Introduction to LaTeX

1 Introduction

LaTeX is a typesetting program that allows ordinary users to produce high quality mathematical and other output using characters typed on a standard keyboard. Initially it can look intimidating compared to other WSIWYG[1] editors such as *MS Word* but it is actually quick and intuitive to learn. With just a few commands you have great flexibility with your presentation and should you desire more in-depth formatting you can easily modify existing code for your own individual purposes.

In this document we will aim to give you a brief overview of the program which should enable you to write a simple report with beautiful and clear presentation. There are however many more complex things you can do with the program which can be found online, see [1] for more information.

Important: Make sure you are using pdfLaTeX to compile your LaTeX files. This will create a .pdf file directly. On the University Managed System we will be using an interface called TeXworks.

2 Getting Started

2.1 Accessing the TeXworks software on the Managed Desktop

To see if TeXworks is already installed on your machine click the button on the bottom left of the screen, scroll down and look for MiKTeX2.9 and then TeXworks. Click on TeXworks if it is there. If it isn't installed, follow the instructions in the Appendix to install MiKTeX and TeXworks on your Managed Desktop machine.

2.2 Downloading additional files

At various points you will be asked to download additional LaTeX files. Please agree to these requests. On the Managed Desktop you may find that this doesn't work the first time. If this is the case try it twice and it should download properly.

2.3 A first document

The input to LaTeX is a source file which contains plain ASCII text. It contains the text you want to display plus instructions which tell the program how to typeset it. To create our first example type

```
% Hello.tex - our first program in LaTeX

\documentclass[a4paper, 11pt]{article}
\begin{document}
Hello World!
\end{document}
```

2.4 Creating the output

Save this file as `HelloWorld.tex` and compile it. See Appendix A for instructions on how to compile depending upon whether you are using the managed desktop or are at home.

3 Document Layout

3.1 Preamble

To create any LaTeX document the source file must start with a preamble which will specify the layout of the document. This normally contains commands which will affect the whole document. We give an example which

[1] What You See Is What You Get

should be sufficient for most applications. It can be copied into any new document. Create a new text file called `LatexEx1.tex` containing

```
% LatexEx1.tex - An introduction to LaTeX

\documentclass[a4paper, 12pt]{article}

\usepackage{amsmath}         % Used for mathematical equations
\usepackage{amssymb}         % As above
\usepackage{natbib}          % Used for bibliography
\usepackage{graphicx}        % Used to include figures
\usepackage{epstopdf}        % Convert .eps files to .pdf

\title{An introduction to \LaTeX}
\author{A.N. Other}
\date{1st Jan 2000}

\begin{document}
\maketitle
You can start writing all your text in here!
\end{document}
```

The above commands should be self-explanatory except for

- The `%` symbol allows comments (as for the # symbol within R). Anything on the line after this symbol is not read by the compiler.
- `\documentclass[options]{class}` — Specifies what class of document you are creating (commonly `article` but other possibilities you may use at other times are `report` or `book`) together with font size and paper size.
- `\usepackage{package}` — Tells LaTeXto utilise some external macros which will ease later typesetting.
- `\maketitle` — This creates the title, it should be the first line after `\begin{document}`

Now compile this document and look at the result.

3.2 Exercises

1. Change the font size to `10pt`;
2. Modify the preamble to also include the package `geometry`;
3. Alter the
   (a) Title to "An Introduction to LaTeX",
   (b) Author to your name,
   (c) Date to today's date.

4 Typesetting and Organisation

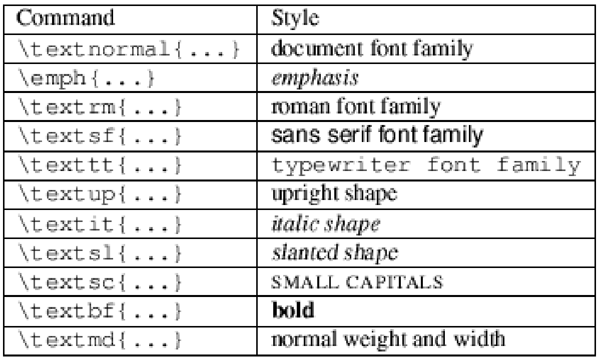
4.1 Creating sections

4.1.1 An example

Any report will need different sections and subsections with appropriate headings. The commands required to make these are,

```
\section{Name}
\subsection{Name}
\subsubsection{Name}
```

It is up to you to use them in the correct order. These commands are inserted at the point in the text where the heading should go. Create an introduction by typing directly after `\maketitle`



Command	Style
\textnormal{...}	document font family
\emph{...}	*emphasis*
\textrm{...}	roman font family
\textsf{...}	sans serif font family
\texttt{...}	`typewriter font family`
\textup{...}	upright shape
\textit{...}	*italic shape*
\textsl{...}	*slanted shape*
\textsc{...}	SMALL CAPITALS
\textbf{...}	**bold**
\textmd{...}	normal weight and width

Figure 1: A list of various Font styles in LaTeX

```
\section{Introduction \label{S:Intro}}
This document is an introduction to the use of \LaTeX. We will show how to use \LaTeX\ to
produce most things which would be needed in a short report. We will show how to produce
tables, include figures and typeset some simple mathematics.
```

Note the use of the `\label{Name}` command within the title so that we can automatically reference the section later if we want — see Section 8 for more details.

