ESA 2010 LaTeXworkshop

Y. Lucero

Contents

1 Basic Typesetting

1.1 Getting started

Exercise 1.1: Set up a basic tex file using the code below. Name the file whatever you like, but make sure it has the suffix .tex.

```
\documentclass[12pt]{article}

\usepackage{amssymb}
\usepackage{amsbsy}
\usepackage{amsmath}

\begin{document}
\section{Hello World!}
filler text
\end{document}
```

Exercise 1.2: Open up the file `LatexWorkshopExercises.tex` in your text editor. At the top of the document is a paragraph of text that is commented out. You can copy and paste this text into your document to use as dummy text.

Exercise 1.3: Use the options in `\documentclass[options]{class}` to change the font size of your document. Note that there are only templates types for three font sizes: 10pt, 11pt and 12 pt, where 10pt is the default.

Exercise 1.4: Add some section headings using `\section{text}`

Exercise 1.5: Add some subsection headings using `\subsection{text}`

Exercise 1.6: Make an unnumbered section heading with the command `\section*{text}`

Exercise 1.7: Similarly, make an unnumbered subsection heading, `\subsection*{text}`

Exercise 1.8: LaTeX uses a run-in heading called paragraph headings. Add paragraph headings using `\paragraph{text} text text`

Exercise 1.9: There are also sub-paragraph headings `\subparagraph{text} text text`

Exercise 1.10: Change some text to bold `\textbf{text}`

Exercise 1.11: Emphasize some text (with italics) using `\emph{text}`

Exercise 1.12: Underline some text using `\underline{text}`

Exercise 1.13: Special characters. Some characters in tex have special uses. For example, '%' is used to comment lines out and '$' is used to offset mathematics within a paragraph (called inline math). If you wanted to use these reserved characters just as characters, you have to place an escape in front of them: `\%`. Add these characters to your document: %, $, &, #, , {.

Exercise 1.14: A quirk of LaTeX is the lack of smart quotes. When you type quotes, you always get closed quotes. In order to get an open quote, you have to use the symbol that is found on the upper left key with the ~ on it: `` ``quoted'' `` yields "quoted"

Exercise 1.15: Here are a few text accents (there are many more available to you): Use `\~n` to get ñ. Also try `\^a` and `\.x`

Exercise 1.16: There are a series of commands that will let you format text sizes. Table 1 (on page 4) gives the details. See what happens when you type `\tiny` anywhere in your text.

Table 1: Absolute Point Sizes in Standard Classes

size	10pt (default)	11pt option	12pt option
`\tiny`	5pt	6pt	6pt
`\scriptsize`	7pt	8pt	8pt
`\footnotesize`	8pt	9pt	10pt
`\small`	9pt	10pt	11pt
`\normalsize`	10pt	11pt	12pt
`\large`	12pt	12pt	14pt
`\Large`	14pt	14pt	17pt
`\LARGE`	17pt	17pt	20pt
`\huge`	20pt	20pt	25pt
`\Huge`	25pt	25pt	25pt

1.2 Introduction to environments

In TeX, many important things are handled with environments. For example to create equations that are typeset formally (centered, numbered, etc.), use the equation environment. The code,

```
\begin{equation}
```

```
E=mc^2
\end{equation}
```

creates:

$$E = mc^2 \tag{1}$$

To use an environment you use the command `\begin{environment.name}` to open the environment and the command `\end{environment.name}` to close it.

Exercise 1.17: For now, try using the `itemize` environment to create a list:

```
\begin{itemize}
\item green
\item blue
\item red
\end{itemize}
```

This should produce:

- green
- blue
- red

Exercise 1.18: Reproduce the list above as a numbered list using the `enumerate` environment

Exercise 1.19: you can also create nested lists by simply nesting these environments:

```
\begin{enumerate}
\item types of fish
   \begin{enumerate}
      \item red
      \item blue
```

```
   \end{enumerate}
\item flavors of jello
   \begin{enumerate}
      \item grape
      \item cherry
      \item raspberry
      \item lime
   \end{enumerate}
\end{enumerate}
```

produces:

1. types of fish
   (a) red
   (b) blue
2. flavors of jello
   (a) grape
   (b) cherry
   (c) raspberry
   (d) lime

2 Mathematics

2.1 Math environments

Make sure that you have the following packages in your preamble:

```
\usepackage{amssymb}
\usepackage{amsbsy}
\usepackage{amsmath}
```

You can create math notation within a paragraph (called inline math) by surrounding the text with dollar symbols, for example, `$c^2$`. The `$` symbols open and close inline math mode.

Exercise 2.1: Reproduce the following inline math text:

The pythagorean equation is sometimes also written as $c^2 = a^2 + b^2$.

Often equations are created with the equation environment, like Equation 1. But there are a few alternatives to the equation environment. You can use an alternate environment called `displaymath`. This functions very similarly to the `equation` environment except that it does not number the equations.

