I. Introduction to LaTeX

LaTeX for Economists
January 31, 2018

Niclas Frederic Poitiers
n.f.poitiers@ub.edu

UNIVERSITAT DE BARCELONA
SCHOOL OF ECONOMICS

LaTeX for Economist

This course will consist of three sessions:

I. Introduction to LaTeX

II. Figures, Tables, and References

III. Presentations and Graphics

At the end of this course participants will be able to create scientific papers and presentations using LaTeX.

Class Objective

In this course we want to learn:

- What LaTeX is.
- How to write LaTeX code.
- How to create and compile a LaTeX document.
- How to structure a LaTeX document.
- How to formate text in LaTeX.
- How to create mathematical formulas in LaTeX.

Outline

What is LaTeX

LaTeX is a descriptive programming language that allows to create PDF text documents. Advantages of using LaTeX are:

- Professional font setting.
- System Independence.
- Easy management of large projects.
- Comprehensive support of mathematical notation.
- Reference management.
- Git support.
- Industry standard.

WYSIWYM

WYSIWYM (What you see is what you mean) vs. WYSIWYG (What you see is what you get).

Outline

Basics of the Language

We create a LaTeX document by creating a text file with the extension `.tex`. The document itself contains the code, which can then be converted by the LaTeX compiler into a PDF.

The basic syntax of the language is the following:

- A command in LaTeX always starts with a `\`
- We put command argument in `{...}` and options in `[...]`
- We can create an environment using

```
\begin{...}
  ...
\end{...}
```

- We can start a comment by using `%`

Text Editors and Compilers

There are a variety of editors that assist you in the creation of LaTeX documents, e. g. by highlighting the syntax, but in principle you can use every text file editor for this.

Special LaTeX editors:

- Cross platform: TeXmaker (free), TeXstudio (free), TeXworks (free).
- Windows: LEd (free), TeXnicCenter (free), WinEdt (free), WinShell (free).
- Mac: TeXShop (free), texpad, TeXnicle (free), Archimedes.
- Linux: LaTeXilla (free), Kile (free)

General purpose editors supporting LaTeX: Atom (free), Sublime, Emacs (free), Vim (free).

Compilers

The `.tex` file is *compiled* into a `.pdf` file using compilers. Different operations in the file need the execution of different compilers (e. g. bibtex for bibliographies), and sometimes subsequent executions of different compilers.

The main compiler that you should use always first and last for "normal" LaTeX documents is *pdflatex*. Each compiler creates intermediate files, which you can ignore for most purposes.

Structure of the Document

A LaTeX file is structured as follows:

- Specification of the document type

```
\documentclass[a4paper, 12pt]{article}
```

- The *preamble*, where we specify document properties and packages.
- The content of the document in an environment of the form

```
\begin{document}
  ...
\end{document}
```

The Preamble

The preamble is used for defining fonts, document and font properties, and for loading packages for additional commands.

- We can load a package by using `\usepackage{...}`.
- We can define the title, author, and date of the document by using

```
\title{...}
\author{...}
\date{...}
```

  if the date is undefined, it will use the date of the last compilation.

Special Text Characters

Special characters that are reserved for the syntax have to be written as commands:

\	`$\backslash$`
{}	`$\{ \}$`
&	`\&`
%	`\%`
$	`\$`

You can create opening quotation marks by using two grave accents and a closing quotation mark using two apostrophes.

```
``...''
```

A single `-` creates a hyphen, a double `--` a dash.

Language Support

LaTeX compiles the document using syllables. Therefore – and in order to specify special characters – you should specify languages in the preamble using the package `babel` and specify the language in the options.

```
\usepackage[Spanish]{babel}
```

To make the usage of special characters easier, you should specify the input encoding to UTF-8 using the package `inputenc`

```
\usepackage[utf8]{inputenc}
```

Debugging

When there is a mistake in the syntax of your `.tex` document, it will not compile. The most common mistakes are:

- Open arguments (not closed `{...}` brackets).
- Open environments (missing `\begin{...}` or `\end{...}`).
- Misspelled commands.
- Wrong options (e. g. in tables).

