

# Document Preparation using L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

Phil Spector

Statistical Computing Facility  
Department of Statistics  
University of California, Berkeley

1

## Word Processing Programs

- WYSIWYG “What You See Is What You Get”
  - sometimes WYSIAYG “What You See Is All You Get”
  - fast, easy-to-use
  - non-portable
  - generally proprietary
- Mark-up Languages
  - human readable, transportable
  - some (like T<sub>E</sub>X/ L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>) are free
  - files must be processed before they are displayed or printed
  - entire document is processed, not just a page at a time

2

## T<sub>E</sub>X vs. L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

- T<sub>E</sub>X
  - developed by Donald Knuth of Stanford (1970s/early 1980s)
  - about 300 basic commands
  - complete control of the printed page
  - additional commands (macros) are generally necessary
- L<sup>A</sup>T<sub>E</sub>X
  - originally developed by Leslie Lamport of DEC (mid 1980s)
  - a set of macros written in T<sub>E</sub>X
  - focuses on the structure of the document
  - widely accepted by journals
  - many additional macros are easily available
  - upgraded in 1990 to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

3

## The Basics of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

- You must specify a document class with `\documentclass`
- Precede text with `\begin{document}` and end with `\end{document}`
- Type sentences in the normal way — extra spaces in your input are ignored
- Leave a blank line for paragraphs
- Beware of the 10 special characters:  
$$\# \ \$ \ \% \ \& \ \_ \ \{ \ \} \ \sim \ \hat \ \backslash$$
- The command `latex` processes a `.tex` file, and creates a `.dvi` (device-independent file)
- Display using `xdvi`; print using `dvips`

4

## A Simple Example

```
\documentclass{article}
\begin{document}
This is a test.
It is only a test.
I understand about 20\% of
\LaTeXe.
% A percent sign is the
% beginning of a comment

\LaTeXe\ is simple to use,
but
there are a few tricks you
should know.
\end{document}
```

This is a test. It is only a test. I understand about 20% of  $\text{\LaTeX} 2_{\epsilon}$ .  $\text{\LaTeX} 2_{\epsilon}$  is simple to use, but there are a few tricks you should know.

5

## The `\usepackage` command

After deciding upon an overall style, you can specify additional features in your document through the `\usepackage` command. Packages can be used to change fonts, provide support for languages other than English, modify existing commands or create new commands.

One popular package is `fullpage`; it modifies the default margins to use more of the page for text. To use this package, include the following line after the `\documentclass` command:

```
\usepackage{fullpage}
```

If you are using more than one package, their names should be separated by commas.

Not all packages will be available on all systems.

6

## Quotes and Special Symbols

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> uses different symbols for open quote and close quote:

Open Quote ‘ ‘ “ “ Close Quote ’ ’ ” ”

There are several different types of dashes:

em-dash --- — en-dash -- - hyphen - -

Seven of the ten special symbols are easy to produce:

`\$` \$ `\&` & `\%` % `\#` #  
`\_` - `\{` { `\}` }

The other three symbols can be produced in math mode:

`$_{tilde{}}$` ~ `$_{backslash}$` \ `$_{grave{}}$` `

## Accents

In	Out	In	Out	In	Out
<code>\'a</code>	á	<code>\u b</code>	ǔ	<code>\~c</code>	ĉ
<code>\"d</code>	ď	<code>\.e</code>	ė	<code>\'f</code>	ƒ
<code>\v g</code>	ǧ	<code>\H h</code>	ĥ	<code>\=\i</code>	ī
<code>\~\j</code>	ĵ	<code>\b k</code>	ķ	<code>\c l</code>	ł
<code>\d m</code>	ṁ	<code>\t no</code>	ñó		

### Notes

1. `\i` and `\j` produce dotless i and j.
2. Most accents operate on one letter, but need spaces if they are defined with a letter
3. An alternative is to provide the argument in curly braces (`{ }`); `\v{g}` results in ǧ.

## Modes

When  $\text{\LaTeX} 2_{\epsilon}$  processes your input, it does so in one of three modes:

1. Paragraph mode - Your input is arranged into paragraphs, and the paragraphs are placed on the page. Line and page breaks are determined using standard typographic rules.  
*Note:* To suppress the usual indentation at the start of a paragraph, use the `\noindent` command.
2. LR mode - Your input is treated as a single “box” running from left to right. No line breaking is done.
3. math mode - Your input is considered as symbols to construct a formula. Spacing is determined by typographic rules for formulas.

## Footnotes

$\text{\LaTeX} 2_{\epsilon}$  uses the `\footnote` command to produce footnotes.

Simply enclose the desired text in curly braces. By default, it starts with the number 1 and increments it each time you use a footnote.<sup>1</sup>

To have the footnotes marked by symbols use the command:

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}
```

Footnotes will cycle through the symbols \* † ‡ ||, then \*\* †† ‡‡ ||||, etc. Be careful of using footnotes in boxes (where they don't work) and in the `minipage` environment, where they are indexed by letter, not number.

---

<sup>1</sup>This is the way footnotes will appear in your document.

## Environments

To temporarily change the way  $\text{\LaTeX} 2_{\epsilon}$  formats your document, you can use one of several different *environments*. The format to change environments is:

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

Inside of environments, there are different commands available, and different for how you compose your input.

You can also change the values of various parameters, and their effect takes place only inside the environment.

Required arguments to environments are placed in curly braces (`{ }`), optional arguments are placed in square brackets (`[ ]`).

If you get a strange error, make sure you've included appropriate `\end{environment}` commands.

## Environments for Controlling Alignment

By default,  $\text{\LaTeX} 2_{\epsilon}$  hyphenates words and inserts interword space to justify text to both the left and right margins. Three environments are useful for overriding this behavior:

- `center` - centers each line of text on the page
- `flushleft` - justifies text to the left margin only
- `flushright` - justifies text to the right margin only

```
\begin{center}  
Text typed in the center  
environment will not be  
justified but will be  
neatly centered on each  
line  
\end{center}
```

Text typed in the center  
environment will not be  
justified but will be neatly  
centered on each line

The  $\text{\TeX}$  command `\centerline{text}` will center *text* on a line.

## The `verbatim` environment

To display text as you type it (in typewriter font), use the `verbatim` environment:

Here is a sample of C code:

```
\begin{verbatim}
```

```
for(i=0;i<n;i++){
    sum += x[i];
    sum2 += x[i] * x[i];
}
```

```
\end{verbatim}
```

Here is a sample of C code:

```
for(i=0;i<n;i++){
    sum += x[i];
    sum2 += x[i] * x[i];
}
```

Inline, you can use `\texttt{}` to change to typewriter font, or, if you need to include symbols, `\verb+text+`, where *text* is the text you want, and the `+` can be any symbol not in *text*.

## Modifying Typefaces

The most common need for modifying typefaces is to provide *emphasis*, achieved through the `\emph{}` command. This will choose the appropriate typeface, regardless of the current default.

A number of other font attributes can be modified, including the family (overall style), shape (slanted, italics, etc.), and series (weight and width).  $\text{\LaTeX} 2_{\epsilon}$  commands to temporarily modify these include:

Family			
<code>\textrm{}</code>	Normal	<code>\texttt{}</code>	typewriter
<code>\textsf{}</code>	sans serif		
Shape			
<code>\textup{}</code>	upright	<code>\textit{}</code>	<i>italics</i>
<code>\textsl{}</code>	<i>slanted</i>	<code>\textsc{}</code>	SMALL CAPS
Series			
<code>\textmd{}</code>	medium	<code>\textbf{}</code>	<b>Bold</b>

## Modifying Typefaces (cont'd)

The command `\textnormal{}` always uses the default font.

Commands from the previous table can be nested. For example `\textsl{\textbf{Slanted Bold}}` results in ***Slanted Bold***; `\textbf{\textsf{Bold Sans Serif}}` results in **Bold Sans Serif**. If a particular combination of attributes can't be found, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> provides a substitute.

For modifying large blocks of text, the following environments are available:

Family: `rmfamily` `ttfamily` `sffamily`  
Shape: `upshape` `itshape` `slshape` `scshape`  
Series: `mdseries` `bfseries`

If necessary, defaults for family, shape and series can be forced with the `\normalfont` declaration.

## Changing The Default Font

By default, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> uses the Computer Modern Times font. To change fonts, use the `\usepackage{fontname}` command.

`computer modern (default)`

The quick brown fox jumped over the lazy white horse.  
bookman

**The quick brown fox jumped over the lazy white horse.**

`newcent`

**The quick brown fox jumped over the lazy white horse.**

`palatino`

**The quick brown fox jumped over the lazy white horse.**

`times`

The quick brown fox jumped over the lazy white horse.

More fonts may be available on your system - any Postscript font can be used with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.



## Modifying Type Size

There are ten declarations in  $\text{\LaTeX} 2_{\epsilon}$  for altering the size of type:

<code>\tiny</code>	UCB Statistics
<code>\scriptsize</code>	UCB Statistics
<code>\footnotesize</code>	UCB Statistics
<code>\small</code>	UCB Statistics
<code>\normalsize</code>	UCB Statistics
<code>\large</code>	UCB Statistics
<code>\Large</code>	UCB Statistics
<code>\LARGE</code>	UCB Statistics
<code>\huge</code>	UCB Statistics
<code>\Huge</code>	UCB Statistics

## Modifying Type Size(cont'd)

The type size declarations on the previous slide can be used before an environment to change the default type size within that environment. For example, to get a smaller version of the `verbatim` environment, use:

```
\small
\begin{verbatim}
. . .
\end{verbatim}
\normalsize
```

For more precise control of the font size, you can pass both a font size and an interline spacing size to the `\fontsize` command:

```
\fontsize{8}{9}
```

would set the font size to 8 points, and the interline spacing to 9 points. Just as with other typesetting commands, you can restore the default size and spacing with `\normalsize`.

## Sectioning Commands

The exact sectioning commands depend on the document style you choose. For example, the `article` document style allows the following choices:

<code>\part</code>	<code>\section</code>
<code>\subsection</code>	<code>\subsubsection</code>
<code>\paragraph</code>	<code>\subparagraph</code>

The commands on the first two lines produced numbered headings and are tabulated in the table of contents. Numbers are automatically incremented as sections are encountered.

## Labels and Crossreferences

You can use a `\label` command to associate a label with a section, equation, figure or table number. If the number changes, the label will automatically change.

You can refer to these labels anywhere in the text using the `\ref` command, or the `\pageref` command to produce the page number.

```
\section{First section}
\label{first}
This is the first section.
\section{Second section}
So far we've only done
Section~\ref{first}\,
(p.~\pageref{first}).
```

### **1 First section**

This is the first section.

### **2 Second section**

So far we've only done Section 1 (p. 20).

## Itemized Lists

An itemized list precedes each list element with a symbol:

```
\begin{itemize}
\item                               • Table of Contents
Table of Contents
\item                               • Mathematical Expressions
Mathematical Expressions          – In-line
\begin{itemize}
\item                               – Display
In-line
\item
Display
\end{itemize}
\end{itemize}
```

## The `itemize` environment

There are four levels of nesting available for itemized lists. The symbols used at each level are stored in `\labelitemi`, `\labelitemii`, etc. For example,

```
\renewcommand{\labelitemi}{\Diamond}
\renewcommand{\labelitemii}{\rightarrow}
```

would change the previous list to

- ◇ Table of Contents
- ◇ Mathematical Expressions
  - In-line
  - Display

You can override these by specifying a symbol in square brackets after the `\item` declaration.

## The `enumerate` Environment

Enumerated lists precede each item with a number or letter. Labels can be used for crossreferencing. There are four levels of nesting available.

<code>\begin{enumerate}</code>	
<code>\item</code>	
Why I like <code>\LaTeXe</code>	1. Why I like <code>\LaTeX</code>
<code>\begin{enumerate}</code>	
<code>\item</code>	(a) It's free
It's free	
<code>\item</code>	(b) You can do anything
<code>\label{anything}</code>	
You can do anything	
<code>\item</code>	(c) It's fun
It's fun	
<code>\end{enumerate}</code>	
<code>\item</code>	
Why I don't like <code>\LaTeXe</code>	2. Why I don't like <code>\LaTeX</code>
<code>\begin{enumerate}</code>	
<code>\item</code>	(a) It's complicated
It's complicated	
<code>\item</code>	(b) It doesn't always listen
It doesn't always listen	
<code>\item</code>	(c) See 1b above.
See <code>\ref{anything}</code> above.	
<code>\end{enumerate}</code>	
<code>\end{enumerate}</code>	

## The `tabular` environment

To begin a table use

```
\begin{tabular}[pos]{cols}
```

*pos* (optional) alignment: default $\Rightarrow$ center **t** $\Rightarrow$ top **b** $\Rightarrow$ bottom.

*cols* column alignment: **l** $\Rightarrow$ left **r** $\Rightarrow$ right **c** $\Rightarrow$ centered

`|,||`  $\Rightarrow$  single or double vertical rule

`@{text}`  $\Rightarrow$  replace intercolumn space with *text*

`*{n}{cols}`  $\Rightarrow$  repeat *cols* *n* times

Follow with the rows of the table, using an ampersand (&) between columns, and two backslashes (`\`) at the end of each line.

Use `\hline` to produce a full horizontal rule, or `\cline{i-j}` to produce a rule spanning columns *i* through *j*.

Finish the table with

```
\end{tabular}
```

## A Simple Example

```
\begin{tabular}{|lllr|}  
\hline  
Name&Description&Directory&Size (Kb)\\  
\hline  
BMDP&Statistical Package&{\tt bmdp}&60616\\  
Lapack&Subroutine Library&{\tt lapack}&27715\\  
Lisp-Stat&Statistical Package&{\tt lisp-stat}&27715\\  
Linda&Parallelization Package&{\tt linda}&7036\\  
SAS&Statistical Package&{\tt sas609}&153159\\  
\hline  
\end{tabular}
```

Name	Description	Directory	Size (Kb)
BMDP	Statistical Package	bmdp	60616
Lapack	Subroutine Library	lapack	27715
Lisp-Stat	Statistical Package	lisp-stat	27715
Linda	Parallelization Package	linda	7036
SAS	Statistical Package	sas609	153159

25

## The Multicolumn command

You can use the `\multicolumn` command anywhere in the rows of a table to span several columns.

The form of the command is

$$\backslash\text{multicolumn}\{num\}\{col\}\{item\}$$

*num* number of columns to span

*col* a replacement for the *col* declaration from the `tabular` environment for the columns being spanned

*item* the contents of the spanned columns.

A `\multicolumn` command must either begin a row or appear after an ampersand (`&`).

26

### A More Complex Table: Output

	Whiteboard		Corkboard	
Size	Aluminum	Oak	Aluminum	Oak
3'×2'	\$ 33.50	\$ 35.00	\$ 30.75	\$ 33.00
4'×3'	56.00	59.75	52.00	55.00
5'×3'	110.00	120.00	93.00	97.00
6'×4'	140.00	168.00	115.00	140.00
8'×4'	183.00	213.00	162.00	185.00

27

### A More Complex Table: Input