Exercise 2.2: Typeset Equation 1 with `displaymath`.

You also have the option of using this notation to create something like the `displaymath` environment:

```
\[ E=mc^2 \]
```

Exercise 2.3: Use the square bracket notation to retypset Equation 1

2.2 Creating mathematical notation

Reproduce the equations in this section using some of the commands given.

Exercise 2.4: Choose whichever math environment you prefer from the examples above and write the equation:

$$S = cA^z \tag{2}$$

Exercise 2.5: use the function `\sqrt{}` to write this formula

$$c = \sqrt{a^2 + b^2} \tag{3}$$

Exercise 2.6: Create a ratio with the function `\frac{numerator}{denominator}`

$$I = \frac{aP}{1 + abP} \tag{4}$$

Exercise 2.7: use the function `\hat{}` to add a modifier to S

$$\hat{S} = \frac{IP}{I + E} \tag{5}$$

Exercise 2.8: Create delimitters that can adjust their size: `\left( content \right)`

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right) \tag{6}$$

Exercise 2.9: Special characters: create the greek letters with `\epsilon`, `\mu` and `\sigma` and use `\sim` to create the tilde symbol

$$\epsilon \sim Norm(\mu, \sigma^2) \tag{7}$$

Exercise 2.10: Create superscripts with more than one character: `e^{-rx}` and create a sum with subscripts `\sum_x`

$$1 = \sum_x e^{-rx} l(x) b(x) \tag{8}$$

Exercise 2.11: Create a sum with multi-character subscripts and a superscript `\sum_{i=0}^k`

$$RSS = \sum_{i=1}^{N} (y_{pred} - y_{obs})^2 \tag{9}$$

Exercise 2.12: Special characters and spaces, use `\cdots` to create a ellipses and `\qquad` to create a space. Also, use the greek `\beta`

$$y \sim \beta_0 + \beta_1 x_1 + \cdots \beta_i x_i + \epsilon \qquad \epsilon \sim Norm(\mu, \sigma^2) \tag{10}$$

Exercise 2.13: Special characters: `\rightarrow` and `\infty` and the function `\lim_{x}`

$$e^x = \lim_{n \to \infty} \left(1 + \frac{x}{n}\right)^n \tag{11}$$

Exercise 2.14: Use `\mid` to create a vertical line and and escape character to get curly braces without an error: `\{ text \}`

$$Pr\{A \mid B\} = \frac{Pr\{A, B\}}{Pr\{B\}} \tag{12}$$

Exercise 2.15: Use `\textup{}` to get the plain text "good."

$$Pr\{J_i(T) = A\} = Pr\{H(A) = \text{good}\} Pr\{S(T) > dist(A, B) \mid J_i(0) = B, t = T\} \tag{13}$$

Exercise 2.16: The equation environment numbers equations. You can suppress the numbering by using `\nonumber` before you close the equation environment. Notice that the equation numbering skips over that equation.

Exercise 2.17: Reformat the equation numbering to be on the left side by adding the document class option "leqno" in the preamble:

```
\documentclass[11pt,leqno]{article}
```

Exercise 2.18: Also, try the document class option "fleqn" to have the equations flush left.

Exercise 2.19: The align environment is used for creating multi-line equations. The the \\ is used to break the lines and the & is used to indicate where to align the different lines.

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
\begin{align}
 f(x) = x^4 &+ 7x^3 +2x^2 \nonumber \\
 & +10x + 12
\end{align}
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

$$\begin{aligned} f(x) = x^4 + 7x^3 + 2x^2 \\ + 10x + 12 \end{aligned} \tag{14}$$

Here, I have used the command `\nonumber` on the first line, so the equation produced only numbers the second line. We could also use the environments `align*` to produce an unnumbered equation.

Exercise 2.20: Notice that in the equation environment the characters in the denominator are sometimes very small:

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}} \tag{15}$$

Let's subvert this by reformating the characters. Use the command `\displaystyle{denominator}` in the denominators to produce:

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + a_4}}} \tag{16}$$

Exercise 2.21: The amsmath package gives us five environments for creating matrices: pmatrix, bmatrix, Bmatrix, vmatrix and Vmatrix. We use these environment within the equation environment. We use the command \\ to break lines and & to separate columns. Use the code below to create a matrix, and then try out all five environments.

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
\begin{equation}
A = \begin{pmatrix}
a_{11}   & a_{12}   & a_{13} \\
a_{21}   & a_{22}   & a_{23} \\
a_{31}   & a_{32}   & a_{33}
\end{pmatrix}
\end{equation}
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Exercise 2.22: The cases environment is used to create piecemeal equations. As before, we use the command \\ to break lines and & to separate columns. For example:

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
\begin{equation}
f(x)=\begin{cases}
0 & \text{for} \; i = 0 \\
1 & \text{for} \; i > 0
\end{cases}
\end{equation}
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

The command `\text{text}` formats the characters as words rather than math. Within any math environment, the command `\;` creates a single blank space. Use the cases environment to reproduce Equation 17. You will require the symbol `\leq` for the lesser-than-or-equal-to symbol.

$$f(x,y) = \begin{cases} 0 & \text{for } x \leq 0\,, \\ \sin x + \phi & \text{for } x > 0\,. \end{cases} \tag{17}$$

3 Figures

In both Word and in LaTeX, figures and tables are usually created as floats. Floats are objects that cannot be broken up into parts and do not need to appear in the exact location where they are placed within the text. Each time we use the figure environment to create a figure, we create a float objects.

