Essential LaTeX ++

Jon Warbrick
with additions by David Carlisle, Michel Goossens, Sebastian Rahtz, Adrian Clark
[Note: Annotations for LaTeX 2.09 Users by Brian Junker –BJ]

June 1995

Contents

1 Introduction **2**

2 How Does LaTeX Work? **3**

3 A Sample LaTeX File **3**

1 Simple Text **5**

4 Document Classes and Options **7**

5 Environments **8**

6 Type Styles **10**

7 Sectioning Commands and Tables of Contents **10**

8 Producing Special Symbols **11**

9 Titles **12**

10 Tabular Material **12**

11 Tables and Figures **14**

12 Cross-References and Citations **14**

13 Mathematical Typesetting **15**

14 What about $'s? **17**

15 Letters **18**

16 Errors **19**

17 A Final Reminder **21**

A Mathematical Symbols **22**

B Horrible Mathematical Examples to Study **27**

C Defining New Commands and Environments **29**

D Creating and Importing Figures **33**

E Whitespace and Hyphenation **35**

List of Tables

1 Greek letters 22
2 Binary operation symbols 22
3 Relation symbols 22
4 Delimiters 23
5 Large delimiters 23
6 Arrow symbols 23
7 Miscellaneous symbols 23
8 Log-like symbols 24
9 Math mode accents 24
10 Variable-sized symbols 24
11 LaTeX math constructs 24
12 AMS miscellaneous symbols 24
13 AMS Greek and Hebrew 25
14 AMS delimiters 25
15 AMS arrows 25
16 AMS negated arrows 25
17 AMS binary operators 25
18 AMS binary relations 26
19 AMS negated binary relations 26

1 Introduction

This document is an attempt to give you all the *essential* information that you will need in order to use the LaTeX document preparation system[1].[1]. Only basic features are covered, and a vast amount of detail has been omitted. In a document of this size it is not possible to include everything that you might need to know, and if you intend to make extensive use of the program you should refer to a more complete reference. Attempting to produce complex documents using only the information found below will require much more work than it should, and will probably produce a less than satisfactory result.

The main reference for LaTeX is *The* LaTeX *User's Guide and Reference Manual* by Leslie Lamport[1]. This contains all the information that you will ever need to know about the program, and you will need access to a copy if you are to use LaTeX seriously.

[1] For those readers who have some familiarity with LaTeX, this document assumes the LaTeX2$_\varepsilon$ release of January 1994 or later. *[Note: Remarks on differences between* LaTeX2$_\varepsilon$ *and* LaTeX *2.09 are noted like this in this document. –BJ]*

Both the Manual and this document avoid mentioning anything that depends on the particular computer system that you are using. This is because LaTeX is available on a number of systems and they all differ in one way or another. Instead, they both refer to a *local guide* for their particular system.

2 How Does LaTeX Work?

In order to use LaTeX, you generate a file containing both the text that you wish to typeset and instructions to tell LaTeX how you want it to appear. You will normally create this file using your system's text editor. You can give the file any name you like, but it should end "`.tex`" to identify the file's contents. You then get LaTeX to process the file, and it creates a new file of typesetting commands; this has the same name as your file but the "`.tex`" ending is replaced by "`.dvi`". This stands for 'Device Independent' and, as the name implies, this file can be used to create output on a range of printing devices. Your *local guide* will go into more detail.

Rather than encourage you to dictate exactly how your document should be laid out, LaTeX instructions allow you to describe its *logical structure*. For example, you can think of a quotation embedded within your text as an element of this logical structure: you would normally expect a quotation to be displayed in a recognizable style to set it off from the rest of the text. A human typesetter would recognize the quotation and handle it accordingly, but since LaTeX is only a computer program it requires your help. There are therefore LaTeX commands that allow you to identify quotations and consequently allow LaTeX to typeset them correctly.

[Note: LaTeX is currently undergoing revisions to streamline some of its features. The most visible corrections are in the way that document styles are declared, and the way that font styles are declared, within a LaTeX document. The current "old" version of LaTeX is LaTeX 2.09. The first revision incorporating these changes is LaTeX2ε, and the final "production" version of the new LaTeX will be LaTeX 3.0. For most routine documents, virtually the only differences are: (1) LaTeX 2.09's `\documentstyle` *command has become LaTeX2ε's* `\documentclass` *command (but it works essentially the same way); and (2) commands to set type styles have changed (see table on page 10). –BJ]*

Fundamental to LaTeX2ε is the idea of a *document class* that determines exactly how a document will be formatted. *[Note: In LaTeX 2.09 this is called a* document style*, and works essentially the same way. –BJ]* LaTeX provides standard document classes that describe how common logical structures (such as quotations) should be formatted. You may have to supplement these styles by specifying the formatting of logical structures peculiar to your document, such as mathematical formulae. You can also modify the standard document classes or even create an entirely new one, though you should know the basic principles of typographical design before creating a radically new style.

There are a number of good reasons for concentrating on the logical structure rather than on the appearance of a document. Doing so prevents you from making elementary typographical errors in the mistaken belief that they improve the æsthetics of a document—you should remember that the primary function of document design is to make documents easier to read, not prettier. It is also more flexible, since you need only alter the definition of the quotation style to change the appearance of *all* the quotations in a document. Most important of all, logical design encourages better writing. A visual system makes it easier to create visual effects rather than a coherent structure; logical design encourages you to concentrate on your writing and makes it harder to use formatting as a substitute for good writing.

3 A Sample LaTeX File

Have a look at the example LaTeX file in Figure 1. It is a slightly modified copy of the standard LaTeX example file `small.tex`. Your local guide will tell you how you can make your own copy of this file. The

```
% SMALL.TEX -- Released 5 July 1985
% USE THIS FILE AS A MODEL FOR MAKING YOUR OWN LaTeX
% INPUT FILE. EVERYTHING TO THE RIGHT OF A  %  IS A
% REMARK TO  YOU AND IS IGNORED BY LaTeX.
%
% WARNING!  DO NOT TYPE ANY OF THE FOLLOWING 10 CHARACTERS
% EXCEPT AS DIRECTED:       &   $   #   %   _   {   }   ^   ~   \

\documentclass[11pt]{article}                          % LaTeX 2.09: \documentstyle[11pt]{article}

\begin{document}   % YOUR INPUT FILE MUST CONTAIN A \documentclass OR \documentstyle
                   % LINE, AS WELL AS \begin{document} AND \end{document} LINES.
\section{Simple Text}                                  % THIS COMMAND MAKES A SECTION TITLE.

Words are separated by one or    more      spaces.  Paragraphs are
    separated by one or more blank lines.  The output is not affected
by adding extra spaces or extra blank lines to the input file.


Double quotes are typed like this: ``quoted text''.
Single quotes are typed like this: `single-quoted text'.

Long dashes are typed as three dash characters---like this.

Italic text is typed like this: \emph{this is italic text}.   % LaTeX 2.09: {\em this is italic text}.
Bold   text is typed like this: \textbf{this is  bold  text}. % LaTeX 2.09: {\bf this is bold text}.

\subsection{A Warning or Two}                          % THIS MAKES A SUBSECTION TITLE.

If you get too much space after a mid-sentence period---abbreviations
like etc.\ are the common culprits)---then type a backslash followed by
a space after the period, as in this sentence.

Remember, don't type the 10 special characters (such as dollar sign and
backslash) except as directed!  The following seven are printed by
typing a backslash in front of them:  \$  \&  \#  \%  \_  \{  and  \}.
The manual tells how to make other symbols.

\end{document}                  % THE INPUT FILE ENDS LIKE THIS
```

Figure 1: A Sample LaTeX File

1 Simple Text

Words are separated by one or more spaces. Paragraphs are separated by one or more blank lines. The output is not affected by adding extra spaces or extra blank lines to the input file.

Double quotes are typed like this: "quoted text". Single quotes are typed like this: 'single-quoted text'.

Long dashes are typed as three dash characters—like this.

Italic text is typed like this: *this is italic text.* Bold text is typed like this: **this is bold text**.

1.1 A Warning or Two

If you get too much space after a mid-sentence period—abbreviations like etc. are the common culprits)—then type a backslash followed by a space after the period, as in this sentence.

Remember, don't type the 10 special characters (such as dollar sign and backslash) except as directed! The following seven are printed by typing a backslash in front of them: $ & # % _ { and }. The manual tells how to make other symbols.

Figure 2: The Result of Processing the Sample File

line numbers down the left-hand side are not part of the file, but have been added to make it easier to refer to various portions. Also have a look at Figure 2 which shows, more or less, the result of processing this file.

3.1 Running Text

Most documents consist almost entirely of running text—words formed into sentences, which are in turn formed into paragraphs—and the example file is no exception. Describing running text poses no problems, you just type it in naturally. In the output that it produces, LaTeX will fill lines and adjust the spacing between words to give tidy left and right margins. The spacing and distribution of the words in your input file will have no effect at all on the eventual output. Any number of spaces in your input file are treated as a single space by LaTeX. It also regards the end of each line as a space between words (see lines 15–17) of the example. A new paragraph is indicated by a blank line in your input file, so don't leave any blank lines unless you really wish to start a paragraph.

LaTeX reserves a number of the less common keyboard characters for its own use. The ten characters

```
# $ % & ~ _ ^ \ { }
```

should not appear as part of your text—if they do, LaTeX will get confused.

3.2 LaTeX Commands

There are a number of words in the file that start with '\' (see lines 9, 11 and 13) of the example. These are LaTeX *commands* and they describe the structure of your document. There are several things that you should realize about these commands:

- All LaTeX commands consist of a '\' followed by one or more characters.
- LaTeX commands should be typed using the correct mixture of upper- and lower-case letters. `\BEGIN` and `\Begin` are *not* the same as `\begin`.

- Some commands are placed within your text. These are used to switch things, like different typestyles, on and off. The `\em` command is used like this to emphasize text, normally by changing to an *italic* typestyle (see line 25 of the example). The command and the text are always enclosed between '`{`' and '`}`'—the '`{\em`' turns the effect on and and the '`}`' turns it off. In LaTeX2ε, All the typeface-changing switches like this have a corresponding *command* with an argument — all the possibilities are shown on page 10.

- There are other commands that look like

  ```
  \command{text}
  ```

  In this case, `text` is called the "argument" of the command. The `\section` command is like this (see line 13 of the example). Sometimes you have to use curly brackets '`{}`' to enclose the argument, sometimes square brackets '`[]`', and sometimes both. There is method behind this apparent madness, but for the time being you should be sure to copy the commands exactly as given.

- When a command's name is made up entirely of letters, you must make sure that the end of the command is marked by something that isn't a letter. This is usually either the opening bracket around the command's argument, or a space. When it's a space, that space is always ignored by LaTeX. We shall see later that this can sometimes be a problem.

3.3 Overall Structure

There are some LaTeX commands that must appear in every document. The actual text of the document always starts with a `\begin{document}` command and ends with an `\end{document}` command (see lines 11 and 39) of the example. Anything that comes after the `\end{document}` command is ignored. Everything that comes before the `\begin{document}` command is called the *preamble*. The preamble should contain only LaTeX commands to describe the document's style.

One command that must appear in the preamble is the `\documentclass` command (see line 9 of the example). *[Note: In* LaTeX *2.09 this is the* `\documentstyle` *command. –BJ]* This command specifies the overall style for the document. Our example file is a simple technical document and uses the `article` style, modified to print in eleven-point fonts. There are other styles that you can use, as you will discover in Section 4.