4.2 Paragraphs and Changing Text Formats

4.2.1 Paragraphs and New Lines

In order to create a new paragraph in LaTeX you need to leave a blank line in the text spearating the two paragraphs. This is illustrated in the introduction created in Section 4.1.1. LaTeX will automatically fit your text to the width of the page and deal with identation of new paragraphs.

4.2.2 Text Formats

There are often times when you will want to emphasize certain words for the reader. The standard way to do this is to use *italics*. LaTeX provides the command `\emph{text}` to do this. There are also lots of other ways to change the text format as shown in Figure 1.

4.3 Lists

4.3.1 Itemize

You may often want to create a list in an academic document to present clearly an algorithm or methodology. LaTeX provides two options for this. If you want a standard bulleted list then you should use the itemize environment,

```
\begin{itemize}
\item First item in list
\item Second item in list
\item Third item in list
\end{itemize}
```

- First item in list
- Second item in list
- Third item in list

4.3.2 Enumerate

Alternatively, if you want a numbered list then you will want the enumerate environment,

```
\begin{enumerate}
\item First item in list
\item Second item in list
\item Third item in list
\end{enumerate}
```

1. First item in list
2. Second item in list
3. Third item in list

4.4 Exercises

1. Create a new section in your document entitled “Typesetting”. To this you should add subsections “Changing Text Formats” and “Creating Lists”. Split the List subsection further into “Itemize” and “Enumerate” subsubsections.

2. Within your “Changing Text Formats” subsection, write two paragraphs describing what you had for your breakfast and lunch. List the breakfast items using a **bold** font and the lunch items with an *emphasized* font.

3. In the “enumerate” subsubsection create an enumerated lists of items you had for dinner last night. It should start with the sentence, “For dinner last night I had: . . . ”.

   In the “itemize” subsubsection create an itemized list of you favourite sports.

5 Mathematics

Typesetting mathematics is perhaps the most powerful tool within LaTeX and what has made it so popular. LaTeX typesets equations differently from text and so needs to know when it should expect mathematics. To do this it uses maths environments. There are two types depending upon whether you want to typeset the equation

- *text style* — equations are displayed on the same line as the text itself.
- *display style* — equations will break the paragraph and be shown on a separate line.

It will be easiest to show how to typeset equations with some simple examples. Create another section with a suitable title and try the following

5.1 Text and Display Style

To typeset an equation so that it appears on the same line as the *text* it should be entered between $ and $. For example

```
Add $a$ and $b$ to get $c$,
or written more formally $a + b = c$.
```

Add $a$ and $b$ to get $c$, or written more formally $a+b = c$.

If you have larger equations and you want to *display* them on a separate line then they should be entered between `\begin{equation}` and `\end{equation}`. This will also give the equation a number so that you can refer to it later. For example

```
Add $a$ and $b$ to get $c$,
or written more formally
\begin{equation}
a + b = c. \label{E:SimpleEx}
\end{equation}
```

Add $a$ and $b$ to get $c$, or written more formally

$$a + b = c. \tag{1}$$

Again note the use of the `\label{Name}` command within the equation environment so that we can refer to the equation later — see Section 8.

If you do not want a number on the equation then you should enclose the equation between `\begin{equation*}` and `\end{equation*}` or more simply `\[` and `\]`. For example

```
Add $a$ and $b$ to get $c$,
or written more formally
\[
a + b = c.
\]
```

Add $a$ and $b$ to get $c$, or written more formally

$$a + b = c.$$

5.2 Common examples of equations

5.2.1 Symbols

There are a lot of symbols and brackets which you can use within equations.

```
\alpha, \beta, \lambda, \mu,
\Gamma, \Delta
=, <, \geq, \neq
\cos, \sin, \lim
\mathbf{R}, \forall
(x+2), [x+2], \{x+2\}
```

$\alpha, \beta, \lambda, \mu, \Gamma, \Delta$
$=, <, \geq, \neq$
$\cos, \sin, \lim$
$\mathbf{R}, \forall$
$(x+2), [x+2], \{x+2\}$