Make sure that you have `\usepackage{graphicx}` in your preamble.

Exercise 3.1: To include figures, we will use the command `\includegraphics[options]{filename}`. Make sure that you have a file named fig1.pdf in the same directory as your tex file, then use the following code to create a basic figure.

```
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
\begin{figure}
\begin{center}
\includegraphics[width=3in]{fig1}
\caption{A very basic figure.}
\end{center}
\end{figure}
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

By default, `\includegraphics{filename}` assumes that the file given is a pdf file. You can use files in many formats, including jpg, gif, png and ps. If you are using a file format other than pdf, then you must include the file suffix when you specify the file name.

Exercise 3.2: You may specify a longer pathname in the command `\includegraphics{pathname}`. Create subdirectory named figs in your current folder, and move fig1.pdf to that subdirectory. Then specify `\includegraphics{figs/fig1}`. You should also be able to move up a directory using the pathname `./fig1`

Exercise 3.3: In the above example, we used the option "width" for includegraphics. Try these other options:

```
\includegraphics[scale=0.4, height=4in, angle=90]{filename}
```

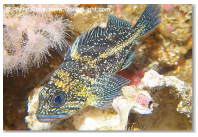
Exercise 3.4: The figure environment has options for figure placement: h for here, t for top of page, b for bottom of page and p for place it on a floats only page. For example, try `\begin{figure}[t]`.

LaTeX tries to follow your placement directions, but it will only do so if the figure "fits." I.e., if you specify that you want to place a five inch figure "here," but the page is already mostly filled with text, LaTeX will override your specification and find another place to fit the figure. Therefore, you can specify several placement options at once, and LaTeX will follow them in the order that you specify. For example, the default figure option is often written: [htbp]. Note, LaTeX will try to find a place where the figure fits in both width and height. This means that if your figure is too wide for the page, it will not fit anywhere and LaTeX will place it at the end of your document. Furthermore because floats are placed in order, this will cause ñall of your remaining figures will be placed at the end of the document as well.

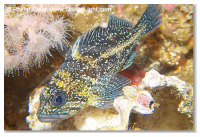
Exercise 3.5: Of course, in manuscript form you *want* to place the figures at the end of the document and on its own page. To do this, place your figures at the end of your document and use the command `\newpage` before each figure.

Exercise 3.6: Make sure you have the subfigure package loaded and recreate Figure 1 (on p14) with the following text. Use the command `\\` to break between rows of figures.

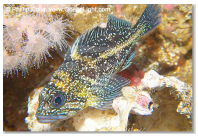
```
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
\begin{figure}[htbp]
\begin{center}
  \subfigure[cap 1]{\includegraphics[width=1in]{figs/fig1}}
  \subfigure[cap 2]{\includegraphics[width=1in]{figs/fig1}} \\
  \subfigure[cap 3]{\includegraphics[width=1in]{figs/fig1}}
  \subfigure[cap 4]{\includegraphics[width=1in]{figs/fig1}}
  \subfigure[cap 5]{\includegraphics[width=1in]{figs/fig1}}
\end{center}
\caption{Master caption.}
\label{fig.subfigs}
\end{figure}
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```



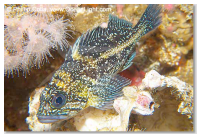
(a) cap 1



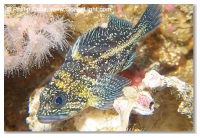
(b) cap 2



(c) cap 3



(d) cap 4



(e) cap 5

Figure 1: Master caption.

4 Tables

Table 2 is a very basic table produced by this code:

```
%$$$$$$$$$$$$$$$$$$$$$
\begin{table}[htdp]
\caption{default}
\begin{center}
\begin{tabular}{ c c c }
\hline
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\hline
\end{tabular}
\end{center}
\end{table}
%$$$$$$$$$$$$$$$$$$$$$
```

Opening the table environment creates a float. Notice the same placement specifiers as used for the figure environment. The `\caption{}` command creates a caption. We open a centering environment so that the table is centered on the page. Opening the tabular environment starts the table itself. The tabular arguments are how the columns should be

aligned: c is for centerd, r is for right aligned, and l is for left aligned. The command `\hline` creates a horizontal line. Rows are broken with \\ and columns are broken with &.

Table 2: default

1	2	3
4	5	6
7	8	9

Exercise 4.1: Re-create Table 3. You will need to use the inline math mode for the greek letters, i.e. `$\gamma$`. Also, center the columns with symbols in them, but left align the columns with text.