3.4 Other Things to Look At

LaTeX can print both opening and closing quote characters, and can manage either of these either single or double. To do this, it uses the two quote characters from your keyboard: ` (which sometimes resembles a grave accent or back-quote) and ' (apostrophe). You type these characters once for single quote (see line 21 of the example), and twice for double quotes (see line 20 of the example). The double quote character itself " is almost never used.

LaTeX can produce three different kinds of dashes. A long dash, for use as a punctuation symbol, as is typed as three dash characters in a row, like this '`---`' (see line 23 of the example). A shorter dash, used between numbers as in '10–20', is typed as two dash characters in a row, while a single dash character is used as a hyphen.

From time to time, you will need to include one or more of the LaTeX special symbols in your text. Seven of them can be printed by making them into commands by proceeding them by backslash (see line 36 of the example). The remaining three symbols can be produced by more advanced commands, as can symbols that do not appear on your keyboard such as †, ‡, §, £, ©, ♯ and ♣.

It is sometimes useful to include comments in a LaTeX file, to remind you of what you have done or why you did it. Everything to the right of a `%`-sign is ignored by LaTeX, and so it can be used to introduce a comment.

4 Document Classes and Options

There are four standard document classes available in LaTeX:

`article` intended for short documents and articles for publication. Articles do not have chapters, and when `\maketitle` is used to generate a title (see Section 9), the title appears at the top of the first page rather than on a page of its own.

`report` intended for longer technical documents. It is similar to `article`, except that it contains chapters and the title appears on a page of its own.

`book` intended as a basis for book publication. Page layout is adjusted assuming that the output will eventually be used to print on both sides of the paper.

`letter` intended for producing personal letters. This style will allow you to produce all the elements of a well laid out letter: addresses, date, signature, etc..

These standard styles can be modified by a number of *options*. They appear in square brackets after the `\documentclass` command *[Note: in LaTeX 2.09 this is* `\documentstyle` *–BJ]*. Only one class can be used in a document but you can have more than one option, in which case their names should be separated by commas. The standard options are:

`11pt` prints the document using eleven-point type for the running text rather that the ten-point type normally used. Eleven-point type is about ten percent larger than ten-point.

`12pt` prints the document using twelve-point type for the running text rather than the ten-point type normally used. Twelve-point type is about twenty percent larger than ten-point.

`twoside` causes documents in the article or report styles to be formatted for printing on both sides of the paper. This is the default for the book style.

`twocolumn` produces two columns of text on each page.

`titlepage` causes the `\maketitle` command to generate a title on a separate page for documents in the `article` style. A separate page is always used in both the `report` and `book` styles.

There is a further option which is very useful in European countries. The `a4paper` option causes the output in all of the standard styles to be adjusted to fit correctly on A4 paper. (LaTeX was designed in America where the standard paper is shorter and slightly wider than A4; without this option you will find that your output looks a little strange.)

[Note: In LaTeX2ε you specify options and document classes using the `\documentclass` *command. In LaTeX 2.09 the same specification is made with the* `\documentstyle` *command. For example,*

LaTeX2ε: `\documentclass[12pt,titlepage]{article}`
LaTeX *2.09:* `\documentstyle[12pt,titlepage]{article}`

–BJ]

5 Environments

We mentioned earlier the idea of identifying a quotation to LaTeX so that it could arrange to typeset it correctly. To do this you enclose the quotation between the commands `\begin{quotation}` and `\end{quotation}`. This is an example of a LaTeX construction called an *environment*. A number of special effects are obtained by putting text into particular environments.

5.1 Quotations

There are two environments for quotations: `quote` and `quotation`. `quote` is used either for a short quotation or for a sequence of short quotations separated by blank lines:

US presidents have been known for their pithy remarks:

> The buck stops here.
>
> I am not a crook.

```
US presidents have been known for
their pithy remarks:
\begin{quote}
The buck stops here.

I am not a crook.
\end{quote}
```

Use the `quotation` environment for quotations that consist of more than one paragraph. Paragraphs in the input are separated by blank lines as usual:

Here is some advice to remember:

> Environments for making quotations can be used for other things as well.
>
> Many problems can be solved by novel applications of existing environments.

```
Here is some advice to remember:
\begin{quotation}
Environments for making quotations
can be used for other things as well.

Many problems can be solved by
novel applications of existing
environments.
\end{quotation}
```

5.2 Centering and Flushing

Text can be centred on the page by putting it within the `center` environment, and it will appear flush against the left or right margins if it is placed within the `flushleft` or `flushright` environments. Notice the spelling of `center`—unfortunately LaTeX doesn't understand the British English spelling.

Text within these environments will be formatted in the normal way; in particular, the ends of the lines that you type are regarded as spaces. To indicate a new line, you type the `\\` command. For example:

one two three
four
five

```
\begin{center}
one
two
three \\
four \\
five
\end{center}
```

5.3 Lists

There are three environments for constructing lists. In each one, each new item is begun with an `\item` command. In the `itemize` environment, the start of each item is given a marker, while, in the `enumerate` environment, each item is marked by a number. These environments can be nested within each other, in which case the amount of indentation used and the marker are adjusted accordingly:

- Itemized lists are handy.
- However, don't forget
  1. The 'item' command.
  2. The 'end' command.

```
\begin{itemize}
\item Itemized lists are handy.
\item However, don't forget
  \begin{enumerate}
  \item The 'item' command.
  \item The 'end' command.
  \end{enumerate}
\end{itemize}
```

The third list-making environment is `description`. In a `description` you specify the item labels inside square brackets after the `\item` command. For example:

Three animals that you should know about are:

gnat A small animal that causes no end of trouble.

gnu A large animal that causes no end of trouble.

armadillo A medium-sized animal.

```
Three animals that you should
know about are:
\begin{description}
  \item[gnat] A small animal that causes
    no end of trouble.
  \item[gnu] A large animal that causes
    no end of trouble.
  \item[armadillo] A medium-sized animal.
\end{description}
```

5.4 Verbatim Output

Sometimes you will want to include text exactly as it appears on a terminal screen. For example, you might want to include part of a computer program. Not only do you want LaTeX to stop playing around with the layout of your text, you also want to be able to type all the characters on your keyboard without confusing LaTeX. The `verbatim` environment has this effect:

The section of program in question is:

```
{ this finds %a & %b }

for i := 1 to 27 do
   begin
   table[i] := fn(i);
   process(i)
   end;
```

```
The section of program in
question is:
\begin{verbatim}
{ this finds %a & %b }

for i := 1 to 27 do
   begin
   table[i] := fn(i);
   process(i)
   end;

\end{verbatim}
```

6 Type Styles

We have already come across the `\emph` command for changing the typeface to emphasis. Here is a full list of the available typeface changing commands *[Note: Corresponding LaTeX 2.09 commands have been added. –BJ]*:

LaTeX2ε *Command*	*or*	LaTeX 2.09 *Command*	*Effect on Text*
`\textrm{...}`	`{\rmfamily...}`	`{\rm...}`	Text is set in roman family
`\textsf{...}`	`{\sffamily...}`	`{\sf...}`	Text is set in sans serif family
`\texttt{...}`	`{\ttfamily...}`	`{\tt...}`	Text is set in `typewriter` family
`\textmd{...}`	`{\mdseries...}`		Text is set in medium series
`\textbf{...}`	`{\bfseries...}`	`{\bf...}`	Text is set in **bold** series
`\textup{...}`	`{\upshape...}`		Text is set in upright shape
`\textit{...}`	`{\itshape...}`	`{\it...}`	Text is set in *italic* shape
`\textsl{...}`	`{\slshape...}`	`{\sl...}`	Text is set in *slanted* shape
`\textsc{...}`	`{\scshape...}`	`{\sc...}`	Text is set in SMALL CAPS shape
`\emph{...}`	`{\em...}`	`{\em...}`	Text is set *emphasized*
`\textnormal{...}`	`{\normalfont...}`		Text is set in the document font
`\mathrm{...}`		`{\rm...}`	Math text is set in roman family
`\mathbf{...}`		`{\bf...}`	Math text is set in **bold** series
`\mathcal{...}`		`{\cal...}`	Math text is set in $\mathcal{CALLIGRAPHIC}$ family

Remember that the declaration versions (second column) are used *inside a pair of braces* to limit the amount of text that they affect. *[Note: In LaTeX2ε, It is recommended that you use the commands (first column) with a parameter of the text to typeset differently. –BJ]* In addition to the typeface commands, there are a set of commands that alter the size of the type. These commands are:

tiny scriptsize footnotesize small normalsize large
Large LARGE huge Huge

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

7 Sectioning Commands and Tables of Contents

Technical documents, like this one, are often divided into sections. Each section has a heading containing a title and a number for easy reference. LaTeX has a series of commands that will allow you to identify different sorts of sections. Once you have done this, LaTeX takes on the responsibility of laying out the title and of providing the numbers.

The commands that you can use are:

```
\chapter        \subsection          \paragraph
\section        \subsubsection       \subparagraph
```

The naming of these last two is rather unfortunate, since they do not really have anything to do with 'paragraphs' in the normal sense of the word: they are just lower levels of section. In most document styles, headings made with `\paragraph` and `\subparagraph` are not numbered. `\chapter` is not available in document style `article`. The commands should be used in the order given, since sections are numbered within chapters, subsections within sections, etc..

A seventh sectioning command, `\part`, is also available. Its use is always optional, and it is used to divide a large document into series of parts. It does not alter the numbering used for any of the other commands.

Including the command `\tableofcontents` in your document will cause a contents list to be included, containing information collected from the various sectioning commands. You will notice that each time your document is run through LaTeX the table of contents is always made up of the headings from the *previous* time you ran LaTeX on it. This is because LaTeX collects information for the table as it processes the document, and then includes it the next time it is run. This can sometimes mean that the document has to be processed through LaTeX *twice* to get a correct table of contents.

8 Producing Special Symbols

You can include in your LaTeX document a wide range of symbols that do not appear on your keyboard. For a start, you can add an accent to any letter:

ò	`\'{o}`	õ	`\~{o}`	ǒ	`\v{o}`
o̧	`\c{o}`	ó	`\'{o}`	ō	`\={o}`
ő	`\H{o}`	ọ	`\d{o}`	ô	`\^{o}`
ȯ	`\.{o}`	o͡o	`\t{oo}`	o̲	`\b{o}`
ö	`\"{o}`	ŏ	`\u{o}`		

Several other symbols are available and are generated using the following commands:

†	`\dag`	§	`\S`	©	`\copyright`
‡	`\ddag`	¶	`\P`	£	`\pounds`
œ	`\oe`	Œ	`\OE`	æ	`\AE`
Æ	`\AE`	å	`\aa`	Å	`\AA`
ø	`\o`	Ø	`\O`	ł	`\l`
Ł	`\E`	ß	`\ss`	¿	`?‘`
¡	`!‘`	...	`\ldots`	LaTeX	`\LaTeX`

There is also a `\today` command that prints the current date. When you use these commands remember that LaTeX will ignore any spaces that follow them, so that you can type '`\pounds 20`' to get '£20'. However, if you type '`\LaTeX is wonderful`' you will get 'LaTeXis wonderful'—notice the lack of space after LaTeX. To overcome this problem you can follow any of these commands by a pair of empty braces and then any spaces that you wish to include, and you will see that `\LaTeX{} really is wonderful!` (LaTeX really is wonderful!).