5.2.2 Common equations - Fractions

```
\[
\frac{x+y}{y-z}
\]
```

$$\frac{x+y}{y-z}$$

5.2.3 Common equations - Powers and indices

```
\[
p^2_{ij}, e^{x}, e^{x^2}
\]
```

$$p^2_{ij}, e^{x}, e^{x^2}$$

5.2.4 Spacing

Mainly LaTeX does a good job of spacing formula but if you want to force more space to appear between two symbols in an equation you can use `\,` `\quad` `\qquad`. For example

```
\int f(x) dx

\int f(x) \, dx

\int f(x) \quad dx

\int f(x) \qquad dx
```

$$\int f(x) dx$$

$$\int f(x) \, dx$$

$$\int f(x) \quad dx$$

$$\int f(x) \qquad dx$$

5.2.5 Limits and Integrals

```
\begin{equation*}
  \lim_{x \rightarrow 0}
  \frac{\sin x}{x}=1
\end{equation*}
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}=1$$

```
\begin{equation*}
\sum_{i=1}^n,
\int_0^{\frac{\pi}{2}},
\prod_\epsilon
\end{equation*}
```

$$\sum_{i=1}^n, \int_0^{\frac{\pi}{2}}, \prod_\epsilon$$

5.3 Equations of multiple lines

Use the align enviroment (or align* if you don't want numbering). The `\\` is used to denote the end of a line and `&` is used to specify what elements you want to line up.

```
\begin{align*}
  a & = b + c \\
   & = d+e
\end{align*}
```

$$\begin{aligned} a &= b + c \\ &= d+e \end{aligned}$$

5.4 Array

To create a matrix you can type

```
\[              % Enter math environment
\left[          % Left brace (square chosen)
\begin{array}   % Start array
{c c}           % Create two, centred columns
a & b \\        % Elements in first row
c & d \\        % Elements in second row
\end{array}     % End array
\right]         % Right brace
\]              $ End math environment
```

$$\left[\begin{array}{cc} a & b \\ c & d \end{array}\right]$$

5.5 Exercises

Typeset the following

- $$\frac{\frac{1}{x}+\frac{1}{y}}{z}$$
- $$x^a x^b = x^{a+b}$$
- $$\int_{\{x^2<1\}} f(x)\,dx$$
- $$\forall x \in \mathbf{R}: \qquad x^2 \geq 0$$
- $$\begin{aligned} f(x) &= (x-a)(x+a) \\ &= x^2 - ax + ax - a^2 \\ &= x^2 - a^2 \end{aligned}$$
- $$\left(\begin{array}{cc} 3 & 1 \\ 9 & 2 \end{array}\right)$$

6 Including Figures — within pdfLaTeX

6.1 Via an Example

The easiest way to import a figure into pdfLaTeX from R will be to save it as a *.pdf* file. This will be taught within the relevant R courses but for the moment you should download the file *BodyBrains.pdf* from MOLE and save it within the same directory as `LatexEx1.tex`. In order to include this Figure within the LaTeX document you should type, in the place you want the figure to appear,

```
\begin{figure}                   % Starts the figure environment
\begin{center}                   % Centres the figure (note spelling)
\includegraphics{BodyBrains.pdf}  % Insert the figures
\caption{Example Figure}          % A caption to appear under the figure
\label{F:Example}                 % A label so you can refer to it
\end{center}                      % Ends the centring
\end{figure}                      % Ends the figure
```

This should produce Figure 2. It should be said that exact figure placement is difficult but generally LaTeX will try and place it in the first suitable space after the text which surrounds it. If the Figure does not appear where you want it to go then we recommend moving the whole `\begin{figure}`...`\end{figure}` earlier or later in the ASCII text file.

Note 1: The pdf viewer within TeXworks seems to have problems viewing figures and the above plot does not appear correctly on the screen. To view the figure properly you need to open the .pdf output using another viewer (for example Adobe Acrobat).

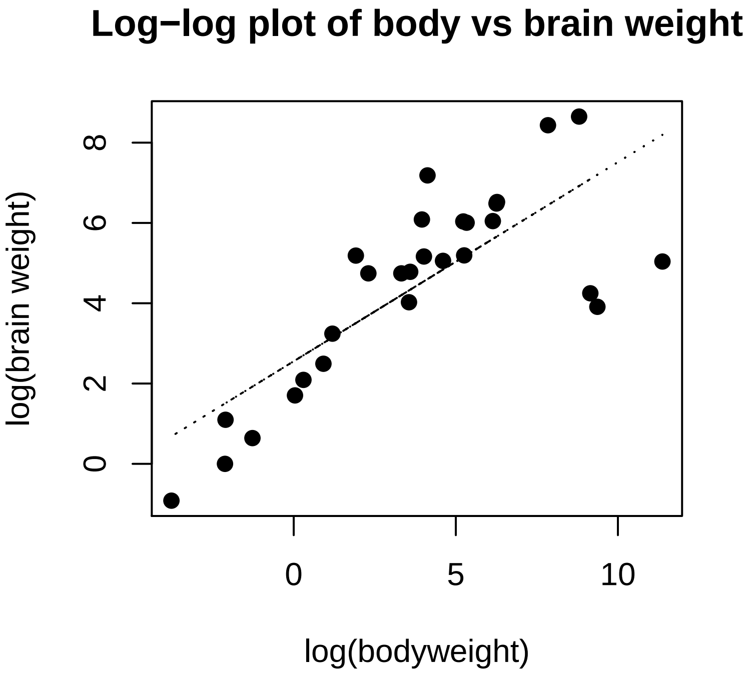


Figure 2: Example Figure

Name	Exam 1	Exam 2
Bob Smith	43	81
Anne Frank	75	70

Table 1: Exam marks of Students in Year 1

Note 2: To save a plot as a .pdf file from R you need to select `>File>Save as>PDF`

6.2 Options

There are lots more options as to how to manipulate the size of the figure, include more than one plot in a figure . . . We do not have the space to go into this here but a quick search of the internet will reveal much more information.

6.3 Exercises

- Alter the figure caption to explain that it is a plot of log average bodyweight and log average brain weight for 28 species of land animals. The solid line is the linear regression fit.

7 Tables

In lots of the documents you might want to be creating, the use of tables to summarize your results will be common. Production of basic tables in LaTeX is again relatively straightforward (although it may initially look a bit confusing). We will give an example of a simple table and then explain each command in turn