Outline

Title

We can create a title with author and date by inserting the command `\maketitle`.

If you want to add a "thank you"-note in a footnote to the title (or to the author), use `\thanks{...}` when defining the title or the author in the preamble.

If you want to suppress the date, put `\date{}` in the preamble.

Sections

Articles are structured using *sections*.

- We create a new section by using `\section{<section title>}`.
- We can create sub- and subsubsections by using `\subsection{...}` and `\subsubsection{...}`.

A new (sub-) section is started with the title of the section.

Table of Contents

Every section and subsection is numbered. We can create a table of content by using `\tableofcontent`.

If you want to have a separate name for a section in the table of content, you can specify it in the options of `section`

```
\section[<title in toc>]{<section title>}
```

We can create a non-indexed section that does not appear in the table of contents by using `\section*{...}`.

Paragraphs, Abstract, and Appendix

If you want a special (named) paragraph, you can use `\paragraph{...}` and `\subparagraph{...}`. The argument of `paragraph` will appear as "title" of the paragraph.

You can create an abstract in the beginning of the document using the `abstract` environment:

```
\begin{abstract}
  ...
\end{abstract}
```

With `\appendix` you start the appendix section of the article. Every section after `\appendix` is numbered with capital letters.

Lists

To create a list use the environment `itemize`, and `\item` for the single items. To create a numerated list, use the `enumerate` environment. You can specify the numbering in the options.

```
\begin{itemize}
  \item Point 1
  \item Point 2
\end{itemize}
```

- Point 1
- Point 2

```
\begin{enumerate}
  \item Point 1
  \item Point 2
\end{enumerate}
```

1. Point 1
2. Point 2

Page Numbering

You can specify the page numbering using the command `\pagenumbering{...}`. You have the options:

- `Roman` capital roman numbers.
- `roman` small roman numbers.
- `arabic` arabic numbers.

You can change the total number counter using `\setcounter{page}{...}`.

Footers and Headers

You can change the "style" of all (following) pages using `\pagestyle{}`, and of only the current page using `\thispagestyle{}`. By default (`plain`), an article has no header and in the footer only the current page number.

- `empty` creates a page without header or footer.
- `headings` creates a page with all information in the header.

Fancyheader

The package `fancyhdr` allows to create custom headers and footers for the pagestyle `fancy`:

```
\lhead[<even left header>]{<odd left header>}
\chead[<even center header>]{<odd center header>}
\rhead[<even right header>]{<odd right header>}
\lfoot[<even left footer>]{<odd left footer>}
\cfoot[<even center footer>]{<odd center footer>}
\rfoot[<even right footer>]{<odd right footer>}
```

Details

You can summon details of the document using the following commands (e. g. for specifying the header/footer):

- `\thepage` current page number.
- `\numpage` total number of pages.
- `\thesection` current section number.
- `\thesubsection` current subsection number.

Margins and Landscape

The package `geometry` allows you to change the margins of the page:

```
\usepackage[top=1in, bottom=1.25in,
            left=1.25in, right=1.25in]{geometry}
```

You can set the whole article in landscape mode by setting the option `landscape` to either `documentclass` or `geometry`. Single landscape pages can be created using the package `pdflscape` and the `landscape` environment.

Importing from other Files

Often, it makes sense to structure large projects within different files. We can import code from another `.tex`-file by using `\input{...}`, where the argument is the path to the file. The code in the file is treated as if it where at the exact place of the command.

Outline

Blanks

The compiler automatically adjusts the white space in the text, double blanks are ignored. If you want to create an additional space you can use `\ `.

LaTeX distinguishes between different types of spaces:

- `\ ` creates a "normal" space, as between words.
- `\quad` `\qquad` create large and very large spaces.
- `\,` creates a small space, e. g. within abbreviations: `e.\,g.`.

After a dot lets LaTeX think that a sentence ends, after which it places a double space. Thus you should use after abbreviations that end on a dot a `\ ` to cause a "normal" space: `e.\,g.\ `.

