Intermediate Skills for LaTeX:

Workshop Notes

Steven Miller

Department of Statistics

University of Auckland

Contents

List of Figures

List of Tables

Introduction

This set of notes is designed for people who already have some understanding of LaTeX and its syntactical structure. It provides the reader with a more advanced set of skills to tailor a document to suit the reader's needs. The techniques outlined in these notes make use of functions available in the simplest LaTeX document. There are a wide range of packages available for supplemental use which allow the techniques mentioned in these notes to be applied, as well as far more complex variations. These notes have been written with the intention of providing the reader with a versatile set of simple skills for formatting a document quickly, and giving the reader some knowledge about the methods applied by these extension packages.

By no means is this a comprehensive guide to all the layout facilities available in LaTeX, and the enthusiastic reader is encouraged to explore the internet or seek the advice of experienced collegues for instruction on more advanced areas. This set of notes does not describe how to write in TeX code, or how to alter predefined document or bibliographic template files.

Section 1 describes how a titlepage or main heading can be added to a document (see Section 1.1), and how to include a table of contents, as well as any lists of tables or figures (see Section 1.2).

Section 2 describes how to alter the layout of a page to suit the style of the document. This could include stretching the default margins to better fit the text or a figure, increase the margin allowance for the binding of a book (Section 2.1), or double-spacing lines for academic documents (Section 2.2).

Section 3 details how to include headers and footers on pages of the document, and how to write footnotes. It also illustrates how to change the style of the page numbers used.

Means of altering the counters used by LaTeX to number objects from pages to equations are addressed in Section 4.

In Section 5, the reader is instructed how to provide references to equations, sections, tables, figures etc. within a document.

Section 6 describes the process of creating a bibliographic record, citing entries, and creating a bibliography at the end of a document.

Finally, Section 7 gives a simple introduction to creating user-defined functions for use within a LaTeX document, which can be used to simplify the process of formatting a document.

This set of notes is completed by a set of basic exercises which practise some of the techniques illustrated herein.

1 Title Pages and Contents Pages

Most reports will be required to have a title of some sort, stating the author and the name of the document. Whether this title is situated at the top of the first page of the article, or on a separate page of the document is left to the discretion of the author. LaTeX provides the facility to include the author's name, the name of the document, the date of publication, and recognition of any contributions that the author wishes to acknowledge.

Longer documents may require a table of contents, and possibly a list of tables and figures.

1.1 Creating Title Pages

There are two common commands for constructing title pages.

The `titlepage` environment leaves the formatting of the titlepage to the user. The text inside the environment is displayed on a blank page with no page-number or header, where the following page is numbered as page one (that is, the title page does not figure in the number of pages of the document).

The `\maketitle` command creates a title page from information provided in the preamble. This information can include authors (`\author{...}`), a title (`\title{...}`), and the date of publication (`\date{...}`). Line breaks can be forced in the title and author information by using the double backslash, "\\". If the date element is omitted, the current system date is used on the title page, as if the command `\date{\today}` had been used. To create a title page without a date, call the date command with nothing inside the braces. It is also possible to make a footnote to the title to provide an acknowledgement to those who helped with the document. This is specified in the `\thanks{...}` function, which must be included *inside* either the `\author` or `\title` functions..

Once the relevent information has been recorded in the preamble, it remains to make a call to the command

```
\maketitle
```

at the beginning of the document (this command will generate the title page wherever it is placed, so should be placed ahead of the rest of the document). Note that `\maketitle` will make a separate title page for every document style *excpet* those of type `article`. Typically journal articles have their title at the top of the first page. In order to have a separate title page for an `article`, it is necessary to specify a pagebreak immediately after creating the title, with one of the commands `\newpage`, `\pagebreak` or `\clearpage`.

1.2 Contents Pages

LaTeX will automatically create a table of contents for you with the command

```
\tableofcontents
```

The table of contents is place where the command is called, which will typically be immediately after a titlepage. It is similarly possible to create lists of figures and tables

```
\listoffigures
\listoftables
```

Note that these commands do not start new pages once they have been generated. To start a new page after each of these commands it is necessary to call one of `\newpage`, `\pagebreak` or `\clearpage`.