Table 3: Some greek letters.

ω	omega	Ω	Omega
λ	lambda	Λ	Lambda
γ	gamma	Γ	Gamma

Exercise 4.2: The booktabs package creates more professional looking tables. Make sure you include the booktabs package and try replacing `\hline` with the booktabs commands `\toprule`, `\bottomrule`, and `\midrule`. Also, create a horizontal line that only stretches across some columns, i.e. for columns 1 through 2: `\cmidrule{1-2}`

Exercise 4.3: `\begin{tabular}{c l c l c}` Note the distinction between the lower case letter "l" and the vertical bar | (located on the '\' key). For left alignment, we use the letter "l." To put a vertical line between columns, use the vertical bar between the column alignment arguments to tabular: `{c | c | c}`

Exercise 4.4: The multirow package gives us two commands for grouping rows and columns: `\multicolumn{#columns}{orientation}{text}` and `\multirow{#rows}{width}{text}`. For the width parameter, we can either specify a width (2in) or we can use `*` to get latex to choose the best fitting width. Add this heading to Table 3 using the multirow package and the booktabs package:

```
\toprule
\multicolumn{2}{c}{lowercase}&\multicolumn{2}{c}{uppercase} \\
\midrule
```

Exercise 4.5: Next, use the multirow command to group rows.

```
\multirow{3}{*}{familiar}&$\omega$   & omega  & $\Omega$ & Omega \\
&$\lambda$   & lambda  & $\Lambda$ & Lambda \\
&$\gamma$  & gamma  & $\Gamma$ & Gamma \\
```

Don't forget to add the empty column on the other grouped rows that don't have the multirow command on them.

Exercise 4.6: Put it all together to reproduce Table 4

Table 4: Some harder greek letters.

	lowercase		uppercase	
familiar	ω	omega	Ω	Omega
	λ	lambda	Λ	Lambda
	γ	gamma	Γ	Gamma
unfamiliar	ξ	xi	Ξ	Xi
	υ	upsilon	Υ	Upsilon

Exercise 4.7: Frequently, it is desirable to produce tables where a column of numbers are aligned so that they center around a decimal point. This can be done using the specifier `@{text}` which inserts some text in every row. This command suppresses the inter-column space normally placed between columns. Try the following table:

```
%$$$$$$$$$$$$$$$$$$$$$
\begin{table}[h]
\begin{center}
\begin{tabular}{r@{.}l}
3    & 14159   \\
16   & 2       \\
123  & 456     \\
\end{tabular}
\end{center}
\end{table}
%$$$$$$$$$$$$$$$$$$$$$
```

Exercise 4.8: LaTeX doesn't wrap text in columns by default. If the text in a column is too wide, it will create a very wide column and maybe run off the page. You can get wrapped text in table columns by using `p{width}` as a specifier in the tabular environment. This defines a special type of column with text wrapping, as in a normal paragraph. For example, try re-creating a version of Table 5 using the following:

```
\begin{tabular}[h]{ l | p{3in}}
```

Exercise 4.9: You can resize tables with `\resizebox{width}{height}{tabular object}`. Using the symbol '!' for the height preserves the original width/height ratio. Try resizing one of your tables with:

```
\resizebox{5in}{!}{
\begin{tabular}{...}
...
\end{tabular}
}
```

Table 5: Example of columns with wrapped text. Also, useful list of packages for making sophisticated tables.

Package	Description
hhline	Do whatever you want with horizontal lines
array	gives you more freedom on how to define columns
colortbl	make your table more colorful
supertab	for tables that need to stretch over several pages
longtable	same as above. Note: footnotes do not work properly in a normal tabular environment. If you replace it with a longtable environment, footnotes work properly
xtabular	Yet another package for tables that need to span many pages
tabulary	modified tabular* allowing width of columns set for equal heights
array	Does several things, including letting you create paragraph columns where text is bottom aligned or centered.

Exercise 4.10: You can also try `\scalebox{ratio}{tabular object}`

Exercise 4.11: Use the landscape environment to rotate your table to landscape orientation. This environment requires the lscape package, so be sure to add lscape in your pre-amble with other packages.

```
\begin{landscape}
\begin{table}
....
\end{table}
\end{landscape}
```

5 Advanced Typesetting

5.1 Working with counters, references and labels

Counters Latex uses counters to keep track of the numbering of sections, equations, tables, figures, etc. Here is a partial list of important counters that are automatically created:

- chapter
- section
- part
- paragraph
- page
- equation
- figure
- table
- footnote
- section
- subsection
- subsubsection

Because these counters are always created automatically, at any time you can ask LaTeX to print out the value of the counter—even if you haven't use the counter before. For example, you could ask what the latest figure number is even before you have created a figure.

Exercise 5.1: See the value of a counter by placing '\the' in front of the counter name, e.g. `\thesection` will print the section number.