Finally, LaTeX 'math' mode, normally used to layout mathematical formulae, gives access to an even larger range of symbols, including the upper and lower case greek mathematical alphabet, calligraphic letters, mathematical operators and relations, arrows and a whole lot more. This will be discussed in Section 13.

9 Titles

Most documents have a title. To title a LaTeX document, you include the following commands in your document, usually just after `\begin{document}`.

```
\title{required title}
\author{required author}
\date{required date}
\maketitle
```

If there are several authors, then their names should be separated by `\and` in the `\author` command; they can also be separated by \\ if you want them to be centred on different lines. If the `\date` command is left out, then the current date will be printed.

```
\title{Essential \LaTeX}
\author{Jon Warbrick \and A N Other}
\date{14th February 1988}
\maketitle
```

Essential LaTeX

Jon Warbrick A N Other

14th February 1988

The exact appearance of the title varies depending on the document style. In the `report` and `book` styles, the title appears on a page of its own. In the `article` style, it normally appears at the top of the first page; use the class option `titlepage` to alter this (see Section 4).

10 Tabular Material

Because LaTeX will almost always convert a sequence of spaces into a single space, it can be rather difficult to lay out tables. See what happens in this example

Income Expenditure Result
20s 0d 19s 11d happiness
20s 0d 20s 1d misery

```
\begin{flushleft}
Income  Expenditure Result    \\
20s 0d  19s 11d     happiness \\
20s 0d  20s 1d      misery
\end{flushleft}
```

The `tabbing` environment overcomes this problem. Within it, you set tab-stops and tab to them much as you would do on a typewriter. Tab-stops are set with the `\=` command, and the `\>` command moves to the next stop. The \\ command is used to separate each line. A line that ends in `\kill` produces no output, and can be used to set tab-stops:

Income Expenditure Result
20s 0d 19s 11d Happiness
20s 0d 20s 1d Misery

```
\begin{tabbing}
Income \=Expenditure \=       \kill
Income \>Expenditure \>Result \\
20s 0d \>19s 11d \>Happiness  \\
20s 0d \>20s 1d  \>Misery
\end{tabbing}
```

Unlike a typewriter's tab key, the `\>` command always moves to the next tab-stop in sequence, even if this means moving to the left. This can cause text to be overwritten if the gap between two tab-stops is too small.

If you have more sophisticated tabular work, you will need the `tabular` environment. This differs from `tabbing` in that you need not worry about widths of columns—LaTeX will look at the whole table and see how wide columns need to be to cope with the widest entries in them. You must, however, tell LaTeX how many columns there are and how you wish to lay them out. You do this by supplying a *template* after `\begin{tabular}` with a letter for each column:

`l`	means the column will be left justified
`r`	means the column will be right justified
`c`	means the column will be centred

Items within a row are separated by the & character (now you know why it is treated specially), and the row is ended, as before, with \\. A simple table with three columns and two rows looks like:

Name	*Age*	*Height*
Sebastian	45	195cm
Mathew	1	68cm

```
\begin{tabular}{lrr}
\em Name & \em Age &\em Height \\
Sebastian & 45 & 195cm         \\
Mathew    & 1 & 68cm
\end{tabular}
```

Various other bells and whistles can be used to make the layout nicer:

1. You can put `\hline` between rows to draw a line across the table.
2. Type a | in the template where you want a vertical bar down every row of the table.
3. You can specify that a column is of a fixed width with text flowing within it by using the symbol `p` in the template, followed by a width between `{` and `}`. The width can be given in any of 'cm', 'mm' or 'in' (but not 'inches'!).

Thus, a more sophisticated table might look like this:

Group	*Type*	Sherds
Groups 1–9	Grey wares	218
Groups 40–44	Black (mostly 'black burnished') wares	116
Groups 61–67	Buff-red-orange wares	46
Groups 81–85	Colour-coated fine wares	67
Groups 91–2, 93–4	Mortaria and miscellaneous	35
Group 96	Samian	56
		538

```
\begin{tabular}{|l|p{1in}|r|} \hline
\em Group &\em Type &Sherds  \\\hline
Groups 1--9   &Grey wares&218\\
Groups 40--44 &Black (mostly
'black burnished')  wares&116\\
Groups 61--67 &Buff-red-orange
wares & 46                  \\
Groups 81--85 &Colour-coated
fine wares & 67             \\
Groups 91--2, 93--4&Mortaria
and miscellaneous & 35       \\
Group 96 &Samian & 56        \\\hline
 &&538                       \\\hline
\end{tabular}
```

Note that line-endings in what you type are not important; rows end only when a \\ is found.

11 Tables and Figures

When you lay out some text and numbers in a table, it is almost always intended that the whole unit stays together, not breaking over a page. What do you do when you are at the bottom of a page with not enough space left? Similarly, what do you do when you want to leave space for a photograph to be pasted in? When you are typing the text, you do not know where LaTeX will start a new page, so it is often very difficult to leave space just where you want. LaTeX solves this by a system of 'floats', objects that will be placed in a nice position on the page, but not necessarily exactly where you put it in your file. They will often be placed at the bottom of the current page, or at the top of the next, or on a page by themselves, if they need that much space. LaTeX lets you specify two sorts of float, namely *tables* and *figures*; in each of them you can specify a caption. The following shows how you would set up space for a figure in which you wanted to leave 3 inches of space, followed by a caption:

```
\begin{figure}
\vspace{3in}
\caption{A Photograph of my Subject}
\end{figure}
```

Figures and tables are automatically numbered, like sections; just as `\tableofcontents` lists your sections, subsections, etc. with their page numbers, `\listoffigures` and `\listoftables` will list the figures and tables with their captions.

12 Cross-References and Citations

One of the most useful things LaTeX can do for you is to to automatically generate a section, page, figure, table, or equation number in a cross-reference. For example, to refer to Section 4 of this report (on document styles) and Figure 2 (the output of processing `sample.tex`) in this sentence, the commands used were:

```
to refer to Section~\ref{sec:styles} of this report (on document
styles) and Figure~\ref{fig:result} (the output of processing
```

You will see that the `\ref` command inserted the cross-references number into the printed document. The `~` simply inserts a space while prohibiting LaTeX from breaking a line.

Of course, to refer to a cross-reference, there must be an indication in the LaTeX input file of the point being referred to. This is done via the `\label` command. The label for a section is most conveniently done immediately after the section heading, as in:

```
\section{Document Classes and Options}
\label{sec:styles}
```

and that for a figure or table in or immediately following the `\caption` command:

```
\caption{The Result of Processing the Sample File}
\label{fig:result}
```

In addition to (sub)-sections, figures and tables, it is common to use `\label` and `\ref` to refer to numbered equations (see Section 13).

Any text may be used in `\label` and `\ref` commands, though the convention impled above is a good one to follow. Page references use `\label` in the same way, but are referred to using `\pageref` instead of `\ref`.

In much the same way that you can refer to other parts of your document from the LaTeX file symbolically, you may cite other documents. To do this, you include the `\cite` command in your text, as in

```
The book by Lamport \cite{Lamport-LaTeX} is the principal
reference work on \LaTeX.
```

The argument to the `\cite` command, known as the *citation key*, is used to uniquely identify a paper, book, or other document that has been described in a *bibliographic database*.

Having cited other work from your document, you can generate a list of references by including the commands

```
\bibliographystyle{plain}
\bibliography{mybib1,mybib2}
```

where you want the list to appear. The `\bibliographystyle` command tells LaTeX how you want the citations to appear, the standard ones being:

`plain` Citations are sorted alphabetically in the list, with each entry being assigned a number; entries in the text are indicated in square brackets.

`unsrt` Citations are ordered in the reference list in the order of their first appearance and assigned a number; entries in the text are indicated in square brackets.

`alpha` Citations are sorted alphabetically in the list but have labels like "Lam86" instead of a number; the same label appears in the text in square brackets.

`abbrev` Citations are similar to `alpha` but are more compact (e.g.,by abbreviating month and journal names).

[Note: In addition to these four, bibliography styles matching many major statistical journals will soon be available on `statlib`*. –BJ]*

The argument of the `\bibliography` command is a comma-separated list of filenames, which should end in `.bib` (i.e., `mybib1.bib` and `mybib2.bib` in the example above). These files contain the expanded references referred to by the `\cite` commands. A full description of the format of these `.bib` files and how the citations are converted to a LaTeX-compatible form are beyond the scope of this (introductory) guide; details are given in Appendix B of Lamport's book.

13 Mathematical Typesetting

13.1 Math, Display-math and Equation

Mathematics is treated by TeX completely differently from ordinary text. There are two special *modes* for mathematics, known as *math mode* and *display math mode*.

Math mode commands are surrounded by `\(...\)` or by `$...$`.

Some mathematics set inline $2 \times 3 = 6$. Note that spaces in the input file are ignored in math mode.

```
Some mathematics set inline
\( 2\times 3 =   6 \).
Note that spaces in the input file
are ignored in math mode.
```

Display math mode commands are surrounded by `\[...\]` or by `$$...$$`.

A larger equation to be displayed on a line by itself.

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x)}{i!}$$

```
A larger equation to be displayed on a line by itself.
\[ f(x) = \sum_{i=0}^{\infty}\frac{f^{(i)}(x)}{i!} \]
```

There is a variant of display math mode, the `equation` environment, which automatically generates an equation number.

$$\left(\begin{array}{cc} 1 & 2 \\ 0 & 1 \end{array}\right)\left(\begin{array}{cc} 2 & 0 \\ 1 & 3 \end{array}\right) = \left(\begin{array}{cc} 4 & 6 \\ 1 & 3 \end{array}\right) \quad (1)$$

```
\begin{equation}
\left(\begin{array}{cc}
1 & 2 \\0 & 1
\end{array}\right)
\left(\begin{array}{cc}
2 & 0\\1 & 3
\end{array}\right)
=
\left(\begin{array}{cc}
4 & 6 \\1 & 3
\end{array}\right)
\end{equation}
```

These short examples show the main types of commands available in math mode. A few things to note are:

1. Subscripts and superscripts are produced with _ and ^, as in `x_{1} = p^{2}` for $x_1 = p^2$.
2. Fractions are produced by the `\frac` command: `\(\frac{a + b}{c}\)` gives $\frac{a+b}{c}$.
3. Various commands give names to mathematical symbols:
   `\infty \Rightarrow \surd \bigotimes` generate $\infty \Rightarrow \surd \bigotimes$
4. Arrays are produced by the `array` environment. This is identical to the `tabular` environment described in Section 10 except that the entries are typeset in math mode instead of LR mode. Note that the `array` environment does not put brackets arround the array, so it can also be used for setting determinants—or even sets of equations in which you want the columns to line up.
5. The commands `\left` and `\right` produce delimiters that grow as large as needed. They can be used with a variety of symbols, e.g.,`\left( \left\{ \left|`. The full set of these delimiters is shown in tables 4 and 5 below.

13.2 Spacing

All spaces in the input file are ignored in math mode. Sometimes you may want to adjust the spacing; use one of the following commands to achieve this.

`\,` thin space	`\:` medium space
`\!` negative thin space	`\;` thick space

A good example of where LaTeX needs some help with spacing is

$\iint z\,dxdy .. \int\int z dxdy$ `\(\int\!\!\int z\, dx dy .. \int\int z dx dy\)`

13.3 Changing Fonts in Math Mode

The default math mode font is *math italic*. This should not be confused with ordinary *text italic*. The font for 'ordinary' letters can be changed with the usual commands, `\emph`, `\textbf`, etc.. Note that lower case Greek letters (`\alpha`, etc.) are regarded as mathematical symbols (which means that they must be typed in math mode) and are not affected by these commands.

