

---

## Making Tables with LaTeX

Making tables with LaTeX is not generally difficult, as long as you know how the necessary commands are used. These examples of each of the commands will help you get started.

### *Starting tabular*

Each letter in the preamble signifies two things: 1) how many columns you want in this table, and 2) how are the contents of the columns to be positioned (left, right, or centered)

Syntax:

```
\begin{tabular}{lcr}rcc
```

↑

Table Preamble

Start a column by giving a letter in the preamble. The letter will start a column and at the same time assign the justification: l, r, or c determines the justification within each column: l to the left, r to the right, c to the center.

(Important! Remember to use the letter 'l', not the number '1')

Number of columns is optional, number of lines is optional

Each new column is started with &

May use less columns than are in preamble, but not more columns

Each line ends with \\

Notice how the preamble sets the way the contents of the columns are positioned? In the first case, the contents of the first column goes to the left, the second column centers, and the third column moves to the right.

```
\begin{tabular}{lcr}
Here&is our&table\\
a&b&c\\
\end{tabular}
```

.....

```
Here is our table
a      b      c
```

You don't need to use every column:

```
\begin{tabular}{lcr}  
Here&&table\\  
a&b&\\  
&&hello  
\end{tabular}
```

---

Here		table
a	b	
		hello

Font changes are contained within the column entry where they are written:

```
\begin{tabular}{lcr}  
\bf Here&\it is our&table\\  
a&b&c\\  
\end{tabular}
```

---

<b>Here</b>	<i>is our</i>	table
a	b	c

## All the things you can do in the preamble

If we want to have paragraphs of wrapping text appear in the table we can use the `p{<wrapping width>}` expression in the preamble:

```
\begin{tabular}{lcrp{1.5in}}
Here& is our&table&Here we have a place where we
can enter a lot of text and it will wrap
within a space of 1.5in.\\
a&b&c\\
\end{tabular}
```

.....

Here	is our	table	Here we have a place where we can enter a lot of text and it will wrap within a space of 1.5in.
a	b	c	

Here we add vertical lines in the preamble:

```
\begin{tabular}{|l||l|l|cr|}
Here& is our&table\\
a&b&c\\
\end{tabular}
```

.....

	Here		is our	table	
	a		b	c	

When we ask one column to go to the right and the next to the left, notice there is still a space between the two columns:

```
\begin{tabular}{r|c}
Here&is our&table\\
a&b&c\\
\end{tabular}
```

.....

Here	is our	table
a	b	c

To get rid of that space we can use an @ expression in the preamble.

Now we can get numbers to line up on the decimal point by using two columns for the numbers, with the part of the number preceding the decimal point going in the first column, and the part of the number following the decimal point going in the next column.

```
\begin{tabular}{r@{}|c}
Here&is our&table\\
a&b&c\\
2130&.456\\
30&.33\\
1&.4568\\
\end{tabular}
```

.....

Here	is our	table
a	b	c
2130	.456	
30	.33	
1	.4568	

If you want to, you can add something between the curly brackets and that will be added to every table line between the two columns:

```
\begin{tabular}{r@{ Hi! }lc}
Here&is our&table\\
a&b&c\\
\end{tabular}
```

.....

Here Hi!	is our	table
a Hi!	b	c

An author might use the following technique to get a decimal point inserted. Here is the previous decimal table with the decimal points added in the preamble:

```
\begin{tabular}{r@{.}lc}
2130&456\\
30&33\\
1&4568\\
\end{tabular}
```

.....

2130.456
30.33
1.4568

## Quick Summary of All Preamble Commands

- `r|c` Letters that show how contents of column should be aligned, left, right, or center.
- `p{<dimen>}` This column should wrap at the width specified. For example, `p[1in]` will make this column wrap at the width of one inch.
  - | Will make vertical line appear in table between the columns where | was used.
- `@{}` Allow user to add something to every line in the table. In addition, it will get rid of space that normally appears between columns, useful if you are lining up decimal points.

## Horizontal lines

Here we add horizontal lines in the table:

```
\begin{tabular}{|l||cr|}  
\hline  
Here& is our&table\\  
\hline  
\hline  
a&b&c\\  
\hline  
\end{tabular}
```

---

Here	is our	table
a	b	c

If we want horizontal lines to extend under some columns but not others we use `\cline{<column numbers>}`

```
\begin{tabular}{|lcr|}  
\hline  
Here& is our&table\\  
\cline{1-2}  
a&b&c\\  
\cline{2-3}  
\end{tabular}
```

---

Here	is our	table
a	b	c

A line may be used under a single column. In that case we use the column number twice, as in `\cline{2-2}`, since `cline` needs two numbers with a dash between them to work correctly.

```
\begin{tabular}{lcr}
\hline
Here& is our&table\\
\cline{1-2}
a&b&c\\
\cline{2-2}
\end{tabular}
```

.....

Here	is our	table
a	<u>b</u>	c

## Spanning Columns

When we want to span more than one column with text, we can use the `multicolumn` command:

```
\multicolumn{<number of columns>}{<position text>}{<text>}
```

```
\begin{tabular}{ccc}
\multicolumn{2}{c}{Hello}&table entry\\
One&Two&Three\\
&\multicolumn{2c}{Hi!}\\
\end{tabular}
```

.....

	Hello	table entry
One	Two	Three
	Hi!	

We can also add vertical lines to the multicolumn command in the second argument

```
\multicolumn{<number of columns>|}{<position text>}{<text>}
```

This may be necessary if you have vertical lines in the table preamble, since the `\multicolumn` command ignores table preamble instructions. Here's an example where `\multicolumn` gets rid of the vertical rules that found in the rest of the table:

```
\begin{tabular}{||c|c|c|}
\multicolumn{2}{c}{Hello}&table entry\\
One&Two&Three\\
More Things&Here&You See\\
\end{tabular}
```

.....

	Hello		table entry	
	One		Two	
	More Things		Here	
			Three	
			You See	

Here's how to fix that problem. Add vertical lines into the `\multicolumn` argument:

```
\begin{tabular}{||c|c|c|}
\multicolumn{2}{||c|}{Hello}&table entry\\
One&Two&Three\\
More Things&Here&You See\\
\end{tabular}
```

.....

