Document Preparation using LaTeX

Phil Spector

Statistical Computing Facility
Department of Statistics
University of California, Berkeley

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 TeX/ LaTeX) are free
  - files must be processed before they are displayed or printed
  - entire document is processed, not just a page at a time

TEX vs. LATEX

- TEX
  - 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
- LATEX
  - developed by Leslie Lamport of DEC (mid 1980s)
  - a set of macros written in TEX
  - focuses on the structure of the document; various *styles* take care of the details
  - widely accepted by journals
  - many additional macros are easily available

The Basics of LATEX

- You must specify a document style with `\documentstyle`
- 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:

  `#  $  %  &  _  {  }  ~  ^  \`
- The command `latex` processes a `.tex` file, and creates a `.dvi` (device-independent file)
- Display using `xdvi`; print using `dvips`

A Simple Example

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

\LaTeX\ 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 LaTeX.

LaTeX is simple to use, but there are a few tricks you should know.

Quotes and Special Symbols

LaTeX 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
\'a	á	\u b	b̆	\^c	ĉ
\"d	d̈	\.e	ė	\`f	f̀
\v g	ǧ	\H h	h̋	\=\i	ī
\~\j	j̃	\b k	k̲	\c l	l̦
\d m	ṃ	\t no	n͡o		

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 LaTeX 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

LaTeX 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.[1]

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.

---

[1] This is the way footnotes will appear in your document.

Environments

To temporarily change the way LaTeX 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, LaTeX 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 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 `\tt` 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

There are seven type changing directives in LaTeX. Generally, their scope is within a set of curly braces. The `\em` command can be used with any of these for *emphasis*.

`\bf`	bold	**Changing typefaces**	*When you need*
`\it`	italic	*can make your text*	a more emphatic
`\rm`	roman	more exciting, but	*typeface, you don't*
`\sc`	small caps	YOU SHOULD BE	*need to worry*
`\sf`	sans serif	aware that it	*about which*
`\sl`	slanted	*is possible to*	typeface you
`\tt`	typewriter	`overdo it!`	*should use.*

For example, the fourth line in the table was produced with `{\sc you should be}` and `{\sc \em need to worry}`

Modifying Type Size

There are ten declarations in LaTeX for altering the size of type:

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

Sectioning Commands

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

`\part`	`\section`
`\subsection`	`\subsubsection`
`\paragraph`	`\subparagraph`

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. 16).

Itemized Lists

An itemized list precedes each list element with a symbol:

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

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

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.

```
\begin{enumerate}
\item
Why I like \LaTeX
\begin{enumerate}
\item
It's free
\item
\label{anything}
You can do anything
\item
It's fun
\end{enumerate}
\item
Why I don't like \LaTeX
\begin{enumerate}
\item
It's complicated
\item
It doesn't always listen
\item
See \ref{anything} above.
\end{enumerate}
\end{enumerate}
```

1. Why I like LaTeX
   (a) It's free
   (b) You can do anything
   (c) It's fun
2. Why I don't like LaTeX
   (a) It's complicated
   (b) It doesn't always listen
   (c) See 1b above.

The `tabular` environment

To begin a table use

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

pos (optional) alignment: default⇒center `t`⇒top `b`⇒bottom.

cols column alignment: `l`⇒left `r`⇒right `c`⇒centered
`|`,`||` ⇒ single or double vertical rule
`@{`*text*`}` ⇒ replace intercolumn space with *text*
`*{`*n*`}{`*cols*`}` ⇒ 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

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

`\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 (`&`).