2 Page Layout

Sometimes it will be necessary to alter the layout of a page by adjusting the margins and/or the spacing of the text. The following subsections describe how to achieve this using some of the basic length elements defined in LaTeX. This way of modifying the page properties is quick, but necessarily simple. For more advanced instruction on altering the page properties it is recommended that a more formal reference be advised, or precompiled package used which has been programmed in accordance with all the subtleties inherent in the default LaTeX style template.

In general, when we make a modification to the layout properties of a document, we want the changes to be document-wide, rather than for certain pages or parts of a page. Therefore it is usual to make the adjustments in the preamble, before executing the `\begin{document}` command. There are two useful ways of adjusting a length element

- `\setlength{\lengthElement}{length}`
- `\addtolength{\lengthElement}{length}`

The `\lengthElement` is the name of the stored quantity which you would like to modify (note that since it is already a defined quantity, the name of the element is preceded by a backslash), followed by the desired value. This value can be specified in any metric understood by LaTeX including millimetres (`mm`), inches (`in`) and points (`pt`).

Setting the length will change the value of that length to whatever new value is specified. The usefulness of adding to the length, rather than making an absolute change, is that the effect can be undone by adding the negative of the value at a later point in time, without ever having to know the original value of the length.

2.1 Page Margins

Often it is useful to adjust the margins of a document to suit the purpose for which the document is intended. Three of the most important attributes associated with the margins are

- `\oddsidemargin`
- `\evensidemargin`
- `\textwidth`

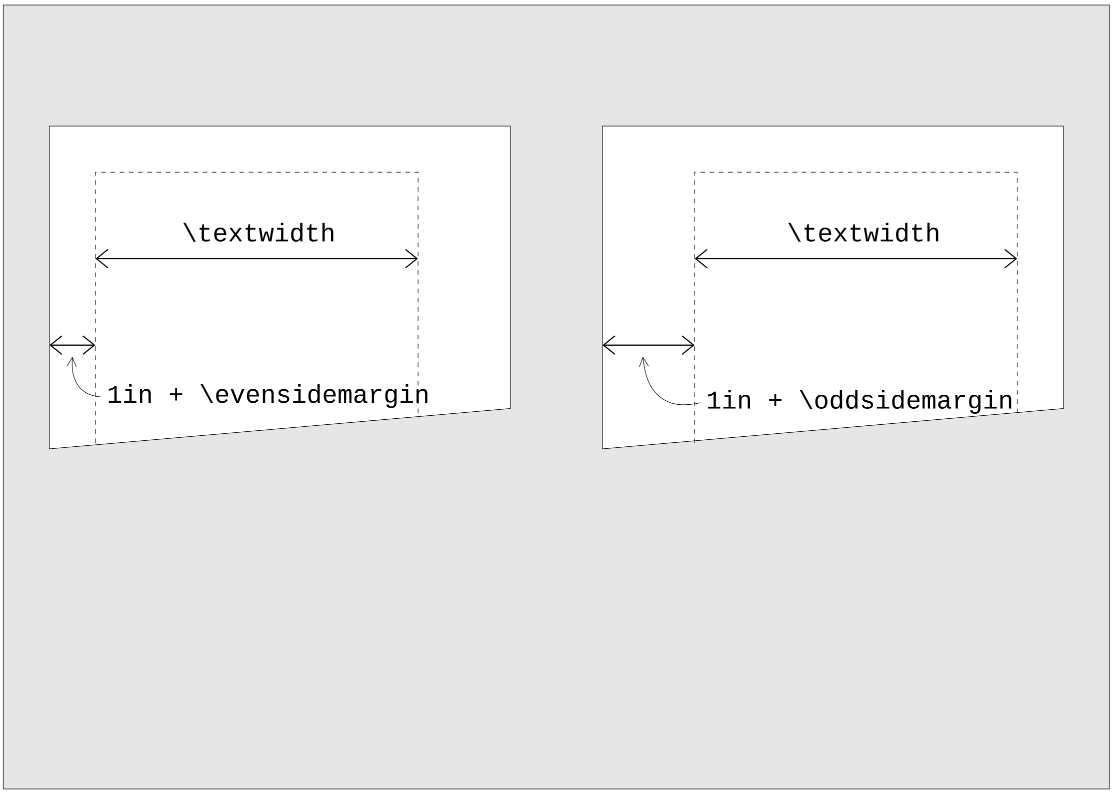

Figure 1: Simple layout of a two-sided document

See Figure 1 for a graphical representation of how these relate to the margins of the page.