	Hello		table entry	
	One		Two	
	More Things		Here	
			Three	
			You See	



Since `\multicolumn` ignores the table preamble instructions, sometimes we use this when we don't want to use something in the preamble. We can even use `\multicolumn{1}{}`, only spanning 1 column, in order to get around something in the preamble. Notice in this example how the double vertical lines go away in the bottom line because of using `\multicolumn`.

(Note: `\multicolumnlc` is the same as `\multicolumn{1}{c}`  
Curly brackets are not needed for grouping, if there is only a single letter or number following the command.)

```
\begin{tabular}{|r||l|c|}
\multicolumn{1}{r}{Text in Multicolumn}&\multicolumnlc{e}
&\multicolumnlc{f}\\
Here&is our&table\\
a&b&c\\
\end{tabular}
```

.....

Text in Multicolumn	e	f
Here	is our	table
a	b	c

We can use `\multicolumn` to make a column header which spans both parts of the decimal number:

```
\begin{tabular}{r@{.}l}
\multicolumn2c{\bf Decimal Numbers}\\
2130&456\\
30&33\\
1&4568\\
\end{tabular}
```

.....

**Decimal Numbers**  
2130.456  
30.33  
1.4568

---

## You Try It!

Please try to make this table:

A Sample Table		hello	341.143
and now	we use cline	hello	78.99
Finally, Multicolumn	This is in multicolumn		

The table includes:

- Vertical lines
- `@{ hello }` in the preamble.
- Making decimal numbers in two adjoining columns; use `@{}` between the two columns so that the two columns don't have a space between them.
- Horizontal lines at the top and bottom of table
- Use `\cline{1-1}` and `\cline{3-4}`.
- Use the `\multicolumn{}{}{}` command.

You might want to type `\vskip .5in` between tables, if you are doing more than one, to see each table more easily.

## *Congratulations!*

That wasn't easy!

*(Code for sample table, but try not to peek)*

```
\begin{tabular}{|l|c|@{ hello }|r@{}}|l|
\hline
A Sample Table&341.&143\\
\cline{1-1}
and now&we use cline&78.&99\\
\cline{3-4}
Finally, Multicolumn&\multicolumn{3}{c}{This is in multicolumn}\\
\hline
\end{tabular}
```

## Adding extra space between lines

If we want extra space between lines we can write `\\[<dimension>]`, which will add space beneath that line:

```
\begin{tabular}{|l|l|l|cr|}  
\hline  
Here& is our&table\\[24pt]  
\hline  
\hline  
a&b&c\\  
\hline  
\end{tabular}
```

---

Here	is our	table
a	b	c

## Using `vrule` in the table

But, we may want to have more space above the table line as well as below it.

In that case, we can set a `vrule` to a specific height and depth, remembering that the height starts from the baseline of the letters and goes up, while the depth starts at the baseline and goes down.

`\vrule` is a Plain  $\TeX$  command. It expects to be followed by one or more of these **keywords** entered in any order: `height`, `width`, and `depth`. Notice that **keywords** do not have a backslash before them.)

```
\begin{tabular}{|l|l|l|cr|}  
\hline  
Here\vrule height 16pt depth 6pt width0pt& is our&table\\  
\hline  
\hline  
a&b&c\\  
\hline  
\end{tabular}
```

---

Here	is our	table
a	b	c

We can add more space between every line by using the `vrule` in the preamble:

```
\begin{tabular}{@{\vrule height 12pt depth4pt width0pt}lrc}
Here&is our&table\\
a&b&c\\
More&Things&Here
\end{tabular}
```

.....

Here	is our	table
a	b	c
More	Things	Here

Another way to add more vertical space to all the tables in a document:

`\arraystretch`.

We can redefine this term.

The default value is `\def\arraystretch{1.0}`. If we want to add 50% more vertical space between lines in the tables, we could redefine it to be

`\def\arraystretch{1.5}`:

```
\begin{tabular}{lrc}\hline
Here&is our&table\\
a&b&c\\
More&Things&Here\\\hline
\end{tabular}
```

```
\def\arraystretch{1.5}
```

```
\begin{tabular}{lrc}\hline
Here&is our&table\\
a&b&c\\
More&Things&Here\\\hline
\end{tabular}
```

```
\botline
```

Here	is our	table
a	b	c
More	Things	Here

Here	is our	table
a	b	c
More	Things	Here

## Using *noalign* between table lines

The `\noalign` command will allow us to temporarily stop the table, enter a space command or even text, and then continue with the table. The skip can either be used to add space or take away space between lines of the table.

```
\begin{tabular}{|l|l|cr|}  
\hline  
Here& is our




```

.....

Here	is our	table
a	b	c

We can even put some text in here.

def	hij	klm
-----	-----	-----

## How to trick the preamble in particular entries

There are times when just one symbol or letter should stick out from the column. We can use `\rlap{}` to make something stick out to the right, and `\llap{}` to get something to stick out to the left.

Compare these two table results to see that `\rlap` and `\llap` are useful:

Not using `\llap` and `\rlap`:

```
\begin{tabular}{lcr}
Here&\it is our&table\\
one&two&three\\
one&two&three*\\
1one&two&three\\
\end{tabular}
```

Using `\llap` and `\rlap`:

```
\begin{tabular}{lcr}
Here&\it is our&table\\
one&two&three\\
one&two&three\rlap{*}\\
\llap{1one}&two&three\\
\end{tabular}
```

.....