```

\begin{center}
\begin{tabular}{|r@{${}\times$}l|r@{.}l|r@{.}l|r@{.}l|}
\hline
\multicolumn{2}{|c|}&
\multicolumn{4}{|c|}{Whiteboard}&
\multicolumn{4}{c|}{Corkboard}\\
\hline
\multicolumn{2}{|c|}{Size}&
\multicolumn{2}{c|}{Aluminum}&
\multicolumn{2}{c|}{Oak}&
\multicolumn{2}{c|}{Aluminum}&
\multicolumn{2}{c|}{Oak}\\
\hline
3'&2'& \$ 33&50& \$ 35&00& \$ 30&75& \$ 33&00\\
4'&3'& &56&&59&75&&52&&55&00\\
5'&3'& &110&&120&&93&&97&&00\\
6'&4'& &140&&168&&115&&140&&00\\
8'&4'& &183&&213&&162&&185&&00\\
\end{tabular}
\end{center}

```

28

## The `tabbing` environment

As an alternative to the `tabular` environment, you can use the `tabbing` environment to make aligned displays. Begin with

```
\begin{tabbing}
```

Follow with a line of text, setting the tab stops with `\=`.

You can end this line with `\kill` to suppress printing.

Finally enter the rows of the display, using `\>` to advance to the next tab stop, and `\\` to mark the end of a line.

Finish the display with

```
\end{tabbing}
```

If you need to change the tab settings, you can store the old ones with the `\pushtabs` command, and restore them with the `\poptabs` command.

*Note:* If you end the last line with a `\\`, it will add extra space.

## Example

```
\begin{tabbing}
Address:X=University of California, BerkeleyX\=\kill
Name:\>Phil Spector\\
Address:\>Department of Statistics\\
\>University of California, Berkeley\\
\>367 Evans Hall \# 3860\\
\>Berkeley, CA 94720-3860\\
Phone:\>(510)-642-9056\>(answering machine)\\
email:\>{\tt spector@stat.berkeley.edu}\>(Internet)
\end{tabbing}
```

---

Name: Phil Spector  
Address: Department of Statistics  
University of California, Berkeley  
367 Evans Hall # 3860  
Berkeley, CA 94720-3860  
Phone: (510)-642-9056 (answering machine)  
email: spector@stat.berkeley.edu (Internet)

## Floats: Tables and Figures

A float is a part of your document which may appear in a different place in the final document than it does in the  $\text{\LaTeX}2_{\epsilon}$  input file.

To specify a float, use either

```
\begin{figure}[pos] text \end{figure}
```

```
or \begin{table}[pos] text \end{table}
```

*pos* up to four letters from among **b**(bottom), **h**(“here”), **p**(separate page), and **t**(top). Including a **!** overrides some of  $\text{\LaTeX}2_{\epsilon}$ 's usual placement rules.

*text* the body of the table or figure

You can include an optional `\caption{}` command to include a caption. Table captions are preceded by the word “Table” while figure captions are preceded with the word “Figure”. If you include a `\label` in a figure or table, it must appear after the `\caption`.

## Example

Suppose we wish to create a float consisting of the table of software used to illustrate the `\tabular` command.

```
\begin{table}[h]
\begin{tabular}
\hline Name&Description&Directory&Size (Kb)\
. . .
\end{tabular}
\caption{Sizes and Locations of Software Packages}
\end{table}
```

---

Name	Description	Directory	Size (Kb)
BMDP	Statistical Package	bmdp	60616
Lapack	Subroutine Library	lapack	27715
Lisp-Stat	Statistical Package	lisp-stat	27715
Linda	Parallelization Package	linda	7036
SAS	Statistical Package	sas609	153159

Table 1: Sizes and Locations of Software Packages



## Including Postscript Figures

PostScript figures can be easily incorporated into  $\text{\LaTeX} 2_{\epsilon}$  using the `epsfig` package. Your input file should be in encapsulated PostScript; specifically, there *must* be a “Bounding Box” defined in the file. You can include the package in your document by inserting

```
\usepackage{epsfig}
```

after the `\documentclass` command.

The basic form of the `\epsfig` command is:

```
\epsfig{figure=postscriptfile, height=..., width=..., ...}
```

Other commands are available to rotate or clip the figure, and to specify the bounding box in the `\epsfig` command.

It is often useful to surround the command with `\centerline{}`.

## More About Floats

One of the more frustrating tasks in producing an attractive document is getting your tables and figures to appear just where you want them. Part of this is due to  $\text{\LaTeX}$ 's default behavior of demanding that there is at least half a page of text on each page of floats. You can override this behavior with the following commands:

```
\renewcommand{\floatpagefraction}{.9}
```

```
\renewcommand{\topfraction}{.9}
```

```
\renewcommand{\bottomfraction}{.9}
```

```
\renewcommand{\textfraction}{.1}
```

The first three values should be the same, and the last value should be equal to  $1 - \text{\floatpagefraction}$ .

## Headers and Footers

Headers and footers are the text which is automatically printed at the top and bottom of each page of your output.

In  $\text{\LaTeX} 2_{\epsilon}$  you control headers and footers using the `\pagestyle` declaration, which accepts one of the following four values.

- `plain` (default) Empty header, page number in footer
- `headings` Section name in header, page number in footer
- `myheadings` Argument of `\markright` or `\markboth` in header, page number in footer
- `empty` Header and footer are both empty

It is often necessary to repeat the `pagestyle` request using `\thispagestyle` for  $\text{\LaTeX} 2_{\epsilon}$  to recognize your request, especially at the beginning of a document.

## Some Unix Commands Related to $\text{\LaTeX} 2_{\epsilon}$

<code>latex</code> <i>or</i> <code>latex2e</code>	processes $\text{\LaTeX} 2_{\epsilon}$ input into a <code>.dvi</code> file
<code>dvi2tty</code>	Displays a <code>.dvi</code> file on a non-windowed system
<code>dviselect</code>	Extracts part of a <code>.dvi</code> file
<code>dviconcat</code>	Puts together multiple <code>.dvi</code> files
<code>dvips</code>	Converts <code>.dvi</code> files to PostScript files (for printing)
<code>dvi2pdf</code>	Converts <code>.dvi</code> files to PDF
<code>xdvi</code>	Xwindow system previewer for <code>.dvi</code> files
<code>ghostview</code>	Xwindow system previewer for PostScript files
<code>metafont</code>	Constructs fonts for use in $\text{\TeX}$ and $\text{\LaTeX} 2_{\epsilon}$ ; may be called automatically by <code>xdvi</code> .

## Including Files in $\text{\LaTeX} 2_{\epsilon}$ Source files

If you are producing a large document, you may want to keep sections or chapters in separate files, and then instruct  $\text{\LaTeX} 2_{\epsilon}$  to include those files as part of its input. The `\input{file}` directive tells  $\text{\LaTeX} 2_{\epsilon}$  to use the named file as its input before returning to processing other directives in the current file. If no extension is given for *file*,  $\text{\LaTeX} 2_{\epsilon}$  uses the extension `.tex`.

By inserting a percent sign (%) before some of the `\input` commands, you can process only certain parts of your document.

$\text{\LaTeX} 2_{\epsilon}$  searches the directories found in the environmental variable `TEXINPUTS` in order to find the files referenced in the `\input` command.

`TEXINPUTS` should be a colon separated list of directories; usually the current directory (`.`) will be the first member of the list.

## Using Math Mode

There are a variety of ways of processing text in math mode:

- In text, surround math input with dollar signs (\$)   
Not `x > 0`, but `$x>0$`.  $\Rightarrow$  Not `x > 0`, but `x > 0`.

- For numbered equations, use   
`\begin{equation} \dots \end{equation}`

You can refer to equations using the `\label` and `\ref` mechanism mentioned previously.

- For unnumbered equations, use   
`\begin{displaymath} \dots \end{displaymath}`

Both the `equation` and `displaymath` environments allow only a single, one line formula.  $\text{\LaTeX}$  will *not* break up your formula to fit on the page.

## Special Math Words

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

## Fonts in Math Mode

If it is necessary to include normal text in a formula, simply enclose it in an `mbox`; this maintains the proper spacings between symbols and words.