The left-side margin of a regular page is governed by the `\oddsidemargin` attribute. Notice that the left margin is *always* offset by `1in`. That is, even when we set `\oddsidemargin` to zero, the left margin will still be one inch from the edge of the page. To reduce the distance of the left margin from the page, it is possible to assign a negative value to `\oddsidemargin`.

Despite the intuitivelly appealing symmetry, `\evensidemargin` *does not* govern the distance of the right-side margin from the edge of the page. Instead, this is controlled by the measure of the width of the text across the page, `\textwidth`. Increasing this length will shorten the right margin.

For example, to allow the document to stretch across the entire width of an A4 page (see Table 2.1 for the dimensions of common paper sizes) we would add to the preamble

```
\setlength{\oddsidemargin}{-1in}
\setlength{\textwidth}{210mm}
```

`\evensidemargin` only comes into play when we have specified in the preamble that we wish our document pages to be two-sided. This means that if the pages are to be bound, odd-numbered pages will require an extra long left margin, while even pages will require extra space on their right margin to avoid the binding covering up any of the writing.

There are other measures which govern the lengths of the top and bottom margins of the page, but these are tricky to manipulate as they are contingent on the presence or absence of headers and footers. On the whole it is usually better to either not mess with these, or use a precompiled package which is designed to allow for any complications which may exist.

2.2 Double Spacing

The minimum distance between two successive lines in a paragraph is stored as an element called `\baselineskip`. However, it is unreliable to change the line spacing directly using this element, as a change to the fontsize elsewhere in the document may unintentionally cause the `\baselineskip` attribute to reset to its default value. In order to permanently alter the line spacing of the whole document, we modify a scaling value to the `\baselineskip` element known as `\baselinestretch`. This applies a multiplier to whatever distance `\baselineskip` takes, and is 1 by default. Therefore, to achieve double spacing, add the following line to the preamble.

```
\renewcommand{\baselinestretch}{2}
```

Table 1: Table of common paper sizes in millimeters and inches.

Papersize	**Metric dimensions (mm)**			**Imperial dimensions (in)**		
2A	1189.0	x	1681.0	46.81	x	66.18
A0	841.0	x	1189.0	33.11	x	46.81
A1	594.0	x	841.0	23.39	x	33.11
A2	420.0	x	594.0	16.54	x	23.39
A3	297.0	x	420.0	11.69	x	16.54
A4	210.0	x	297.0	8.24	x	11.69
A5	148.0	x	210.0	5.83	x	8.24
A6	105.0	x	148.0	4.13	x	5.83
A7	74.0	x	105.0	2.91	x	4.13
A8	53.0	x	74.0	2.09	x	2.91
Super A3/B	330.2	x	482.6	13.00	x	19.00
Ledger (B)	279.4	x	431.8	11.00	x	17.00
Legal	215.9	x	355.6	8.50	x	14.00
Letter (A)	215.9	x	279.4	8.50	x	11.00
Executive	190.5	x	254.0	7.50	x	10.00

Notice that it is necessary to use the `\renewcommand` function, since `\baselinestretch` is not a length. Similarly, it is not necessary to specify units for the scaling factor.

3 Headers, Footers and Footnotes

By default, pages have no header and the footer holds the page number. Other entries in the header and footer of the page are possible using the `\pagestyle{style}` command. There are four style options:

- `plain`
- `empty`
- `headings`
- `myheadings`

The first three style types generate preset headers and footers, while the fourth style option allows the user to specify their own desired content.

The `plain` style is used by default, i.e. there is no header and the footer contains the page number. If you do not want page numbers displayed, then use the `empty` style which has no header or footer content. The `headings` style has no content in the footer, and the header content depends on the document style chosen with the `\documentclass` command. Generally this will be some arrangement of chapter/section title and page number.

To customise the content in the header, choose the `myheadings` style. To specify what the content should be, use the command `\markboth{left-page header}{right-page header}` for headers for both left-sided and right-sided pages, or alternatively `\markright{header}` for headers for right-sided pages only (note: if you are using a style which does not have orientations for the pages, such as `article`, *all* the pages are counted as right-sided for the purposes of headers). Typically the page number will be displayed alongside this customised header content. By specifying the header content in the preamble, you can give every page the same hearer. Alternatively, you could choose to use the `\markboth`/`\markright` command on each page to give the headers page specific content (such as chapter headings). This is particularly useful when used in conjunction with user-specified functions which can pass the section title for a particular page to the header commands (for an example, see Section 7).