Not using `\llap` and `\rlap`:

Here	<i>is our</i>	table
one	two	three
one	two	three*
<sup>1</sup> one	two	three

Using `\llap` and `\rlap`:

Here	<i>is our</i>	table
one	two	three
one	two	three*
<sup>1</sup> one	two	three

## Making table spread out to particular width

If we want the table to spread out to any particular width we use `\begin{tabular*}{<width>}{@{\extracolsep\fill}<rest of preamble>}` and remember to end with `\end{tabular*}`:

```
\begin{tabular*}{3in}{@{\extracolsep\fill}|lcr|}  
\hline  
\bf Here&\it is our&table\  
\hline  
\hline  
a&b&c\  
\hline  
\end{tabular*}
```

---

Here	<i>is our</i>	table
a	b	c

To fit the width of the page, we can use `\textwidth` as the dimension:

```
\begin{tabular*}{\textwidth}{@{\extracolsep\fill}|lcr|}  
\hline  
\bf Here&\it is our&table\  
\hline  
\hline  
a&b&c\  
\hline  
\end{tabular*}
```

---

Here	<i>is our</i>	table
a	b	c

## A table inside another table

Finally, we put one table inside another and see that it will work:

```
\begin{tabular}{|l|cr|}  
\hline  
Here& is our table&  
\begin{tabular}{|l|cr|}  
\hline  
Here& is our&table\\  
\hline  
\hline  
a&b&c\\  
\hline  
\end{tabular}\\\\  
\hline  
\hline  
a&b&c\\  
\hline  
\end{tabular}
```

---

Here	is our table	Here	is our table
a	b	c	
a	b	c	

This is not entirely a toy example- Often knowing that you can put a table within another table can be helpful when making complex column headers.

---

### You Try It!

Now you do it- Try making a little table and try each of the features demo'ed above—have fun!



## Floating a table, and giving it a caption

To give the table a caption, and to make it float to the top or bottom of a page, we have the `\begin{table}` environment. Here is an example. The `\centering` command makes the table center horizontally on the page. Short captions will also center, but if the caption is longer it will form a paragraph.

Notice the square bracket argument used: `\begin{table}[ht]` This argument determines what part of the page will be used for the float. We have these choices:

```
[h] for here  
[t] for top of the page  
[b] for bottom of the page  
[p] for put on its own page
```

You can use more than one choice. In case the first one doesn't work, the second one will be used.

```
\begin{table}[ht]  
\caption{Table caption here.}  
\centering  
\begin{tabular}{|l|l|cr|}  
\hline  
Here& is our&table\\  
\hline  
\hline  
a&b&c\\  
\hline  
\end{tabular}  
\end{table}
```

.....

Table 1.1: Table caption here.

Here	is our	table
a	b	c

Here is a real life table submitted by an author to a journal. He starts with a small macro `\poff` to make it easier to get ‘off’ in roman rather than math italics fonts.

Things to notice are:

The optional argument in square brackets that specifies where the table is to appear on the page; Use of `\rlap{}` to hide the width of the text between `{}` from the table making mechanism; The `\multicolumn` at the end used for making a table note; The caption; The `\label{}` and `\ref{}`.

```
\def\poff{P\hbox{off}}

\begin{table}[h]
\caption{Values of  $\poff_{ij}$  for the model found by MEME in pass 1
on the CRP/LexA dataset at the positions of the known LexA sites.}
\label{tab:pass1}

\centering
\begin{tabular}{lcccccc}\hline
sample & site 1 &  $\poff_{ij}$  & & site 2 & \\
&  $\poff_{ij}$  &  $\rlap{a}$  & &  $\poff_{ij}$  & \\
cloacin DF13 & 97a & 0.998684 & & & \\
colicin Ib & 99a & 0.990472 & & & \\
recN & 71 & 0.999988 & 93 & 0.865704 & 111a \\
& & & & & 0.134281 \\
\hline
\end{tabular}
\multicolumn{7}{l}{aIndicates site known only by sequence similarity to known sites.}
\end{table}
Here is the table reference--\ref{tab:pass1}.
.....
```

Table 1.2: Values of  $Poff_{ij}$  for the model found by MEME in pass 1 on the CRP/LexA dataset at the positions of the known LexA sites.

sample	site 1	$Poff_{ij}$	site 2	$Poff_{ij}$	site 3	$Poff_{ij}$
cloacin DF13	97 <sup>a</sup>	0.998684				
colicin Ib	99 <sup>a</sup>	0.990472				
recN	71	0.999988	93	0.865704	111 <sup>a</sup>	0.134281

<sup>a</sup>Indicates site known only by sequence similarity to known sites.

Here is the table reference–1.2.

## Figure Floats

Figure floats and captions work just like the table floats and captions.

Here is a sample:

Here is a figure reference, `\ref{figsamp}`.

```
\begin{figure}[ht]
\vskip.25in
\caption{Here is the figure caption.}
\label{figsamp}
\end{figure}
```

Here is a figure reference, 1.1.

Figure 1.1: Here is the figure caption.

---

---

## You Try It!

See if you can make the following tables. (The code is on the following page, but try not to look unless you need to.)

Table 1.3: Small Table

one	two	three
Very	Small	Table Here

Table 1.4: Interval Solutions to the Interval Equation System 7.3 Obtained in Three Ways

	Method 1	Method 2	Method 3
$x_1$	[1.2325, 1.2607]	[1.2328, 1.2604]	[1.2415~]
$x_2$	[0.5609, 0.5752] <sup>a</sup>	[0.5610, 0.5750]	[0.5640~] <sup>b</sup>

<sup>a</sup>Refs. 19 and 20.

<sup>b</sup> $\kappa, \lambda > 1$ .

## How to do it...

```
\begin{table}[ht]
\centering
\caption{Small Table}
\vskip12pt
\begin{tabular}{@{\vrule height 12pt depth 6pt width0pt}|ccc|}
\hline
one&two&three\\
\hline
Very&Small&Table Here\\
\hline
\end{tabular}
\end{table}

\begin{table}[ht]
\caption{Interval Solutions to the Interval Equation System
7.3 Obtained in Three Ways}
\vskip3pt
\begin{tabular*}{\textwidth}{@{\extracolsep{\fill}}l@{\vrule
height 12pt depth 5pt width0pt}ccc}
\hline
&{\bf Method 1\hfill}&{\bf Method 2\hfill}&{\bf Method 3\hfill}\\
\hline
 $x_1$ &[1.2325, 1.2607]&[1.2328, 1.2604]&[1.2415\sim{]}\\
 $x_2$ &[0.5609, 0.5752]\r1ap{\^a}&[0.5610, 0.5750]
&[0.5640\sim{]}\r1ap{\^b}\\
\hline
\end{tabular*}
\vskip3pt
 $\^a$ Refs.~19 and 20.

 $\^b$ \kappa, \lambda>1$.
\end{table}
```