```
\begin{table}
\centering
\begin{tabular}{|l||c|r|} \hline
Name & Exam 1 & Exam 2 \\ \hline
Bob Smith & 43 & 81 \\
Anne Frank & 75 & 70 \\ \hline
\end{tabular}
```

```
\caption{Table 1 caption}
\label{T:Exam Marks}
\end{table}
```

This should produce the table seen in Table 1. While this initially looks confusing, each command can be explained in turn and the basic format needs only slight modification for most of the basic tables you are going to produce. Breaking down the commands:

- `\begin{table}` — This will start the floating table environment
- `\centering` — Places the table in the centre of the page.
- `\begin{tabular}{|l||c|r|} \hline` — Starts the table, the `{|l||c|r|}` argument specifies with the number of letters the number of columns (in this case three with an l, a c, and an r) and where to write the text in them (l (left), c (centre) or r (right)). The `|`'s specify whether you want vertical lines between columns and how many.
- `Name & Exam 1 & Exam 2 \\ \hline` — The first row of the table. You separate the different columns with use of an `&`. The `\hline` command states that you want a line to be drawn underneath this row to separate the table.
- `Bob Smith & 43 & 81 \\` — The second row of the table, again the `&`'s separate the entries into each column.
- `Anne Frank & 75 & 70 \\ \hline` — The third row, note that this row will also have a horizontal line drawn underneath (to mark the bottom of the table in this case).
- `\end{tabular}` — End the table.
- `\caption{Exam marks of Students in Year 1}` — a caption to go underneath the Table explaining what is shown.
- `\label{T:Exam Marks}` — A label if we want to refer back to the Table later, see Section 8.
- `\end{table}` — The end of the floating table enivorment.

7.1 More complex tables

LaTeX allows for much more complex tables, a great deal of literature is available online to explain this. Specific examples that might come in useful are merging columns together at certain points or making sure that the decimal points line up throughout a column.

7.1.1 Large Tables

It looks like it might be a bit of a chore to create large tables if you have to copy the numbers in by hand and add `&` characters between each column. To simplify this, we recommend opening the raw data (just the numbers split into columns) within OpenOffice (or Excel) and then using the adding columns functionality to add the `&` and \\'s. This can then be copied into your LaTeX document.

7.2 Exercises

- Change the caption of Table 1 so that it is clearer to the reader what is illustrated.
- Modify the given code to produce Table 2. Note that you will need to include a new row and a new column for the extra student and the marks from the third exam.
- Within MOLE, the file ExamMarksYear2.csv contains a larger table of random, comma separated data. Open this within OpenOffice (or Excel) and add the required `&` and \\'s to separate columns and rows. Import this table into your LaTeX document and create Table 3.

Name	Exam 1	Exam 2	Exam 3
Bob Smith	43	81	9
Anne Frank	75	70	65
John Prescott	49	68	91

Table 2: Exam marks of Students in Year 1

Student Name	Exam 1	Exam 2	Exam 3	Exam 4
Angelica Huston	97	64	38	5
Brian May	71	46	32	1
Sid Vicious	27	59	13	11
Sania Mirza	92	55	79	26
David Cameron	41	11	28	93
Bill Murray	87	77	77	52
James Herriot	21	3	80	12
Jane Austen	0	23	38	14
Katie Price	93	65	25	52
Julia Roberts	27	71	1	5

Table 3: Exam Marks of Students in Year 2

8 Labelling

One more benefit of LATEX is that it enables you to automatically refer to other parts of the text when you are writing. As we have gone along you should have noticed that when we created a new section, included a numbered equation or a figure we used the command `\label{name}`, see the input described in Sections 4.1.1, 5.1 and 6.1. If at any point in the document we then want to refer back to this Section, equation or figure we can then use the command

```
\ref{name}
```

For example, suppose that we wanted to refer back to the Introduction in this document to which I have attached the label `S:Intro` or the equation (1) to which we attached the label `E:SimpleEx`.