`\mathbf` produces **bold face roman** letters. If you wish to have ***bold face math italic*** letters, and bold face Greek letters and mathematical symbols, use the `\boldmath` command *before* going into math mode. This changes the default math fonts to bold. *[Note: LaTeX 2.09 has only the* `\boldmath` *command. –BJ]*

$x = 2\pi \Rightarrow x \simeq 6.28$ $\mathbf{x = 2\pi \Rightarrow x \simeq 6.28}$ $\boldsymbol{x = 2\pi \Rightarrow x \simeq 6.28}$

```
\( x = 2\pi \Rightarrow x \simeq 6.28 \)
\(\mathbf{x} = 2\pi \Rightarrow x \simeq 6.28 \)
{\boldmath \(x = \mathbf{2}\pi \Rightarrow x
          \simeq{\mathbf{6.28}}          \)}
```

There is also a calligraphic font for upper case letters; these are produced by the `\mathcal` command:

$\mathcal{F}$

```
\( \mathcal{F} \)
```

[Note: In LaTeX 2.09 one writes `{\cal F}` *to achieve the same effect. –BJ]*

14 What about $'s?

If you have converted to LaTeX from plain or AMSTeX, you will probably be wondering why there has been no mention of $ and $$.

In these systems math mode is surrounded by `$`'s and display math mode is surrounded by `$$`. This has certain drawbacks over the LaTeX system as it is difficult for your text editor to match `$`'s as it is hard to tell which ones are starting math mode and which are ending it. TeX will also get confused if you miss a $ out.

The (incorrect) input

```
let (a,b,c)$ be a Pythagorean triple, i.e.\ three
integers such that $a^{2}+b^{2}=c^{2}$.
```

produces the slightly mysterious error message

```
! Missing $ inserted.
<inserted text>
                $
<to be read again>
                   ^
l.56 ...triple, i.e.\ three integers such that $a^
                                                {2}+b^{2}=c^{2}$
?
```

Note that it reports the wrong error and in the wrong place, the use of the `^` command out of math mode. TeX has typeset 'be a ... such that' in math mode and *exited* math mode at the `$` after 'such that'. If you had made the equivalent LaTeX error, LaTeX has a better idea of what you intended:

```
let (a,b,c)\) be a Pythagorean triple, i.e.\ three
integers such that \(a^{2}+b^{2}=c^{2}\)
```

The error message may still be unintelligable, but at least it reports the error in the right place, you have used `\)` to end math mode when you were not in math mode (as you omitted the `\(` which should have been before the `(a,b,c)`).

```
LaTeX error.  See LaTeX manual for explanation.
              Type  H <return>  for immediate help.
! Bad math environment delimiter.
\@latexerr ...for immediate help.}\errmessage {#1}

\)...ifinner $\else \@badmath \fi \else \@badmath
                                                \fi
l.56 let (a,b,c)\)
                   be a Pythagorean triple, i.e.\ three integers such that \...

?
```

The single dollar is sometimes useful for small sections of math.

Let G be a p-group

```
Let $G$ be a $p$-group
```

The double dollar is *not* always equivalent to `\[ ... \]`, and so should not be used if you want your LaTeX file to be compatible with different styles and options (try the *fleqn* class option).

14.1 Symbols

Tables 1 through 11 in Appendix A show most of the symbols available from the standard LaTeX symbol fonts. Negations of the relational symbols can be made with the `\not` command:

$G \not\equiv H$

```
$G \not\equiv H$
```

Tables 12 through 19 show supplementary $\mathcal{AMS}$TeX symbols, which are available when you specify the `amssymb` package.

[Note: Most of the same characters are available by specifying the `amssymbols.sty` *package on the* `\documentstyle` *line in* LaTeX *2.09, like this:* `\documentstyle[amssymbols]{article}`*. Some characters not available with the* `amssymbols.sty` *option using* LaTeX *2.09 are omitted from the tables. –BJ]*

15 Letters

Producing letters is simple with LaTeX. To do this you use the document style `letter`. You can make any number of letters with a single input file. Your name and address, which are likely to be the same for all letters, are given once at the top of the file. Each letter is produced by a `letter` environment, having the name and address of the recipient as its argument. The letter itself begins with an `\opening` command to generate the salutation.

The letter ends with a `\closing` command. You can use the commands `\encl` and `\cc` to generate lists of enclosures and people to whom you are sending copies. Any text that follows the `\closing` must be

proceeded by a `\ps` command. This command produces no text—you'll have to type "P.S." yourself—but is needed to format the additional text correctly.

An example will make this clearer:

```
\documentstyle{letter}
\begin{document}

\address{1234 Avenue of the Armadillos \\
         Gnu York, G.Y. 56789}
\signature{R. (Ma) Dillo \\ Director of Cuisine}

\begin{letter}{G. Natheniel Picking \\
               Acme Exterminators \\
               Illinois}

\opening{Dear Nat,}

I'm afraid that the armadillo problem is still with us.
I did everything ...

... and I hope that you can get rid of the nasty
beasts this time.

\closing{Best Regards,}
\cc{Jimmy Carter\\Richard M. Nixon}
\end{letter}

\end{document}
```

16 Errors

When you create a new input file for LaTeX, you will probably make mistakes. Everybody does, and it's nothing to be worried about. As with most computer programs, there are two sorts of mistake that you can make: those that LaTeX notices and those that it doesn't. To take a rather silly example, since LaTeX doesn't understand what you are typing, it isn't going to be worried if you mis-spell some of the words in your text; you will just have to accurately proof-read your printed output. On the other hand, if you mis-spell one of the environment names in your file then LaTeX won't know what you want it to do.

When this sort of thing happens, LaTeX prints an error message on your terminal screen and then stops and waits for you to take some action. The error messages that it produces may seem user-unfriendly and not a little frightening at first. Nevertheless, if you know where to look they will probably tell you where the error is and what went wrong.

Consider what would happen if you mistyped `\begin{itemize}` as `\begin{itemie}`. When LaTeX processes this instruction, it displays the following on your terminal:

```
LaTeX error.  See LaTeX manual for explanation.
              Type  H <return>  for immediate help.
! Environment itemie undefined.
\@latexerr ...for immediate help.}\errmessage {#1}
                                                  \endgroup
l.140 \begin{itemie}

?
```

After typing the '?' LaTeX stops and waits for you to tell it what to do.

The first two lines of the message just tell you that the error was detected by LaTeX. The third line, the one that starts '!' is the *error indicator*. It tells you what the problem is, though until you have had some

experience of LaTeX this may not mean a lot to you. In this case it is just telling you that it doesn't recognise an environment called `itemie`. The next two lines tell you what LaTeX was doing when it found the error; they are irrelevant at the moment and can be ignored. The final line is called the *error locator*, and is a copy of the line in your file that caused the problem. It starts with a line number to help you to find it in your file. If the error was in the middle of a line, it will be shown broken at the point where LaTeX realized that there was an error. As with all computer programs, LaTeX can sometimes pass the point where the real error was before discovering that something is wrong, but it doesn't usually get very far.

At this point you could do several things. If you knew enough about LaTeX you might be able to fix the problem, or you could type '`X`' and press the return key to stop LaTeX running while you go and correct the error. The best thing to do, however, is just to press the return key. This will allow LaTeX to go on running as if nothing had happened. If you have made one mistake, then you have probably made several and you may as well try to find them all in one go. It's much more efficient to do it this way than to run LaTeX over and over again fixing one error at a time. Don't worry about remembering what the errors were—a copy of all the error messages is being saved in a *log* file so that you can look at them afterwards. See your *local guide* to find out what this file is called.

If you look at the line that caused the error, it's normally obvious what the problem was. If you can't work out what your problem is, look at the hints below and, if they don't help, consult Chapter 6 of Lamport's book: it contains a list of all of the error messages that you are likely to encounter, together with some suggestions as to what may have caused them.

Some of the most common mistakes that cause errors are:

- A mis-spelt command or environment name.
- Improperly matched '`{`' and '`}`'—remember that they should always come in pairs.
- Trying to use one of the ten special characters `# $ % & _ { } ~ ^` and `\` as an ordinary printing symbol.
- A missing `\end` command.
- A missing command argument (that's the bit enclosed in '`{`' and '`}`').

A single error can get LaTeX so confused that it reports a series of spurious errors as a result. If you have an error that you understand, followed by a series that you don't, then try correcting the first error—the rest may vanish as if by magic.

Sometimes LaTeX may write a `*` and stop without an error message. This is normally caused by a missing `\end{document}` command, though other errors can cause it. If this happens type `\stop` and press the return key.

Finally, LaTeX will sometimes print *warning* messages. They report problems that were not bad enough to cause LaTeX to stop processing, but nevertheless may require investigation. The most common problems are 'overfull' and 'underfull' lines of text. A message like:

```
Overfull \hbox (10.58649pt too wide) in paragraph at lines 172--175
[]\tenrm Mathematical for-mu-las may be dis-played. A dis-played
```

indicates that LaTeX could not find a good place to break a line when laying out a paragraph. As a result, it was forced to let the line stick out into the right-hand margin, in this case by 10.6 points. Since a point is 1/72.27 inches, this may be rather hard to see, but it will be there nonetheless.

This particular problem happens because LaTeX is rather fussy about line breaking, and it would rather generate a line that is too long than generate a paragraph that doesn't meet its high standards. The simplest way around the problem is to enclose the entire offending paragraph between `\begin{sloppypar}`

and `\end{sloppypar}` commands. This tells LaTeX that you are happy for it to break its own rules while it is working on that particular bit of text.

Alternatively, messages about "Underfull `\hbox`es" may appear. These are lines that had to have more space inserted between words than LaTeX would have liked. In general there is not much that you can do about these. Your output will look fine, even if the line looks a bit stretched. About the only thing you could do is to re-write the offending paragraph!

17 A Final Reminder

You now know enough LaTeX to produce a wide range of documents. But this document has only scratched the surface of the things that LaTeX can do. This entire document was itself produced with LaTeX (with no sticking things in or clever use of a photocopier) and even it hasn't used all the features that it could. From this you may get some feeling for the power that LaTeX puts at your disposal.

Please remember what was said in the introduction: if you *do* have a complex document to produce then *go and read Lamport's book*. You will be wasting your time if you rely only on what you have read here. Once you have used LaTeX for some time, you might want to know how to use the many style file extensions, which have been developed to aid LaTeX users to typeset their documents. This and much else (like how to use *MakeIndex*, for generating indices, or BibTeX, for managing your bibliographies) is covered in *The* LaTeX *Companion*[3]. Finally the final word about TeX is Knuth's *TeX book*[2].

And one other warning: having dabbled with LaTeX your documents will never be the same again...