It is also possible to specify the page style for any particular page, while leaving the style for all the other pages unchanged. For this, use the command `\thispagestyle{style}`

On a related note, it is also possible to change the style of the page-numbering. This can be useful to suit the style of the sectioning of the document, or perhaps to suit the style demanded by a publication. To do this, use the command `\pagenumbering{style}`. The style options are

- `arabic`
- `Roman` or `roman`
- `Alpha` or `alpha`

`arabic` gives you plain arabic numerals (1, 2, 3, and so on), `Roman`/`roman` uses roman numerals (i, ii, iii, and so on), either uppercase or lowercase, and `Alpha`/`alpha` returns alphabetic sequencing (a, b, c, etc), again in uppercase or lowercase. To specify which number to start counting from, see the section on counters (Section 4).

Finally, to generate footnotes in the text, use the command `\footnote{text}` immediately after the word/sentence you would like to be footnoting. A numerical counter will appear next to the reference, and the `text` will appear at the bottom of the page. If you would prefer the footnotes to be indicated by symbols rather than numbers, see the section on counters (Section 4).

4 Counters

Everything that LaTeX automatically increments has an associated counter which keeps track of which number is to come next in the sequence. These counters have the same names as the objects for which they provide an index. For example, the number of the next section will have be stored as the counter `section`. Other counters include

`chapter`
`section`
`subsection`
`paragraph`
`page`
`equation`
`figure`
`table`
`footnote`

It is also possible to access the counters used by the `\enumerate` environment using the counters `enumi,` `enumii,` `enumiii,` and `enumiv` for up to four nested enumerated lists.

In addition, it is possible to create your own counters using the command

```
\newcounter{mycounter}[counter]
```

This creates a counter called whatever name is specified as `mycounter` which can be modified as any other counter. Note that all counters are intialised with a value of zero, which may be invalid depending on the form of display the counter takes (e.g. there is no 0th alphabet letter). Note also that this counter will not be incremented automatically, so the user must indicate when they would like this counter to change. The optional argument `counter` is the name of any other counter. When this counter is incremented, `mycounter` will be reset.

To set a counter to be a particular value, use the command

```
\setcounter{counter}{value}
```

This sets the value of the counter specified by `counter` to the value given by `value`. Note that this way of setting the value of the counter is generally most appropriate when the current value of the counter is irrelevant.

To manually increment a counter, use the function

```
\addtocounter{counter}{value},
```

which adds the value given by `value` to `counter`. `value` is allowed to be a negative integer, which causes the counter to be decremented. This method of changing the value of the counter is most useful when the new value depends on the current value of the counter, but this does not require the current value of the counter to be known by the user.

A useful function that can be used with counters is `\value{counter}`, which will return the current value of the counter which is specified as an argument.

LaTeX stores the current value of counters as integers. However, the displayed counters can take the form of roman or arabic numerals, alphabetic characters, or symbolic characters.

To display a counter as an alphabetic character, use the commands

```
\Alph{counter}, \alph{counter}
```

for uppercase and lowercase characters respectively.

Similarly, for an uppercase or lowercase roman numeral, use the commands

```
\Roman{counter}, \roman{counter}
```

For a typical numeral, use the function

```
\arabic{counter}
```

Counters can also be displayed as a symbolic character from a set of nine symbols, respectively $*$, †, ‡, §, ¶, ‖, $**$, ††, and ‡‡ . This is particularly useful for footnotes, but since the symbols have no generally understood order, would not usually be appropriate for counters for which an increment is informative. To achieve this display,

```
\fnsymbol{counter}
```

Note that the value for this counter must be between 1 and 9 inclusive.

If you wish the command `\footnote` to generate the footnote reference in a different style, it is necessary to add a line to the preamble. For example, if you would like alphabetic footnote characters, you would insert the line

```
  \renewcommand{\thefootnote}{\Alph{footnote}}
```

A call to `\footnote` would then produce the appropriate alphabetic character as a reference. Similarly for roman characters, or symbols.