Line Breaks

A single line break in the code does not cause a line-break in the output. You can create line breaks by using `\\` or by using double line breaks. If you want an extra large line break, you can specify that in the options of `\\`, e. g. `\\[12pt]`.

You can start a new page by using `\pagebreak`.

If you want to keep two words in one line, use `~` instead of a space between the words. You can also use `\mbox{...}` for that.

Paragraph, Indent, and Break

A normal line break causes the text to start further to the right. You can suppress that by using `\noindent` at the start of the new line.

You can specify the length of the indent of new paragraphs using

```
\setlength{\parindent}{<length>}
```

You can specify space after a paragraph using (this also affects lists, equations etc.)

```
\setlength{\parskip}{<space after paragraph>
          plus<variance> minus<variance>}
```

To change the style to just space after a paragraph without any indent you can use the package `parskip`.

Bold and Italic

In order to *italicise* rest of the text or the text in an environment you can write `\it`, to put the rest of the text or the text in an environment **bold** you can write `\bf`. You can also create a bold or italic test environment using

```
\textit{
   ...
}
\textbf{
   ...
}
```

Use `\sc` and `\textsc{}` in the same way for Small Caps. If you want to emphasise text dynamically use `\emph{...}`.

Text Size

The text size is specified at the beginning of the document as an option of the `documentclass`. Allowed as text-sizes are 10pt, 11pt and 12pt. In order to change the size of the text, you can use the following commands

`\tiny`	`\large`
`\scriptsize`	`\Large`
`\footnotesize`	`\LARGE`
`\small`	`\huge`
`\normalsize`	`\HUGE`

In order to only change some part of text you can use `{...}`

```
{\Large Large text} normal text
```

Orientation

By default, text in LaTeX is fully-justified. If you want to centre the text you can use `\centering` within an environment, or the environment

```
\begin{center}
  ...
\end{center}
```

The same way, you can also have left- or right-justified text using `flushleft` and `flushright` as commands or environments.

Footnotes, Superscripts and Subscripts

You can create a footnote in the text using the command

```
\footnote{<text of the footnote>}
```

It will be automatically aligned and numbered.

With `\textsubscript{...}` and `\textsuperscript{...}` you can create subscripts and superscripts, e. g. the 15$^{\text{th}}$ century.

Ligatures

LaTeX creates ligatures, i. e. it adjust the space between some characters such that they overlap. This is not always wanted (e. g. in composite words) and can be suppressed using `{}` between two characters.

fire flower fjord shelfful *figure*
fire flower fjord shelf{}ful *figure*

Fonts

LaTeX specifies four types of fonts:

- `\normalfont` font type of the main text.
- `\rmdefault` serif font.
- `\sfdefault` sans-serif font.
- `\ttdefault` typewriter font.

By default, `normalfont` is set to be `rmdefault`. You can change to the sans-serif font using

```
\renewcommand{\familydefault}{\sfdefault}
```

If you want to change one of the fonts, use e. g. for the serif font:

```
\renewcommand*\rmdefault{<fontname>}
```

Fonts

The standard font family is *Computer Modern*, but more are installed by default, including

- latin modern: `lmr`, `lmss` & `lmtt` (to change all fonts use the package `lmodern`).
- times: `ptm`
- helvetica: `phv`

For some fonts you need to change the encoding to `T1` before specifying the font. You can do this by using the package `fontenc`:

```
\usepackage[T1]{fontenc}
\usepackage{lmodern}
```

A variety of different fonts are available at `http://www.tug.dk/FontCatalogue/`

Outline

Mathematical Formula-rs

One of the main reasons to use LaTeX is its comprehensive abilities to create mathematical formulas.

In order to create mathematical formulas, you need to load either the `amsmath` package or the `mathtools` package.