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

A More Complex Table: Input

```
\begin{center}
\begin{tabular}{|r@{$\times$}l|r@{.}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&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\\
\end{tabular}
\end{center}
```

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 LaTeX 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)

text the body of the table or figure

You can include an optional `\caption{}` command to include a caption with a table or figure. The only difference between the two environments is that captions of tables are preceded by the word "Table" while those of figures 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}[hhhh]
\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 LaTeX using the `\psfig` macro, written by Trevor J. Darrel. Your input file should be in encapsulated PostScript; specifically, there *must* be a "Bounding Box" defined in the file. You can include the macros in your document by either including the statement

`\input psfig`

near the beginning of your document or by specifying `psfig` as an option to the `\documentstyle` statement (*i.e.*`\documentstyle[...,psfig]{...}`) The basic form of the `\psfig` command is:

`\psfig{figure=`*postscriptfile*`, height=..., width=..., ...}`

It is often useful to surround the command with `\centerline{}`. For more information, consult the help file, `help psfig`.

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 LaTeX 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 LaTeX to recognize your request, especially at the beginning of a document.

Some Unix Commands Related to LaTeX

`latex`	processes LaTeX input into a `.dvi` file
`dvi2tty`	Displays a `.dvi` file on a non-windowed system
`dviselect`	Extracts part of a `.dvi` file
`dviconcat`	Puts togther multiple `.dvi` files
`dvips`	Converts `.dvi` files to PostScript files (for printing)
`xdvi`	Xwindow system previewer for `.dvi` files
`ghostview`	Xwindow system previewer for PostScript files
`metafont`	Constructs fonts for use in TeX and LaTeX ; may be called automatically by `xdvi`.

Including Files in LaTeX Source files

If you are producing a large document, you may want to keep sections or chapters in separate files, and then instruct LaTeX to include those files as part of its input. The `\input{`*file*`}` directive tells LaTeX 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*, LaTeX uses the extension `.tex`.

LaTeX 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 $x>0$.` ⇒ Not x ¿ 0, but $x > 0$.
- For numbered equations, use
  `\begin{equation}` ··· `\end{equation}`
  You can refer to equations using the `\label` and `\ref` mechanism mentioned previously.
- For unnumbered equations, use
  `\begin{displaymath}` ··· `\end{displaymath}`

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

Special Math Words

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

Math Symbols

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

Arrows

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

Accents

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

Relation Symbols

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

You can create negations using the \not command:

\not\equiv $\Rightarrow \not\equiv$ \not\in $\Rightarrow \notin$

Binary Operation Symbols

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

Greek Letters (lowercase)

\alpha	α	\beta	β	\gamma	γ	\delta	δ
\epsilon	ϵ	\varepsilon	ε	\zeta	ζ	\eta	η
\theta	θ	\vartheta	ϑ	\gamma	γ	\kappa	κ
\lambda	λ	\mu	μ	\nu	ν	\xi	ξ
\pi	π	\varpi	ϖ	\rho	ρ	\varrho	ϱ
\sigma	σ	\varsigma	ς	\tau	τ	\upsilon	υ
\phi	ϕ	\varphi	φ	\chi	χ	\psi	ψ
\omega	ω						

Greek letters (uppercase)

\Gamma	Γ	\Delta	Δ	\Theta	Θ	\Lambda	Λ
\Xi	Ξ	\Pi	Π	\Sigma	Σ	\Upsilon	Υ
\Phi	Φ	\Psi	Ψ	\Omega	Ω		

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 {}^{...} or {}_{...}

x_i	x_i	x_{i-1}	x_{i-1}	x^2	x^2
x_i^2	x_i^2	x^2_i	x_i^2	x^{i^2}	x^{i^2}
a^{b^{c^d}}	$a^{b^{c^d}}$	{}_nC_m	${}_nC_m$	{}_n^rC_m^i	${}_n^rC_m^i$

Examples

```
\lim_{n \rightarrow \infty}
A^c_n=\bigcup_{j=1}^\infty A^c_j
```

$$\lim_{n \to \infty} A_n^c = \bigcup_{j=1}^{\infty} A_j^c$$

```
z_1z_2=r_1r_2
[\cos(\theta_1+\theta_2)+i\sin(\theta_1+\theta_2)]
```

$$z_1 z_2 = r_1 r_2[\cos(\theta_1 + \theta_2) + i \sin(\theta_1 + \theta_2)]$$

More on Superscripts

You can use the apostrophe (') to produce primes, or the symbol `\prime` as part of a superscript, but they can't be combined. (`f''^{2}` results in a "`Double subscript error`".)