A Mathematical Symbols

α	\alpha	β	\beta	γ	\gamma	δ	\delta
ϵ	\epsilon	ε	\varepsilon	ζ	\zeta	η	\eta
θ	\theta	ϑ	\vartheta	ι	\iota	κ	\kappa
λ	\lambda	μ	\mu	ν	\nu	ξ	\xi
o	o	π	\pi	ϖ	\varpi	ρ	\rho
ϱ	\varrho	σ	\sigma	ς	\varsigma	τ	\tau
υ	\upsilon	ϕ	\phi	φ	\varphi	χ	\chi
ψ	\psi	ω	\omega				
Γ	\Gamma	Δ	\Delta	Θ	\Theta	Λ	\Lambda
Ξ	\Xi	Π	\Pi	Σ	\Sigma	Υ	\Upsilon
Φ	\Phi	Ψ	\Psi	Ω	\Omega		

Table 1: Greek letters

$\pm$	\pm	$\cap$	\cap	$\diamond$	\diamond	$\oplus$	\oplus
$\mp$	\mp	$\cup$	\cup	$\triangle$	\bigtriangleup	$\ominus$	\ominus
$\times$	\times	$\uplus$	\uplus	$\bigtriangledown$	\bigtriangledown	$\otimes$	\otimes
$\div$	\div	$\sqcap$	\sqcap	$\triangleleft$	\triangleleft	$\oslash$	\oslash
$*$	\ast	$\sqcup$	\sqcup	$\triangleright$	\triangleright	$\odot$	\odot
$\star$	\star	$\vee$	\vee	$\lhd$[a]	\lhd	$\bigcirc$	\bigcirc
$\circ$	\circ	$\wedge$	\wedge	$\rhd$[a]	\rhd	$\dagger$	\dagger
$\bullet$	\bullet	$\setminus$	\setminus	$\unlhd$[a]	\unlhd	$\ddagger$	\ddagger
$\cdot$	\cdot	$\wr$	\wr	$\unrhd$[a]	\unrhd	$\amalg$	\amalg

[a] Not predefined in LaTeX2$_\varepsilon$. Use the packages `latexsym` or `amssymb`.

[Note: These symbols are predefined and work fine in LaTeX 2.09. –BJ]

Table 2: Binary operation symbols

$\leq$	\leq	$\geq$	\geq	$\equiv$	\equiv	$\models$	\models	$\prec$	\prec
$\succ$	\succ	$\sim$	\sim	$\perp$	\perp	$\preceq$	\preceq	$\succeq$	\succeq
$\simeq$	\simeq	$\mid$	\mid	$\ll$	\ll	$\gg$	\gg	$\asymp$	\asymp
$\parallel$	\parallel	$\subset$	\subset	$\supset$	\supset	$\approx$	\approx	$\bowtie$	\bowtie
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong	$\bowtie$	\Join	$\sqsubset$	\sqsubset
$\sqsupset$	\sqsupset	$\neq$	\neq	$\smile$	\smile	$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq
$\doteq$	\doteq	$\frown$	\frown	$\in$	\in	$\ni$	\ni	$\propto$	\propto
$=$	=	$\vdash$	\vdash	$\dashv$	\dashv	$<$	<	$>$	>

Table 3: Relation symbols

$\uparrow$	\uparrow	$\Uparrow$	\Uparrow	$\downarrow$	\downarrow	$\Downarrow$	\Downarrow
$\{$	\{	$\}$	\}	$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow
$\lfloor$	\lfloor	$\rfloor$	\rfloor	$\lceil$	\lceil	$\rceil$	\rceil
$\langle$	\langle	$\rangle$	\rangle	/	/	\	\backslash
$\vert$	|	$\Vert$	\\|				

Table 4: Delimiters

$\rmoustache$	\rmoustache	$\lmoustache$	\lmoustache	$\rgroup$	\rgroup	$\lgroup$	\lgroup
$\arrowvert$	\arrowvert	$\Arrowvert$	\Arrowvert	$\bracevert$	\bracevert		

Table 5: Large delimiters

$\leftarrow$	\leftarrow	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\rightarrow$	\rightarrow	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow

Table 6: Arrow symbols

$\ldots$	\ldots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots	$\aleph$	\aleph
$\prime$	\prime	$\forall$	\forall	∞	\infty	$\hbar$	\hbar	$\emptyset$	\emptyset
$\exists$	\exists	∇	\nabla	$\surd$	\surd	$\Box$	\Box[a]	$\triangle$	\triangle
$\Diamond$	\Diamond[a]	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell	$\neg$	\neg
$\top$	\top	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp	$\wp$	\wp
$\bot$	\bot	$\clubsuit$	\clubsuit	$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\spadesuit$	\spadesuit
$\mho$	\mho[a]	$\Re$	\Re	$\Im$	\Im	$\angle$	\angle	∂	\partial

[a] Not predefined in LaTeX2ε. Use the packages `latexsym` or `amssymb`.

[Note: These symbols are predefined and work fine in LaTeX 2.09. –BJ]

Table 7: Miscellaneous symbols

\arccos	\cos	\csc	\exp	\ker	\limsup	\min	\sinh
\arcsin	\cosh	\deg	\gcd	\lg	\ln	\Pr	\sup
\arctan	\cot	\det	\hom	\lim	\log	\sec	\tan
\arg	\coth	\dim	\inf	\liminf	\max	\sin	\tanh

Table 8: Log-like symbols

$\hat{a}$	\hat{a}	$\acute{a}$	\acute{a}	$\bar{a}$	\bar{a}	$\dot{a}$	\dot{a}	$\breve{a}$	\breve{a}
$\check{a}$	\check{a}	$\grave{a}$	\grave{a}	$\vec{a}$	\vec{a}	$\ddot{a}$	\ddot{a}	$\tilde{a}$	\tilde{a}

Table 9: Math mode accents

$\sum$	\sum	$\prod$	\prod	$\coprod$	\coprod	$\int$	\int	$\oint$	\oint
$\bigcap$	\bigcap	$\bigcup$	\bigcup	$\bigsqcup$	\bigsqcup	$\bigvee$	\bigvee	$\bigwedge$	\bigwedge
$\bigodot$	\bigodot	$\bigotimes$	\bigotimes	$\bigoplus$	\bigoplus	$\biguplus$	\biguplus		

Table 10: Variable-sized symbols

$\widetilde{abc}$	\widetilde{abc}	$\widehat{abc}$	\widehat{abc}
$\overleftarrow{abc}$	\overleftarrow{abc}	$\overrightarrow{abc}$	\overrightarrow{abc}
$\overline{abc}$	\overline{abc}	$\underline{abc}$	\underline{abc}
$\overbrace{abc}$	\overbrace{abc}	$\underbrace{abc}$	\underbrace{abc}
$\sqrt{abc}$	\sqrt{abc}	$\sqrt[n]{abc}$	\sqrt[n]{abc}
f'	f'	$\frac{abc}{xyz}$	\frac{abc}{xyz}

Table 11: LaTeX math constructs

$\hbar$	\hbar	$\hslash$	\hslash	$\vartriangle$	\vartriangle
$\triangledown$	\triangledown	$\square$	\square	$\lozenge$	\lozenge
Ⓢ	\circledS	$\angle$	\angle	$\measuredangle$	\measuredangle
[?] [b]	\nexists	$\mho$	\mho	[?] [b]	\Finv [a]
[?] [b]	\Game [a]	[?] [b]	\Bbbk [a]	`	\backprime
$\varnothing$	\varnothing	$\blacktriangle$	\blacktriangle	$\blacktriangledown$	\blacktriangledown
$\blacksquare$	\blacksquare	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\sphericalangle$	\sphericalangle	$\complement$	\complement	[?] [b]	\eth
[?] [b]	\diagup [a]	[?] [b]	\diagdown [a]		

[a] Not defined in style amssymb, define using the LaTeX2ε \DeclareMathSymbol command.

[b] *[Note: This symbol is not available with* amssymbols.sty *using* LaTeX *2.09. –BJ]*

Table 12: AMS miscellaneous symbols

ϝ	\digamma	ϰ	\varkappa	ℶ	\beth	ℸ	\daleth	ℷ	\gimel

Table 13: AMS Greek and Hebrew

⌜	\ulcorner	⌝	\urcorner	⌞	\llcorner	⌟	\lrcorner

Table 14: AMS delimiters

⇢[b]	\dashrightarrow	⇠[b]	\dashleftarrow	⇇	\leftleftarrows
⇆	\leftrightarrows	⇚	\Lleftarrow	↞	\twoheadleftarrow
↢	\leftarrowtail	↫	\looparrowleft	⇋	\leftrightharpoons
↶	\curvearrowleft	↺	\circlearrowleft	↰	\Lsh
⇈	\upuparrows	↿	\upharpoonleft	⇃	\downharpoonleft
⊸	\multimap	↭	\leftrightsquigarrow	⇉	\rightrightarrows
⇄	\rightleftarrows	⇉	\rightrightarrows	⇄	\rightleftarrows
↠	\twoheadrightarrow	↣	\rightarrowtail	↬	\looparrowright
⇌	\rightleftharpoons	↷	\curvearrowright	↻	\circlearrowright
↱	\Rsh	⇊	\downdownarrows	↾	\upharpoonright
⇂	\downharpoonright	⇝	\rightsquigarrow		

[b] *[Note: This symbol is not available with* `amssymbols.sty` *using* LaTeX *2.09. –BJ]*

Table 15: AMS arrows

↚	\nleftarrow	↛	\nrightarrow	⇍	\nLeftarrow
⇏	\nRightarrow	↮	\nleftrightarrow	⇎	\nLeftrightarrow

Table 16: AMS negated arrows

∔	\dotplus	∖	\smallsetminus	⋒	\Cap
⋓	\Cup	⩒	\barwedge	⩟	\veebar
⩖	\doublebarwedge	⊟	\boxminus	⊠	\boxtimes
⊡	\boxdot	⊞	\boxplus	⋇	\divideontimes
⋉	\ltimes	⋊	\rtimes	⋋	\leftthreetimes
⋌	\rightthreetimes	⋏	\curlywedge	⋎	\curlyvee
⊖	\circleddash	⊛	\circledast	⊚	\circledcirc
⋅	\centerdot	⊺	\intercal		

Table 17: AMS binary operators

≦	\leqq	⩽	\leqslant	⪕	\eqslantless
≲	\lesssim	⪅	\lessapprox	≊	\approxeq
⋖	\lessdot	⋘	\lll	≶	\lessgtr
⋚	\lesseqgtr	⪋	\lesseqqgtr	≒	\doteqdot
≓	\risingdotseq	≒	\fallingdotseq	∽	\backsim
⋍	\backsimeq	⫅	\subseteqq	⋐	\Subset
⊏	\sqsubset	≼	\preccurlyeq	≽	\curlyeqprec
≾	\precsim	⪷	\precapprox	◁	\vartriangleleft
⊴	\trianglelefteq	⊨	\vDash	⊪	\Vvdash
⌣	\smallsmile	⌢	\smallfrown	≏	\bumpeq
≎	\Bumpeq	≧	\geqq	⩾	\geqslant
⪖	\eqslantgtr	≳	\gtrsim	⪆	\gtrapprox
⪀	\gtrdot	⋙	\ggg	≷	\gtrless
⋛	\gtreqless	⪌	\gtreqqless	≗	\eqcirc
≗	\circeq	◁	\triangleq	∼	\thicksim
≈	\thickapprox	⫆	\supseteqq	⋑	\Supset
⊐	\sqsupset	≽	\succcurlyeq	≼	\curlyeqsucc
≿	\succsim	⪸	\succapprox	▷	\vartriangleright
⊵	\trianglerighteq	⊩	\Vdash	∣	\shortmid
∥	\shortparallel	≬	\between	⋔	\pitchfork
∝	\varpropto	◀	\blacktriangleleft	∴	\therefore
϶	\backepsilon	▶	\blacktriangleright	∵	\because