```
\begin{displaymath}
y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad \text{for } i = 1, \dots, n
\end{displaymath}
```

results in

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad \text{for } i = 1, \dots, n$$

To change the font of individual symbols, use one of the following commands:

<code>\mathrm{}</code>	Normal	<code>\mathtt</code>	Typewriter
<code>\mathbf{}</code>	<b>Bold</b>	<code>\mathsf</code>	SansSerif
<code>\mathit{}</code>	<i>Italics</i>	<code>\mathcal</code>	<i>CALLIGRAPHIC</i>

## Math Symbols

<code>\ldots</code>	...	<code>\aleph</code>	$\aleph$	<code>\hbar</code>	$\hbar$
<code>\imath</code>	$i$	<code>\jmath</code>	$j$	<code>\ell</code>	$\ell$
<code>\wp</code>	$\wp$	<code>\Re</code>	$\Re$	<code>\Im</code>	$\Im$
<code>\mho*</code>	$\mho$	<code>\cdots</code>	...	<code>\prime</code>	$'$
<code>\emptyset</code>	$\emptyset$	<code>\nabla</code>	$\nabla$	<code>\surd</code>	$\surd$
<code>\top</code>	$\top$	<code>\bot</code>	$\perp$	<code>\ </code>	$\ $
<code>\angle</code>	$\angle$	<code>\vdots</code>	$\vdots$	<code>\forall</code>	$\forall$
<code>\exists</code>	$\exists$	<code>\neg</code>	$\neg$	<code>\flat</code>	$\flat$
<code>\natural</code>	$\natural$	<code>\sharp</code>	$\sharp$	<code>\backslash</code>	$\backslash$
<code>\partial</code>	$\partial$	<code>\ddots</code>	$\ddots$	<code>\infty</code>	$\infty$
<code>\Box*</code>	$\square$	<code>\Diamond*</code>	$\diamond$	<code>\triangle</code>	$\triangle$
<code>\clubsuit</code>	$\clubsuit$	<code>\diamondsuit</code>	$\diamondsuit$	<code>\heartsuit</code>	$\heartsuit$
<code>\spadesuit</code>	$\spadesuit$				

\*requires `\usepackage{latexsym}`

## Arrows

<code>\leftarrow</code>	$\leftarrow$	<code>\Leftarrow</code>	$\Leftarrow$	<code>\rightarrow</code>	$\rightarrow$
<code>\Rightarrow</code>	$\Rightarrow$	<code>\leftrightarrow</code>	$\leftrightarrow$	<code>\Leftrightarrow</code>	$\Leftrightarrow$
<code>\mapsto</code>	$\mapsto$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\leftharpoonup</code>	$\leftharpoonup$
<code>\leftharpoondown</code>	$\leftharpoondown$	<code>\rightleftharpoons</code>	$\rightleftharpoons$	<code>\longleftarrow</code>	$\longleftarrow$
<code>\Longleftarrow</code>	$\Longleftarrow$	<code>\longrightarrow</code>	$\longrightarrow$	<code>\Longrightarrow</code>	$\Longrightarrow$
<code>\longleftrightarrow</code>	$\longleftrightarrow$	<code>\Longleftrightarrow</code>	$\Longleftrightarrow$	<code>\longmapsto</code>	$\longmapsto$
<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\rightharpoonup</code>	$\rightharpoonup$	<code>\rightharpoondown</code>	$\rightharpoondown$
<code>\leadsto*</code>	$\leadsto$	<code>\uparrow</code>	$\uparrow$	<code>\Uparrow</code>	$\Uparrow$
<code>\downarrow</code>	$\downarrow$	<code>\Downarrow</code>	$\Downarrow$	<code>\updownarrow</code>	$\updownarrow$
<code>\Updownarrow</code>	$\Updownarrow$	<code>\nearrow</code>	$\nearrow$	<code>\searrow</code>	$\searrow$
<code>\swarrow</code>	$\swarrow$	<code>\nwarrow</code>	$\nwarrow$		

## Accents

<code>\hat{a}</code>	$\hat{a}$	<code>\check{a}</code>	$\check{a}$	<code>\acute{a}</code>	$\acute{a}$	<code>\grave{a}</code>	$\grave{a}$	<code>\bar{a}</code>	$\bar{a}$
<code>\vec{a}</code>	$\vec{a}$	<code>\dot{a}</code>	$\dot{a}$	<code>\ddot{a}</code>	$\ddot{a}$	<code>\breve{a}</code>	$\breve{a}$	<code>\tilde{a}</code>	$\tilde{a}$

\*requires `\usepackage{latexsym}`

## Relation Symbols

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

You can create negations using the `\not` command:

`\not\equiv`  $\Rightarrow$   $\neq$                       `\not\in`  $\Rightarrow$   $\notin$

---

\*requires `\usepackage{latexsym}`

## Binary Operation Symbols

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

---

\*requires `\usepackage{latexsym}`

## Greek Letters (lowercase)

<code>\alpha</code>	$\alpha$	<code>\beta</code>	$\beta$	<code>\gamma</code>	$\gamma$	<code>\delta</code>	$\delta$
<code>\epsilon</code>	$\epsilon$	<code>\varepsilon</code>	$\varepsilon$	<code>\zeta</code>	$\zeta$	<code>\eta</code>	$\eta$
<code>\theta</code>	$\theta$	<code>\vartheta</code>	$\vartheta$	<code>\gamma</code>	$\gamma$	<code>\kappa</code>	$\kappa$
<code>\lambda</code>	$\lambda$	<code>\mu</code>	$\mu$	<code>\nu</code>	$\nu$	<code>\xi</code>	$\xi$
<code>\pi</code>	$\pi$	<code>\varpi</code>	$\varpi$	<code>\rho</code>	$\rho$	<code>\varrho</code>	$\varrho$
<code>\sigma</code>	$\sigma$	<code>\varsigma</code>	$\varsigma$	<code>\tau</code>	$\tau$	<code>\upsilon</code>	$\upsilon$
<code>\phi</code>	$\phi$	<code>\varphi</code>	$\varphi$	<code>\chi</code>	$\chi$	<code>\psi</code>	$\psi$