## Table MakeOver

We start with a crowded table:

```
\begin{tabular}{|c|c|c|} \hline
Index &  $J\Delta$  Representation &  $J\Phi$  Representation & \ \ \hline \hline
1 &  $d=1, f=1, g=0, h=1$  &  $d=1, h=1$  & \ \ \hline
2 &  $d=1, e=1, f=1, g=0, h=1, i=1$  &  $d=1, e=1$  & \ \ \hline
3 &  $b=0, d=1, e=1, f=1, g=0, h=1, i=1$  &  $b=0$  & \ \ \hline
\end{tabular}
```

Index	$J\Delta$ Representation	$J\Phi$ Representation
1	$d = 1, f = 1, g = 0, h = 1$	$d = 1, h = 1$
2	$d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$d = 1, e = 1$
3	$b = 0, d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$b = 0$

Add extra horizontal space with `tabular*{<size>}`

```
\begin{tabular*}{\textwidth}{@{\extracolsep\fill}|c|c|c|}\hline
Index &  $J\Delta$  Representation &  $J\Phi$  Representation & \ \ \hline \hline
1 &  $d=1, f=1, g=0, h=1$  &  $d=1, h=1$  & \ \ \hline
2 &  $d=1, e=1, f=1, g=0, h=1, i=1$  &  $d=1, e=1$  & \ \ \hline
3 &  $b=0, d=1, e=1, f=1, g=0, h=1, i=1$  &  $b=0$  & \ \ \hline
\end{tabular*}
```

Index	$J\Delta$ Representation	$J\Phi$ Representation
1	$d = 1, f = 1, g = 0, h = 1$	$d = 1, h = 1$
2	$d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$d = 1, e = 1$
3	$b = 0, d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$b = 0$

Now, we take out those ugly vertical lines and add more vertical space. Notice how we got rid of the extra space on the right side of the table by adding `@{}` to the end of the preamble?

```
\begin{tabular*}{\textwidth}{@{\vrule height 12pt depth 5pt width0pt
\extracolsep\fill}ccc@{}}\hline
\bf Index & \bf  $J\Delta$  Representation & & \\
& \bf  $J\Phi$  Representation & \ \ \hline \hline
1 &  $d=1, f=1, g=0, h=1$  &  $d=1, h=1$  & \ \ \\
2 &  $d=1, e=1, f=1, g=0, h=1, i=1$  &  $d=1, e=1$  & \ \ \\
3 &  $b=0, d=1, e=1, f=1, g=0, h=1, i=1$  &  $b=0$  & \ \ \hline
\end{tabular*}
```

Index	$J\Delta$ Representation	$J\Phi$ Representation
1	$d = 1, f = 1, g = 0, h = 1$	$d = 1, h = 1$
2	$d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$d = 1, e = 1$
3	$b = 0, d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$b = 0$

*Second table to improve:*

```
\begin{tabular}{rrrrrrrr}
\multicolumn{3}{l}{\bf Parameters}&
\multicolumn{5}{c}{\bf Averaged Results}\cr
\hline
$n$&$S^*_{MAX}$&$t_1$&$r_1$&$m_1$&$t_2$&$r_2$&$m_2$\\cr
\hline
10&1\quad &4&235.0007&4&4&.0020&4\\cr
10&5\quad &50&.0008&8&50&.0020* &12\\cr
\hline
\end{tabular}
```

---

Parameters			Averaged Results				
$n$	$S^*_{MAX}$	$t_1$	$r_1$	$m_1$	$t_2$	$r_2$	$m_2$
10	1	4	235.0007	4	4	.0020	4
10	5	50	.0008	8	50	.0020*	12

The problem with the table above is that the column headers are not centered above the columns. The asterisk also mispositions one column entry. Let's add extra vertical space to only the column header line, and a different amount of space for the other lines in the table.

```
\begin{tabular*}{\hsize}{@{\extracolsep{\fill}}ccrrcc@{\vrule height 11pt
depth 4pt width0pt}}
\multicolumn{3}{l}{\bf Parameters}&
\multicolumn{5}{c}{\bf Averaged Results}\\\noalign{\vskip4pt}\hline
\vrule height 11pt depth 6pt width0pt
$n$&$S^*_{MAX}$&$t_1$&\multicolumn{1c}{r_1}&$m_1$&$t_2$&
r_2&$m_2$\\hline
10&1\quad &4&235.0007&4&4&.0020&4\\
10&5\quad &50&.0008&8&50&.0020\rlap{*}&12\\hline
\end{tabular*}
```

---

Parameters			Averaged Results				
$n$	$S^*_{MAX}$	$t_1$	$r_1$	$m_1$	$t_2$	$r_2$	$m_2$
10	1	4	235.0007	4	4	.0020	4
10	5	50	.0008	8	50	.0020*	12

---

## Illustrations

Encapsulated PostScript files, \*.eps files, may be included. This is done with the added use of the file graphics.sty or graphicx.sty. This package allows the user to specify the driver program being used, so that the same .sty file will work with different driver programs.

```
\documentclass{book}
\usepackage[dvips]{graphicx}
\begin{document}
\begin{figure}[h]
\centering
\includegraphics[height=1in]{1-fig2.eps}
\caption{Vector addition  $v+w$  produces the diagonal
of a parallelogram.}
\end{figure}
```

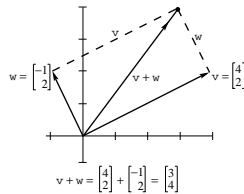


Figure 1.2: Vector addition  $v + w$  produces the diagonal of a parallelogram.

### Inserting a .tif file

Some authors prefer to use a .tif file, thinking that the color reproduces better. If you must use a .tif file, you must supply **both the height and width of the graphic** for it to print. For example,

```
\centerline{\includegraphics[width=3.4in, height=3in]{tifsamp.tif}}
```

### When using PDFLaTeX, No EPS!

PDFLaTeX is a very convenient program that will turn your .tex file directly into PDF. The only difficulty is that it requires .pdf, or .png, or .jpg files for illustrations, but will not accept .eps files.

So if you have .eps files that you want to use for illustrations, you must change them into pdf by running the Distiller program or another program on them.