Table 18: AMS binary relations

≮	\nless	≰	\nleq	⩽̸	\nleqslant
≨	\nleqq	≨	\lneq	⪇	\lneqq
≨	\lvertneqq	⋦	\lnsim	⪉	\lnapprox
⊀	\nprec	⪵	\npreceq	⪹	\precnsim
⪹	\precnapprox	≁	\nsim	∤	\nshortmid
∤	\nmid	⊬	\nvdash	⊭	\nvDash
⋪	\ntriangleleft	⋬	\ntrianglelefteq	⊈	\nsubseteq
⊊	\subsetneq	⫋	\varsubsetneq	⫋	\subsetneqq
⫋	\varsubsetneqq	≯	\ngtr	≱	\ngeq
≱	\ngeqslant	≱	\ngeqq	⪈	\gneq
⪈	\gneqq	≩	\gvertneqq	⋧	\gnsim
⪊	\gnapprox	⊁	\nsucc	⪰̸	\nsucceq
⋩	\succnsim	[?] [b]	\succnapprox	[?] [b]	\ncong
∦	\nshortparallel	∦	\nparallel	⊯	\nVDash
⊯	\nVDash	⋫	\ntriangleright	⋭	\ntrianglerighteq
⊋	\nsupseteq	⫌	\nsupseteqq	⊋	\supsetneq
⊋	\varsupsetneq	⫌	\supsetneqq	⫌	\varsupsetneqq

[b] *[Note: This symbol is not available with* amssymbols.sty *using* LaTeX *2.09. –BJ]*

Table 19: AMS negated binary relations

B Horrible Mathematical Examples to Study

$$\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{t} e^{-x^2/2} dx \tag{2}$$

```
\begin{equation}
  \Phi(t)=\frac{1}{\sqrt{2\pi}}
  \int_\infty^t e^{-x^2/2} dx
\end{equation}
```

$$\prod_{j\geq 0} \left(\sum_{k\geq 0} a_{jk} z^k \right) = \sum_{k\geq 0} z^n \left(\sum_{\substack{k_0,k_1,\ldots\geq 0 \\ k_0+k_1+\ldots=n}} a_0 k_0 a_{1k_1} \ldots \right) \tag{3}$$

```
\begin{equation}
\prod_{j\geq 0}
\left(\sum_{k\geq 0}a_{jk} z^k\right)
= \sum_{k\geq 0} z^n
\left( \sum_{{k_0,k_1,\ldots\geq 0}
          \atop{k_0+k_1+\ldots=n}    }
       a{_0k_0}a_{1k_1}\ldots  \right)
\end{equation}
```

$$\pi(n) = \sum_{m=2}^{n} \left\lfloor \left(\sum_{k=1}^{m-1} \lfloor (m/k)/\lceil m/k \rceil \rfloor \right)^{-1} \right\rfloor \tag{4}$$

```
\begin{equation}
\pi(n) = \sum_{m=2}^{n}
  \left\lfloor \left(\sum_{k=1}^{m-1}
       \lfloor(m/k)/\lceil m/k\rceil
       \rfloor \right)^{-1}
  \right\rfloor
\end{equation}
```

$$\{\overbrace{a,\ldots,a}^{k\ a's},\overbrace{b,\ldots,b}^{l\ b's}\}_{k+l\ \text{elements}} \tag{5}$$

```
\begin{equation}
\{\underbrace{%
    \overbrace{\mathstrut a,\ldots,a}^{k\ a's},
    \overbrace{\mathstrut b,\ldots,b}^{l\ b's}}
  _{k+l\ \mathrm{elements}}                    \}
\end{equation}
```

$$\mathrm{W}^+ \begin{array}{l} \nearrow \mu^+ + \nu_\mu \\ \rightarrow \pi^+ + \pi^0 \\ \rightarrow \kappa^+ + \pi^0 \\ \searrow \mathrm{e}^+ + \nu_\mathrm{e} \end{array}$$

```
\begin{displaymath}
\mbox{W}^+\
\begin{array}{l}
\nearrow\raise5pt\hbox{$\mu^+ + \nu_{\mu}$}\\
\rightarrow         \pi^+ +\pi^0         \\[5pt]
\rightarrow \kappa^+ +\pi^0              \\
\searrow\lower5pt\hbox{$\mathrm{e}^+
          +\nu_{\scriptstyle\mathrm{e}}$}
\end{array}
\end{displaymath}
```

$$F(x,y)=0 \quad \text{and} \quad \left|\begin{array}{ccc} F''_{xx} & F''_{xy} & F'_x \\ F''_{yx} & F''_{yy} & F'_y \\ F'_x & F'_y & 0 \end{array}\right| = 0$$

```
\begin{displaymath}
{F}(x,y)=0\quad\mathrm{and}\quad
 \left|\begin{array}{ccc}
  F_{xx}'' & F_{xy}'' & F_{x}' \\
  F_{yx}'' & F_{yy}'' & F_{y}' \\
  F_{x}'   & F_{y}'   & 0
       \end{array}      \right| =0
\end{displaymath}
```

$$\frac{\pm\left|\begin{array}{ccc} x_1-x_2 & y_1-y_2 & z_1-z_2 \\ l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \end{array}\right|}{\sqrt{\left|\begin{array}{cc} l_1 & m_1 \\ l_2 & m_2 \end{array}\right|^2+\left|\begin{array}{cc} m_1 & n_1 \\ n_1 & l_1 \end{array}\right|^2+\left|\begin{array}{cc} m_2 & n_2 \\ n_2 & l_2 \end{array}\right|^2}}$$

```
\begin{displaymath}
\frac{\pm
\left|\begin{array}{ccc}
x_1-x_2  & y_1-y_2 & z_1-z_2 \\
l_1      & m_1     & n_1     \\
l_2      & m_2     & n_2
\end{array}\right|}{
\sqrt{\left|\begin{array}{cc}l_1&m_1\\
l_2&m_2\end{array}\right|^2
+     \left|\begin{array}{cc}m_1&n_1\\
n_1&l_1\end{array}\right|^2
+     \left|\begin{array}{cc}m_2&n_2\\
n_2&l_2\end{array}\right|^2}}
\end{displaymath}
```

$$\begin{aligned}\sigma_0^f(Q,T_{3R},\beta,s) &= \frac{4\pi\alpha^2}{3s}\beta\times\left[\frac{Q^2\left\{\frac{3-\beta^2}{2}\right\}-2QC_{\rm V}C'_{\rm V}s(s-M_{\rm Z)}^2}{(s-M_{\rm Z)}^2{}^2+M_{\rm Z}^2\Gamma_{\rm Z}^2\left\{\frac{3-\beta^2}{2}\right\}}\right. \\ &+ \left.\frac{(C_{\rm V}^2+C_{\rm A}^2)s^2}{(s-M_{\rm Z)}^2{}^2+M_{\rm Z}^2\Gamma_{\rm Z}^2\left\{C'^2_{\rm V}\left\{\frac{3-\beta^2}{2}\right\}+C'^2_{\rm A}\{\beta^2\}\right\}}\right]\end{aligned} \tag{6}$$

```
\newcommand{\CA}{C_{\rm A}}     \newcommand{\CV}{C_{\rm V}}
\newcommand{\CPA}{{C'}_{\rm A}} \newcommand{\CPV}{{C'}_{\rm V}}
\newcommand{\GZ}{\Gamma^2_{\rm Z}}
\newcommand{\MZ}{M^2_{\rm Z}}   \newcommand{\MZs}{{(s-M^2_{\rm Z})}}
\newcommand{\BE}{\left\{\frac{\displaystyle 3-\beta^2}{\displaystyle 2}\right\}}
\begin{eqnarray}
\sigma^f_0(Q,T_{3R},\beta,s) & = &
      \frac{4\pi\alpha^2}{3s}\beta \times
        \left[ \frac{Q^2 \BE - 2Q \CV \CPV s \MZs}{\MZs^2 + \MZ \GZ \BE}
            \right.                                        \nonumber \\[-3mm]
                                &   &                                \\[-3mm]
                                & + &
      \left.\frac{(\CV^2 + \CA^2) s^2}%
                 {\MZs^2+\MZ\GZ\left\{\CPV^2 \BE+\CPA^2 \{\beta^2\}\right\}}
        \right]                                            \nonumber
\end{eqnarray}
```

C Defining New Commands and Environments

[Note: Added for June 1995 Interface meeting. –BJ]

In the last example in Appendix B, you may have noticed the use of the `\newcommand` command. This the standard LaTeX way to define new commands (also called *macros*) to lessen your typing load and to easily repeat typesetting effects that you are interested in. In this section we will show you how to define new commands and new environments in LaTeX, as well as how to save them for future use in your documents.

You can define new commands and environments anywhere in the document that you like, but for ease of reference it is usually best to make all your definitions in the prologue (the lines between the `\documentclass` (or `\documentstyle`) and `\begin{document}` lines at the beginning of your document).

C.1 New Commands

The basic syntax of `\newcommand` is `\newcommand{\mycommand}{stuff that \mycommand stands for}` Our first example is just like the use of `\newcommand` in Appendix B.

We will denote the set of Real numbers by $\mathbb{R}$, the Natural numbers by $\mathbb{N}$, and the Rational numbers by $\mathcal{Q}$. Similarly, Euclidean n-space will be denoted $\mathbb{R}^n$.

```
\newcommand{\Reals}{{\rm I\kern-.1667em{}R}}
\newcommand{\Nats}{{\rm I\kern-.1667em{}N}}
\newcommand{\Rats}{{\cal Q}}

We will denote the set of Real numbers by $\Reals$,
the Natural numbers by $\Nats$, and the Rational
numbers by $\Rats$.  Similarly, Euclidean $n$-space
will be denoted $\Reals^n$.
```

If you want to pass arguments or parameters to your new command, put the number of arguments in square braces before defining what the command stands for, and refer to the arguments as `#1`, `#2`, etc. Up to nine arguments can be passed to a command in this way.

If you really want to save time you can make a macro that accepts arguments. And a macro can accept more than one argument.

For example, everyone knows that

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k} .$$

```
\newcommand{\ul}[1]{\underline{#1}}
\newcommand{\combos}[2]{\left(\begin{array}{@{}c@{}}
#1 \\ #2 \end{array}\right)}

If you \ul{really} want to save time you can make a
macro that accepts arguments.  And a macro can accept
more than one argument.

\vspace{1.2em}

For example, \ul{everyone} knows that
\begin{displaymath}
    \combos{n-1}{k-1} + \combos{n-1}{k} = \combos{n}{k} .
\end{displaymath}
```

If you use `\newcommand` to define a command that has already been defined, LaTeX will complain. For example, `\bar` is aready defined by LaTeX, so if you have in your file the lines

```
\newcommand{\bar}[1]{\overbar{#1}}
The sample average is $\bar X$.
```

then LaTeX will complain:

```
LaTeX error.  See LaTeX manual for explanation.
              Type  H <return>  for immediate help.
! Command name 'bar' already used.
\@latexerr ...for immediate help.}\errmessage {#1}

\@ifundefined ...me #1\endcsname \relax #2\else #3
                                                  \fi
l.2064 \newcommand{\bar}[1]{\overbar{#1}}

?
```

If you really want to redefine `\bar`, you can do so by replacing the `\newcommand` command word with `\renewcommand`. This tells LaTeX to replace the old definition of `\bar` with your new definition.

