
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.

Syntax:

```
\begin{tabular}{lcr rrrccc}
```

↑

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}
```

which produces...

```
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}
```

which produces...

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}
```

which produces...

Here	<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}
```

which produces...

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}
```

which produces...

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

which produces...

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

which produces...

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}
```

which produces...

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}
```

which produces...

2130.456
30.33
1.4568

Horizontal lines

Here we add horizontal lines in the table:

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

which produces...

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}
```

which produces...

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}
which produces...
```

Here	is our	table
a	b	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{2}{c}{Hi!}\\
\end{tabular}
which produces...
```

	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}
```

which produces...

	Hello		table entry
	One	Two	Three
	More Things	Here	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}
```

which produces...

		Hello		table entry
		One	Two	Three
		More Things	Here	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: `\multicolumn1c` 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|||c|}
\multicolumn{1}{r}{Text in Multicolumn}&\multicolumn1c{e}
&\multicolumn1c{f}\\
Here&is our&table\\
a&b&c\\
\end{tabular}
```

which produces...

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}
```

which produces...

Decimal Numbers
2130.456
30.33
1.4568

You Try It!

Please make a table that includes:

- Use vertical lines
- Try `@{hello}` in the preamble.
- Try making decimal numbers in two adjoining columns; use `@{}` between the two columns so that the two columns don't have a space between them.
- Try horizontal lines
- Try using `\cline{2-3}` and `\cline{4-4}`.
- Try the `\multicolumn{...}{...}` command; add vertical lines in the multicolumn preamble.

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!

Adding extra space between lines in the table

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

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

which produces...

Here	is our	table
a	b	c

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, which uses the keywords following it `height`, `width`, and `depth`. These terms do not have a backslash before them and may be used in any order.)

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

which produces...

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}
```

which produces...

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&table\\
\hline
\noalign{\vskip-2pt}
\hline
a&b&c\\
\noalign{\vspace{6pt}}
We can even put some text in here.\vspace{6pt}}
def&hij&klm\\
\hline
\end{tabular}
```

which produces...

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

Using `\llap` and `\rlap`:

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

which produces...

Not using <code>\llap</code> and <code>\rlap</code> :	Using <code>\llap</code> and <code>\rlap</code> :																								
<table border="1"><tr><td>Here</td><td><i>is our</i></td><td>table</td></tr><tr><td>one</td><td>two</td><td>three</td></tr><tr><td>one</td><td>two</td><td>three*</td></tr><tr><td>¹one</td><td>two</td><td>three</td></tr></table>	Here	<i>is our</i>	table	one	two	three	one	two	three*	¹ one	two	three	<table border="1"><tr><td>Here</td><td><i>is our</i></td><td>table</td></tr><tr><td>one</td><td>two</td><td>three</td></tr><tr><td>one</td><td>two</td><td>three*</td></tr><tr><td>¹one</td><td>two</td><td>three</td></tr></table>	Here	<i>is our</i>	table	one	two	three	one	two	three*	¹ one	two	three
Here	<i>is our</i>	table																							
one	two	three																							
one	two	three*																							
¹ one	two	three																							
Here	<i>is our</i>	table																							
one	two	three																							
one	two	three*																							
¹ 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*}
```

which produces...

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*}
```

which produces...

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




```

which produces...

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

which produces...

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. 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 put text into when you don't want their width to be noticed by 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}{lrrrrrr}\hline
{\em sample} & {\em site 1} &  $\$ \poff_{ij} \$$  & {\em site 2}
&  $\$ \poff_{ij} \$$  & {\em site 3} &  $\$ \poff_{ij} \$$  \\ \hline
cloacin DF13 & 97a & 0.998684 & & & & \\
colicin Ib & 99a & 0.990472 & & & & \\
{\em 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}.
which produces...
```

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 ^a	0.998684				
colicin Ib	99 ^a	0.990472				
recN	71	0.999988	93	0.865704	111 ^a	0.134281

^aIndicates 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] ^a	[0.5610, 0.5750]	[0.5640~] ^b

^aRefs. 19 and 20.

^b $\kappa, \lambda > 1$.

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]\rlap{ $\hat{a}$ &[0.5610, 0.5750]
&[0.5640\sim{]}\rlap{ $\hat{b}$ }\\
\hline
\end{tabular*}
\vskip3pt
 $\hat{a}$ Refs.~19 and 20.

 $\hat{b}$ \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
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
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? The `vrule` in the preamble adds extra vertical space to every line.

```
\begin{tabular*}{\textwidth}{@{\vrule height 10pt depth 5pt width0pt
\extracolsep\fill}ccc@{}}\hline
\bf Index & \bf  $J\Delta$  Representation & \bf  $J\Phi$  Representation & \\\ \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	$b = 0, d = 1, e = 1, f = 1, g = 0, h = 1, i = 1$	$b = 0$

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 & 4 & 235.0007 & 4 & 4 & .0020 & 4 \cr
10 & 5 & 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 now, make it into a floating table with a label and reference.

```

\begin{table}[ht]
\caption{This is the table caption.}
\vskip12pt
\begin{tabular*}{\hsize}{\extracolsep{\fill}}{ccrrcrrc}
\multicolumn{3}{l}{\bf Parameters}&
\multicolumn{5}{c}{\bf Averaged Results} \\ \hline
\vrule height 11pt depth 6pt width 0pt
 $n$  &  $S_{MAX}^*$  &  $t_1$  &  $r_1$  &  $m_1$  &  $t_2$  &  $r_2$  &  $m_2$  \\ \hline
10 & 1 & 4 & 235.0007 & 4 & 4 & .0020 & 4 \\ \hline
10 & 5 & 50 & .0008 & 8 & 50 & .0020* & 12 \\ \hline
\end{tabular*}
\label{tableref}
\end{table}
Here is a reference to the table above: \ref{tableref}.

```

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

Here is a reference to the table above: 1.5.

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=1.5in]{1-fig2.eps}
\caption{Vector addition  $v+w$  produces the diagonal
of a parallelogram.}
\end{figure}
```

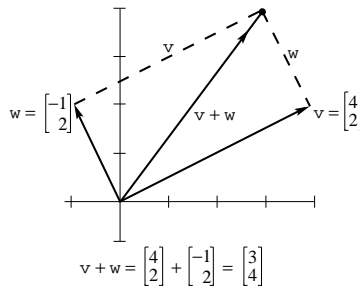


Figure 1.2: Vector addition $v + w$ produces the diagonal of a parallelogram.

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}}
```

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}`.

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.

```
\begin{table}[p]
  \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}
```

which produces...

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

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.

```
\usepackage[dvipsone]{graphicx}
\usepackage[dvipsone]{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}
```

which produces...

one	two	three	red
four	five	six	more red

Row Colors

```
\begin{tabular}{1c}
\rowcolor[gray]{.9}
one&two\\
\rowcolor[gray]{.5}
three&four
\end{tabular}
```

which produces...

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}}1c}
\rowcolor[gray]{.8}
one&two\\
\rowcolor[gray]{.5}\color{white}
three&four
\end{tabular}
```

which produces...

one	two
three	four

Cell Colors

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

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

which produces...

one	two
three and four	

Or, used to override another column or row color:

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

which produces...

one	two
three	four

A few more details, for making colored ruled lines, and positioning the colored screens within the table columns, may be found in `colortbl.pdf`, available from CTAN.