

MSC MODULE RESEARCH SKILLS

SCIENTIFIC WRITING WITH LATEX

These lecture notes are part of the course material for the MSc module Research Skills (website [here](#)). First edition 2015, this version was created March 12, 2018.

[Florian Hartig](#)

University of Regensburg Germany

If you find typos or have suggestions, please report them by creating a ticket [here](#)

These notes were created 2014 by Paul Bauche based on the existing RS LaTeX exercise of FH. Updates 2015 by Severin Hauenstein and Florian Hartig. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

Contents

1	<i>Introduction</i>	5
2	<i>Setting up \LaTeX on your computer</i>	6
3	<i>Starting a new \LaTeX project</i>	7
3.1	<i>Creating a new directory for the project</i>	7
3.2	<i>Preparing your editor</i>	7
3.3	<i>Creating a \LaTeX document</i>	7
4	<i>Start editing your document</i>	9
5	<i>The preamble</i>	12
6	<i>Equations in \LaTeX</i>	13
7	<i>Labels and referencing</i>	16
8	<i>Including figures</i>	18
9	<i>Tables</i>	21

10	<i>References in \LaTeX</i>	25
11	<i>Going further</i>	30
	<i>Appendices</i>	31
A	<i>FAQ</i>	32
B	<i>Useful packages</i>	34
B.1	<i>Encoding - inputenc</i>	34
B.2	<i>Typesetting - fontenc</i>	34
B.3	<i>Language specific labels and hyphenation - babel</i>	34
B.4	<i>Including pdfs in your text - pdfpages</i>	34
B.5	<i>Proper appendices - the appendix package</i>	35
B.6	<i>For Formulae: Amsmath, amsfons and amssymb</i>	35
B.7	<i>The color package</i>	35
B.8	<i>The xcolor package</i>	35
B.9	<i>Underline - the ulem package</i>	35
B.10	<i>Spacing with setspace and linespread</i>	36
B.11	<i>Hyphenation</i>	36
B.12	<i>The preprint bundle</i>	36
B.13	<i>figcaps</i>	36

1

Introduction

This tutorial will help you start working with \LaTeX . You will learn how to set up a basic \LaTeX document which contains all the elements typically used in a scientific article, for example references, graphics, tables or formulae.

\LaTeX is a script-based typesetting program which allows you to set the design of your report, article, book or whatever written statement via written commands. This may appear a bit cryptic in the beginning, but after a short time of getting used to this way of working you will realize how much time you can save and how much nicer and more professional your documents look.

Basically, when it comes to typesetting software there are two distinguishable types. The first and most common is called "*What You See Is What You Get*" (WYSIWYG). This describes software such as OpenOffice Writer, Microsoft Word etc. where you can see how your document is going to look in the end while you are working on it, but it might be difficult to create a document exactly the way you wanted it to be. The second type is called "*What You Get Is What You Want*" (WYGIWYW). This is the category \LaTeX belongs to. It means that while you are working on your document you do not get a visual representation of the final result, but your final result will be the way you imagined it when you started.

Templates Since the basic structure of \LaTeX documents of the same class is similar in most cases, you may save some time by using existing templates. Prof. Dormann created such a template for a Master's thesis. It is available [here](#).

Other templates for the Tufte-style book (book_1), the Legrand Orange book (book_2) or some stylish article (article_3) can be found [here](#).

The \LaTeX document we use for the \LaTeX practical of this course can also be used as template for a simple article.

2

Setting up \LaTeX on your computer

First you need to set up \LaTeX on your computer. Basically, a \LaTeX system consists of two components: a \LaTeX distribution, and an editor.

Distribution - There are different distributions, also depending on your operating system. For Windows, the clear choice is MikTeX. For Mac, MacTeX is probably the best choice, but there are other options as well. The \LaTeX distribution under Linux is called TeX Live and can be installed through the package managers of all major distributions.

Editor - The second thing you need is an editor. In principle you can use any editor (\LaTeX documents are just text files), but it is more convenient to use a specialised \LaTeX editor that is already set up to work together with your \LaTeX distribution. We recommend using one of the free \LaTeX editors TexMaker or TeXstudio, and some of the comments later in this document will assume that you will be working with a MikTeX / TexMaker or TeXstudio combination (Most of the advice, however, as well as any proper \LaTeX document, can be used on any system).

To get MikTeX and TexMaker or TeXstudio, search for it on the web, then download and install it on your computer. Make sure you install from the project website and not from third-party sites that often offer pretty old versions.

Before you start your project you have to keep a couple of things in mind that will later on help you to keep track of what you are doing and to avoid unnecessary errors.

If you use MikTeX, it is helpful to be connected to the internet in some way to enable \LaTeX to download packages on the fly. For other distributions, you will have to install missing packages by hand.

Texmaker and TeXstudio are very similar (TeXstudio originated from a previous version of Texmaker). TeXstudio is a bit more automatized and offers more buttons.

If you install MacTeX, choose the full installation because missing packages are not installed automatically.

3

Starting a new L^AT_EX project

3.1 Creating a new directory for the project

First of all, create a folder on your computer and make sure that you have the necessary rights to read and write in that location. Do not simply start your project on the desktop since L^AT_EX creates additional files as you compile your project and you will make a mess of things.

When creating the folder and all following files make sure you do not use spaces or special symbols. Nowhere in the path to your files should be any of those symbols. Stick to letters, numbers and underscores.

Always create a new project folder with a distinct name when starting a L^AT_EX document!

No spaces or special symbols in file or folder names anywhere in your file path!

3.2 Preparing your editor

In TexMaker you have several options to configure the user interface. The most important ones shall be explained briefly.

It is advisable to set the spell check to the language you want to write your article in. You can do that under "Options -> Configure TexMaker -> Editor -> Spelling Dictionary". The most common languages are already part of your software and you can just choose the one you need.