In many TeX and some LaTeX documents you will see the command `\def` used to define new commands. The syntax is similar to but not identical with `\newcommand` and `\renewcommand`. `\def` is the underlying TeX command for defining new commands; `\newcommand` and `\renewcommand` are simply interfaces to TeX's `\def` command. Using `\def`, you are working without a net: LaTeX will not check to see if you are redefining an already-defined command, and LaTeX is much less careful about checking your syntax, consistency of definitions, etc. The moral of the story is, use `\newcommand` and `\renewcommand` unless you have a really good reason for switching to `\def`.

C.2 New Environments

Like `\newcommand` and `\renewcommand`, you can also create new environments with `\newenvironment`. The syntax is similar to `\newcommand`: `\newenvironment{myenvironment}{stuff to do at the beginning of the environment}{stuff to do at the end of the environment}`.

The only difference between LaTeX's quote and my declaration is that I have defined a declaration to automatically emphasize the text.

The only difference between LaTeX's quote and my declaration is that I have defined a declaration to automatically emphasize the text.

```
\newenvironment{declaration}{\begin{quote}\em}{\end{quote}}

\begin{quote}
The only difference between \LaTeX's quote and my declaration
is that I have defined a declaration to automatically
emphasize the text.
\end{quote}

\begin{declaration}
The only difference between \LaTeX's quote and my declaration
is that I have defined a declaration to automatically
emphasize the text.
\end{declaration}
```

It is also possible to pass arguments or parameters to an environment.

Conventional Books

Lamport. L. (1985). LaTeX—*A Document Preparation System—User's Guide and Reference Manual.* Addison-Wesley, Reading, MA, USA.

Goossens, M., Mittelbach, F., and Samarin, A. (1993). *The* LaTeX *Companion.* Addison-Wesley, Reading, MA, USA.

On the Internet

`http://www.stat.ucla.edu/develop/tex/`: UCLA Dept. of Statistics TeX References.

`ftp://ftp.SHSU.edu/tex-archive/`: Sam Houston State University's Comprehensive TeX Archives.

```
\newenvironment{references}[1]{\vspace{0.5em}
\begin{center}\bf #1\end{center}
\begin{description}\labelsep=0pt}{\end{description}\hrule}

\begin{references}{Conventional Books}
   \item Lamport. L. (1985). {\em \LaTeX---A Document
     Preparation System---User's  Guide and Reference
     Manual}.  Addison-Wesley, Reading, MA, USA.
   \item Goossens, M., Mittelbach, F., and Samarin, A.
     (1993).  {\em The \LaTeX\ Companion.}
     Addison-Wesley, Reading, MA, USA.
\end{references}

\begin{references}{On the Internet}
   \item {\tt http://www.stat.ucla.edu/develop/tex/}:
     UCLA Dept. of Statistics \TeX\ References.
   \item {\tt ftp://ftp.SHSU.edu/tex-archive/}:
     Sam Houston State University's Comprehensive
     \TeX\ Archives.
\end{references}
```

Once again, if an environment is already defined, you can re-define it with the `\renewenvironment` command.

A special shorthand version of the `\newenvironment` command, called `\newtheorem`, can be used to define environments for definitions, theorems, corollaries, and the like, that automatically generate theorem numbers, labels for referencing, etc. The example below shows what can be done.

Theorem 1 *Be careful, or you will get what you ask for.*

Corollary 1 *Never volunteer to teach a* LaTeX *course unless you intend to do it.*

Proof. As everyone knows, this follows directly from Theorem 1. □

```
\newtheorem{theorem}{Theorem}
\newtheorem{cor}{Corollary}

\newenvironment{proof}{\par\noindent{\bf Proof. }}{$\Box$\par}

\begin{theorem}\label{careful}
Be careful, or you will get what you ask for.
\end{theorem}

\begin{cor}
Never volunteer to teach a \LaTeX\ course
unless you intend to do it.
\end{cor}

\begin{proof}
As everyone knows, this follows
directly from Theorem~\ref{careful}.
\end{proof}
```

C.3 Style Files

Many LaTeX users never define new commands and environments, or only make definitions that are useful for a single document.

If you find that you are making the same command or environment definitions over and over again in each document you typeset with LaTeX, you can collect these definitions together in a *style file* (also called a *macro package*). The file can have any name you want, but should have the extension `.sty`, like `bj.sty`. You can use the file's definitions in your document by including its name in square brackets on the `\documentstyle` line. *[Note: In LaTeX2ε, style files can be included with the* `\usepackage` *command immediately after the* `\documentclass` *line. –BJ]* Here is an example, showing a `.sty` file and the `.tex` file that uses it.

Theorem 1 *For every pair of integers n and k such that $0 \leq k-1 \leq n-1$,*

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k} .$$

Proof. This can be proved by induction. □

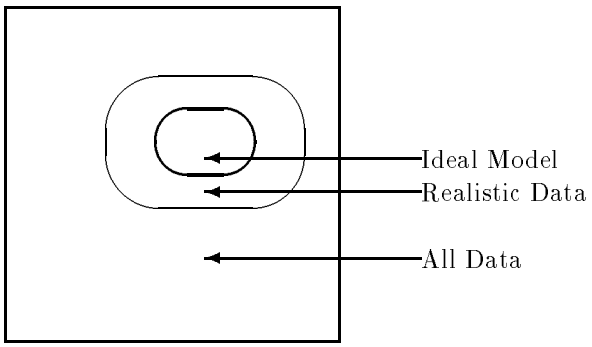
```
> type bj.sty

%%%%%%%%%%%%%%%%%%%%%%%% beginning of bj.sty

\newcommand{\ul}[1]{\underline{#1}}
\newcommand{\combos}[2]%
{\left(\begin{array}{@{}c@{}}
#1 \\ #2 \end{array}\right)}

\newtheorem{theorem}{Theorem}

\newenvironment{proof}%
{\par\noindent{\bf Proof. }}{$\Box$\par}

%%%%%%%%%%%%%%%%%%%%%%%% end of bj.sty

> type my-example.tex

%%%%%%%%%%%%%%%%%%%%%%%% beginning of my-example.tex

\documentstyle[12pt,bj]{article}
\begin{document}

\begin{theorem}
For every pair of integers $n$ and $k$
such that $0\leq k-1 \leq n-1$,
\begin{displaymath}
    \combos{n-1}{k-1} + \combos{n-1}{k}
    = \combos{n}{k} .
\end{displaymath}
\end{theorem}

\begin{proof}
This can be proved by \ul{induction}.
\end{proof}

\end{document}

%%%%%%%%%%%%%%%%%%%%%%%% end of my-example.tex

> latex my-example
```

D Creating and Importing Figures

[Note: Added for June 1995 Interface meeting. –BJ]

D.1 The Picture Environment

LaTeX has a limited facility for creating figures, called the `picture` environment. For a detailed explanation of this environment, see the LaTeX *User's Guide and Reference Manual*. Here is an example of what can be done.




```
\setlength{\unitlength}{1.25pt}
\begin{picture}(100,100)(0,0)
\thicklines
\put(0,0){\framebox(100,100){}}
%
\thinlines
\put(60,60){\oval(60,40)}
%
\thicklines
\put(60,60){\oval(30,20)}
%
\put(125,55){\vector(-1,0){65}}
\put(125,55){\makebox(0,0)[l]{Ideal Model}}
%
\put(125,45){\vector(-1,0){65}}
\put(125,45){\makebox(0,0)[l]{Realistic Data}}
%
\put(125,25){\vector(-1,0){65}}
\put(125,25){\makebox(0,0)[l]{All Data}}
\end{picture}
```

D.2 Including PostScript Images

For many purposes it is simpler to create your figures with some other package and include them in your document. Statisticians often create `PostScript` figures using *S-Plus*, and computational scientists generally may produce such figures with *Mathematica*, *Maple*, etc.

To include an external image in a formatted TeX or LaTeX document requires two generic steps:

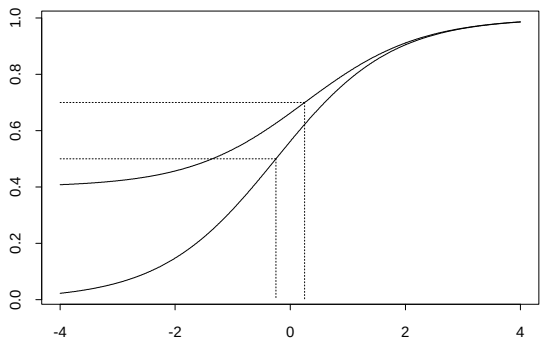
- Tell TeX or LaTeX where to place the figure and how much space to reserve for it.
- "Include" the figure using a TeX command that the program you use to process your `.dvi` file for printing or screen viewing will understand.

Because the second step involves the `.dvi` program you use, the exact method for including `PostScript` and other figures in your documents varies from computer to computer. You should refer to your *local guide*, or ask your favorite LaTeX guru, to find out exactly what has to be done where you are.

Here are some examples, to give you a feel for what you might need to do. Both of these examples work with `dvips`, developed by Tomas Rokicki, which is used on MS-DOS and UNIX machines to print formatted documents from `.dvi` files on PostScript printers.

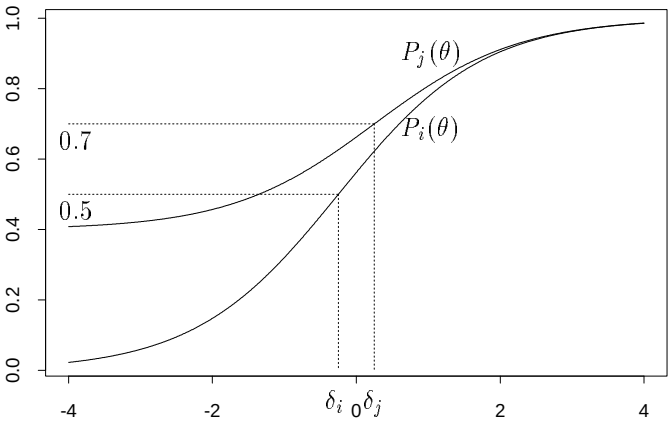
The first example just includes the raw figure with no frills. The `\special` command is a TeX command that passes its argument directly to the `.dvi` processing program, so it is actually `dvips` that is reading

the information "`psfile=fig2.ps hscale=40 vscale=40`" to import and resize the figure in the formatted document (it is also possible to rotate the figure by including a phrase like `angle=270` as part of the `\special` argument).




```
\vspace*{3.0in}
\hspace*{-0.5in}
\special{psfile=fig2.ps hscale=40 vscale=40}
\vspace*{-1.25in}
```

The second example places a `PostScript` figure in a `picture` environment. This allows us to do more fine-tuned placement of the figure, as well as to add labelling that is not included in the original `PostScript` image.



Two IRF's for which $P_i(\theta) \leq P_j(\theta)$ but $\delta_i < \delta_j$ using the definition of δ in logistic IRT models.

```
\begin{center}
\setlength{\unitlength}{0.66in}
\begin{picture}(4,3)(0.3,-0.2)
\put(-1.05,-2.8){\special{psfile=fig2.ps hscale=50 vscale=50}}
\put(2.1,-0.25){\mbox{$\delta_i$}}
\put(0,1.25){\mbox{0.5}}
\put(2.4,-0.25){\mbox{$\delta_j$}}
\put(0,1.80){\mbox{0.7}}
\put(2.7,1.9){\mbox{$P_i(\theta)$}}
\put(2.7,2.5){\mbox{$P_j(\theta)$}}
\end{picture}

\vspace{0.5em}
Two IRF's for which $P_i(\theta)$ $\leq$ $P_j(\theta)$
but $\delta_i$ $<$ $\delta_j$ using the definition of
$\delta$ in logistic IRT models.
\end{center}
```

`Psfig/TeX` is a widely-available macro package for TEX and LATEX that many experienced LATEX users prefer for including `PostScript` figures into LATEX documents. With `psfig`-compatible `.dvi` processing programs like `dvips` (DOS/UNIX) and `OzTeX` (Macintosh), `PostScript` figures are automatically scaled and positioned on the page, and the proper amount of space is reserved. To include a figure in LATEX with `psfig`, add `psfig` as a style option in square brackets on the `\documentstyle` or `\documentclass` line of your document. Then invoke the macro `\psfig{figure=filename}`, where filename is the name of a `PostScript` file. `psfig` will automatically position the figure at the current point on the page, and reserve the proper amount of space. Your *local guide* or TEX guru will have detailed information for you if your installation has `psfig`.