For an extensive list of operators and symbols, go to `https://en.wikibooks.org/wiki/LaTeX/Mathematics`

Align Environment

The main environment for maths is `align`. It creates aligned multi-row mathematical formulas. It aligns the formulas left and right of the `&` symbol:

```
\begin{align}
  1 + 2 &= 3\\
  2 + 3 &= 5
\end{align}
```

$$1 + 2 = 4 - 1 \tag{1}$$

$$1 + 2 + 3 = 6 \tag{2}$$

You can suppress row-numbering by an asterisk: `\begin{align*}`.

Alignat Environment

The `alignat` environment, which doesn't add space between "equation columns", allows you to add comments to formulas. You need as argument the number of "equation columns": count the maximum number of `&` in any row, add 1 and divide by 2.

```
\begin{alignat}{2}
  x &= y_1-y_2+y_3-\dots &\quad& \text{Axiom 3}\\
  &= \sum_{i=0}^n (-1)^i y_{i+1}&& \text{by Axiom 4}
\end{alignat}
```

$$\begin{aligned} x &= y_1 - y_2 + y_3 - y_5 + y_8 - \dots && \text{by Axiom 3} && (3)\\ &= \sum_{i=0}^{n} (-1)^i y_{i+1} && \text{by Axiom 4} && (4) \end{aligned}$$

Maths in Text

In order to put formulas within text, you can use the `$` sign. Everything between two dollar signs is considered maths text. This allows you to incorporate maths symbols (e. g. Greek letters) into text.

Functions

Text in the maths environment is set italic. Some functions have their own command:

`\sum`	$\sum$
`\prod`	$\prod$
`\sqrt[n]{x}`	$\sqrt[n]{x}$
`\log`	log
`\max`	max
`\int`	$\int$
`\lim`	lim
`\sin`	sin
`\cos`	cos
`\tan`	tan

Operators

There is also an extensive number of operators available in the maths environment:

`\leq`	$\leq$	`\ast`	$*$
`\geq`	$\geq$	`\times`	$\times$
`\equiv`	$\equiv$	`otimes`	$\otimes$
`\in`	$\in$	`\pm`	$\pm$
`\notin`	$\notin$	`\wedge`	$\wedge$
`\sim`	$\sim$	`\vee`	$\vee$
`\simeq`	$\simeq$	`\Rightarrow`	$\Rightarrow$
`\cap`	$\cap$	`\Leftarrow`	$\Leftarrow$
`\cup`	$\cup$	`\Leftrightarrow`	$\Leftrightarrow$
`\cdot`	$\cdot$	`\infty`	∞
`\dots`	$\dots$	`\forall`	$\forall$
`\%`	%	`\exists`	$\exists$

Indices and Powers

You can take the power and set upper indices using `^`, the first character after the symbol is take as power. If you want to have multiple characters in the upper index, use brackets `^{...}`.

You can set an lower index using `_`, the first character after the symbol is take as lower index. If you want to have multiple characters in the lower index, use brackets `_{...}`.

Brackets

You can create brackets in latex maths environment using

- `(...)` for (...)
- `[...]` for [...]
- `\{...\}` for $\{...\}$
- `|...|` for $|...|$
- `\|...\|` for $\|....\|$

In case you need larger brackets, e. g. because you use some special character, use `\left<specify bracket>` and `\right<specify bracket>`. This must always be closed (it can be closed without prompting a bracket using a point).

Accents

You can use the following accents in order to distinguish variables:

`a'`	a'	`\hat{a}`	$\hat{a}$
`\bar{a}`	$\bar{a}$	`\grave{a}`	$\grave{a}$
`\acute{a}`	$\acute{a}$	`\dot{a}`	$\dot{a}$
`\vec{a}`	$\vec{a}$	`\overrightarrow{AB}`	$\overrightarrow{AB}$
`\tilde{a}`	$\tilde{a}$	`\overleftarrow{AB}`	$\overleftarrow{AB}$
`\overline{aaa}`	$\overline{aaa}$	`\widehat{AAA}`	$\widehat{AAA}$
`\underline{a}`	$\underline{a}$	`\widetilde{AAA}`	$\widetilde{AAA}$