Here you can also choose the text encoding of your editor. It is recommended to use UTF-8.

Under "Options -> Interface Language" you can choose the language for TexMaker. This does not affect the spell check but is solely for your convenience.

3.3 Creating a L^AT_EX document

Now that you have created your project folder, you can create your first L^AT_EX document. Navigate to your project folder and create a new text file. Open this file and type the following line:

% This is my first Latex document. There are many documents like it, but this one is mine.

Make sure you do not forget the percent symbol %. Then save your document as `my_first_article.tex` and close the editor. With the extension `.tex` you make sure that your system recognizes the file as a \LaTeX document. Now open `my_first_article.tex` with your \LaTeX editor (e.g. Texmaker). You will notice that the line you wrote now looks gray.

The % symbol is used for comments, lines of text that are not interpreted by \LaTeX !

```
% This is my first Latex document. There are many documents like it , but
this one is mine.
```

The gray color symbolizes a *comment*. The comment is declared by using the % symbol at the start of the line. Everything written after the % is not interpreted by \LaTeX and does not appear in your document. Comments can be used to structure your document for yourself and others using it. You can explain functions and why you include packages. This is especially useful if you want to reuse the structure of our document again for other papers.

4

Start editing your document

To start your actual document, you first need to declare the type of document you want to create. This is done in the so called "preamble" of the \LaTeX document. There you can also declare additional options and include packages to increase the functionality of \LaTeX .

For the purpose of this tutorial you are going to create a scientific article. This type of document is declared by `\documentclass{article}`. This defines a few design characteristics such as line spacing and the width of the borders of the page for your document. The backslash `\` is used to indicate to \LaTeX that the following is a command. In \LaTeX commands are used for everything from setting your font type to including pictures or graphics.

Now you can begin with your document. To do so, you start with the line `\begin{document}`. This should result in two lines being created: in addition to `\begin{document}` there should also be a line saying `\end{document}`. If this is not the case, just type it yourself. Now you should have a document looking like this:

Every *begin* statement requires an *end* statement!

```
% This is my first Latex document. There are many documents like it , but
this one is mine.

\documentclass{article}

\begin{document}

\end{document}
```

This is the most basic setup for your document. To start adding some text, you can now write a few lines between the begin- and end-document statements. Since this is supposed to be a scientific article, it should start with an abstract. The abstract is also framed by a begin- and an end-statement. They are `\begin{abstract}` and `\end{abstract}`. For your abstract try:

This is my first \LaTeX document. I am going to learn a lot about typesetting and writing articles in \LaTeX .

Now your document should look like this:

```
% This is my first Latex document. There are many documents like it, but
  this one is mine.

\documentclass{article}

\begin{document}

\begin{abstract}
This is my first \LaTeX\ document. I am going to learn a lot about
typesetting and writing articles in \LaTeX\ and my papers will look
really nice from now on.
\end{abstract}

\end{document}
```

You should now be able to compile your document and create a pdf file from it. To do so, you just have to press compile in your L^AT_EX editor. You should notice the L^AT_EX symbol. This was created by using the \LaTeX command. After the abstract you can start your first section. This is usually the introduction. You can add sections via \section{}. Here no begin- or end-statement is needed. One section lasts until the next one starts or the document ends. To add the section "Introduction" to your document, now type \section{Introduction} below \end{abstract} in your document and write a few lines like

```
This is my \textbf{first} \emph{sentence}.

This is my \textit{second} sentence.

This is my third sentence. You can also put some distance between lines
.\[3mm]
Test
```

Now your document should look like this:

```
% This is my first Latex document. There are many documents like it, but
  this one is mine.

\documentclass{article}

\begin{document}
\begin{abstract}
This is my first \LaTeX\ document. I am going to learn a lot about
typesetting and writing articles in \LaTeX! and my papers will look
really nice from now on.
\end{abstract}

\section{Introduction}

This is my \textbf{first} \emph{sentence}.

This is my \textit{second} sentence.

This is my third sentence. You can also put some distance between lines
.\[3mm]
Test

\end{document}
```

Compile your document to notice the effect of the new commands. \textbf{} leads to **bold** fonts, \textit{} leads to *italic* fonts, and \emph{} *emphasizes* what ever you put into the curly brackets. Ad-

ditionally, you learned two ways to add a line break to your document. The first one is to simply leave a blank line after the point at which you want to break the line. This also leads to an indent at the beginning of the next line, which makes for better readability. The second one is to add `\\`. This method can be refined by adding the exact spacing either in mm, pt or relative to the default line height in square brackets, for example `\\[3mm]`. No indent follows this type of line break.

5

The preamble

As mentioned before, every \LaTeX document must have a preamble. The preamble is everything you write before the `\begin{document}` statement. Here you declare the packages you want to use, define styles, and you can add some metadata to your document, such as the author or the title of your article. These two are added by using the `\author{}` and `\title{}` commands. Add them to your document right below the `\documentclass{}` command, write a title in the curly brackets after `\title` and your name in the curly brackets after `\author` and compile again. There should be no noticeable difference. Now right under the `\begin{document}` statement add `\maketitle` and compile again. Your pdf document should now have a title, your name and the current date at the top. The `\maketitle` command uses the information you provided in the preamble to create a title for your article. If everything worked, your \LaTeX document should now look like this:

In the preamble, you can include new packages and set general parameters and metadata for your document!

```
% This is my first Latex document. There are many documents like it , but
this one is mine.

\documentclass{article}
\author{FirstName LastName}
\title{My first \LaTeX\ document}

\begin{document}
\maketitle

\begin{abstract}
\noindent This is my first \LaTeX\ document. I am going to learn a lot
about typesetting and writing articles in \LaTeX\ and my papers will
look really nice from now on.
\end{abstract}

\section{Introduction}
This is my \textbf{first} \emph{sentence}.

This is my \textit{second} sentence.

This is my third sentence. You can also put some distance between lines
.\allowbreak
Test

\end{document}
```