<code>\omega</code>	$\omega$						

## Greek letters (uppercase)

<code>\Gamma</code>	$\Gamma$	<code>\Delta</code>	$\Delta$	<code>\Theta</code>	$\Theta$	<code>\Lambda</code>	$\Lambda$
<code>\Xi</code>	$\Xi$	<code>\Pi</code>	$\Pi$	<code>\Sigma</code>	$\Sigma$	<code>\Upsilon</code>	$\Upsilon$
<code>\Phi</code>	$\Phi$	<code>\Psi</code>	$\Psi$	<code>\Omega</code>	$\Omega$		

## Subscripts and Superscripts

- Subscripts and Superscripts are available only in math mode.
- For subscripts, use `_` or `\sb`
- For superscripts, use `^` or `\sp`
- If the script is more than one character, use curly braces (`{}`)
- For both subscripts and superscripts, use both in either order
- For scripts *before* symbols, use `{ }^{\dots}` or `{ }_{\dots}`

<code>x_i</code>	$x_i$	<code>x_{i-1}</code>	$x_{i-1}$	<code>x^2</code>	$x^2$
<code>x_i^2</code>	$x_i^2$	<code>x^2_i</code>	$x^2_i$	<code>x^{i^2}</code>	$x^{i^2}$
<code>a^{\{b^{\{c^d\}}\}}</code>	$a^{b^{c^d}}$	<code>{ }_nC_m</code>	${}_nC_m$	<code>{ }_n^rC_m^i</code>	${}_n^rC_m^i$





## Variable-Sized Symbols

<code>\sum</code>	$\Sigma$	<code>\prod</code>	$\Pi$	<code>\coprod</code>	$\coprod$
<code>\int</code>	$\int$	<code>\oint</code>	$\oint$		

There are two styles for limits on the variable-sized symbols.

Input: `\sum_{i=1}^n x_i`  
`\int_0^{\infty} f(x)dx`  
`\prod_{i=1}^n l(\mathbf{x}_i)`

Inline:  $\sum_{i=1}^n x_i$        $\int_0^{\infty} f(x)dx$        $\prod_{i=1}^n l(\mathbf{x}_i)$

Display:

$$\sum_{i=1}^n x_i \quad \int_0^{\infty} f(x)dx \quad \prod_{i=1}^n l(\mathbf{x}_i)$$

## Display-style limits

You can force display-style limits by following the symbol with `\limits`; following the symbol with `\nolimits` forces inline style.

In fact, you can change the meaning of subscripts and superscripts for any symbol by enclosing it in `\mathop{...}`, and following it with `\limits`.

Example:

<code>F_n(x)</code>	
<code>\mathop{\longrightarrow}\limits_{n \rightarrow \infty} F(x)</code>	$F_n(x) \xrightarrow[n \rightarrow \infty]{} F(x)$
<code>F_n(x)\longrightarrow_{n \rightarrow \infty} F(x)</code>	$F_n(x) \xrightarrow{n \rightarrow \infty} F(x)$

## Variable Size Delimiters

Many delimiters in L<sup>A</sup>T<sub>E</sub>X change size depending on their context. They are used in pairs:  $\backslash\left\mathit{delim}_1 \mathit{formula} \backslash\right\mathit{delim}_2$  where  $\mathit{delim}_1$  and  $\mathit{delim}_2$  are chosen from:

$($	$($	$)$	$\backslash{$	$\{$	$\}$
$[$	$[$	$]$	$\backslashbackslash$	$\backslash$	$/$
$\lgroup$	$\lgroup$	$\rgroup$	$\lceil$	$\rceil$	$\rceil$
$\lfloor$	$\lfloor$	$\rfloor$	$\uparrow$	$\Uparrow$	$\Uparrow$
$\updownarrow$	$\updownarrow$	$\Updownarrow$	$\downarrow$	$\Downarrow$	$\Downarrow$
$\lgroup$	$\lgroup$	$\rgroup$	$\arrowvert$	$\Arrowvert$	$\Arrowvert$
$\langle$	$\langle$	$\rangle$	$\vert$	$\Vert$	$\Vert$

$\mathit{delim}_1$  or  $\mathit{delim}_2$  equal to a period ( $.$ ) gives an “invisible” delimiter.

## Other Constructions

$\sqrt{x}$	$\sqrt{x}$	$\sqrt[n]{abc}$	$\sqrt[n]{abc}$
$\frac{x}{y}$	$\frac{x}{y}$	$x \over y$	$\frac{x}{y}$
$x \atop y$	$\frac{x}{y}$	$x \choose y$	$\binom{x}{y}$
$x \brack y$	$\left[ \begin{array}{c} x \\ y \end{array} \right]$	$x \brace y$	$\left\{ \begin{array}{c} x \\ y \end{array} \right\}$
$x \above 2pt y$	$\frac{x}{y}$	$\widetilde{xyz}$	$\widetilde{xyz}$
$\overleftarrow{xyz}$	$\overleftarrow{xyz}$	$\overline{xyz}$	$\overline{xyz}$
$\overbrace{xyz}$	$\overbrace{xyz}$	$\underbrace{xyz}$	$\underbrace{xyz}$
$\widehat{xyz}$	$\widehat{xyz}$	$\overrightarrow{xyz}$	$\overrightarrow{xyz}$
$\underline{xyz}$	$\underline{xyz}$	$x'$	$x'$

In display mode, a sub or superscript on many of these constructions will center the argument under or over the construction. For example,

$\underbrace{X_j, \dots, X_k}_{i1}$  will produce  $\underbrace{X_j, \dots, X_k}_{i1}$ .

## Examples

$$J_n(x) = \frac{x^n}{2^n n!} \left\{ 1 - \frac{x^2}{2^2 \cdot 1!(n+1)} + \frac{x^4}{2^4 \cdot 2!(n+1)(n+2)} - \dots \right\}$$

$$W^2 = \int_{-\infty}^{\infty} \{S_n(x) - F_0(x)\}^2 dF_0(x)$$

$$W^2 = \int_{-\infty}^{\infty} \{S_n(x) - F_0(x)\}^2 dF_0(x)$$

53

## Examples

$$P(T) = \frac{1}{N^{[n]}} \sum_{j=0}^n \left\{ \binom{n}{j} (Np)^{[j]} (Nq)^{[n-j]} t^j \right\}$$

$$P(T) = \frac{1}{N^{[n]}} \sum_{j=0}^n \left\{ \binom{n}{j} (Np)^{[j]} (Nq)^{[n-j]} t^j \right\}$$

$$f(z) = \frac{n^n z^{n-1}}{a^n \Gamma(n)} \left( \log \frac{a}{z} \right)^{n-1}$$

$$f(z) = \frac{n^n z^{n-1}}{a^n \Gamma(n)} \left( \log \frac{a}{z} \right)^{n-1}$$

54

## Examples