```
In Section \ref{S:Intro} we
explain what this document aims
to do.

We display a very simple
relationship in equation
\ref{E:SimpleEx}
```

In Section 1 we explain what this document aims to do.

We display a very simple relationship in equation 1.

One of the really nice features is that if we include new Sections or Equations which alter the numbering then LATEX will automatically alter the reference.

8.1 Exercises

- Write a sentence that references Figure 2 within the main text.
- Write a sentence that references Equation 1.
- Write a new numbered equation (using `\begin{equation}`. . .) with a new label and place it in the text before equation 1. Now recompile your code, the equation number should have changed but the reference should still be correct.

9 Citations using BibTEX

BIBTEX is an auxiliary database program that makes handling references in LATEX efficient and flexible. It requires creating an external file, with the file extension .bib, containing all the information about your references. You can then refer to this from your LATEX file and BIBTEX will print out the relevant reference in summary form in the text of your document and prepare a references list or bibliography section at the end of the document giving full details about all the sources to which you have referred.

This approach has the advantage that one can create a single .bib file containing all the possible references you have come across which can then be transferred between LaTeX documents and reused.

Creating your .bib file

To use BIBTeX you first need to have set up a database of references. The database should be a text file whose name ends in the extension `.bib` but otherwise can be anything you choose. The file `refdatabase.bib` in the 'MAS6002 LaTeX Resources' page on MOLE is an example.
Most references you will use will have one of two forms: a `book` or an `article` (from a journal). Other possibilities include `incollection` (for a part of a book in which the entry has its own title), `inproceedings` (for an article in conference proceedings), `manual`, `mastersthesis`, `unpublished` (for an unpublished work with an author and a title), `misc` (for something that doesn't fit anywhere else e.g. webpages) . . . . Each form has a slightly different set of fields for you to fill in. Some of these are required and others optional.
The `.bib` file can be produced in any text editor. Both WinEdt and TeXMaker make the task easy by providing templates for the different entry types, indicating the possible field names for each, and which fields are optional, which mandatory[2].

Entry for a book

To enter in the database the information for a book one writes in the `.bib` text file

```
@BOOK{cramerleadbetter:1967,
AUTHOR = {Cram\'{e}r, H. and Leadbetter, M.R.},
TITLE = {Stationary and Related Stochastic Processes},
YEAR = {1967},
PUBLISHER = {Wiley}
}
```

The first part, beginning with `@`, determines the type of entry; here a book. These entry types may be written in capitals or lower case letters or a mixture of both. The entry type itself is followed by a *key*, a unique identifier for the reference that will be used whenever it is referred to in the document; in the above it is `cramerleadbetter:1967`. It should be preceded by a `{` and followed by a comma. The *name:year* format is not mandatory, but many people find it useful.
The subsequent lines contain *field names* (`AUTHOR, TITLE`, etc) and values assigned to them enclosed in `{ }` or double quotation marks. The final line in the entry consists of a closing `}`, balancing the initial `{` on the first line, and indicating the end of the reference.

Entry for an article

In order to enter the information for an article in a journal, the required fields are author, title, journal and year, with optional fields for volume, number, pages, month and note. An example is

```
@ARTICLE{Baxrychwil:2005,
author={A. Baxevani and I. Rychlik and R. Wilson},
year= {2005},
title={A new method for modelling the space variability of
            significant wave height},
journal={Extremes},
volume={8},
pages={267--294}
}
```

Including the bibliography

There are many possible styles of bibliographies that can be used. We recommend the `rmd` style. To use this, download `rmd.bst` from the 'MAS6002 LaTeX Resources' page on MOLE and place it in the same directory as your LaTeX source file. Now enter into the LaTeX source file the lines

```
\usepackage{natbib}
\bibpunct[, ]{(}{)}{;}{a}{,}{,}
```

[2] In WinEdt see BibTeX Items in the Insert menu, and in TeXMaker see the Bibliography menu

in the preamble (the `\usepackage{natbib}` will already be present if you are using the template from previous sheet). Also enter the lines

```
\bibliographystyle{rmd}
\bibliography{databasename}
```

at the place in the main text where you wish the list of references to appear (*databasename*`.bib` here being the name of the `.bib` file). Note that you do *not* include the ".bib" part of the filename in the `bibliography` declaration.

Having included the above commands, to put a reference at any point in the source file use one of