6

Equations in \LaTeX

Here you will learn how to write equations in \LaTeX . There are two basic types of equations. First, there are inline equations which can be used to write down basic mathematical expressions and equations like $\alpha = 3.5$. or $\lambda = \sqrt{\alpha}$ within your text. Second are the numbered equations. They will appear separated from your text like this:

```
\begin{equation}
\lambda = \int_{m_0}^{\infty} f(\theta) d\theta
\end{equation}
```

They will be numbered (in this case as the first equation in chapter six), which allows you to easily reference the equation later on in your text. You can use this to write down more complex equations or expressions. As an exercise, you can now create those two types within your own document and put them in a new section named "Equations". You will start with an inline equation. An inline equation is surrounded by $\$$ symbols to separate it from the rest of the text. For this purpose, start a new subsection called "Inline Equations". A subsection can be created with the `\subsection{}` command. Your first inline equations will be $\alpha = 3.5$ and $\lambda = \sqrt{\alpha}$. To emphasize the inline effect, put a short sentence before the equation. As you can see, Greek letters can be written by typing a \backslash before the full name of the letter. Additionally the `\sqrt{}` command is used for the square root symbol. Include this section in your document:

```
\subsection{Inline equations}
This is our first inline equation  $\alpha = 3.5$ .  $\lambda = \sqrt{\alpha}$ .
\end{document}
```

Next come your numbered equations. A numbered equation starts with the `\begin{equation}` statement and ends with `\end{equation}`. In between you can put text as well as all the statements used in equations. Try it yourself now. Start a new subsection called "Numbered Equations" and write down a short sentence followed by:

Use inline equations for short expressions and numbered equations for more complex equations!

```
\begin{equation}
m = \left( \frac{a \cdot b}{c} \right)
\end{equation}
```

and another short sentence followed by:

```
\begin{equation}
\lambda = \int_{m_0}^{\infty} f(\theta) d\theta
\end{equation}.
```

Include this section in your document:

```
\subsection{Numbered equations}
This is our first numbered equation.

\begin{equation}
m = \left( \frac{a \cdot b}{c} \right)
\end{equation}

This is our second numbered equation.

\begin{equation}
\lambda = \int_{m_0}^{\infty} f(\theta) d\theta
\end{equation}
```

Your document should now look like this:

```
% This is my first Latex document. There are many documents like it , but
this one is mine.

\documentclass{article}
\author{FirstName LastName}
\title{My first \LaTeX\ document}

\begin{document}
\maketitle

\begin{abstract}
This is my first \LaTeX! document. I am going to learn a lot about
typesetting and writing articles in \LaTeX! and my papers will look
really nice from now on.
\end{abstract}

\section{Introduction}
This is my \textbf{first} \emph{sentence}.

This is my \textit{second} sentence.

This is my third sentence. You can also put some distance between lines
.\[3mm]
Test

\section{Doing equations}

\subsection{Inline equations}
This is our first inline equation  $\alpha = 3.5$ .  $\lambda = \sqrt{\alpha}$ .

\subsection{Numbered equations}
This is our first numbered equation.

\begin{equation}
m = \left( \frac{a \cdot b}{c} \right)
\end{equation}

This is our second numbered equation.

\begin{equation}
\lambda = \int_{m_0}^{\infty} f(\theta) d\theta
\end{equation}
```

```
\end{document}
```

As you can see, there are a lot of new symbols added, but if you look at them, they are pretty self explanatory. `\infty` is used for the infinity symbol ∞ , `\frac{}{}` is used for fractions such as $\frac{x}{y}$ and `\cdot` is used for the multiplication dot \cdot . A bit more complicated is `\int`, which is used for integrals. The lower bound is indicated by an underscore "`_`" and the upper bound by the `^`. The full expression for an integral is `\int_x^y` which then looks like \int_x^y . Brackets can be placed with the commands `\left(` and `\right)`.

7

Labels and referencing

When adding graphs, tables, pictures or equations to your document, it is common to number these and refer to the number in the text to allow the reader to easily connect your text to the respective visualization. In addition, you might also want to number sections to be able to later refer to them. The numbering is done automatically by \LaTeX for all these types. To reference them in your document you just need to label them. The commands used for this purpose are `\label{}` and `\ref{}`. You assign a specific label to the object you want to reference later on by simply placing `\label{}` behind the object you want to reference. Here it is common to use a short prefix to indicate the type of object you want to label such as *sec* for sections, *fig* for figures, *tab* for tables and so on. This makes it easy to distinguish between different objects connected to the same topic (e.g. a graph and a table displaying the same kind of data). You can also make your references interactive. This means you can link your references to the actual content to make navigating the pdf even easier. To do so you need to include the *hyperref* package in your document preamble. You can then use the `\hypersetup` command to define the formatting of your references. Here we colored them in dark blue. You can use the packages *color* and *xcolor* to increase the spectrum of colors you can use and to enable the definition of custom colors from RGB values. To see how this is done, look at the following:

```
\usepackage[{}]{hyperref}
\usepackage{color}
\usepackage{xcolor}
\definecolor{darkblue}{rgb}{0,0,.5}
\hypersetup{colorlinks=true, breaklinks=true,
linkcolor=darkblue, menucolor=darkblue,
urlcolor=darkblue, citecolor=darkblue}
```

As a quick exercise, you are now going to label your numbered equations and the introduction section and reference them in a new section called *Labels and referencing*. Do so by placing a label behind the `\section{}` and the `\begin{equation}` statements. The label you assign here is just for you, to help you distinguish it from the others.