And, on the other hand, if you are using the

```
dvips ==> ps ==> pdf
```

method of producing PDF, remember to use .eps or .tif files, **not PDF** files for illustrations.

## More on Compiling with *pdflatex*

You cannot insert EPS files. You can insert:

**JPG** widely used on Internet, digital cameras, etc. They are the best choice if you want to insert photos

**PNG** a very common format (even if not as much as JPG); it's a lossless format and it's the best choice for diagrams (if you were not able to generate a vector version) and screenshots

**PDF** is widely used for documents but can be used to store images as well. It supports both vector and bit-map images, but it's not recommended for the latter, as JPG or PNG will provide the same result using less disk space.

JPG and PNG are supported by any image processing program, so you just have to use the one you prefer. If you want to create high quality vector PDF to embed within your PDF document, you can use Inkscape: it supports many vector formats and so you can use it to convert from one to other. You could also create your graphics directly with Inkscape. If you want to make mathematical plots, then Gnuplot can save in any format.

Note, that EPS files can not be used with *pdflatex*, however they can be converted to PDF using the *epstopdf* utility, included in most LaTeX distributions. This can be called automatically by LaTeX using the *epstopdf* package. In Windows, multiple files can be converted by placing the following line in a batch file (a text file with a .BAT extension) in the same directory as the images:

```
for %%f in (*.eps) do epstopdf %%f
```

which can then be run from the command line. If *epstopdf* produces whole page with your small graphics somewhere on it, use

```
$ epstopdf -gsopt=-dEPSCrop foo.eps
```

or try using *ps2pdf* utility to crop final PDF.

```
$ ps2pdf -dEPSCrop foo.eps
```

Images can be saved in multiple formats for different purposes. For example, a directory can have "diagram.pdf" for high-resolution printing, while "diagram.png" can be used for previewing on the monitor. You can specify which image file is to be used by *pdflatex* through the preamble command:

```
\DeclareGraphicsExtensions{.pdf, .png, .jpg}
```

which specifies the files to include in the document, if files with the same basename exist, but with different extensions.



## ***Getting Figures and Tables to appear where you want them***

Usually illustrations and tables will print near the place where they are found in text, floating to the top, bottom, or on to their own page.

Sometimes, a large illustration will push all the following ones to the end of the article. Occasionally tables will accumulate as well, and not print until the end.

The `\clearpage` command will kick out the remaining floats and will clear out that part of LaTeX's memory. This command works similarly to `\newpage` in that it will fill the bottom of the existing page with white space, starting from the place where `\clearpage` was written.

If you want to make sure that there is no extra space at the bottom of the page where you typed `\clearpage`, you can run LaTeX once without the `\clearpage` command and see where the page would naturally end. At this point, if it is in the middle of a paragraph you can type `\break` to end the page mid-paragraph, then `\clearpage`, then before the text that follows, so it is clear that it is a continuation of the paragraph.

```
text in paragraph\break
\clearpage
\noindent
continuing text...
```

If you only wanted one illustration to appear at the top of the page, you could follow a similar path and then specifically call for the illustration at the top of the new page:

```
text in paragraph\break
\newpage
\noindent
\begin{figure}[h]
...\caption{}
\end{figure}
continuing text...
```

## ***Aggregating a number of illustrations or tables***

It may not be immediately obvious that you can have a number of illustrations and captions in one figure environment or a number of tables and captions in one table environment, but this will work.

This knowledge can be helpful to keep multiple illustrations or tables from drifting away from the first place they are mentioned in the text. You could use the [t] option to make the large illustration float to the top of the page, or [p] if you need even more space so that the illustrations will be on their own page.

```
\begin{figure}[p]
illustration
\caption{}
illustration
\caption{}
illustration
\caption{}
illustration
\caption{}
\end{figure}
```

A similar technique can be used for multiple tables.

---

## Rotating Tables and Figures

If you have included `\usepackage{graphicx}`, you will have the `\rotatebox` command available. Start your `\begin{table}[p]` or `\begin{figure}[p]` environment first, then rotate the contents. Usually you will want these rotated figures or tables to be placed on their own page, so you would use the `[p]` option, for 'page'. Use `\rotatebox{(angle)}{\vbox{ table or figure }}`; then end with `\end{table}` or `\end{figure}`.

```
\begin{table}[p]
  \rotatebox{-90}{\vbox{
\caption{This is the table caption.}
\begin{tabular}{@{\vrule height 10pt depth 2pt width0pt}ccrrcrrc@{}}
\multicolumn{3}{l}{\bf Parameters}&&...
\end{tabular}
}}
\end{table}
```

.....

Table 1.5: This is the table caption.

Parameters		Averaged Results					
$n$	$S_{MAX}^*$	$t_1$	$r_1$	$m_1$	$t_2$	$r_2$	$m_2$
10	1	4	235.0007	4	4	.0020	4
10	5	50	.0008	8	50	.0020*	12

If you need to position the table or figure up or down on the page, you can add in `\vskip <num>` points above the `\rotatebox` command. A positive dimension will move the illustration down on the page; a negative dimension, ie. `\vskip-1in` will move it up on the page.

To move the table to the left or right on the page, add a `\hskip` before the rotated table environment, as seen below. A positive dimension will move the table to the right; a negative dimension to the left.

```
\begin{table}[p]
\vskip-1in %% <== this will move table up or down, since it is
           %%      used before the box is rotated
%%
\hskip3in %% <== this will move the table left or right depending
           %%      on whether the dimension is positive or negative
\rotatebox{-90}{\vbox{\caption{This is the table caption.}
\begin{tabular}{@{\vrule height 10pt depth 2pt width 0pt}ccrrccrcc@{}}
\multicolumn{3}{l}{\bf Parameters}&&...
\end{tabular}
}}
\end{table}
```

Table 1.6: This is the table caption.

Parameters		Averaged Results					
$n$	$S_{MAX}^*$	$t_1$	$r_1$	$m_1$	$t_2$	$r_2$	$m_2$
10	1	4	235.0007	4	4	.0020	4
10	5	50	.0008	8	50	.0020*	12

---

## Using Color

In the following section we will see how to add color tables, but before that, let's see the related topic, how to use color anywhere in your .tex document.