Exercise 5.2: You can use the command `\setcounter{counter.name}{new value}` to manually change the value of a counter. Try using it to change the current page number to page ten: `\setcounter{page}{10}`. Then change it back!

Exercise 5.3: Change the table numbering so that it skips Table 1 and labels the first table as Table 2. Place `\setcounter{table}{2}` before the first table float is created.

Exercise 5.4: You can simply advance a counter using `\addtocounter{counter}{1}`

Exercise 5.5: Importantly, you can reformat how numbering appears using `\renewcommand`. (This command can be used to change any LaTeX command.) To reformat how a counter appears, you can use any of the following: `\arabic{counter}`, `\alph{counter}`, `\Alph{counter}`, `\roman{counter}`, `\Roman{counter}`. For example place the following in your preamble:

```
\renewcommand{\thetable}{I-\arabic{table}}
```

Exercise 5.6: Now experiment with some of the other letter formating functions to see what they do: `\arabic{counter}`, `\alph{counter}`, `\Alph{counter}`, `\roman{counter}`, `\Roman{counter}`.

Labels and References Any counter can be dynamically referenced. This means that you can reference figures, equations and any other counter throughout your document and the numbers will automatically update in the text. First, you use mark the counter with the command `\label{lab.text}`. Then later, you can refer to this label with the command `\ref{lab.text}` to reference it.

Exercise 5.7: Label one of the equations in your document by placing `\label{eq.test}` inside the equation environment. Then reference that label elsewhere by writing `\ref{eq.test}`. Notice that you have to compile twice for the reference to work (the first time you compile you get: **??**). This is because LaTeX collects the labels on the first pass, and goes through and fills in the references on the second pass.

Exercise 5.8: The placement of the label can matter. For equations, the label should be placed within the equation environment. For sections, you place the label anywhere after the section has been created. For figures and tables, the label should be placed after the `\caption{text}` command, because this is when the counter is officially advanced. Try placing the labels in the wrong place and see what happens.

Exercise 5.9: It is a good idea to have a naming system for choosing labels. For example, I usually follow this convention:

- tab.name for tables
- fig.name for figures
- eq.name for equations
- sec.name for sections

where "name" is something short and memorable. The goal is to choose labels that are easy to remember so that you spend minimal time going back to look them up. Go through and give all of your figures and tables systematic and memorable names of your choosing.

Exercise 5.10: Try out `\pageref{your.label.here}`. Instead of the counter number, this returns the page number with your label on it.

Exercise 5.11: There is a built-in command to make creating appendix headings easier. Place the command `\appendix` towards the end of your document and then create some new section headings afterwards. Notice that you now have a new style of section numbering for your "appendix."

5.2 Titles and other front matter

Exercise 5.12: You can automatically create a formal title by defining a few fields and then adding the command `\maketitle` where you want the title to show up (usually immediately below `\begin{document}`. To define the title page fields, add the following to your preamble:

```
\title{Best Article Ever}
\author{your name here!}
\date{December 31, 1999}
...
\begin{document}
\maketitle
```

Exercise 5.13: If you fail to define the date field, the latex will default to today's date. If you prefer no date, you can put `\date{}`

Exercise 5.14: Insert a table of contents into your document by adding the command `\tableofcontents` after `\maketitle`. You will need to compile twice. Notice that this command creates a file with the .toc suffix. This is a simple text file, and if you open it you will see the latex code used to typeset your table of contents.

Exercise 5.15: In exactly the same way, you can add `\listoffigures` or `\listoftables`. Again, text files are produced with the suffixes .lof and .lot and you will need to compile twice. Latex will place the list where you place the commands. You may prefer to have these lists at the end of your document.

Exercise 5.16: These two commands draw on the captions to generate the lists. If you prefer a shorter title be used in the list, you can go back to the caption in the figure or table and use the options: `\caption[short caption for list]{real caption}`

5.3 Line numbers, headers, and footers

Exercise 5.17: You can add line numbers with `\usepackage{lineno}` and the following command inside your document: `\linenumbers`

Exercise 5.18: Creating a footnote is simple: `\footnote{text}`

Exercise 5.19: You can also create margin notes pretty easily: `\marginpar{text}` will place a note in the outside margin.