Labels and references are important for you and for others to easily link written information to graphic content!

Use abbreviated prefixes to quickly identify your labels!

hyperref allows you to create interactive links between your references and the referenced content. This makes navigating large documents a lot easier!

The `\ref{}` command only returns the number of the object. To reference your labels simply type `\ref{}` with the label you assigned within the curly brackets. Your document should now look like this:

```
% This is my first Latex document. There are many documents like it , but
this one is mine.

\documentclass{article}
\author{FirstName LastName}
\title{My first \LaTeX\ document}

\usepackage[{}]{hyperref}
\usepackage{color}
\usepackage{xcolor}
\definecolor{darkblue}{rgb}{0,0,.5}
\hypersetup{colorlinks=true, breaklinks=true,
linkcolor=darkblue, menucolor=darkblue,
urlcolor=darkblue, citecolor=darkblue}

\begin{document}
\maketitle

\begin{abstract}
This is my first \LaTeX! document. I am going to learn a lot about
typesetting and writing articles in \LaTeX! and my papers will look
really nice from now on.
\end{abstract}

\section{Introduction}
This is my \textbf{first} \emph{sentence}.

This is my \textit{second} sentence.

This is my third sentence. You can also put some distance between lines
.\[3mm]
Test

\section{Doing equations} \label{equations}

\subsection{Inline equations}
This is our first inline equation  $\alpha = 3.5$ .  $\lambda = \sqrt{\alpha}$ .

\subsection{Numbered equations}
This is our first numbered equation.

\begin{equation} \label{eq: definition of m}
m = \left( \frac{a \cdot b}{c} \right)
\end{equation}

This is our second numbered equation.

\begin{equation} \label{eq: definition of lambda}
\lambda = \int_{m_0}^{\infty} f(\theta) d\theta
\end{equation}

\section{Labels and referencing}
In LaTeX, everything can be labeled, and after that it can be referenced
. Consider for example section~\ref{sec: equations}, where eq.~\ref{eq:
definition of lambda} and eq.~\ref{eq: definition of m} are defined.

\end{document}
```

8

Including figures

As mentioned before, you can also include graphs and pictures in your \LaTeX document. This can be done via the *graphicx* package. The easiest way to include figures in your document is by placing the file containing your figure (e.g. a PDF or JPG) in your project folder. That way you only need to specify the name of the file when including it and not the full path. You can however place it in another location in your file system or in a subfolder of your project folder and give the file path to \LaTeX (but keep in mind not to use spaces or special symbols in your file and path names).

To include images in your \LaTeX document, you first need to define a figure space. This can be achieved by using the `\begin{figure}` and `\end{figure}` statements. In between these statements you can now include your image and add a caption to it to briefly summarize what is displayed here. To do so, you need two commands: `\includegraphics[scale=]{}`, with the name or file path of the image to include without the file extension, and `\caption{}`. Most of the time you want your image to be displayed in the center of the lines you are placing it in. For that you can use the command `\centering` one line above the `\includegraphics[scale=]{}` statement.

There should be two image files in your training material, a picture of Albert Einstein in his study and a boxplot graph. Include them into your document. They should be 6 cm wide.

```
\begin{marginfigure}
\includegraphics[width=6cm]{../example/einstein}
\caption{This photograph by Ferdinand Schmutzer shows Einstein in his
study}
\end{marginfigure}
```

To include two pictures side by side, just put them into the same figure and make sure to set their width to 3 cm. For each figure, you just need one caption even if it includes several images. And do not forget to label your figures.

Your document should now look like this:

Remember that you can include packages in the preamble before your *begin document* statement!

```

% This is my first Latex document. There are many documents like it, but
  this one is mine.

\documentclass{article}

\usepackage{graphicx}

\usepackage[]{}{hyperref}
\usepackage{color}
\usepackage{xcolor}
\definecolor{darkblue}{rgb}{0,0,.5}
\hypersetup{colorlinks=true, breaklinks=true,
linkcolor=darkblue, menucolor=darkblue,
urlcolor=darkblue, citecolor=darkblue}

\author{FirstName LastName}
\title{My first \LaTeX\ document}

\begin{document}
\maketitle

\begin{abstract}
This is my first \LaTeX! document. I am going to learn a lot about
typesetting and writing articles in \LaTeX! and my papers will look
really nice from now on.
\end{abstract}

\section{Introduction}
This is my \textbf{first} \emph{sentence}.

This is my \textit{second} sentence.

This is my third sentence. You can also put some distance between lines
.\[3mm]
Test

\section{Doing equations} \label{equations}

\subsection{Inline equations}
This is our first inline equation  $\alpha = 3.5$ .  $\lambda = \sqrt{\alpha}$ .

\subsection{Numbered equations}
This is our first numbered equation.

\begin{equation} \label{eq: definition of m}
m = \left( \frac{a \cdot b}{c} \right) \right)
\end{equation}

This is our second numbered equation.

\begin{equation} \label{eq: definition of lambda}
\lambda = \int_{m_0}^{\infty} f(\theta) d\theta
\end{equation}

\section{Labels and referencing}

In LaTeX, everything can be labeled, and after that it can be referenced.
Consider for example section~\ref{sec: equations}, where eq.~\ref{eq:
definition of lambda} and eq.~\ref{eq: definition of m} are defined.

\section{Including figures}

\begin{figure}
\centering
\includegraphics[width=6cm]{einstein} % without file extension
\caption{This shows Einstein in his study} \label{fig: einstein}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=3cm]{einstein}
\includegraphics[width=3cm]{einstein}
\caption{This shows Einstein in his study} \label{fig: einstein2}
\end{figure}