The complete documentation is found in grfguide.pdf, which you can find with Google.

You must use

```
\usepackage{color}
```

To get the 68 standard colors, and if you are using dvips or pdflatex, you should add the following options:

```
\usepackage[usenames,dvipsnames]{color}
```

(See <http://en.wikibooks.org/wiki/LaTeX/Colors>)

Like `\usepackage{graphicx}`, you can tune the color macros to your particular LaTeX implementation by specifying the name of your driver program. Choose from this list:

```
dvips, xdvi, dviptf, dviptfm, dviptfmx, pdftex, dvipsone,  
dviwindo, emtex, dviwin, pctexps, pctexwin, pctexhp, pctex32,  
truetex, tcidvi, vtex, oztex, textures, xetex
```

and use the term within square brackets:

```
\usepackage[dvips]{color}
```

## Set your own colors

Should you wish to mix your own colors, use the `\definecolor` command.

You can define colors using one of three color models;

`rgb` (red green blue, used for on screen colors)

`cmyk` (cyan, magenta, yellow, black, for printed colors)

`gray` (percent of black and white).

Usually you will use `cmyk`.

The colors are specified in terms of percent of 1. For example, `.5` would mean half color and half white, `.9` would mean 90% of full color.

```
\definecolor{lightblue}{cmyk}{.5,0,0,0}
```

```
\definecolor{puregreen}{rgb}{0,1,0}
```

Used:\\

```
\color{lightblue} Here is light blue in cmyk.\\
```

```
\color{puregreen} Here is green in rgb.
```

.....

Used:

Here is light blue in cmyk.

Here is green in rgb.

You can find a color chart with the `cmyk` values for many colors here:

<http://www.color-chart.org/cmyk-color-chart.php>

## Ready mixed colors

The colors black, white, red, green, blue, cyan, magenta, yellow are predefined.

In addition, if you use

```
\usepackage[usenames,dvipsnames]{color}
```

You will have access to all the colors listed below. The definition is found to your right, which you can copy and vary if you want to make a new color based on one of these:

---

<b>GreenYellow</b>	<code>\definecolor{GreenYellow}{cmyk}{0.15,0,0.69,0}</code>
<b>Yellow</b>	<code>\definecolor{Yellow}{cmyk}{0,0,1,0}</code>
<b>Goldenrod</b>	<code>\definecolor{Goldenrod}{cmyk}{0,0.10,0.84,0}</code>
<b>Dandelion</b>	<code>\definecolor{Dandelion}{cmyk}{0,0.29,0.84,0}</code>
<b>Apricot</b>	<code>\definecolor{Apricot}{cmyk}{0,0.32,0.52,0}</code>
<b>Peach</b>	<code>\definecolor{Peach}{cmyk}{0,0.50,0.70,0}</code>
<b>Melon</b>	<code>\definecolor{Melon}{cmyk}{0,0.46,0.50,0}</code>
<b>YellowOrange</b>	<code>\definecolor{YellowOrange}{cmyk}{0,0.42,1,0}</code>
<b>Orange</b>	<code>\definecolor{Orange}{cmyk}{0,0.61,0.87,0}</code>
<b>BurntOrange</b>	<code>\definecolor{BurntOrange}{cmyk}{0,0.51,1,0}</code>
<b>Bittersweet</b>	<code>\definecolor{Bittersweet}{cmyk}{0,0.75,1,0.24}</code>
<b>RedOrange</b>	<code>\definecolor{RedOrange}{cmyk}{0,0.77,0.87,0}</code>
<b>Mahogany</b>	<code>\definecolor{Mahogany}{cmyk}{0,0.85,0.87,0.35}</code>
<b>Maroon</b>	<code>\definecolor{Maroon}{cmyk}{0,0.87,0.68,0.32}</code>
<b>BrickRed</b>	<code>\definecolor{BrickRed}{cmyk}{0,0.89,0.94,0.28}</code>
<b>Red</b>	<code>\definecolor{Red}{cmyk}{0,1,1,0}</code>
<b>OrangeRed</b>	<code>\definecolor{OrangeRed}{cmyk}{0,1,0.50,0}</code>
<b>RubineRed</b>	<code>\definecolor{RubineRed}{cmyk}{0,1,0.13,0}</code>
<b>WildStrawberry</b>	<code>\definecolor{WildStrawberry}{cmyk}{0,0.96,0.39,0}</code>
<b>Salmon</b>	<code>\definecolor{Salmon}{cmyk}{0,0.53,0.38,0}</code>
<b>CarnationPink</b>	<code>\definecolor{CarnationPink}{cmyk}{0,0.63,0,0}</code>
<b>Magenta</b>	<code>\definecolor{Magenta}{cmyk}{0,1,0,0}</code>
<b>VioletRed</b>	<code>\definecolor{VioletRed}{cmyk}{0,0.81,0,0}</code>
<b>Rhodamine</b>	<code>\definecolor{Rhodamine}{cmyk}{0,0.82,0,0}</code>
<b>Mulberry</b>	<code>\definecolor{Mulberry}{cmyk}{0.34,0.90,0,0.02}</code>
<b>RedViolet</b>	<code>\definecolor{RedViolet}{cmyk}{0.07,0.90,0,0.34}</code>
<b>Fuchsia</b>	<code>\definecolor{Fuchsia}{cmyk}{0.47,0.91,0,0.08}</code>
<b>Lavender</b>	<code>\definecolor{Lavender}{cmyk}{0,0.48,0,0}</code>
<b>Thistle</b>	<code>\definecolor{Thistle}{cmyk}{0.12,0.59,0,0}</code>
<b>Orchid</b>	<code>\definecolor{Orchid}{cmyk}{0.32,0.64,0,0}</code>
<b>DarkOrchid</b>	<code>\definecolor{DarkOrchid}{cmyk}{0.40,0.80,0.20,0}</code>
<b>Purple</b>	<code>\definecolor{Purple}{cmyk}{0.45,0.86,0,0}</code>
<b>Plum</b>	<code>\definecolor{Plum}{cmyk}{0.50,1,0,0}</code>
<b>Violet</b>	<code>\definecolor{Violet}{cmyk}{0.79,0.88,0,0}</code>
<b>RoyalPurple</b>	<code>\definecolor{RoyalPurple}{cmyk}{0.75,0.90,0,0}</code>
<b>BlueViolet</b>	<code>\definecolor{BlueViolet}{cmyk}{0.86,0.91,0,0.04}</code>
<b>Periwinkle</b>	<code>\definecolor{Periwinkle}{cmyk}{0.57,0.55,0,0}</code>
<b>CadetBlue</b>	<code>\definecolor{CadetBlue}{cmyk}{0.62,0.57,0.23,0}</code>
<b>CornflowerBlue</b>	<code>\definecolor{CornflowerBlue}{cmyk}{0.65,0.13,0,0}</code>
<b>MidnightBlue</b>	<code>\definecolor{MidnightBlue}{cmyk}{0.98,0.13,0,0.43}</code>
<b>NavyBlue</b>	<code>\definecolor{NavyBlue}{cmyk}{0.94,0.54,0,0}</code>
<b>RoyalBlue</b>	<code>\definecolor{RoyalBlue}{cmyk}{1,0.50,0,0}</code>
<b>Blue</b>	<code>\definecolor{Blue}{cmyk}{1,1,0,0}</code>
<b>Cerulean</b>	<code>\definecolor{Cerulean}{cmyk}{0.94,0.11,0,0}</code>