There is really nothing special or unique about `PostScript` figures here. Other types of figures (`.gif`, `.jpeg`, etc.) can, in principle, be included in a formatted LATEX document also. The main question to ask is, will your `.dvi` processing program know what to do with the figure once it has it?

E Whitespace and Hyphenation

[Note: Added for June 1995 Interface meeting. –BJ]

This appendix introduces you to two nuisances that all LaTeX users eventually run into: whitespace and hyphenation. Unlike much of the rest of LaTeX, solving the problems of whitespace and hyphenation involves a lot of tricks and special cases that don't fit in well with the overall structure of LaTeX.

E.1 Working with Whitespace

LaTeX usually makes reasonable judgements about how much and where to put whitespace in your document, but sometimes you want to over-ride LaTeX's judgements for one reason or another. In some of the examples in Appendix D you probably recognized some commands, such as `\vspace` and `\hspace`, that insert or delete whitespace.

E.1.1 Lengths

Commands to insert or delete space need to be told how much space to insert or delete. Almost any reasonable units of measure will work, e.g. `1in` = 1 inch, `2.5mm` = 2.5 millimeters, etc. Some printer's units that are useful are:

Name	*Unit*	*Meaning*
Point	`1pt`	Approx. $1/72^2$ inch.
Em	`1em`	About the width of a capital "M" in the current type size.
Ex	`1ex`	Approx. `0.5em`, this is about the height of a lower-case "x" in the current type size.

Note that the `em` and the `ex` change size with the type size you are using. Although these may be hard to get used to, it is usually better to use these units of measure to specify spacing within a document, than to use the fixed units `in`, `mm` and `pt`. The reason is that if you have to change the type size of your document—say, from `12pt` type to `10pt` type—the spacing that you have specified in `em` and `ex` units will change too, and it will probably still look good in the new type size. If you have used `in` or `mm`, you will have to sweep through your whole document using trial-and-error to adjust all the spacing to look good in the new type size.

One other length of importance is `\fill`. This is a special kind of length that is known in TeX as a *rubber length*. It can shrink to zero length or expand to the full length or width of a page. As we shall see, you can use two `\fill`'s together to center text horizontally or vertically on a page.

E.1.2 Page Margins

LaTeX's defaults leave a lot of whitespace between the text area and the four edges of the printed page. There are special length commands that can be used to change the amount of whitespace in each of these margins. For example, this document was printed using the following commands (place immediately after the `\documentclass` or `\documentstyle` line).

```
\setlength{\topmargin}{0in}
\setlength{\oddsidemargin}{0in}
\setlength{\evensidemargin}{0in}
\setlength{\textheight}{8in}
\setlength{\textwidth}{6.5in}
```

[2] There are 72.27 points per inch.

For obscure reasons, TeX measures the `\topmargin` and `\sidemargin` from a point one inch down and one inch to the right of the upper left corner of the page. Thus, the commands above give a 1 inch margin at the top and left sides of the page, and specify that the main text area should be a 6.5"×8" rectangle. On an 8.5"×11" page, the text area is large but not too large, and nicely centered on the page.

(The separate `\oddsidemargin` and `\evensidemargin` lengths are useful for specifying different left margins on odd- and even-numbered pages, with certain style options. This is useful for formatting material that will be bound into a book, for example.)

E.1.3 Vertical Space

You can direct LaTeX to add or remove vertical space with the `\vspace` command. The space will be added or removed starting with the *output* line following the `\vspace`. LaTeX normally adjusts the whitespace at the beginning and end of each page, and in doing so may ignore your `\vspace` request. A vertical space request made with `\vspace*` is never ignored. Here are some examples:

- Effect of `\vspace`:

Here are two lines with no vspace. You can see that there is no extra space here. But there is lots

of space between the second line and this line.

```
Here are two lines with no vspace.
You can see that there is no extra
space here. \vspace{1.2em} But there
is lots of space between the second
line and this line.
```

- Difference between `\vspace` and `\vspace*`:

The `vspace` above is ignored since it is at the top of a page.

```
\vspace{1.2em}
The {\tt vspace} above is ignored
since it is at the top of a page.
```

The `vspace*` above is obeyed although it is at the top of a page.

```
\vspace*{1.2em}
The {\tt vspace*} above is obeyed
although it is at the top of a page.
```

- Centering text vertically on the page. Note the use of `\fill`:

This text is centered vertically on the page.

```
\vspace*{\fill}
This text is centered vertically
on the page.
\vspace*{\fill}
```

- Effect of negative `\vspace` ("back-up" on the page):

This text will appear higher on the page.
This text will appear lower on the page.

```
\vspace*{3em}
This text will appear
lower on the page.

\vspace*{-2.5em}
This text will appear
higher on the page.
```

E.1.4 Horizontal Space

You can also direct LaTeX to add or remove horizontal space with the `\hspace` command. The space will be added or removed immediately following the `\hspace` on the current *output* line. LaTeX normally adjusts the whitespace space at the beginning and end of each line, and in doing so may ignore your `\hspace` request. A horizontal space request made with `\hspace*` is never ignored.

Just as with `\vspace`, you can use two `\hspace*{\fill}`'s to center material on a line, and you can use negative lengths with `\hspace` to shift material to "back up" on a line. You should be careful of whitespace before or after `\hspace`; The `\hspace` command is considered a word, and LaTeX will insert an inter-word space before and after it if you type spaces before and after it.

Here is a 0.5 inch space.
Here is a 0.5 inch space.
Here is a 0.5 inch space.
Negative `hspace` is a backspace—like this.

```
Here\hspace{0.5in}is a 0.5 inch space.

Here \hspace{0.5in}is a 0.5 inch space.

Here \hspace{0.5in} is a 0.5 inch space.

Negative {\tt hspace} is a backspace---like
this.\hspace{-0.5in}/////
```

E.1.5 Forced End-of-Line

You have already seen the command \\ used to separate lines in `tabular` and other environments (see Section 10), as well as in array structures in mathematics (see Appendix B). In regular text you can also use \\ to force a line break. In either setting, you can add vertical space after the line break by specifying a length in square brackets after the \\.

Here is a sentence that keeps going...
and going...
and going...

and going...

and going!!!

```
Here is a sentence that keeps
going\ldots \\ and going\ldots \\[0.5em]
and going\ldots \\[1.0em] and going\ldots
\\[1.5em] and going!!!
```

You can specify positive or negative lengths in the brackets, and the effects will be much the same as with `\vspace`.

E.1.6 Forced End-of-Page

Sometimes you would like LaTeX to start a new page before it has completely filled the page you are on. You can type either `\newpage` and `\clearpage` on a line by itself where you want the new page to start. I prefer `\clearpage`, since it also prints any *floats* (see Section 11) that haven't been printed yet, before starting the new page.

E.1.7 Space Between Lines

For draft copies of a document, it is often useful to widen the distance between lines of text to space-and-a-half, or even double-spaced, width, to facilitate copy editing and other physical marking of the document. You can achieve space-and-a-half formatting by redefining LaTeX's `\baselinestretch` command, like this:

```
\renewcommand{\baselinestretch}{1.5}\tiny\normalsize
```

This is adequate for most copy-editing purposes. Other spacing can be achieved by placing the appropriate integer (2 for double-spacing, etc.) where the 1.5 is in this line. For obscure technical reasons, the two type size changes are needed to ensure that the new spacing takes effect.

E.1.8 Space Between Words

Usually LaTeX allows itself to start a new output line within a paragraph between any two words in the paragraph. For some word combinations, such as two-word last names like "von Beethoven" or section names like "Section 2", you do not want the two words split across two lines. To prevent this you can use the ~ symbol to indicate an "unbreakable inter-word space", e.g., `von~Beethoven` or `Section~2`.

E.2 Hyphenation

LaTeX posesses many spelling rules that help it to hyphenate words, and will rarely hyphenate a word in the wrong place. More commonly LaTeX will fail to hyphenate a word, or you will find that in some context any hyphenation at all would not be appropriate.

E.2.1 Too Little Hyphenation

The most common situation in which LaTeX will fail to hyphenate is with words that are themselves hyphenated, like "x-radiation," since LaTeX has been taught that is bad form to hyphenate hyphenated words. You can force LaTeX to consider hyphenation by using the `\-` command at each possible hyphenation point.

All hyphenated words, in particular "x-radiation", frustrate LaTeX's spelling rules, and LaTeX compensates by adding or subtracting space between words on the line to force the word to fit without hyphenation. This can be fixed by inserting possible hyphenation points, e.g. "x-radiation".

```
All hyphenated words, in particular ``x-radiation'',
frustrate \LaTeX's spelling rules, and \LaTeX\
compensates by adding or subtracting space between
words on the line to force the word to fit without
hyphenation.  This can be fixed by inserting possible
hyphenation points, e.g. ``x-ra\-di\-a\-tion''.
```

E.2.2 Too Much Hyphenation

Finally, there are times when you don't want LaTeX to hyphenate a word at all. The simplest way to prevent this is to use the `\mbox` command. This causes LaTeX to treat the material in the `\mbox` as an unbreakable chunk of text.

If there are particular words, perhaps "major-domo," that you want un-hyphenated, try: "major-domo."

```
If there are particular words, perhaps ``major-domo,''
that you want un-hyphenated, try: ``\mbox{major-domo}.''
```

References

[1] Leslie Lamport. LaTeX—*A Document Preparation System—User's Guide and Reference Manual.* Addison-Wesley, Reading, MA, USA, 1985.

[2] Donald E. Knuth. *The TeXbook*, volume A of *Computers and Typesetting.* Addison-Wesley, Reading, MA, USA, 1986.

[3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The* LaTeX *Companion* Addison-Wesley, Reading, MA, USA, 1993.

Electronic Resources

- The Comprehensive TeX Archive Network at Sam Houston State University:

  ftp: `ftp.shsu.edu`, login with user name `anonymous` and send your e-mail address as the password.

  WWW: `ftp://ftp.SHSU.edu/tex-archive/`

- UCLA Department of Statistics TeX and LaTeX documentation:

  WWW: `http://www.stat.ucla.edu/develop/tex/`

- `Statlib` at Carnegie Mellon University, Department of Statistics (will soon publish bibliography style files for many major statistical journals.)

  Email: Send the one-line email message "`send index`" to `statlib@stat.cmu.edu`

  ftp: `lib.stat.cmu.edu (128.2.241.142)`, login with user name `statlib` and send your e-mail address as the password.

  gopher: Port 70 on `lib.stat.cmu.edu`

  WWW: `http://lib.stat.cmu.edu/` or `http://lib.stat.cmu.edu/.index.html`