Fractions and Binomials

You can generate a fraction using the command `\frac{<numerator>}{<denominator>}`. You can generate nested fractions using multiple fractions. Similarly, you can create binomial expressions: `\binom{n}{k}`

```
\frac{n!}{k!(n-k)!} = \binom{n}{k}
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

Text in Maths

When you want to write text in maths, you can use the command

- `\text{...}` text in maths font: A
- `\mathsf{...}` text in sans-serif font: A
- `\mathrm{...}` text in serif font: A
- `\mathit{...}` test in italic maths font: A
- `\mathbf{...}` text in bold maths font (e. g. vectors): $\mathbf{A}$
- `\mathbb{...}` text in blackboard bold (e. g. expectation symbol): $\mathbb{A}$
- `\mathcal{...}` text in calligraphy: $\mathcal{A}$
- `\mathfrak{...}` text in fracture: $\mathfrak{A}$

Greek Characters and Symbols

LaTeX supports the Greek alphabet in maths mode: just write the name of the Greek letter as command (with small initial letter for small Greek letters and capital initial letter for capital Greek letters). E. g. write `\delta` for δ and `\Delta` for Δ.
There are some extra Greek characters:

- Use `\varepsilon` to get a statistical error symbol: ε.
- Use `\partial` to get a delta for partial derivatives: ∂.

For generating bold Greek symbols use `\boldsymbol`.

Currency Symbols

The dollar symbol can be generated using `\$`, the pound symbol by using `\pounds`.

In order to generate euro symbol use the package `eurosym` which include the command symbol `\euro{}` and `\EUR{}` (which has different spacing. In order to use those symbols in maths mode, you need to use `\text{\euro{}}`.

Matrices

You can generate matrices using the `matrix` environment.

```
\begin{matrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{matrix}
```

In order to get brackets around the matrix, use

- `pmatirx` for $(\ldots)$
- `bmatirx` for $[\ldots]$
- `Bmatirx` for $\{\ldots\}$
- `vmatirx` for $|\ldots|$
- `Vmatirx` for $\|\ldots\|$

Cases

You can create a case distinction using

```
y = \begin{cases}
a & \text{if}\ x<1\\
a + 2 & \text{if}\ x\leq1
\end{cases}
```

$$y = \begin{cases} a & \text{if } x < 1 \\ a + 2 & \text{if } x \leq 1 \end{cases}$$

Comment Expressions

`overbrace{...}_{...}` and `\underbrace{...}_{...}` allows you to put comments under and over a mathematical term. `mathclap{...}` (for which you need the `mathtools` package) helps if the text is longer than the term.

```
Y = f(\underbrace{x}_{\mathclap{\text{Long text}}})
```

$$Y = f(\underbrace{x}_{\mathclap{\text{Long text}}}) \tag{5}$$

Overset and Underset

If you want to type expressions above or below expressions, use `\overset{}{}` and `\underset{}{}`:

```
A \overset{!}{=} B
```

$$A \overset{!}{=} B$$

The package `mathtools` provides you with a number of different arrows extending to the length of a subscript,

```
A \xleftarrow{\text{this way}} B
  \xrightarrow[\text{or that way}]{ } C
```

$$A \xleftarrow{\text{this way}} B \xrightarrow[\text{or that way}]{} C$$

Labels

You can reference numbered equations easily using `\label{...}` and `ref{...}`. If you put `\label{<name>}` in a line of an equation, the corresponding number of the equation is linked to the name. You can prompt the number in the text using `\ref{<name>}`.

If you want only to index one line of a multi-line equation, use `\nonumber` in the other lines.

Generating New Commands

You can generate new commands using

```
\newcommand{<command>}[<num. of arg.>]{<def.>}
```

You can insert an argument using `#<number of the argument>`. `operatorname{...}` allows you to easily create new functions:

```
\newcommand{\var}[1]{\operatorname{var}(#1)}
```

creates the operator `\var{x}` which prompts now var(x).

Writing your own Packages

When often reuse the same preamble with the same packages, custom commands etc., it can make sense to save them in your own package. You can do that by putting the packages, commands, etc. in a `.sty` file, which you than can import using `\usepackage{}`.