Two possible solutions:

`\mbox{$f''$}^2` $\Rightarrow$ f''^2

`f^{\prime\prime 2}` $\Rightarrow$ f''^2

Note that LaTeX uses the group of characters before the ^ operator to determine the height of the superscript, so there is a difference between `(x+1)^3` ($(x+1)^3$) and `{(x+1)}^3` ($\left(x+1\right)^3$).

Variable-Sized Symbols

`\sum`	$\sum$	`\prod`	$\prod$	`\coprod`	$\coprod$
`\int`	$\int$	`\oint`	$\oint$		

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({\bf x}_i)
```

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

Display:

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

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:

```
F_n(x)
\mathop{\longrightarrow}\limits_
{n \rightarrow\infty}F(x)
```

$$F_n(x) \mathop{\longrightarrow}\limits_{n \rightarrow\infty} F(x)$$

```
F_n(x)\longrightarrow_
{n \rightarrow\infty}F(x)
```

$$F_n(x) \longrightarrow_{n \rightarrow\infty} F(x)$$

Variable Size Delimiters

Many delimiters in LaTeX change size depending on their context. They are used in pairs: `\left`*delim*$_1$ *formula* `\right`*delim*$_2$ where *delim*$_1$ and *delim*$_2$ are chosen from:

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

delim$_1$ or *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}`	$\genfrac{}{}{0pt}{}{x}{y}$	`{x \choose y}`	$\binom{x}{y}$
`{x \brack y}`	$\genfrac{[}{]}{0pt}{}{x}{y}$	`{x \brace y}`	$\genfrac{\{}{\}}{0pt}{}{x}{y}$
`{x \above 2pt y}`	$\genfrac{}{}{2pt}{}{x}{y}$	`\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,\cdots,X_k}_{i1}` will produce $\underbrace{X_j,\cdots,X_k}_{i1}$.

Examples

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

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

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

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

Examples

```
P(T)={1 \over N^{[n]}}
     \sum_{j=0}^n\left\{{n \choose 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{a \over z}\right)^{n-1}
```

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

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

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

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}
```

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

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]
```

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

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.}\\
\gamma_{31}&\gamma_{32}&\cdots&\gamma_{3q}\\
\end{array}
\right)
```

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

Examples

```
{\bf P}_2\equiv
\left[\begin{array}
{cc@{\hspace{2ex}}|@{\hspace{2ex}}c
@{\hspace{2em}}c@{\hspace{2em}}c}
1&0&0&\cdots&0\\ 0&1&0&\cdots&0\\ \hline 0&0&\\
\vdots&\vdots&\multicolumn{3}{c}{{}^{(n-2)}{\bf P}_2}\\ 0&0
\end{array}
\right]
```

$$\mathbf{P}_2 \equiv \left[\begin{array}{cc|ccc} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & {}^{(n-2)}\mathbf{P}_2 & \\ 0 & 0 & & & \end{array}\right]$$

The eqnarray environment

The **eqnarray** environment is like a table with a right justified column, a centered (and aligned) column, and a left justified column, using display style math mode. In addition to arrays of equations, the **eqnarray** environment can be used for multiline formulas.

The **eqnarray*** environment is like the **eqnarray** environment except that it does not number the equations. To suppress equation numbers for some of the equations in an **eqnarray** environment, include a **\nonumber** command somewhere on the input line.

```
\begin{eqnarray*}
(x+1)^2&=&(x+1)(x+1)\\
&=&x^2+x+x+1\\
&=&x^2+2x+1
\end{eqnarray*}
```

$$\begin{aligned} (x+1)^2 &= (x+1)(x+1) \\ &= x^2+x+x+1 \\ &= x^2+2x+1 \end{aligned}$$