```

```
\begin{figure}  
\centering  
\includegraphics[width=6cm]{boxplot} % without file extension  
\caption{This shows a boxplot}\label{fig: boxplot}  
\end{figure}  
  
\end{document}
```

9

Tables

A very common way of displaying data is to use tables. They are a very efficient way to sort large amounts of data and to make them easy to understand. In L^AT_EX there are different ways to create and display tables. The easiest way is to use the `\begin{tabular}{}` command. In the curly brackets you can define the number of columns with the following syntax `\begin{tabular}{|c|c|}`. This would for example create a table with two columns and centered text. The columns are later separated by the `&` (ampersand) symbol. To add rows you have to break the line with `\\` and optionally add a horizontal line with the `\hline` command. Try to recreate the table on the right and keep in mind that every *begin* statement needs an *end* statement. As you can see, this kind of table is very simplistic and does not look professional or well suited for large amounts of data that require some explanation and additional information. To create proper tables you need additional packages. These packages are called *multicol*, *multirow* and *booktabs*. Include them in your document now. Look at the following code snippet:

1	1
2	1

This is a very simple table created with the *tabular* command!

```
\begin{table*}\label{Table: Fit types}
\centering
\begin{tabular}{l@{\hspace{0.2cm}}l@{\hspace{0.2cm}}l} \toprule
\textsc{Case} & \textsc{Explanation} & \textsc{Dimensions} \\ \midrule
\addlinespace[0.2cm]
\multicolumn{3}{l}{Parameterization to virtual data, 3 PFTs:} \\
$V_1$ & Data: SDD, GRO, reduced parameters & $12,96$ \\
$V_2$ & Data: SDD, GRO, full parameters & $26,96$ \\
$V_3$ & Data: SDD, reduced parameters & $12,48$ \\
$V_4$ & Data: total SDD, reduced parameters & $12,16$ \\
$V_5$ & Data: BM, reduced parameters & $12,3$ \\
\multicolumn{3}{l}{Parameterization to Ecuadorian field data, 7 PFTs:} \\
$E_1$ & Data: SSD & $18,112$ \\ \bottomrule
\end{tabular}
\caption{Table caption}
\end{table*}
```

As you can see, it is way more complex than the simple table, but it also allows a more informative and more structured table design. You do not need to specify the dimensions of the table in advance.

The columns are still separated by the & symbol. The rows are separated simply by a line break. Additionally, the table is divided into different vertical sections which are indicated by the *toprule*, *midrule* and *bottomrule* statements. These sections can be used to display additional information such as a header for the columns or to aggregate different treatments in one table. The `\multicolumn{}{}{}` command can be used to write over several columns e.g. for table headlines or explanations. The first curly bracket defines the number of columns you want to aggregate, the second one the style ("l" means left-justified) and in the third curly bracket you can put your text. In this case the rows are enclosed in \$ symbols to give them the look of a formula and separate them from the headlines. When compiled correctly the table should look like this:

CASE EXPLANATION		DIMENSIONS
Parameterization to virtual data, 3 PFTs:		
V1	Data: SDD, GRO, reduced parameters	12,96
V2	Data: SDD, GRO, full parameters	26,96
V3	Data: SDD, reduced parameters	12,48
V4	Data: total SDD, reduced parameters	12,16
V5	Data: BM, reduced parameters	12,3
Parameterization to Ecuadorian field data, 7 PFTs:		
E1	Data: SSD	18,112

use *multicolumn* to aggregate multiple columns into one e.g. for footnotes or headers

Stick to this kind of table whenever you write a scientific article, report or thesis!

This is what a table for scientific data should look like. Generally, there should be no vertical lines and horizontal lines should only be used to separate the headline from the data. Try to create this table in your own document. Your document should now look like this:

```
% This is my first Latex document. There are many documents like it , but
this one is mine.

\documentclass{article}

\usepackage{graphicx}

\usepackage{multicol}
\usepackage{multirow}
\usepackage{booktabs}

\usepackage[{}]{hyperref}
\usepackage{color}
\usepackage{xcolor}
\definecolor{darkblue}{rgb}{0,0,.5}
\hypersetup{colorlinks=true, breaklinks=true,
linkcolor=darkblue, menucolor=darkblue,
urlcolor=darkblue, citecolor=darkblue}

\author{FirstName LastName}
\title{My first \LaTeX\ document}

\begin{document}
\maketitle
```

```

\begin{abstract}
This is my first \LaTeX! document. I am going to learn a lot about
typesetting and writing articles in \LaTeX! and my papers will look
really nice from now on.
\end{abstract}

\section{Introduction}
This is my \textbf{first} \emph{sentence}.

This is my \textit{second} sentence.

This is my third sentence. You can also put some distance between lines
.\!\![\!3mm]
Test

\section{Doing equations} \label{equations}

\subsection{Inline equations}
This is our first inline equation  $\alpha = 3.5$ .  $\lambda = \sqrt{\alpha}$ .

\subsection{Numbered equations}
This is our first numbered equation.

\begin{equation} \label{eq: definition of m}
m = \left( \frac{a \cdot b}{c} \right)
\end{equation}

This is our second numbered equation.

\begin{equation} \label{eq: definition of lambda}
\lambda = \int_{m_0}^{\infty} f(\theta) d\theta
\end{equation}

\section{Labels and referencing}

In LaTeX, everything can be labeled, and after that it can be referenced.
. Consider for example section~\ref{sec: equations}, where eq.~\ref{eq:
definition of lambda} and eq.~\ref{eq: definition of m} are defined.

\section{Including figures}

\begin{figure}
\centering
\includegraphics[width=6cm]{einstein} % without file extension
\caption{This shows Einstein in his study} \label{fig: einstein}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=3cm]{einstein}
\includegraphics[width=3cm]{einstein}
\caption{This shows Einstein in his study} \label{fig: einstein2}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=6cm]{boxplot} % without file extension
\caption{This shows a boxplot} \label{fig: boxplot}
\end{figure}

\section{Tables}

\subsection{Simple table}
\begin{tabular}{|c|c|}
\hline
1 & 1 \\
\hline
2 & 1 \\
\hline
\end{tabular}

\subsection{Proper table}
\begin{table*} \label{Table: Fit types}
\centering