5 Making Reference to Objects

Inevitably it will be necessary to point the reader to a particular chapter, section, table, figure etc in your document. To do this it is necessary to know the index of that object, for example "Chapter 6" or "Table 2.2.1". LaTeX handles this operation particularly well by keeping track of the indicies of objects by means of counters (see Section 4) so that the author does not have to. Also, if the order of the objects is altered, perhaps by adding a new picture at the beginning of a document or switching two thematical sections around, LaTeX will take this in to consideration and modify all references to these objects accordingly.

In order for LaTeX to keep track of these objects, it is necessary to assign each of them a *unique* label. This is done by using the following command at some point between the `\begin{...}` and `\end{...}` portions of the object environment:

```
\label{uniqueKey}
```

It is necessary for the label to be unique so that LaTeX can distinguish this object from any other in the document. The label can be absolutely anything, but it can be good practice to indicate what sort of object the label is associated with, and a description of the object. For example,

- `fig:bargraph` for a figure of a bargraph;
- `sec:apples` for a section of an article dealing with apples;
- `tab:results` for a table of experimental results.

By using informative labels similar to these, it can be easier to remember which key is associated with a particular object when it comes time to reference this object. To do this, use the following command

```
\ref{uniqueKey}
```

This will return the index number of the object which has the same label. Note that LaTeX will not know what type of object is being referred to, so it is necessary to specify this yourself. For example, say we would like to point the reader to a table of prices and stock quantities. First, we will have assigned a label to the table

```
\begin{table} \label{tab:prices}
\begin{tabular}
...
```

We can then reference this table anywhere in the document

```
[...] , please see Table~\ref{tab:prices}.
```

The tilde character between "Table" and the reference is a non-breakable empty space. That is, LaTeX will put a space between "Table" and the reference, but is not allowed to put "Table" at the end of one line and the index at the beginning of the next; the whole reference is treated as one word despite there being a space in the middle. This is simply good practice, and is not essential for the reference to work.

If we were now to add or subtract another table before this table, LaTeX would increase or decrease the index of the table as appropriate and change any reference to this table to the amended value.

Because LaTeX assumes the responsibility for maintaining the index numbers of the objects and keeping them up to date, it will usually be necessary to compile the document twice before the references are shown properly. This is because a reference could be made before an object is specified in the document, so LaTeX needs one pass through the code to assign the correct index values to the objects before it can associate these values with the appropriate references, which occurs during the second pass through the code.

6 Bibliographies

There are several ways to produce a bibliography in LaTeX, but we will be restricting attention to the use of the BibTeX package. BibTeX requires the construction of only one bibliography database file that can then be accessed by any LaTeX document. BibTeX will only extract those references which have been cited in the document. In this way, it is possible to construct a master bibliography file which can serve for any document.

BibTeX requires two files to operate. The first is a style file, which tells BibTeX how to format the entries in the bibliography list, and the second is the database with the information about each reference. The following description will explain how to obtain these files.

To create a reference within a document, it is necessary to use the `\cite` command. This tells LaTeX to indicate that the preceding text is associated with a particular reference. The exact manner of this indication depends on the style of bibliography that is being used. For example, it could be a superscript number, or maybe a number enclosed in square brackets in line with the text. The reference number does not need to be specified as BibTeX will work this out when it constructs the bibliography, since the exact order of the references in the bibliography can also depend on the style (e.g. the alphabetic order of the first authors, the order in which the articles were cited etc). The only information required is the label of the desired reference in the database. This label is at the author's discretion, but it is often handy to make it informative about the object being referred to, for example a contraction of the first author and the date of publication. Thus if we were citing evidence from a 1998 paper by F Katz, we might have given this paper the label `katz98` in the database file. We could then apply the citation with the command

```
\cite{katz98}
```

6.1 Bibliography Style Files

There are a multitude of ways that bibliographies can be styled. There are four standard styles which are included with any distribution of BibTeX. These are

- `alpha`: the cited works are ordered alphabetically in the bibliography, and are indicated in the text by taking either the first three characters of a single author's surname, or a combination of the initials of multiple authors surnames, and concatenating this with the last two digits of the year of publication within square brackets.

- `plain`: the cited works are ordered alphabetically in the bibliography, and are indicated in the text by plaing their numerical order in the bibliography in square brackets in the text (so the citation numbers in the text will not necessarily start at 1)
- `unsrt`: the cited works are ordered in the order they appear in the text in the bibliography, and are indicated in the text by plaing their numerical order in the bibliography in square brackets in the text (so the citation numbers in the text will start at 1 and increase monotonically).
- `abbrv`: this is similar to the `plain` style, but the information about the cited wroks in the bibliography are abbreviated.