<b>Cyan</b>	<code>\definecolor{Cyan}{cmyk}{1,0,0,0}</code>
<b>ProcessBlue</b>	<code>\definecolor{ProcessBlue}{cmyk}{0.96,0,0,0}</code>
<b>SkyBlue</b>	<code>\definecolor{SkyBlue}{cmyk}{0.62,0,0.12,0}</code>
<b>Turquoise</b>	<code>\definecolor{Turquoise}{cmyk}{0.85,0,0.20,0}</code>
<b>TealBlue</b>	<code>\definecolor{TealBlue}{cmyk}{0.86,0,0.34,0.02}</code>
<b>Aquamarine</b>	<code>\definecolor{Aquamarine}{cmyk}{0.82,0,0.30,0}</code>
<b>BlueGreen</b>	<code>\definecolor{BlueGreen}{cmyk}{0.85,0,0.33,0}</code>
<b>Emerald</b>	<code>\definecolor{Emerald}{cmyk}{1,0,0.50,0}</code>
<b>JungleGreen</b>	<code>\definecolor{JungleGreen}{cmyk}{0.99,0,0.52,0}</code>
<b>SeaGreen</b>	<code>\definecolor{SeaGreen}{cmyk}{0.69,0,0.50,0}</code>
<b>Green</b>	<code>\definecolor{Green}{cmyk}{1,0,1,0}</code>
<b>ForestGreen</b>	<code>\definecolor{ForestGreen}{cmyk}{0.91,0,0.88,0.12}</code>
<b>PineGreen</b>	<code>\definecolor{PineGreen}{cmyk}{0.92,0,0.59,0.25}</code>
<b>LimeGreen</b>	<code>\definecolor{LimeGreen}{cmyk}{0.50,0,1,0}</code>
<b>YellowGreen</b>	<code>\definecolor{YellowGreen}{cmyk}{0.44,0,0.74,0}</code>
<b>SpringGreen</b>	<code>\definecolor{SpringGreen}{cmyk}{0.26,0,0.76,0}</code>
<b>OliveGreen</b>	<code>\definecolor{OliveGreen}{cmyk}{0.64,0,0.95,0.40}</code>
<b>RawSienna</b>	<code>\definecolor{RawSienna}{cmyk}{0,0.72,1,0.45}</code>
<b>Sepia</b>	<code>\definecolor{Sepia}{cmyk}{0,0.83,1,0.70}</code>
<b>Brown</b>	<code>\definecolor{Brown}{cmyk}{0,0.81,1,0.60}</code>
<b>Tan</b>	<code>\definecolor{Tan}{cmyk}{0.14,0.42,0.56,0}</code>
<b>Gray</b>	<code>\definecolor{Gray}{cmyk}{0,0,0,0.50}</code>
<b>Black</b>	<code>\definecolor{Black}{cmyk}{0,0,0,1}</code>
<b>White</b>	<code>\definecolor{White}{cmyk}{0,0,0,0}</code>

---



## Using the color

Here's a neat one: `\pagecolor{color name}` will set the current and following pages in that color. You can set it back to normal with `\pagecolor{white}`. (Case is important!)

`\color{color name}` will make text that color within grouping.

---

```
{\color{red} Here is some text.} Back to normal.  
Here is some text. Back to normal.
```

---

`\textcolor{color name}{text}` will take an extra argument for the text.

---

```
\textcolor{red}{Here is some text} Back to normal.  
Here is some text Back to normal.
```

---

`\colorbox{color name}{text}` will set the background color.

---

```
\colorbox{red}{\color{white}  
here is some text.}
```

**here is some text.**

---

`\fcolorbox{border color}{background color}{text here}` will add a frame in the first color:

---

```
\fcolorbox{red}{yellow}{\color{blue}  
here is some text.}
```

**here is some text.**

---

---

## You Try It!

Define a number of cmyk colors, and use them with the commands listed above. Experiment and enjoy!

---

## Adding Color Screens to Table

To get colored columns, rows, or cells in your table, you must start with these packages. If you don't already have them on your system, you can download them from CTAN.

The documentation for color table commands, `colortabl.pdf`, is available from CTAN, <http://www.ctan.org>

```
\usepackage[dvips]{graphicx}
\usepackage[dvips]{color}
\usepackage{array}
\usepackage{colortbl}
```

### Column Colors

Enter the color for the column background in the table preamble, using the `>{ }` that is defined with the `array` package. `>{ }` is used before the column letter (rlc).