```

```

\begin{tabular}{l@{\hspace{0.2cm}}l@{\hspace{0.2cm}}l} \toprule
\textsc{Case} & \textsc{Explanation} & \textsc{Dimensions} \\ \midrule
\addlinespace[0.2cm]
\multicolumn{3}{l}{Parameterization to virtual data, 3 PFTs:} \\
$V_1$ & Data: SDD, GRO, reduced parameters & $12,96$ \\
$V_2$ & Data: SDD, GRO, full parameters & $26,96$ \\
$V_3$ & Data: SDD, reduced parameters & $12,48$ \\
$V_4$ & Data: total SDD, reduced parameters & $12,16$ \\
$V_5$ & Data: BM, reduced parameters & $12,3$ \\
\multicolumn{3}{l}{Parameterization to Ecuadorian field data, 7 PFTs:} \\
\\
$E_1$ & Data: SSD & $18,112$ \\ \bottomrule \\
\end{tabular}
\caption{Table caption}
\end{table*}
\end{document}

```


References in \LaTeX

In every scientific article it is important to clearly state your sources. You need to cite within your article and list all sources used at the end in your bibliography. Basically, there are two options to use references in \LaTeX . The first is to include references in the \LaTeX file directly, the second is to maintain an external file with references in the BibTeX format. For most cases, the latter will be preferred. There are few reasons to keep references directly in the \LaTeX file. One reason could be that a journal requires the submission in one \LaTeX file only (but in most cases you are allowed to include a .bbl file, which is created by BibTeX). In the following, we focus on the link to the BibTeX format.

An excellent program to build your BibTeX database is Jabref, a free, open-source literature manager, which uses the BibTeX file format. As it is written in Java and distributed as a jar file, you can use it on any operating system with a Java Virtual Machine (JVM) installed. It is the first choice for everyone who works with \LaTeX , but it can also be used to insert citations in LibreOffice Writer or Microsoft Word.

This is an example of how an entry in your BibTeX database could look like:

```
@ARTICLE{Ovaskainen-Spaceandstochasticity-2006,
author = {Ovaskainen, O. and Cornell, S. J.},
title = {Space and stochasticity in population
dynamics},
journal = {PNAS},
year = {2006},
volume = {103},
pages = {12781--12786},
number = {34},
month = aug,
}
```

You will almost never have to type this yourself into your literature database in Jabref. Here are some examples of how to automatically import BibTeX entries from (1) journals, (2) databases (google scholar, ISI, ...) or even (3) package citations from R:

Comprehensive information on how to link your Jabref database with Microsoft Word can be found at: <http://www.medicalnerds.com/how-to-use-jabrefbibtex-with-microsoft-word-2003/>

1. Importing BibTeX entries from journal websites is highly recommended, since these are most likely to be correct. Instead of directly reading the paper (pdf) you may search for an “export citation” button or similar on the journal website. After clicking on it, several format options should be provided. If you drag-and-drop the source to Jabref, it does not matter which format you choose (Jabref can handle it). If you use the import menu option in Jabref, choose the BibTeX format.
2. Faster, but with higher risk of misspelling or other mistakes is the citation export directly from:
 - (a) **Google Scholar:** Click on “cite” below the respective Google Scholar entry. At the bottom of the pop-up window you may choose the source format (see above).
 - (b) **ISI Web of Knowledge:** Tick the box on the left of the article you want to cite and choose from the drop-down list at the top of the page “Save to EndNote desktop”. A window pops up asking you to choose the contents to record. Hit send and open the output with an editor. From there drag-and-drop the source into jabref.
3. If you want to cite R or an R-package, you may obtain the BibTeX source directly from R. For citing the base R program simply type `citation()` in the R console and hit Enter. If you wish to cite a specific package, e.g. MASS, the command to use is: `citation("MASS")`. Copy the source from the R console to jabref.

The default notepad on windows does not let you drag-and-drop. Use [notepad++](#) instead.

Now that you have set up a BibTeX database, you can easily include it in your document and link to the entries in your .bib file. As with the inclusion of graphics, you can either put your .bib file in the same directory as the L^AT_EX file and just refer to the .bib file by its name or put the .bib file elsewhere, but then you need to specify the whole path. You can include a database in your L^AT_EX document by using the command `\bibliography{}` at the end of your document, with the name of your database in curly brackets. The style of your bibliography is determined by the `\bibliographystyle{}` command before the `\bibliography{}` command, with the name of the style file in curly brackets. Here we use the *chicago* citation style. To produce a pdf with all citations and the complete bibliography, you need to compile BibTeX as well as L^AT_EX at least two times. Also make sure you are using the correct file encoding (e.g. UTF-8). Once everything is set, you can reference your sources within your document using the `\citep{}` and `\cite{}` commands from the *natbib* package. The classical citation command is `\citep{}`. It leads to your cited source enclosed in brackets while `\cite{}` and `\citet{}` only put the date

of the citation in brackets allowing for an in-text citation. Once you cite a source in your document it will be listed in your bibliography as well. In this tutorial it is the last step but it is generally advisable to set up your bibliography at the beginning for your project to make citing fluent and to avoid mistakes. Now add a chapter about citations to your document and after that add your bibliography. A literature database is supplied in your training material. To make your citations look nicer and allow for more options include the *natbib* package in your document. You can adjust the punctuation of your references by using the `\bibpunct` command in your preamble. After including the *natbib* package add the following line:

`\bibpunct{{}{}}{;}{a}{}{,}` Now everything is set for you to compile your document and have a look at your citations and your bibliography. Your \LaTeX file should now look like this:

```
% This is my first Latex document. There are many documents like it , but
this one is mine.

\documentclass{article}

\usepackage{graphicx}

\usepackage{multicol}
\usepackage{multirow}
\usepackage{booktabs}

\usepackage{natbib}
\bibpunct{{}{}}{;}{a}{}{,}

\usepackage[{}]{hyperref}
\usepackage{color}
\usepackage{xcolor}
\definecolor{darkblue}{rgb}{0,0,.5}
\hypersetup{colorlinks=true, breaklinks=true,
linkcolor=darkblue, menucolor=darkblue,
urlcolor=darkblue, citecolor=darkblue}

\author{FirstName LastName}
\title{My first \LaTeX\ document}

\begin{document}
\maketitle

\begin{abstract}
This is my first \LaTeX! document. I am going to learn a lot about
typesetting and writing articles in \LaTeX! and my papers will look
really nice from now on.
\end{abstract}

\section{Introduction}
This is my \textbf{first} \emph{sentence}.

This is my \textit{second} sentence.

This is my third sentence. You can also put some distance between lines
.\!\!.[3mm]
Test

\section{Doing equations} \label{equations}

\subsection{Inline equations}
This is our first inline equation  $\alpha = 3.5$ .  $\lambda = \sqrt{\alpha}$ .

\subsection{Numbered equations}
```

This is our first numbered equation.

```
\begin{equation} \label{eq: definition of m}
m = \left( \frac{a \cdot b}{c} \right)
\end{equation}
```

This is our second numbered equation.

```
\begin{equation} \label{eq: definition of lambda}
\lambda = \int_{m_0}^{\infty} f(\theta) d\theta
\end{equation}!!\
```

```
\section{Labels and referencing}
```

In LaTeX, everything can be labeled, and after that it can be referenced. Consider for example `\ref{sec: equations}`, where `\ref{eq: definition of lambda}` and `\ref{eq: definition of m}` are defined.

```
\section{Including figures}
```

```
\begin{figure}
\centering
\includegraphics[width=6cm]{einstein} % without file extension
\caption{This shows Einstein in his study} \label{fig: einstein}
\end{figure}
```

```
\begin{figure}
\centering
\includegraphics[width=3cm]{einstein}
\includegraphics[width=3cm]{einstein}
\caption{This shows Einstein in his study} \label{fig: einstein2}
\end{figure}
```

```
\begin{figure}
\centering
\includegraphics[width=6cm]{boxplot} % without file extension
\caption{This shows a boxplot} \label{fig: boxplot}
\end{figure}
```

```
\section{Tables}
```

```
\subsection{Simple table}
```

```
\begin{tabular}{|c|c|}
\hline
1 & 1 \\
\hline
2 & 1 \\
\hline
\end{tabular}
```

```
\subsection{Proper table}
```

```
\begin{table*} \label{Table: Fit types}
\centering
\begin{tabular}{|l@{\hspace{0.2cm}}l@{\hspace{0.2cm}}l| \toprule
\textsc{Case} & \textsc{Explanation} & \textsc{Dimensions} \\ \midrule
\addlinespace[0.2cm]
\multicolumn{3}{|l|}{Parameterization to virtual data, 3 PFTs:} \\
$V1$ & Data: SDD, GRO, reduced parameters & $12,96$ \\
$V2$ & Data: SDD, GRO, full parameters & $26,96$ \\
$V3$ & Data: SDD, reduced parameters & $12,48$ \\
$V4$ & Data: total SDD, reduced parameters & $12,16$ \\
$V5$ & Data: BM, reduced parameters & $12,3$ \\ \midrule
\multicolumn{3}{|l|}{Parameterization to Ecuadorian field data, 7 PFTs:} \\
$E1$ & Data: SSD & $18,112$ \\ \bottomrule
\end{tabular}
\caption{Table caption}
\end{table*}
```

```
\section{References}
```

Referenced are extremely important `\citep[see also][for more references]` [Gintis–Costlysignalingand–2001, Archetti–Economicgametheory–2011]. However `\citet{Cooper–CommunicationInCoordination–1992}` note that this

```
is not the case.  
\bibliographystyle{chicago}  
\bibliography{yourbibtexfile}  
\end{document}
```

Going further

While expanding your document or when creating a new one you will probably get to a point where you want to do something that has not been covered by this tutorial. Here are three options how to find additional information and help:

- The first thing to do is to search for a solution on the web. \LaTeX has a very active community and most of the things you want to do will have already been done by someone and posted in some forum or wiki.
- Secondly, you can find so called "cheat sheets" (just search for them online) these are short scripts which list most of the functions and commands with a short explanation about what they do and how they are used.
- Finally, I provide some information on a list of useful packages in the [appendix](#) of this document.

When you encounter problems with \LaTeX search engines and forums are your best friends!

Appendices

A

FAQ

Q: Why does the referencing not work in my document?

A₁: The path to your reference file is not correct:

We recommend the use of relative file paths, such that your project folder contains different folders for different files, such as your .tex document (e.g. in a folder called "thesis") and your .bib literature database (e.g. in a folder called "literature"). If you use such a structure you may set `\bibliography{../literature/yourbibtexfile}` at the end of your .tex document in order to link your .tex document with your .bib database.

Another option is to store all files in only one folder. Then you only need to set `\bibliography{yourbibtexfile}` at the end of the .tex document. If you have many additional files, such as figures, this is not recommended.