```
\citet{key}
\citep{key}
```

which produce, respectively, a standard citation and a citation in brackets. In Table 4 we show the output and options for each. Note that in these commands *key* is the unique identifier of the publication as specified in the `.bib` file.

Sample command	Output
shown by `\citet{Baxrychwil:2005}`	shown by Baxevani et al., (2005)
shown by `\citet[pg.~8]{Baxrychwil:2005}`	shown by Baxevani et al., (2005, pg. 8)
results show `\citep{Baxrychwil:2005}`	results show (Baxevani et al., 2005)
results show `\citep[pg.~22]{Baxrychwil:2005}`	results show (Baxevani et al., 2005, pg. 22)
show `\citep[e.g.][]{Baxrychwil:2005}`	show (e.g. Baxevani et al., 2005)
show `\citep[e.g.][pg.~22]{Baxrychwil:2005}`	show (e.g., Baxevani et al., 2005, pg. 22)

Table 4: Citation types possible using the natbib package.

Finally to include the bibliography and the correct references you will need to do either of the following to the source code (*.tex).

- Select pdfLaTeX+makeIndex+BiBTeX as the compilation method (from the drop down menu next to the green arrow). if you then compile the .tex file **twice** the references should appear.

  If this does not work or your editor does not give that option then you should instead:

- 1. Select pdfLaTeX and compile once (doing this your compiler will realise it needs to include a bibliography).
  2. Select BiBTeX and compile once (this will pre-compile the bibliography separately).
  3. Select pdfLaTeX and compile twice (this will first include the bibliography in the compilation step and then secondly run the cross-referencing to include the references you are using).

Note: The package `natbib` referred to above produces amongst other things the style of references used in statistical and mathematical documents. It may be available already on your system, but if it isn't you will receive a warning message on attempting to LaTeX a source file referring to it. The warning will be accompanied by an invitation to download `natbib` from a suitable CRAN mirror site, which you should accept. If your TeXworks is failing to install `natbib` from this mirror then instead of choosing a random internet depository select the ftp site called mirror.ox from the options in the pop-up window.

Exercise

The file `bibexample.tex` on the 'MAS6002 LaTeX Resources' page in MOLE is an example of a LaTeX source file; it calls for `refdatabase.bib` as its bibliography database (also available in MOLE). Download `refdatabase.bib`, `bibexample.tex`, `rmd.bst` (from MOLE) and, if necessary, `natbib` (from CRAN). You should store `rmd.bst` and `natbib` in a directory on the WinEdt or TeXMaker search path, or in your working directory.

Compile `bibexample.tex` to produce a document containing citations and a references list. Add some of your own references to `refdatabase.bib` and modify `bibexample.tex` to generate a document incorporating them. Note that only the references you have cited appear in the references list, not all those in the `.bib` file.

If you now look at the .pdf output you should see that the citations have appeared along with an automatic bibliography which will include all the details of the references used.

Note: Make sure that the files you download from MOLE (bibexample.tex, refdatabase.bib and rmd.bst) are recognised as .tex, .bib and .bst files respectively by your computer. The easiest way to do this is probably downloading them by right clicking on them and selecting"save target

as". If your computer does not allow you to do this then an alternative is to copy and paste them into TeXworks (do not use notepad as they will by default be saved as .txt files) and saving them within that program choosing the 'all files' format with the suitable (.tex, .bib or .bst) extension. You should be able to tell what type of file your computer thinks they then are within the file manager. You will also need to remove the (1) within each filename (e.g. rmd(1).bst) if one has been added by MOLE.

10 Creating More Complex Tables

As we learnt earlier, the `tabular` environment is used to build the raw tables themselves. The basic command tells LaTeX what pattern of columns is needed, then the user has to build up the rows of the table line by line. Basic syntax is:

```
\begin{tabular}{cols}
... (individual row entries)
\end{tabular}
```

where *cols* formats the columns. The *cols* entry is a sequence of the following symbols:

`|`	meaning draw a vertical line from top to bottom of the table
`c`	meaning create a column whose entries are centred within it
`r`	meaning create a column whose entries are right-justified
`l`	meaning create a column whose entries are left-justified.

Individual row entries consist of the specific column items that you wish to print, separated by & characters and ended by \\. A horizontal line is drawn between rows by `\hline`.
For example the code