Many publications require bibliographies to be formatted a very particular way. BibTeX copes with this variation by allowing the user to specify a non-standard style file which provides the template from which bibliographic items are created. These style files have the file extension "`.bst`". It is quite an advanced skill to be able to write a style file, and a subject that is not covered here. However, it is generally easy to find a style which is suitable for your purpose from the wide range of files that are available on the internet. Indeed, if a publication requires a certain format for bibliographies, it will usually provide the style template itself.

To implement the bibliography style file, you need to tell LaTeX which style you are using immediately before asking LaTeX to construct the bibliography. Say we want to use a style file called `genericStyle.bst`. First, it is necessary to copy this style file to the current directory (or else, include the path information where the file is stored in the following command). Then at the point we want the bibliography to be placed, we would include the command

```
\bibliographystyle{genericStyle}
```

Notice that if the style file has the file extension "`.bst`", it is not necessary to apply this extension in the LaTeX code.

6.2 Bibliography Database

It is necessary to provide a database file which has all the necessary information about each reference, as well as a unique label which will be used in the document to refer to each reference when citing them. These database files typically have the extension "`.bib`". A database file consists of a set of separate objects, each holding the information about a bibliographic element. These element can be one of many types, including

- journal articles
- books
- technical reports
- theses
- unpublished documents

Each type of element has a particular format of display in the bibliography, and requires certain information. Some of this information is mandatory, while other fields are optional.

Each element in a bibliography database is preceeded by an "@" followed by the element type that is deemed most suitable (note: although the element type determines how the document is displayed in the bibliography, it is usually not too important to get the type right - if you're unsure, choose the type which can include the most important bits of information, or alternatively use the `misc` type). The information about the element is enclosed within braces. For example, to include a journal article we would write

```
@article{ ... }
```

The first argument must be the unique key used to refer to this object. Other required fields for an article are `author`, `title`, `journal`, and `year`, while optional fields include `volume`, `number`, `pages`, `month`, and `note`.

Thus we could get away with

```
@article{carle79,
         author = "Carle, E.",
         title = "The Hungry Caterpillar",
         journal = "Entymological Ecology",
         year = "1979"}
```

although there is scope to provide a lot more information. Notice the layout of the above entry is just for ease of reading - the whole entry could be on one line, so long as there are commas between the fileds. Also, it is not strictly necessary for numbers to be in quotation marks. Also, had there been multiple authors it would have been necessary to separate each by an "and".

Using the `plain` bibliography style, citing the above article would appear as:

[1] E. Carle. The hungry caterpillar. *Entymological Ecology,* 1979.

Notice that the journal name has been italicised, the authors name has the surname last and the title of the article has a capitalisation only at the beginning. These all conform to the style template dictated by the `plain` style. If these alterations are disagreeable, it is necessary to use another style, except in the case of the capitalisation, where it is possible to preserve the original format by putting the capital letters in braces, i.e. {`[...]  title = "The {H}ungry {C}aterpillar" [...]`}. Other such modifications are possible by using other such TeX escape sequences, but they are not described here.

See Table 6.2 for a list of other biblography entity types.

Table 2: Table of BibTeX object types.

Entity Type	**Required Fields**	**Optional Fields**
`article`	author, title, journal, year	volume, number, pages, month, note
`book`	author or editor, title, publisher, year	volume or number, series, address, edition, month, note
`booklet`	title	author, howpublished, address, month, year, note
`inbook`	author or editor, title, chapter and/or pages, publisher, year	volume or number, series, type, address, edition, month, note
`incollection`	author, title, booktitle, publisher, year	editor, volume or number, series, type, chapter, pages, address, edition, month, note
`inproceedings`	author, title, booktitle, year	editor, volume or number, series, pages, address, month, organization, publisher, note
`manual`	title	author, organization, address, edition, month, year, note
`mastersthesis`	author, title, school, year	type, address, month, note
`misc`	none	author, title, howpublished, month, year, note
`phdthesis`	author, title, school, year	type, address, month, note
`proceedings`	title, year	editor, volume or number, series, address, month, organization, publisher, note
`techreport`	author, title, institution, year	type, number, address, month, note
`unpublished`	author, title, note	month, year

Once the bibliography database file is ready, it is included in the document using the command

```
\bibliography{myBib}
```

where `myBib` is the name of the bibliography style. Again note that using the typically file extension, it is not necessary to specify the extension in the LaTeX code.