There is one other option, which works as long as you are only working on one computer: The use of full file paths. This may look like this: `\bibliography{C:/Users/User/documents/literature/yourbibtexfile}`, or the path to wherever yourbibtexfile.bib is stored. The obvious problem is that this specific file path must exist.

A₂: You mistakenly provided the file extension:

In `\bibliography{}` you must give the file name of your database without the extension (e.g. .bib). This means it must be `\bibliography{yourbibtexfile}` instead of `\bibliography{yourbibtexfile.bib}`

A₃: You need to do a LaTeX–BibTeX–LaTeX–LaTeX sequence:

When compiling your .tex document the first time after adding a link to your literature database (.bib) you need to compile in LaTeX (F1 or F6), then in BibTeX (F8 or F11), and then twice in LaTeX again. This is necessary because each step produces auxiliary files that the next

step uses. The first compiling in LaTeX generates an .aux file, which the citations need. The BibTeX compiling generates a .bbl file. The third compiling incorporates these references into the document, and the fourth is necessary to ensure that the insertion of these references still produces correct formatting.

Q: Why are my figures or tables showing up in the wrong place?

A: Figures and tables are placed in floats:

In \LaTeX figures and tables are positioned in the text always following the same principles or rules (if not specified differently). To do this figures and tables are put into containers called floats. These also prevent the container items to be broken over pages.

Now, the answer to this question may seem a bit circumventive: Simply don't care about the positioning of your tables and figures. If you finished your document and you still don't like the arrangement check [this](#) out.

B

Useful packages

This is a selection of packages, which may help you with everyday L^AT_EX problems:

B.1 Encoding - inputenc

The inputenc package allows for extended character encoding – in particular for characters like Ä, Ü, etc..

```
\usepackage[latin 1]{inputenc}
```

B.2 Typesetting - fontenc

Extends the standard font family of L^AT_EX to allow special characters.

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

B.3 Language specific labels and hyphenation - babel

The babel package changes the standard labels of LaTeX like e.g. "Table of Contents" or "References" to the specified language. It also changes the hyphenation.

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

B.4 Including pdfs in your text - pdfpages

The pdfpages package allows you to include pdf documents (e.g. a paper) in your document.

Documentation [here](#)

Conflicts/Issues Works only with pdflatex

B.5 Proper appendices - the *appendix* package

The *appendix* package adds some useful features, which you will probably need when writing an appendix. It defines an `appendices` as new environment

```
\usepackage[toc,page]{appendix}
...
\begin{appendices}
...
\end{appendices}
```

It is also possible to include subappendices within a section in the document

```
\usepackage{appendix}
...
\section{Some stuff}
...
\begin{subappendices}
\subsection{Some boring stuff}
...
\end{subappendices}
\section{Some new stuff}
...
```

B.6 For Formulae: *Amsmath*, *amsmaths* and *amssymb*

The standard packages for extended mathematical symbols. Careful though, there is a conflict with the *lineno* package!

```
\usepackage{amsmath, amsfonts, amssymb}
```

Documentation [here](#)

Conflicts/Issues Conflict with the *lineno* package!

B.7 The *color* package

Enables colors in L^AT_EX.

B.8 The *xcolor* package

Extends color commands in latex. Enables in particular mixing of colors and provides some predefined colors.

B.9 Underline - the *ulem* package

Surprisingly, L^AT_EX does not include an option for underlining text by default. This can be resolved using the *ulem* package. Ulem replaces italics with underlining in emphasized text given by *em* or *emph*. I recommend to use it with the *normalem* option, which restores the normal *em* behavior.

```
\usepackage[normalem]{ulem}
...
\uuline{Underlined text}
```

B.10 Spacing with setspace and linespread

Many journals demand double spaced text for submission.

```
\usepackage{setspace}
```

Double spacing can be activated by

```
\doublespacing
```

A number of other spacing options is available and can be found in the manual. However, one drawback is that setspace does not change the spacing in the footnotes and floating environments like figures captions, tables etc. If double spacing is required throughout all text, tables etc. the best option is to use the linespread command

```
\linespread{1.6}
```

Note that although you want to have double spacing, the multiplication factor is 1.6, due to the way L^AT_EX measures the spacing.

B.11 Hyphenation

To switch off hyphenation, use the hyphenat package. Careful: this will sometimes cause text to cross the border of your text box

```
\usepackage[none]{hyphenat}
```

B.12 The preprint bundle

Preprint is a bundle of packages which helps submitting preprints to journals. The most important package inside preprint is "figcaps", which is described in the next section. The package is quite old, but the MPI for solar seems to use the package and has made some changes which may be interesting to modify for your own purposes, too. Find their website at: [preprint](#)

B.13 figcaps

Figcaps is part of the preprint bundle. It puts all tables, figure captions, and (optional) figures at the end of the document on separate pages.

```
\usepackage[]{}{figcaps}
```

Figcaps has sometimes problems with the references to figure captions, which results in ?? when you refer to a figure number in your text. Put the code below somewhere after the `\usepackage{figcaps}` command. This should resolve the problem (code from [Ted Pavlic's blog](#)).

```
\makeatletter
% This lets figcaps work with \ref
% However, forces \label outside of \caption
\def\phantomsection{\relax}
\let\oldfigurepage\@figurepage
\def\@figurepage{%
\ifundefined{tf@pof}{}{%
\let\oldlabel\label%
\let\oldinput\@input%
\def\@input{\def\label{\oldlabel}\oldinput}%
\phantomsection%
\addcontentsline{toc}{section}{\figurepagename}}%
\oldfigurepage%
}
\let\@makefcaption\@makecaption
\let\oldtablepage\@tablepage
\def\@tablepage{%
\ifundefined{tf@pot}{}{%
\clearpage%
\phantomsection%
\addcontentsline{toc}{section}{\tablepagename}}%
\oldtablepage%
}
\makeatother
```