```
\begin{center}
\begin{tabular}{|l|crl|}
    \hline
                & dist &  climb   & time   \\  \hline
    Greenmantle  &  2.5 &   650    & 16.083 \\
    Carnethy     &  6.0 &  2500    & 48.350 \\
    Craig Dunain &  6.0 &   900    & 33.650 \\  \hline
\end{tabular}
\end{center}
```

produces

	dist	climb	time
Greenmantle	2.5	650	16.083
Carnethy	6.0	2500	48.350
Craig Dunain	6.0	900	33.650

Note that blank entries are allowed, as in the first line of the above. The `\begin{center}` and `\end{center}` are needed to centre the table on the page. LaTeX otherwise treats it as a giant letter and so tries to join it to the current line of text.
More elaborate dividing lines may be drawn with:

`\cline{`$i-j$`}`	which draws a horizontal line across columns $i-j$
`\multicolumn{`*n*`}{`*cols*`}{...}`	which spans the next *n* columns with *cols* format using the LaTeX commands given in ... (which could be plain text).

Thus

Fell race	Data		
	dist	climb	time
Greenmantle	2.5	650	16.083
Carnethy	6.0	2500	48.350
Craig Dunain	6.0	900	33.650

is produced by

```
\begin{center}
\begin{tabular}{lcrl}
     \multicolumn{1}{l}{Fell race} &  \multicolumn{3}{c}{Data}\\
                            &   dist &  climb   & time \\
         \cline{2-4}
               Greenmantle  &    2.5 &   650    & 16.083  \\
               Carnethy     &    6.0 &  2500    & 48.350  \\
               Craig Dunain &    6.0 &   900    & 33.650  \\  \cline{2-4}
\end{tabular}
\end{center}
```

When entering data which contain decimal points, it is aesthetically pleasing if the decimal point itself lines up all the way down a column. This may not automatically be the case if the numbers are of different lengths (try altering the time for Carnethy in the above example to 148.350). To make sure that the decimal points do line up with LaTeXline up you need to create separate columns for the integer (which should be right aligned) and the decimal part (which should be left aligned) and use the `@{.}` to fix a decimal point as the separator between them. Thus

Fell race	Data		
	dist	climb	time
Greenmantle	2.5	650	16.083
Carnethy	6.0	2500	148.350
Craig Dunain	6.0	900	3.650

is produced by

```
\begin{center}
\begin{tabular}{lcrr@{.}l}
      \multicolumn{1}{l}{Fell race} &  \multicolumn{3}{c}{Data}\\
                            &   dist &  climb   & \multicolumn{2}{c}{time} \\
         \cline{2-5}
               Greenmantle  &    2.5 &   650    & 16 & 083  \\
               Carnethy     &    6.0 &  2500    & 148 & 350  \\
               Craig Dunain &    6.0 &   900    & 3 & 650  \\  \cline{2-5}
\end{tabular}
\end{center}
```

Notes:

Two further commands sometimes useful in the *cols* specification are:

`@{`*text*`}` which inserts *text* in every line of the table between the two columns between which it is placed (and so can generate inter-column spacing of *width* by use in the form `@{\hspace{`*width*`}}`)

`*{`*n*`}{...}` equivalent to *n* copies of ...

Exercise

Use `tabular` to generate the following table:

Fell race	Data		
	dist	climb	time
Greenmantle	2.5	650	16.083
Carnethy	6.0	2500	48.350
Craig Dunain	6.0	900	33.650

noting the larger-than-standard space between the first two columns.

11 Placing Tables and Figures - Floats

When incorporating both Figures and Tables within documents we will want to put them within the `figure` or `table` environment respectively. The purpose of both is to help in placement of the graphic or table on the printed page. LaTeX will try to position a table or graph at the point in the text where the `tabular` or `includegraphics` command is issued. However, if the table or graph is too high to fit on the page after the text written there already, then the page is terminated and the table or graph is put on the next page, leaving empty space on the original page. The `table` and `figure` environments are designed to avoid this by allowing *floating* placement of tables and figures so that surrounding text can flow smoothly round them.

They are invoked by

`\begin{figure}`   *figure*   `\end{figure}`

	dist	climb	time
Greenmantle	2.5	650	16.083
Carnethy	6.0	2500	48.350
Craig Dunain	6.0	900	33.650

Table 5: Some Scottish Hill Races

and

`\begin{table}` *figure* `\end{table}`

Here *figure* consists of the `\tabular` or `\includegraphics` instructions that generate the table or graphic, plus possibly a `\caption` and/or a `\label` command as described below.
If a table or figure is not placed where you wish, some things to try are:

(i) make the table or figure slightly smaller – sometimes that's enough to allow it to fit; a figure may be re-sized by the arguments in `\includegraphics`, and a table by enclosing it in `{\small ...}` or even `{\footnotesize ...}`;

(ii) try moving the `\begin{figure}` or `\begin{table}` command slightly earlier or later in the sourcefile – again that may make enough space.

With a bit of experimentation one or other of these fixes usually works.
A caption for the table or graph may be produced by including

`\caption{`*caption_text*`}`

within the `\table` or `\figure` environment. This both assigns a number to the table or graph and prints a title of the form 'Table *n*: *caption_text*' or 'Figure *n*: *caption_text*'. If the `\caption` command comes before the *figure* instructions, the title is printed above the table or graph; otherwise, below.
To create a label for the table or graph so that it can be referred to from elsewhere in the document, include

`\label{`*name*`}`

anywhere within the `\caption` text. As usual you can then refer to the table or graph by using `\ref{`*name*`}` anywhere else in the document.
The following is an example of a typical use of the `\table` environment. The resulting table is shown in Table 5.