In order for LaTeX to combine with BibTeX to process all the directive correctly, it is necessary to compile the code in a specific order. For a LaTeX document called `myDocument.tex`, that order is

```
1. latex myDocument

2. bibtex myDocument

3. latex myDocument

4. latex myDocument
```

That is, it is necessary to run LaTeX once, then BibTeX once, then LaTeX twice more before the bibliography displays properly. Do not be overly concerned if there are errors involving the citations at the beginning of this sequence as this is normal.

7 User Defined Functions

One of the very useful features of LaTeX is the ability to create your own functions to perform several sets of tasks simulataneously. To do this, we must define our new command with the function

```
\newcommand{\name}[numberOfArguments]{commands}
```

The `\name` of the new command must be unique and not match any other command that could be used by LaTeX in this document. The `\name` must also begin with a backslash, “\”. The `numberOfArguments` tells LaTeX how many separate pieces of information can be given to the command. A maximum of 9 arguments is allowed. LaTeX will use these pieces of information in the next part of the function, where each piece of information is referenced by the order in which it appears when the function is called. By default, no arguments are passed to the function. All the `commands` specified in the next part of the function are called when the function is run. The nth argument can be referred to inside the function as “$\#n$”.

For example, suppose we would like to design a header for our document which includes the name of the section, the name of the document and the page number for each page. First, we must choose the pagestyle `myHeadings`, which allows customised headers as described in Section 3. The title of the document will not be changing, so we do not need to include this as an argument to the function. The page numbering is done automatically by LaTeX. We will want the name of the section to change whenever we start a new section. Therefore, in the preamble we define the new command

```
\newcommand{\mySection}[1]{
  \section{#1}
  \markright{Section \arabic{section}: #1 \hfill
              Intermediate Skills for \LaTeX \hspace{1in}}
}
```

This defines a new function `\mySection` that takes only one argument. That argument is used by the traditional `\section` function as the name for a new section which starts when our new function is called. Additionally, when this new function is called, the heading of the page is modified to include the information we wanted.

Exercises

This series of exercises is intended to provide practise with many of the techniques outlined in this set of notes. The purpose is to create a document following the directions given in the exercises.

1. (a) Create a document of type `article`.
   (b) Specify the use of double spacing.
   (c) Decrease the margin sizes by 1cm on each side (hint: this will allow the text to stretch *2cm* further, 1cm on either side).
2. (a) Use the `\title{}` command to give the paper the title "Latex Exercises".
   (b) Use the `\author{}` command to record your name and the department with which you most closely associate yourself (practise separating your name and department onto different lines).
   (c) Include today's date.
   (d) Construct the titlepage.
3. (a) Create three sections filled with anything you want.
   (b) Label each section.
   (c) Add some footnotes.
   (d) At the end of each section, refer to the previous section (refer to Section 3 from Section 1); for example, the last line in Section 2 should read "Please refer to Section 1", or something similar.
4. (a) Create a table-of-contents immediately after the title page.
   (b) If you haven't done so already, compile the document and look at the output.
   (c) Cut and paste Section 3 to immediately follow the table-of-contents, then recompile the document (notice how the references and the table-of-contents update themselves).
5. (a) Construct a bibliography database file called `myBib.bib` with the following entries:
   - The book "Jonathon Livingston Seagull" by Richard Bach, published by Avon in 1976.

- "Transcendental Aeronautics" by Jackson Sparrow, Cheryl Crow, and Chick Korea; found in *Journal of the Society of Philosophical Ornithologists*, volume 4, March 1996, pages 997-1016.
- C. Robin's Masters thesis, "The eagle has taken off again: population dynamics of the Northern Californian Bald Eagle", submitted to Beakeley University in 1987.
- "One Flue Over a Cuckoo's Nest" by A. V. Ian, chapter 9 in the collected articles "Birds of a Feather: Avian Heating Strategies in the 21st Century", published by Tookan in 2003.

(b) Cite a selection of these documents in your article.

(c) Create a bibliography in the `plain` style.