Exercise 5.20: You can also create customized headers and footers using the fancy header package. Add the following to your preamble

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhead[L]{Date updated: \today{}}
\fancyhead[R]{\LaTeX~Exercises}
```

There are a couple of new things going on in the example above. I am using the command `\today{}` to print out the current date in the heading. I am using the command `\LaTeX` (case sensitive) to get the name to print out in the official LaTeXway. And I am using the symbol `~` to preserve a space in between LaTeXand Exercises. Try removing the `~` and see that the words are pushed together.

Exercise 5.21: Add a footer using `\fancyfoot[selector]{text}`. The selectors can be L for left, R for right, or C for center. Also, try leaving the selector out altogether and see what happens.

Exercise 5.22: Suppose you want to write the page number as 'page x of y.' By default, latex only keeps track of the current page and not the total number of pages. But you can use the `lastpage` package to keep track of the total number of pages. Like with `\ref{}`, you will need to compile twice to make this work. Try adding this to your preamble:

```
\usepackage{lastpage}
\cfoot{\thepage\ of \pageref{LastPage}}
```

Exercise 5.23: Notice that latex adds page numbers to your document by default. Try removing the page numbers by adding the command `\pagestyle{empty}` in your preamble.

5.4 Long documents: books and dissertations

Sometimes when you are working with a very long document the file can grow quite unwieldy. LaTeX has a natural advantage here because the figures files are not embedded; this limits the problems you can have with computer memory because even an extremely long simple text file is not a particularly large file. Nonetheless, by now you have noticed that a long text file can be difficult to navigate.

For this problem, there is a handy function called `\include{}` that allows you spread your document out among multiple text files. So, typically, you might have a separate file for each chapter in your thesis. You would use `include` to bring all of these chapters into a single document.

The central file might look something like this:

```
\documentclass[12pt]{article}

\usepackage{booktabs}

\begin{document}
\maketitle
\tableofcontents
\include{chapter1}
\include{chapter2}
\include{chapter3}
\end{document}
```

In the above example, there should be three files in your directory named `chapter1.tex, chapter2.tex,` and `chapter3.tex`. The command acts exactly as though you copy and pasted the content of those files into the central document. Therefore, the three called tex files should have no preamble of their own! There is only one pre-amble and that is in the central file. This also means that you cannot compile the sub files independently (unless you add a separate preamble temporarily in order to do so).

Exercise 5.24: Create a new tex file and insert some code into it, but do not give it a preamble. Then use the `include` function to insert this text into this document.

5.5 Working with style files

So far, we have been working with one document class in LaTeX, the article class. (Notice that the very first line your .tex file specifies the document class.) This format is based on a style file named `article.cls` that came with you basic installation. While I don't find them very useful, there are several other built in styles, such as `book` and `letter`. While the default styles have limited use in my experience, there are several very useful specialized style files that are worth learning about.

When you write your dissertation in LaTeX, one of the first things you should do is locate a style file for your university's dissertation format. Someone has written one and made it available, likely an engineering graduate student. Similarly, many journals can provide you with a style file that formats your documents according to their journals specifications. Today, we are going to work with a style file for the journal Ecological Applications.

Exercise 5.25: Go get the ecologyLatexClass.tar. This is a style file and supporting materials for the journal Ecological Applications written by Todd Jobe, a UNC postdoc. Unzip these files and compile the example file.

Exercise 5.26: Next, put a copy of ecology.cls into the same directory as your exercises.tex file. Change the document class of your document from "article" to "ecology." Compile this document, ignoring the errors that come up. Observe how this style file changes your document. Particularly note what it does with figures and tables.

Exercise 5.27: Fill out the empty fields of the title page (Running Head, etc.). Look to the preamble in the example file to learn how to do this.

Exercise 5.28: Use the environments for abstract and keywords to add an abstract and keywords to your document. Look to the example file for a demonstration of this.

5.6 Manipulating white space

Latex recognizes a few very specific measurements. Table 6 lists the types of units that are available to you when using spacing commands.

Table 6: Measurement units used in LaTeX

pt	a point is 1/72.27 inch, that means about 0.0138 inch or 0.3515 mm.
bp	a big point is 1/72 inch, that means about 0.0139 inch or 0.3527 mm.
mm	a millimeter
cm	a centimeter
in	inch
ex	roughly the height of an 'x' in the current font
em	roughly the width of an 'M' (note the uppercase) of the current font

Exercise 5.29: Add a vertical space of three inches between two paragraphs: `\vspace{3in}`

Exercise 5.30: Add a small blank space inside the middle of a word: `\hspace{1em}`

Here is a list of several built in length parameters that control different parts of the page layout:

- `\baselineskip` The normal vertical distance between lines in a paragraph
- `\baselinestretch` Multiplies `\baselineskip`
- `\pagewidth` The width of the page
- `\pageheight` The height of the page
- `\parindent` The normal paragraph indentation
- `\parskip` The extra vertical space between paragraphs
- `\tabcolsep` The default separation between columns in a tabular environment
- `\textheight` The height of text on the page
- `\textwidth` The width of the text on the page
- `\topmargin` The size of the top margin (bottom margin is pageheight-topmargin)

Exercise 5.31: Choose a parameter and use the set length command to simply change the value: `\setlength{\parskip}{5ex}`. Notice that you can place this command anywhere in your document (including the preamble) and it will only change the parameter after the command is issued.

Exercise 5.32: Alternatively, make changes relative to the the default value:

```
\addtolength{\baselineskip}{1in}
```

Exercise 5.33: By default, latex will indent new paragraphs. I generally prefer that paragraphs not be indented, but that there be white space between paragraphs instead. To achieve my preferred formatting, I add this to the preamble:

```
\setlength{\parindent}{0in}
\setlength{\parskip}{2ex}
```

Exercise 5.34: Use the geometry package options in the preamble to change the margin size of the document, e.g.:

```
\usepackage[left=1in,right=1in]{geometry}
```

5.7 Using environments for special formatting

There are several specialized environments built into latex that can be used to create parts of your document that need special formatting. For example, I used the abstract environment to create the following:

Abstract

> This paper is about a very important and general subject. In our resereach, we did this that and the other thing. Some of our methods are totally unique. Our results showed this and that. We conclude that more people should do research like ours and that this work will save the planet.

The commands to do this were:

```
\begin{abstract}
insert words here
\end{abstract}
```

Exercise 5.35: Add an abstract to your document using the the abstract environment.

Exercise 5.36: Use the description environment to create a small glossary

```
\begin{description}
\item[Love] a many splendored thing
\item[One] the loneliest number
\item[Black] the color of my true love's hair
\end{description}
```

Exercise 5.37: There are two environments for quotations, with small differences between them. The `quote` environment is meant for shorter quotes or a series of short quotes.

```
\begin{quote}
I had a dream \emph{(Martin Luther King)}
\end{quote}
```

While the `quotation` environment is for formatting longer quotations. Use one or both of these environments to add a quote to your document.

Exercise 5.38: There are a number of packages for formatting algorithms and pseudocode. For example, make sure that you have the package `algorithmic` in your preamble and try typesetting the following:

```
\begin{algorithmic}
\IF {$i\geq maxval$}
        \STATE $i\gets 0$
```

```
\ELSE
        \IF {$i+k\leq maxval$}
                \STATE $i\gets i+k$
        \ENDIF
\ENDIF
\end{algorithmic}
```

to produce:

if $i \geq maxval$ **then**
  $i \leftarrow 0$
else
  if $i + k \leq maxval$ **then**
    $i \leftarrow i + k$
  end if
end if

Exercise 5.39: You can also create your own environments to suite your own purposes with the command `\newenvironment{name}{begin}{end}`. You give your environment a name, and then you specify which commands will occur when the environment is opened (begin), and which will occur when the environment is closed (end). Usually, you piggy-back on another environment.

```
\newenvironment{giantQuote}
{\begin{quote} \Huge}
{\end{quote}}

\begin{giantQuote}
I had a dream - \emph{MLK}
\end{giantQuote}
```

I had a dream - *MLK*

6 Bibliographies

6.1 Bibliographies without BibTeX

The very simplest bibliographies are created with an environment called thebibliography.

Exercise 6.1: Try this code to generate a short reference list:

```
\begin{thebibliography}{}
\bibitem{lamport.1994}
  Leslie Lamport,
  \emph{\LaTeX: A Document Preparation System}.
  Addison Wesley, Massachusetts, 2nd Edition, 1994.
\end{thebibliography}
```

Each bibliographic item is started with `\bibitem{cite.key}`. Next you write out the reference. The line breaks above are for readability, latex will actually ignore them.

Exercise 6.2: if you would like to cite this reference, type `\cite{lamport.1994}`. As before, you will need to compile twice. By default, latex uses numbered citations. To change this we require the bibtex and the natbib package.

6.2 BibTeX

Make sure you have these files in the same directory as your tex file: example.bib, ea.bst

Exercise 6.3: Now we will use bibtex to create a reference list. There are four steps.

1. Cite the references in your document. Place this code in somewhere in your document

```
\nocite{ives.2003,kuparinen.2009,kuparinen.2009a,maechler.2005,zwillinger.1992}
```

2. Call the bibliography style file:

```
\bibliographystyle{ea}
```

The command `\bibliographystyle{}` can be placed anywhere in the document, including the preamble, so long as it appears before you issue the command `\bibliography{}`.

3. Call the bibliography data file