```
\begin{table}
{\small
\begin{center}
\begin{tabular}{|l|crl|}
     \hline
                 & dist &  climb   & time   \\  \hline
     Greenmantle &  2.5 &   650    & 16.083 \\
     Carnethy    &  6.0 &  2500    & 48.350 \\
     Craig Dunain &  6.0 &   900    & 33.650 \\  \hline
\end{tabular}
\end{center}
\caption{Some Scottish Hill Races \label{hills}}   }
\end{table}
```

12 Multi-line equation environments

Sometimes we may wish to typeset an equation where we are defining a matrix or a function and one side of the equation needs brackets that cover more than a single line.
For example,

```
\begin{equation*}
     \mathbf{X} =
     \left( \begin{array}{ccc}
          x_{11} & x_{12} & \ldots \\
          x_{21} & x_{22} & \ldots \\
           \vdots & \vdots & \ddots
```

```
        \end{array} \right)
    \end{equation*}
```

gives

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

The same environment can be used to typeset expressions that have one big delimiter by using a "." as an invisible right delimiter. Thus

```
    \begin{equation*}
         y = \left\{ \begin{array}{ll}
              a & \textrm{if $d>c$}\\
              b+x & \textrm{in the morning}\\
              l & \textrm{all day long}
         \end{array} \right.
    \end{equation*}
```

gives

$$y = \left\{ \begin{array}{ll} a & \textrm{if $d>c$}\\ b+x & \textrm{in the morning}\\ l & \textrm{all day long} \end{array} \right.$$

13 Page headings

To put a heading 'my registration no.' at the top of every page of a document, include

```
    \pagestyle{myheadings}
    \markright{my registration no.}
```

in the preamble. Use `\pagestyle{headings}` alone to put the section heading and page number at the top of every page – something that may be useful in dissertations.

14 Cross Referencing Pages

In addition to the cross-referencing of Tables and Figures described in §11 (and of equations as discussed in `template.tex`) the `\label` command may be used to cross-reference to sections and pages in a document. For example, the reference to §11 in the preceding sentence was produced by putting a `\label{floats}` entry in the LaTeX source code in the 'Floating Figures and Tables' section:

```
    \section{Floating Figures and Tables}
    \label{floats}
```

and then in the present section using

```
  ... described in \S\ref{floats} (and ...
```

to give the above (the `\S` symbol generating §).

The number of the page on which an item labelled with a `\label` command appears may be printed with the `\pageref` command. Thus

```
    \pageref{hills}
```

prints the number of the page (i.e. page 14) on which the LaTeX code used to generate the example table (in Section 11) was given.

A Getting LaTeX and Compilation

Once MiKTeX has been installed, you can find TeXworks in the MiKTeX folders.

A.1 Managed Desktop

A.1.1 Installing MiKTeX

1. On the desktop, double-click on the 'Software Centre' icon.
2. Select the MiKTeX application from the list and install it.
3. Click on the button at the bottom left of the screen, select MiKTeX and then TeXworks.

A.1.2 Compilation

Once you have created your ASCII text source code within TeXworks, to compile all you need do is click the large green arrow. It will ask you to save the document somewhere before it compiles (the pdf and other output files created during compilation will also be saved to this location). It is sensible to create a separate folder for each project as otherwise you will end up with a lot of confusing files.

Make sure that the compiler is set to pdfLaTeX.

A.2 LaTeX at home

A.2.1 Installation and Compilation

A free version of LaTeX is available for download at the MiKTeX website `http://miktex.org/`. This will also include a copy of TeXworks and so you can follow the compilation instructions as for the managed desktop

A.2.2 WinEdt and TeXnicCenter

While you can use any text editor to create your LaTeX files we recommend that you consider purchasing WinEdt (`http://www.winedt.com/`). In the author's opinion, this is an excellent program which interfaces well with LaTeX and provides easy-click buttons to help you typeset code, mathematical equations and compile everything. It has menus which will remind you of all the symbols you are ever likely to need as well as the syntax. It also makes bibliographies much simpler to create in providing templates for all the different types of articles or books to which you might want to refer.

There is also a freeware text editor called TeXnicCenter which can be downloaded at `http://www.texniccenter.org/`. This has also been recommended to the author although he has no direct experience of it himself.

References

[1] T. Oetiker, H. Partl, I Hyna, and E. Schlegl. The not so short introduction to LaTeX2e. Online, 2010.