```
\lambda_2=\frac{\prod_i\left[\sum_{\alpha}(x_{i\alpha}-\bar{x}_i)^2\right]^{\frac{1}{2}N}}{\left[\sum_{i,\alpha}(x_{i,\alpha}-\bar{x}_i)^2/p\right]^{\frac{1}{2}pN}}
```

---

$$\lambda_2 = \frac{\prod_i \left[ \sum_{\alpha} (x_{i\alpha} - \bar{x}_i)^2 \right]^{\frac{1}{2}N}}{\left[ \sum_{i,\alpha} (x_{i,\alpha} - \bar{x}_i)^2 / p \right]^{\frac{1}{2}pN}}$$


---

```
\lambda_2=\frac{\prod_i\left[\sum_{\alpha}(x_{i\alpha}-\bar{x}_i)^2\right]^{\frac{1}{2}N}}{\left[\sum_{i,\alpha}(x_{i,\alpha}-\bar{x}_i)^2/p\right]^{\frac{1}{2}pN}}
```

---

$$\lambda_2 = \frac{\prod_i \left[ \sum_{\alpha} (x_{i\alpha} - \bar{x}_i)^2 \right]^{\frac{1}{2}N}}{\left[ \sum_{i,\alpha} (x_{i,\alpha} - \bar{x}_i)^2 / p \right]^{\frac{1}{2}pN}}$$

## The array environment

The `array` environment is like the `tabular` environment, but text is set in math mode. The `array` environment can be used inside other math environments.

Delimiters can be placed around arrays in the usual way.

```
\left[
\begin{array}{cccc}
x_{11}&x_{12}&\cdots&x_{1p} \\
x_{21}&x_{22}&\cdots&x_{2p} \\
\vdots&\vdots&\ddots&\vdots \\
x_{n1}&x_{n2}&\cdots&x_{np}
\end{array}
\right]_{n \times p}
```

## Examples

```

\left[\begin{array}{c}y_1\\y_2\\ \vdots \\y_n\end{array}\right]=
\left[\begin{array}{cccc}x_{11}&x_{12}&\cdots&x_{1p}\\x_{21}&x_{22}&\cdots&x_{2p}\\ \vdots &\vdots &\ddots &\vdots \\x_{n1}&x_{n2}&\cdots&x_{np}\end{array}\right]
\left[\begin{array}{c}\beta_1\\\beta_2\\ \vdots \\ \beta_p\end{array}\right]+
\left[\begin{array}{c}\epsilon_1\\\epsilon_2\\ \vdots \\ \epsilon_n\end{array}\right]

```

---

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

## Examples

```

\left(
\begin{array}{cccc}
\gamma_{11}&\gamma_{12}&\cdots&\gamma_{1q}\\
\gamma_{21}&\gamma_{22}&\cdots&\gamma_{2q}\\
\multicolumn{4}{c}{\hfill.\hfill.\hfill.\hfill.\hfill.}\\
\gamma_{31}&\gamma_{32}&\cdots&\gamma_{3q}\\
\end{array}
\right)

```

---

$$\begin{pmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1q} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2q} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \gamma_{31} & \gamma_{32} & \cdots & \gamma_{3q} \end{pmatrix}$$



## Multi-line formulas

To get a formula which spans more than one line, you have to decide where to break the formula. You can then use the alignment characters (`&`) to place the continuation in the appropriate place.

```
\begin{eqnarray*}
\{1 \over 2\}\{1+M_1(h)\}&=&(b-a)\sum_{t=1}^{M/2}\{\exp(-h^2s_t^2)\} \\
&&\qquad\{-2\exp(\{1\over 2\}h^2s_t^2)\}|Y_i|^2 \\
&&+n^{-1}h^{-1}(2\pi)^{-1/2}\}. \\
\end{eqnarray*}
```

$$\frac{1}{2}\{1 + M_1(h)\} = (b - a) \sum_{t=1}^{M/2} \left\{ \exp(-h^2 s_t^2) - 2 \exp\left(\frac{1}{2} h^2 s_t^2\right) |Y_i|^2 + n^{-1} h^{-1} (2\pi)^{-1/2} \right\}.$$

Note the use of the empty brackets (`{}`) before the “-” in the continuation line, to insure proper spacing. The `\qqquad` indents the line by an appropriate amount of space.

## More on the `eqnarray` environment

To use the `eqnarray` environment inside a complex construction you must surround it in a `parbox`.

For example, this technique can be used to place a large curly brace to the right of a set of equations.

```
\begin{displaymath}
\left.
\begin{array}{l}
\int_0^\infty \frac{\sin px \cos qx}{x} dx = \frac{1}{2}\pi, \quad |p| > |q| \\
= \frac{1}{4}\pi, \quad |p| = |q| \\
= 0, \quad |p| < |q|
\end{array}
\right\}.
```

## How Math Mode Works

To determine the spacing between math symbols,  $\text{\LaTeX}$  first classifies them into one of 8 categories of “atoms”:

1. ordinary characters (Ord) - like numbers and greek letters
2. operators (Op) - like summation signs and integrals
3. binary operators (Bin) - like plus and minus signs
4. relations (Rel) - like less than and greater than
5. open (Open) - like left bracket
6. close (Close) - like right bracket
7. punctuation (Punct) - like comma or period
8. inner (Inner) - like fractions and similar constructions

## Spacing between Math Atoms

	Ord	Op	Bin	Rel	Open	Close	Punct	Inner
Ord	0	1	(2)	(3)	0	0	0	(1)
Op	1	1	*	(3)	0	0	0	(1)
Bin	(2)	(2)	*	*	(2)	*	*	(2)
Rel	(3)	(3)	*	0	(3)	0	0	(3)
Open	0	0	*	0	0	0	0	0
Close	0	1	(2)	(3)	0	0	0	(1)
Punct	(1)	(1)	*	(1)	(1)	(1)	(1)	(1)
Inner	(1)	1	(2)	(3)	(1)	0	(1)	(1)

1=thin space

2=medium space

3=thick space

Values in parentheses become 0 in sub and superscripts.



## Units of Length

Unit	Symbol	Inches	Centimeters
big point	<b>bp</b>	0.0139	0.0353
cicero	<b>cc</b>	0.1777	0.4514
centimeter	<b>cm</b>	0.3937	1.0000
didot point	<b>dd</b>	0.0148	0.0376
inch	<b>in</b>	1.0000	2.5400
millimeter	<b>mm</b>	0.0394	0.1000
pica	<b>pc</b>	0.0166	0.0422
point	<b>pt</b>	0.0138	0.0351
scaled point	<b>sp</b>	$2.11 \times 10^{-7}$	$5.34 \times 10^{-7}$

- em**      approximately the width of an uppercase “M”.
- ex**      approximately the height of a lowercase “x”.
- mu**      (math mode only)mathematical unit (1/18 of an **em**).

Each digit is  $\frac{1}{2}$  em, and an em-dash (---) is exactly one em.

## Spacing

- `\_`              literal space
- `~`                interword space, preventing line break
- Example: with references, use `Section~\ref{...}`
- `\newline` or `\\`    forces linebreak
- `\\[len]`            forces linebreak and adds *len* vertical space
- `\newpage`         forces pagebreak
- `\hspace{len}`        horizontal space            suppressed at line break
- `\hspace*{len}`      horizontal space            not suppressed
- `\vspace{len}`        vertical space                suppressed at page break
- `\vspace*{len}`      vertical space                not suppressed

Using *len* equal to `\fill` expands to fill the line or page

The commands `\hfill` and `\vfill` are abbreviations for

`\hspace*{\fill}` and `\vspace*{\fill}`, respectively

## Manipulating Lengths

You can create a length variable equal to the length of specified text by first creating a length variable with `\newlength` and then setting its value with `\settowidth`. For example:

```
\newlength{\mystring}
\settowidth{\mystring}{\bf Whatever I want}
```

The length variable `\mystring` will now represent the length of the phrase “Whatever I want” set in bold face type, and can now be used in any context where a length is required.

You can increase or decrease the size of a length parameter using the `\addtolength` command. Suppose we wish to make `\mystring` one inch longer than its current length:

```
\addtolength{\mystring}{1in}
```

## Spacing in Math Mode

```
\,          thin space
\int_0^{\infty}\int_0^{\infty}f(x)g(y)dx\,dy
\int_0^{\infty}\int_0^{\infty}f(x)g(y)dx dy
\mbox{var}\,x=\sigma^2
var x = \sigma^2
\:          medium space          used less often than \,
\;          thick space
\!          negative thin space
\int\!z\,dx\,dy instead of \int z\,dx\,dy
\int z dx dy instead of \int z dx dy
\quad      2 em-spaces
x_i>\quad\hbox{for $i=1,\ldots,n$}
x_i > 0    for i = 1, \dots, n
```

## Fine Adjustments in Paragraph Mode

$\text{\LaTeX}$  will always try to produce a page which minimizes the badness of the output using complex rules.

You can modify some of these rules using the following commands.

In each case the optional argument  $i$  ranges from 0(does nothing) to 4(strong suggestion).

- `\linebreak[i]` - encourages a line break.
- `\nolinebreak[i]` - discourages a line break.
- `\pagebreak[i]` - encourages a page break.
- `\nepagebreak[i]` - discourages a page break.
- `\-` - discretionary hyphen. Inserting in a word encourages hyphenation at that point if necessary. You can use the `\hyphenation` command to provide a list of hyphenation preferences.

## Using LR mode

By default,  $\text{\LaTeX}$  tries to break lines in a reasonable way. You can override this by putting text in a box with the `\mbox` command.

Text is usually broken at convenient points, but you can force text to stay together if you need to.

Text is usually broken at convenient points, but you can force text to stay together if you need to.

---

Text is usually broken at convenient points, but you can `\mbox{force text to stay together}` if you need to.

---

Text is usually broken at convenient points, but you can force text to stay together if you need to.

## Fine Adjustments in Math Mode

Two commands which are useful for moving things a small amount:

`\raiselen`            raises a box by the specified amount  
`\mkernlen`            adds horizontal space (*len* must be in mu)

It is often necessary to put text to be moved in a box using

`\hbox{text}`            creates a box (in LR mode) containing text

To “trick” L<sup>A</sup>T<sub>E</sub>X into thinking something is there when it’s not, use

`\vphantomchar`        creates an empty box big enough to hold *char*

A special case is `\mathstrut`, a phantom parenthesis ()).

To control the size of type in math mode use:

`\displaystyle`        size used in `displaymath` environment  
`\textstyle`            size used in inline equations  
`\scriptstyle`         size used for subscripts  
`\scriptscriptstyle`    size used for sub-subscripts

## Case Study: Continued Fraction

(Note: `\newcommand{\ee}[1]{e^{-#1\pi\sqrt{5}}}`)

$\frac{1}{1+\frac{\ee{2}}{1+\frac{\ee{4}}{1+\frac{\ee{6}}{1+\ddots}}}}$	$\frac{1}{1+\frac{e^{-2\pi\sqrt{5}}}{1+\frac{e^{-4\pi\sqrt{5}}}{1+\frac{e^{-6\pi\sqrt{5}}}{1+\ddots}}}}$
---	--

We need to force the “1”s to be larger, and move the `\ddots` down:

$\frac{1}{\displaystyle 1+\frac{\ee{2}}{\displaystyle 1+\frac{\ee{4}}{\displaystyle 1+\frac{\ee{6}}{\displaystyle 1+\raise-1.5ex\hbox{\$ \ddots \$}}}}}$	$\frac{1}{1+\frac{e^{-2\pi\sqrt{5}}}{1+\frac{e^{-4\pi\sqrt{5}}}{1+\frac{e^{-6\pi\sqrt{5}}}{1+\ddots}}}}$
--	--

## Case Study: “Bar notation”

The goal is to produce an expression like:

$$f(x, y) \Big|_{\substack{x=a \\ y=b}}$$

A first try might be

$$f(x, y) |_{\{x=a \ \text{atop} \ y=b\}} \qquad f(x, y) \Big|_{\substack{x=a \\ y=b}}$$

We can make the bar larger by using `\left.` and `\vphantom`:

$$\left. \vphantom{\hbox{\Large {}}} f(x, y) \right|_{\{x=a \ \text{atop} \ y=b\}} \qquad f(x, y) \Big|_{\substack{x=a \\ y=b}}$$

Finally, we can raise the “subscript” a little:

$$\left. \vphantom{\hbox{\Large {}}} f(x, y) \right|_{\substack{\raise3pt\hbox \\ {\tiny $x=a \ \text{atop} \ y=b$}}}} \qquad f(x, y) \Big|_{\substack{x=a \\ y=b}}$$

## Defining Commands (Macros)

You can define new commands with

`\newcommand{cmdname}[n]{def}`

*cmdname* name of the new command (with leading `\`)

*n* (optional) number of parameters

*def* definition of the new command

In the simplest case, `\newcommand` performs text substitution.

`\newcommand{\eg}{for example}` would result in

when, `\eg`, you do this  $\Rightarrow$  when, for example, you do this

`\newcommand{\implies}{\Rightarrow}` lets you type

`$A\implies B$`  $\Rightarrow A \Rightarrow B$

To redefine an existing command, use the `\renewcommand` command.

## Macros with Arguments

Suppose we wish to display fractions as  $\frac{a}{b}$  instead of as  $a/b$  or  $\frac{a}{b}$ .

We wish to write a macro called `\lfrac`, that works like `\frac`, but using smaller type.

`\newcommand{\lfrac}[2]{\scriptstyle\frac{#1}{#2}}` could be used as:

`-\lfrac{1}{2}x^2` \$ not `-\frac{1}{2}x^2` \$  $\Rightarrow$   $-\frac{1}{2}x^2$  not  $-\frac{1}{2}x^2$

Note the “extra” curly braces to contain the `\scriptstyle` declaration.

Macros are useful whenever an expression appears many times:

`\newcommand{\fsum}[3]{\sum_{#2=0}^{#3}#1_{#2}e^{i\omega#2}}` results in:

$$\fsum{\alpha}{j}{q} \Rightarrow \sum_{j=0}^q \alpha_j e^{i\omega j}$$

## Case Study: Matrix Notation

Suppose we want to display the dimensions of a matrix below its symbol, by using `\mathop` and `\limits`:

$$\mathop{X}\limits_{n \times p} \Rightarrow X_{n \times p}$$

This works fine until we use more than one symbol:

$$y_{n \times 1} = X_{n \times p} \beta_{p \times 1} + \epsilon_{n \times 1}$$

This can be fixed by inserting a `\mathstrut` before each symbol.

Finally, the dimensions can be set in smaller type, and the subscript raised a small amount before writing the macro:

`\newcommand{\mat}[3]{\mathop{\mathstrut #1}\limits_{\raise.5em\hbox{\scriptscriptstyle #2\times#3}}}`

This allows us to type `\mat{y}{n}{1}=\mat{X}{n}{p} \dots` to get:

$$y_{n \times 1} = X_{n \times p} \beta_{p \times 1} + \epsilon_{n \times 1}$$

## Style Parameters

One of the attractive features of L<sup>A</sup>T<sub>E</sub>X is that the overall style of the document is defined for you — you just have to worry about the content.

Sometimes it is necessary to modify some of the parameters which define a style, for example, to change margins, paragraph indentation, etc.

To change a style parameter, use the `\setlength` command. For example, the `\textheight` parameter determines the height of text on a page. To set its value to 2 inches you would use:

```
\setlength{\textheight}{2in}
```

Alternatively, the T<sub>E</sub>X form of the command can be used:

```
\textheight=2in
```

## Rigid and Rubber Lengths

Some of the style parameters have values which contain tolerances; that is, they can be specified to be within a range, rather than a fixed value, and are known as rubber length parameters. For example `\parskip`, the vertical space between paragraphs is a rubber length parameter.

To specify a rubber length parameter, use a syntax like the following:

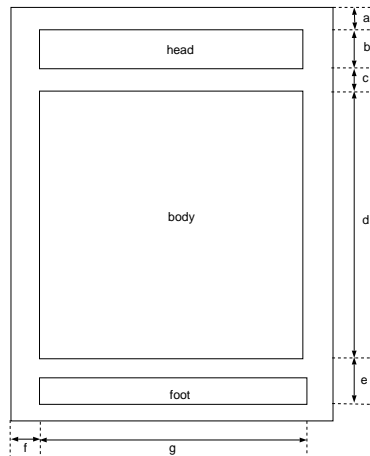
```
\parskip=12pt plus 4pt minus 2pt
```

or

```
setlength{\parskip}{12pt plus 4pt minus 2pt}
```

A full list of rigid and rubber length parameter can be found in the L<sup>A</sup>T<sub>E</sub>X manual.

## Page Layout

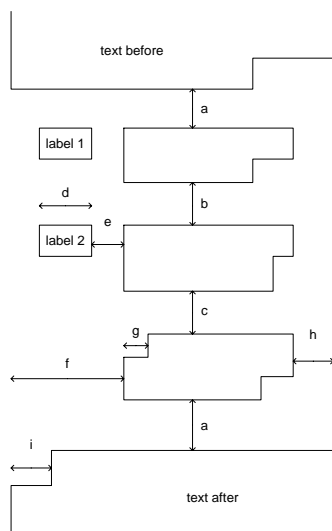


<i>a</i>	<code>\topmargin + 1 in.</code>
<i>b</i>	<code>\headheight</code>
<i>c</i>	<code>\headsep</code>
<i>d</i>	<code>\textheight</code>
<i>e</i>	<code>\footskip</code>
<i>f</i>	<code>\footheight</code>
<i>g</i>	<code>\oddsidemargin + 1 in.</code>
<i>g</i>	<code>\evensidemargin + 1 in.</code>
<i>h</i>	<code>\textwidth</code>

from **L<sup>A</sup>T<sub>E</sub>X Line by Line**, by Antoni Diller

79

## List Layout



<i>a</i>	<code>\topsep + \parskip</code> <i>or</i> <code>\topsep + \parskip + \partsep</code>
<i>b</i>	<code>\itemsep + \parsep</code>
<i>c</i>	<code>\parsep</code>
<i>d</i>	<code>\labelwidth</code>
<i>e</i>	<code>\labelsep</code>
<i>f</i>	<code>\leftmargin</code>
<i>g</i>	<code>\listparindent</code>
<i>h</i>	<code>\rightmargin</code>
<i>i</i>	<code>\parindent</code>

from **L<sup>A</sup>T<sub>E</sub>X Line by Line**, by Antoni Diller

80



## Other Modifications

In addition to length parameters, some aspects of a document's appearance are controlled by variables which are not lengths. For example, `\baselinestretch` is a value whose default is 1, and which is used to multiply the length parameter `\baselineskip` to determine the interline spacing.

If you wanted to get a double-spaced document, you could include the command:

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

Another useful variable is `\arraystretch`, which controls the space between lines in an array or tabular environment. It also defaults to 1, and can be modified with `\renewcommand`.

## Books

- Lamport, Leslie (1994), *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*, Addison-Wesley, Reading, MA.
- Knuth, Donald (1986) *The T<sub>E</sub>Xbook*, Addison-Wesley, Reading, MA.
- Diller, Antoni (1993) *L<sup>A</sup>T<sub>E</sub>X Line by Line*, John Wiley & Sons, New York, NY.
- Kopka, H. and Daly, P. *A Guide to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*, Addison-Wesley, Wokingham, England.
- Hahn, Jane (1991) *L<sup>A</sup>T<sub>E</sub>X for everyone*, Personal TEX, Inc., Mill Valley, CA.
- Buerger, D. J. (1990) *L<sup>A</sup>T<sub>E</sub>X for Engineers and Scientists*, McGraw-Hill, New York, NY.
- Goossens, Michael *et. al.* (1994) *The L<sup>A</sup>T<sub>E</sub>X Companion*, Addison-Wesley, Reading, MA.

## Internet Resources

- Local Help Files: type `help latex`  
Files in the subdirectories of `/usr/local/TeX/tex`.
- FAQ for the `comp.text.tex` newsgroup on `rtfm.mit.edu` in `pub/usenet/news.answers/tex-faq` (use anonymous ftp)
- CTAN (Comprehensive T<sub>E</sub>X Archive Network)  
`ftp.dante.de`      `ftp.tex.ac.uk`      `pip.shsu.edu`  
Accessible by anonymous ftp, and contain source for T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X as well as macros, articles, and lots of other information.
- Cameron Etezadi's L<sup>A</sup>T<sub>E</sub>X short course  
(<http://riceinfo.rice.edu/Computer/Documents/Classes/Unix/class/class.html> on the WWW)

## Internet Resources (Continued)

- Essential L<sup>A</sup>T<sub>E</sub>X by John Warbrick  
(`info/latex-essential/essential.tex` and `info/elm.tex` at the CTAN archives)
- An Introduction to T<sub>E</sub>X and friends by Gavin Maltby  
(`info/maltby-intro.tex` at the CTAN archives)
- Hypertext L<sup>A</sup>T<sub>E</sub>X Help System  
(<http://www.stat.ucla.edu/develop/tex/help/latex/LaTeX.html> ON the WWW)
- Texhax Digest - a regular collection of problems and solutions from users, distributed by email  
Subscription requests: `TeXhax-request@tex.ac.uk`  
(Send message of `subscribe texhax` or `unsubscribe texhax`)  
Archived at CTAN archives in `digests/texhax`.