The `\columncolor` command may be followed with a color model in square brackets, with an argument of the percent of that color to be used:

```
\columncolor[(color type)]{(percent of that color: how dark?)}
```

or a color name in curly brackets:

```
\columncolor{color name}
```

```
\begin{tabular}{>{\columncolor[gray]{.8}\color{blue}}c
>{\columncolor[gray]{.2}\color{white}}c
>{\columncolor[gray]{.5}\color{white}}c
>{\columncolor{red}\color{white}}c}
one&two&three&red\\
four&five&six& more red
\end{tabular}
```

.....

one	two	three	red
four	five	six	more red

## Row Colors

```
\begin{tabular}{lc}  
  \rowcolor[gray]{.9}  
  one&two\\  
  \rowcolor[gray]{.5}  
  three&four  
 \end{tabular}
```

---

one	two
three	four

## What happens if you use both?

Seeing what happens if we use both row and column coloring– the most recent declaration is used:

```
\begin{tabular}{>{\columncolor{blue}\color{blue}}lc}  
  \rowcolor[gray]{.8}  
  one&two\\  
  \rowcolor[gray]{.5}\color{white}  
  three&four  
 \end{tabular}
```

---

one	two
three	four

## Cell Colors

The command `\cellcolor{(color)}` can be used at any point:

```
\begin{tabular}{|c|}
one&\cellcolor{blue}\color{white} two\\
\multicolumn{2}{\cellcolor{red}\color{white} three and four}
\end{tabular}
```

.....

one	two
three and four	

Or, used to override another column or row color:

```
\begin{tabular}{|c|}
\rowcolor[gray]{.8}
one&\cellcolor{blue}\color{white} two\\
\rowcolor[gray]{.5}
three&four
\end{tabular}
```

.....

one	two
three	four

Now let's see how we can use these tools for some sophisticated tables:

**Table 1.7: Growth assumptions for the euro area (in percent)**

	2007	3008	2009	2010
European Commission forecast 10/2008	2.7	1.2	0.1	0.9
IMF forecast 11/2008	2.6	1.2	-0.5	n/a
Bruegel scenario		1.2	-0.9	0.4

Source: European Commission, IMF, Bruegel calculations.

(From *European Recovery Programme, Bruegel Policy Brief*)

---

## You Try It!

Try typing in the code so that you can see the results in color, and practice some of our table techniques at the same time.

```
\definecolor{Salmon}{cmyk}{0,0.53,0.38,0}
\definecolor{ltSalmon}{cmyk}{0,0.13,0.095,0} % .75 percent lighter than Salmon
\begin{table}[h]
\def\arraystretch{2.0}
\fcolorbox{black}{ltSalmon}{\vtop{\hspace=.925\textwidth}
\color{red}
\caption{Growth assumptions for the euro area (in percent)}
\color{black}
\begin{tabular}{lcccc}
\rowcolor{Salmon}
&2007&3008&2009&2010\\
European Commission forecast 10/2008&2.7&1.2&0.1&0.9\\
IMF forecast 11/2008&2.6&1.2&\llap{--}0.5&n/a\\
Bruegel scenario &&1.2&\llap{--}0.9&0.4\\
\multicolumn{5}{\sf}
Source: European Commission, IMF, Bruegel calculations.}\\
\end{tabular}
}}
\end{table}
```

(From \it European Recovery Programme, Bruegel Policy Brief)

This table uses `\noalign{\vskip2pt}` between rows in the table to get the white spaces that function like lines. It uses `\begin{figure}...\end{figure}` even though the contents is a table.

Code to make this table is on the facing page.

Figure 1.3: Evaluate Vendors: US Search Marketing Agencies' Information and Selection Criteria

Vendor	Enterprise-level clients	Search marketing revenues (2007)	Percent of clients using vendor for paid search and SEO	Date evaluated
360i	78%	\$34 million	38%	Nov 2008
iCrossing	73%	\$79 million	30%	Nov 2008
IMPAQT	59%	\$18 million	41%	Nov 2008
iProspect	65%	\$40 million*	40%	Nov 2008
Oneupweb	40%	\$13 million*	26%	Nov 2008
Razorfish	77%	\$31 million	25%	Nov 2008

\*The vendor cannot publicly disclose revenues. This is Forrester's estimate of the vendor's 2007 search revenue.

**Vendor selection criteria**

Revenue for the vendor's search business is greater than \$10 million.

Percent of the vendor's clients served that are enterprise level is greater than 40%.

Percent of the clients that work with the vendor for both paid search and SEO.

Source: Forrester Research, Inc.

```

\definecolor{ltblue}{cmyk}{.2,0,0,0}
\definecolor{medblue}{cmyk}{.5,0,0,0}
\font\smnormal=lsd at 10pt
\font\bf=lsb at 10pt
\begin{figure}
\vskip6pt\hrule
\caption{Evaluate Vendors: US Search Marketing Agencies' Information
and Selection Criteria}
\hrule\vskip12pt
\def\arraystretch{2}
\smnormal %% change to smaller font, defined above
\begin{tabular}{lcccc}
&&&\bf Percent of clients\\[-12pt]
&\bf Enterprise-&\bf Search marketing&\bf using vendor for&
\bf Date\\[-10pt]
\bf Vendor&\bf level clients&\bf revenues (2007)&\bf paid search and SEO
&\bf evaluated\\
\rowcolor{ltblue}i&78\%&\$34 million&38\%&Nov 2008\\
\rowcolor{ltblue}i&Crossing&73\%&\$79 million&30\%&Nov 2008\\
\rowcolor{ltblue}i&IMPAQT&59\%&\$18 million&41\%&Nov 2008\\
\rowcolor{ltblue}i&Prospect&65\%&\$40 million&40\%&Nov 2008\\
\rowcolor{ltblue}i&Oneupweb&40\%&\$13 million&26\%&Nov 2008\\
\rowcolor{ltblue}i&Razorfish&77\%&\$31 million&25\%&Nov 2008
\end{tabular}
*The vendor cannot publicly disclose revenues.
This is Forrester's estimate of the vendor's 2007 search revenue.

\vskip18pt
\centerline{\bf Vendor selection criteria}\vskip4pt
\begin{tabular}{l}
\rowcolor{medblue}
Revenue for the vendor's search business is greater than
\$10 million.\\
\rowcolor{medblue}
Percent of the vendor's clients served that are enterprise level is
greater than 40\%.\\
\rowcolor{medblue}
Percent of the clients that work with the vendor for both paid search and
SEO.\\
\hline
\end{tabular}
\vskip2pt
\hbox to\textwidth{\hfill Source: Forrester Research, Inc.}
\end{figure}

```