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^2_t)\\
&&\qquad{}-2\exp({1\over 2}h^2s^2_t)\}|Y_i|^2
+n^{-1}h^{-1}(2\pi)^{-1/2}.
\end{eqnarray*}
```

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

Note the use of the empty brackets (`{}`) before the "−" in the continuation line, to insure proper spacing. The `\qquad` 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.
\parbox{.75\textwidth}{
\begin{eqnarray*}
\int^\infty_0 \frac{\sin px
\cos qx}{x}dx
& = & \half\pi,\qquad |p|>|q|\\
& = & \frac{1}{4}\pi, \qquad |p|=|q|\\
& = & 0,\qquad |p|<|q|
\end{eqnarray*}
}
\right\}.
\end{displaymath}
```

$$\left.\begin{array}{rcll} \displaystyle\int_0^\infty \frac{\sin px \cos qx}{x}dx & = & \frac{1}{2}\pi, & |p|>|q| \\ & = & \frac{1}{4}\pi, & |p|=|q| \\ & = & 0, & |p|<|q| \end{array}\right\}.$$

How Math Mode Works

To determine the spacing between math symbols, 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	`bp`	0.0139	0.0353
cicero	`cc`	0.1777	0.4514
centimeter	`cm`	0.3937	1.0000
didot point	`dd`	0.0148	0.0376
inch	`in`	1.0000	2.5400
millimeter	`mm`	0.0394	0.1000
pica	`pc`	0.0166	0.0422
point	`pt`	0.0138	0.0351
scaled point	`sp`	2.11×10^{-7}	5.34×10^{-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

Example: suppress extra space after periods

`Dr.\ Smith of the Stat.\ Dept.`	Dr. Smith of the Stat. Dept.
`Dr. Smith of the Stat. Dept.`	Dr. Smith of the Stat. Dept.

`~` interword space, preventing line break

Example: References

`Section~\ref{···}`

`\newline`	forces linebreak
`\\`	
`\\[`*len*`]`	forces linebreak and adds *len* vertical space
`\newpage`	forces pagebreak

Spacing (cont'd)

`\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

Spacing with Font Changes

`\/` italic correction (from slanted to non-slanted)

`{\it in vitro} test or {\it in vitro\/} test` ⇒ *in vitro* test or *in vitro* test

Sometimes it's helpful to include the space in the other font:

`{\tt em} space or {\tt em }space` ⇒ `em` space or `em` space

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
```

$\mathrm{var}\,x = \sigma^2$

`\:` 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 $\int z\,dx\,dy$

`\qquad` 2 em-spaces

```
x_i>0\qquad\hbox{for $i=1,\ldots,n$}
```

$x_i > 0 \qquad \text{for } i = 1, \ldots, n$

Fine Adjustments in Paragraph Mode

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.
- `\nopagebreak[`*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, 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:

`\raise`*len*	raises a box by the specified amount
`\mkern`*len*	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" LaTeX into thinking something is there when it's not, use

`\vphantom`*char*	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}}}}
```

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

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{$\ddots$}}}}}
```

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

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 \atop y=b}
```

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

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

```
\left.\vphantom{\hbox{\Large (}}
f(x,y)\right|_{x=a \atop y=b}
```

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

Finally, we can raise the "subscript" a little:

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

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

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` ⇒ when, for example, you do this

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

`$A\implies B$` ⇒ $A \Rightarrow B$

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}` $\Rightarrow$ $\displaystyle\sum_{j=0}^{q}\alpha_j e^{i\omega j}$

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}` $\Rightarrow \underset{n\times p}{X}$

This works fine until we use more than one symbol:

$$\underset{n\times 1}{y} = \underset{n\times p}{X}\ \underset{p\times 1}{\beta} + \underset{n\times 1}{\epsilon}$$

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}`$\cdots$ to get:

$$\underset{n\times 1}{y} = \underset{n\times p}{X}\ \underset{p\times 1}{\beta} + \underset{n\times 1}{\epsilon}$$

Style Parameters

One of the attractive features of LaTeX 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 TeX 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:

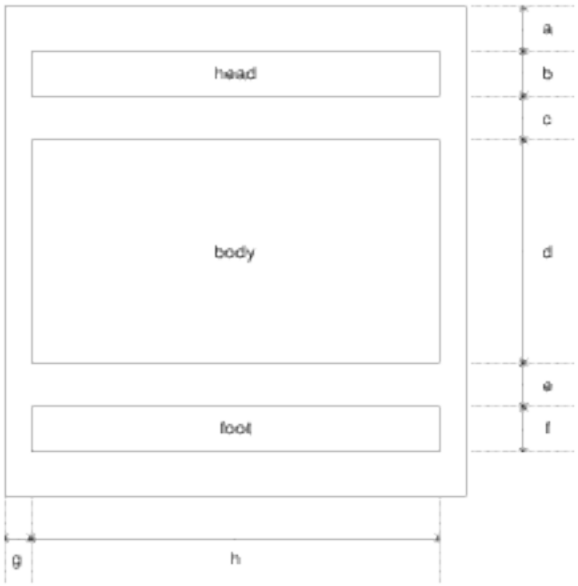
```
\parskip=12pt plus 4pt minus 2pt
```

or

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

A full list of rigid and rubber length parameter can be found in the LaTeX manual.

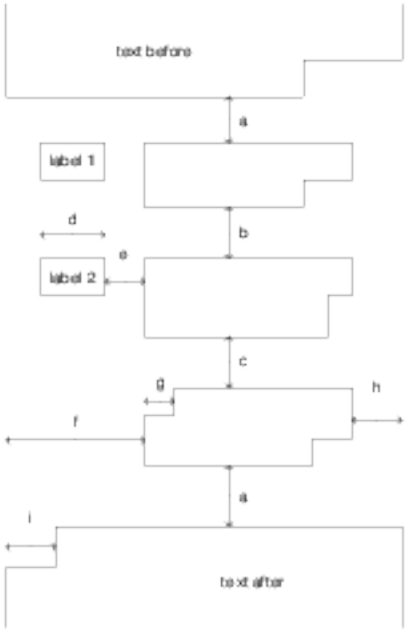
Page Layout



a	\topmargin + 1 in.
b	\headheight
c	\headsep
d	\textheight
e	\footskip − \footheight
f	\footheight
g	\oddsidemargin + 1 in.
g	\evensidemargin + 1 in.
h	\textwidth

from **LaTeX Line by Line**, by Antoni Diller

List Layout



a	\topsep + \parskip
	or \topsep + \parskip + \partopsep
b	\itemsep + \parsep
c	\parsep
d	\labelwidth
e	\labelsep
f	\leftmargin
g	\listparindent
h	\rightmargin
i	\parindent

from **LaTeX Line by Line**, by Antoni Diller

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`.

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 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 - `\floatpagefraction`.

Books

- Lamport, Leslie (1994), LaTeX: *A Document Preparation System*, Addison-Wesley, Reading, MA.
- Knuth, Donald (1986) *The* TeX*book*, Addison-Wesley, Reading, MA.
- Diller, Antoni (1993) LaTeX *Line by Line*, John Wiley & Sons, New York, NY.
- Hahn, Jane (1991) LaTeX *for everyone*, Personal TEX, Inc., Mill Valley, CA.
- Buerger, D. J. (1990) LaTeX *for Engineers and Scientists*, McGraw-Hill, New York, NY.
- Goossens, Michael *et. al.* (1994) *The* LaTeX *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 TeX Archive Network)
  `ftp.dante.de` `ftp.tex.ac.uk` `pip.shsu.edu`
  Accessible by anonymous `ftp`, and contain source for TeX and LaTeX as well as macros, articles, and lots of other information.
- Cameron Etezadi's LaTeX short course
  (`http://riceinfo.rice.edu/Computer/Documents/Classes/Unix/class/class.html` on the WWW)

Internet Resources (Continued)

- Essential LaTeX by John Warbrick
  (`info/latex-essential/essential.tex` and `info/elm.tex` at the CTAN archives)
- An Introduction to TeX and friends by Gavin Maltby
  (`info/maltby-intro.tex` at the CTAN archives)
- Hypertext LaTeX 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`.