).

Using .bib files

- You will need to load bibliography style and citation packages in the preamble.
  - e.g. `usepackage{cite}`
  - e.g. `usepackage{harvard}`
- In the main body of the .tex document you specify the particular citation styles.
  - e.g. `\bibliographystyle{apsr}`
- To create your references list, you call the .bib file you wish to use in the location where the references will appear.
  - e.g. `\bibliography{DissertationReferences}`
- Remember to make sure your .bib file, .tex file *and* .sty file (for special styles like apsr) are in the same folder!

Using .bib files

- You will need to load bibliography style and citation packages in the preamble.
  - e.g. `usepackage{cite}`
  - e.g. `usepackage{harvard}`
- In the main body of the .tex document you specify the particular citation styles.
  - e.g. `\bibliographystyle{apsr}`
- To create your references list, you call the .bib file you wish to use in the location where the references will appear.
  - e.g. `\bibliography{DissertationReferences}`
- Remember to make sure your .bib file, .tex file *and* .sty file (for special styles like apsr) are in the same folder!

Using .bib files

- You will need to load bibliography style and citation packages in the preamble.
  - e.g. `usepackage{cite}`
  - e.g. `usepackage{harvard}`
- In the main body of the .tex document you specify the particular citation styles.
  - e.g. `\bibliographystyle{apsr}`
- To create your references list, you call the .bib file you wish to use in the location where the references will appear.
  - e.g. `\bibliography{DissertationReferences}`
- Remember to make sure your .bib file, .tex file *and* .sty file (for special styles like apsr) are in the same folder!

Using .bib files

- You will need to load bibliography style and citation packages in the preamble.
  - e.g. `usepackage{cite}`
  - e.g. `usepackage{harvard}`
- In the main body of the .tex document you specify the particular citation styles.
  - e.g. `\bibliographystyle{apsr}`
- To create your references list, you call the .bib file you wish to use in the location where the references will appear.
  - e.g. `\bibliography{DissertationReferences}`
- Remember to make sure your .bib file, .tex file *and* .sty file (for special styles like apsr) are in the same folder!

Citation formats

- Citation technique depends on which package you use, and each package relies on a unique citation key for each work referenced.
- For the `cite` package, in text citations are as follows:
  - `\cite{key1}`: In text author-year citation.
    - e.g. (Grossman & Helpman 1994)
  - `\cite{key1,key2}`: In text author-year citation for two works.
    - e.g. (Grossman & Helpman 1994, Jones 2012)
  - `\citeyear{key1}`: In text year only citation.
    - e.g. (1994)
  - `\citeaffixed{key1}{comment}`: In text citation with comment.
    - e.g. (comment Grossman & Helpman 1994)
  - `\cite[#]{key1}`: In text author-year citation with page number #.
    - e.g. (Grossman & Helpman 1994, 842)

Citation formats

- Citation technique depends on which package you use, and each package relies on a unique citation key for each work referenced.
- For the `cite` package, in text citations are as follows:
  - `\cite{key1}`: In text author-year citation.
    - e.g. (Grossman & Helpman 1994)
  - `\cite{key1,key2}`: In text author-year citation for two works.
    - e.g. (Grossman & Helpman 1994, Jones 2012)
  - `\citeyear{key1}`: In text year only citation.
    - e.g. (1994)
  - `\citeaffixed{key1}{comment}`: In text citation with comment.
    - e.g. (comment Grossman & Helpman 1994)
  - `\cite[#]{key1}`: In text author-year citation with page number #.
    - e.g. (Grossman & Helpman 1994, 842)

Typesetting with .bib files

- To fully integrate references and citations into your document, you need to typeset your document FOUR times in this order:
  1. LaTeX
  2. BibTeX
  3. LaTeX
  4. LaTeX
- Note: Some LaTeX front-end programs will have options that allow you to typeset all at once.

Beamer

- Beamer is a package that allows you to prepare slides in LaTeX.
  - Enables easy transfer of text, equations, tables from papers to slides.
- Slides are created in a .tex file and typeset into a .pdf file (like this one).
- Use document class: `beamer`

Setting up the Preamble

- In the preamble, you will need to control a few color and style options.

```
\mode<presentation>{
\usetheme{Madrid}           %Style of slides
\usecolortheme{lily}        % Color Scheme
\useinnertheme{rectangles}  % Shape of bullet points
\useoutertheme{default} %Syle of top/bottom of slides
\setbeamerfont{block title}{size={}}
}
```

- More slide and color themes can be found online, e.g. `http://www.hartwork.org/beamer-theme-matrix/`

Setting up the Preamble

- In the preamble, you will need to control a few color and style options.

```
\mode<presentation>{
\usetheme{Madrid}           %Style of slides
\usecolortheme{lily}        % Color Scheme
\useinnertheme{rectangles}  % Shape of bullet points
\useoutertheme{default} %Syle of top/bottom of slides
\setbeamerfont{block title}{size={}}
}
```

- More slide and color themes can be found online, e.g. `http://www.hartwork.org/beamer-theme-matrix/`

Setting up the Preamble

- In the preamble, you will need to control a few color and style options.

```
\mode<presentation>{
\usetheme{Madrid}           %Style of slides
\usecolortheme{lily}        % Color Scheme
\useinnertheme{rectangles}  % Shape of bullet points
\useoutertheme{default} %Syle of top/bottom of slides
\setbeamerfont{block title}{size={}}
}
```

- More slide and color themes can be found online, e.g. `http://www.hartwork.org/beamer-theme-matrix/`

Title Slide Set-up

- This comes before `begin{document}`:

```
\author{Matt Hitt \&\\
  Ben Jones}
\institute{PRISM \\
  Department of Political Science \\
  \textit{ hitt.23@osu.edu}\\
  \textit{jones.2781@osu.edu}}
\title[Introduction to \LaTeX]{PRISM Introduction
  to \LaTeX}
\date{January 20, 2011}
```

Basic Slide Set-up

- For each slide:

  ```
  \begin{frame}
  \frametitle{title}
  (frame contents)
  \end{frame}
  ```

- The frame contents are the same as any other document. You can use regular text, bullet points (itemize environment), tables, figures etc.
- For the title slide, there is no frametitle and frame contents is only: `\titlepage`.

Basic Slide Set-up

- For each slide:

  ```
  \begin{frame}
  \frametitle{title}
  (frame contents)
  \end{frame}
  ```

- The frame contents are the same as any other document. You can use regular text, bullet points (itemize environment), tables, figures etc.
- For the title slide, there is no frametitle and frame contents is only: `\titlepage`.

Basic Slide Set-up

- For each slide:

```
\begin{frame}
\frametitle{title}
(frame contents)
\end{frame}
```

- The frame contents are the same as any other document. You can use regular text, bullet points (itemize environment), tables, figures etc.
- For the title slide, there is no frametitle and frame contents is only: `\titlepage`.