```
\bibliography{example}
```

Place the command where you want the reference list to appear and be sure the leave off the .bib suffix.

4. Compile. In TeXnic Center, the inclusion of bibtex happens in a single step along with the usual compile command. However in any unix-like system, including TeXshop on OSX, this is done in four steps:

1. compile LaTeX (shift-apple-L): gather cite keys
2. compile BibTeX (shift-apple-B): looks up cite keys in .bib file, generates .bbl file
3. compile LaTeX (shift-apple-L): inserts reference list
4. compile LaTeX (shift-apple-L): inserts citations where cite keys are

This should produce the following:

References

Ives, A. R., B. Dennis, K. L. Cottingham, and S. R. Carpenter. 2003. Estimating community stability and ecological interactinos from time-series data. Ecological Monographs **73**:301–330.

Kuparinen, A., C. G. de Leaniz, S. Consuegra, and J. MerilÃd'. 2009*a*. Growth-history perspective on the decreasing age and size at maturation of exploited atlantic salmon. MEPS **376**:245–252.

Kuparinen, A., S. Kuikka, and J. Meril. 2009*b*. Estimating fisheries-induced selection: traditional gear selectivity research meets fisheries-induced evolution. Evolutionary Applications .

Maechler, M., P. Rousseeuw, A. Struyf, and M. Hubert. 2005. Cluster analysis basics and extensions. Rousseeuw et al provided the S original which has been ported to R by Kurt Hornik and has since been enhanced by Martin Maechler: speed improvements, silhouette() functionality, bug fixes, etc. See the 'Changelog' file (in the package source).

Zwillinger, D. 1992. Handbook of differential equations. Second edition. Academic Press, San Diego, California.

Exercise 6.4: Bibtex generates a .bbl file. This is also a simple text file. You can open it up and see how your reference list is generated.

Exercise 6.5: Let's cite some of these references within the text now. We would like to use author/year format and so we require the natbib package. Make sure that you have `\usepackage{natbib}` in your preamble. Now try: `\citep{maechler.2005}`. This should produce (Maechler et al., 2005)

Exercise 6.6: Use the command `\bibpunct` to change the citation style:

```
\bibpunct{[}{]}{,}{a}{}{;}
```

It has six mandatory arguments:

1. the opening bracket symbol, default is '('
2. the closing bracket symbol, default is ')'
3. the punctuation between multiple citations, default is ';'
4. the letter 'n' for numerical style, or 's' for numerical superscript style, any other letter for author/year, default is author/year;
5. the punctuation that comes between the author names and the year, the default is none
6. the punctuation that comes between years or numbers when common author names are suppressed (default is ',')

Exercise 6.7: Now, refer to Table 7 for the full list of citation commands in natbib. Try each one out.

Table 7: Natbib commands

Citation command	Output
`\citet{goossens93}`	Goossens et al. (1993)
`\citep{goossens93}`	(Goossens et al., 1993)
`\citet*{goossens93}`	Goossens, Mittlebach, and Samarin (1993)
`\citep*{goossens93}`	(Goossens, Mittlebach, and Samarin, 1993)
`\citeauthor{goossens93}`	Goossens et al.
`\citeauthor*{goossens93}`	Goossens, Mittlebach, and Samarin
`\citeyear{goossens93}`	1993
`\citeyearpar{goossens93}`	(1993)

Exercise 6.8: We provided several bibliography style files: ecology.bst, science.bst, nature.bst and plos.bst. Make sure these files are in your directory and experiment with the various style files. For example,

```
\bibliographystyle{science}
```

7 Sweave

Sweave is a package that integrates R and LaTeX. What this means, is that you type R code directly into a LaTeX file and Sweave formats and inserts the R results into your text.

Exercise 7.1: Create a new sweave file. Name the file whatever you like, but make sure it has the suffix .Rnw. Copy and paste a LaTeX preamble from your current file. Don't forget to add the `\end{document}` command at the end.

Exercise 7.2: Type in the following simple Sweave input and then compile your document *with the sweave commands*. This step is often buggy, so take a moment now to get it working before we put in any interesting R code.

```
<<>>=
a = 8
@
```

Exercise 7.3: Use the sweave command to generate `filename.tex`. Open up this file to see how it is typeset using the environment `Schunk`. Now compile this file with your LaTeX back end.

Exercise 7.4: Now try a little bit more R code:

```
<<>>=
randoms = rnorm(100)
summary(randoms)
mean(randoms)
sd(randoms)
@
```

Debugging in Sweave can be a problem. Usually it is easier to debug from the R code itself. You can generate a file that extracts the R code from your sweave file.

Exercise 7.5: Go to R. Call the function `Stangle(filename.Rnw)`. This should generate the file `filename.R`. This file contains all of the R code from your .Rnw file.

Exercise 7.6: Go ahead and open up `filename.R` and look at it. Then source it in R by typing `source("filename.R")`

Exercise 7.7: Sometimes you want to change something around, but you don't want it to print out in your document. In this case you set the option `echo` to false.

```
<<echo=FALSE>>=
setwd("my.directory.of.choice")
@
```

Exercise 7.8: You can name sweave sections, and then call them later by their name. By default, the first argument in a sweave call is interpreted as the name.

```
<<randomstep>>=
randoms = rnorm(100)
@
```

And now, everytime I want to get a new set of random numbers I can call this sweave chunk:

```
<<>>=
mean(randoms)
<<randomstep>>
mean(randoms)
<<randomstep>>
mean(randoms)
@
```

Exercise 7.9: Inserting a figure is a two step process. First you create the figure. Here you use the argument name to give the figure file a name.

```
<<figRandoms,fig=TRUE,echo=FALSE,include=FALSE>>=
plot(randoms)
@
```

And then you have to insert it into your document, in the usual latex way:

```
%++++++++++++++++++++++++++
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=3in]{filename-figRandoms}
```

```
\caption{}
\label{fig.random}
\end{center}
\end{figure}
%++++++++++++++++++++++++++
```