Stat 290: Lab 3
Introduction to LaTeX

Lab Objectives

1. To introduce basic LaTeXcommands

Assignment

There is nothing to be turned in for this lab, however, remember that your homework for this week must be done in LaTeX.

Intro to LaTeX

Copy the files `example.tex, cdfs.ps, pdfs.ps`, all available on the class Homework and Labs web page. We will make some modifications to the file `example.tex` to see what else we can do with LaTeX. Bring up this file in emacs (with `C-x C-f` if emacs is already open, or open emacs with `emacs example.tex &` at the shell prompt).

In case you are wondering about any typesetting in this lab, the file `lab3.tex` is also available on the web page, and you can peruse it at your leisure.

Compile the file

Emacs has a special LaTeX mode. You should see it say "LaTeX" on the info bar near the bottom. In this mode, you can use `C-c C-c` to access a useful set of LaTeX commands. For example, `C-c C-c LaTeX` will run LaTeX on the current file. If it compiles successfully and completely (sometimes you need to run it more than once if there are internal references), then `C-c C-c View` will bring up an xdvi window. Note that Emacs is pretty good at guessing what you want to do when you hit `C-c C-c`, so if the default is correct, you can just hit enter. Other useful commands are `Print` and `Spell`. For a complete list of choices, you can hit `C-c C-c` and hit the tab key. Note that you can use tab-completions for the options.

Maketitle

The `\maketitle` command produces a title heading for the beginning of your document, assuming you have already specified the title and author (as in this example). What if you want a particular date, instead of the default of today? You can specify a date with

```
\date{September 1, 2000}
```

You could also specify the current date with `\today`, but that isn't necessary since the current date is the default. `\maketitle` is also useful in conjunction with `\begin{titlepage}` and `\begin{abstract}`.

Numbered and Unnumbered Sections

You will note that all the sections and subsections in the example document are numbered, but in this lab they are not. If you didn't want numbered sections, you can instead use `\section*` and `\subsection*`. Try changing the example and re-compiling to see the effect.

Type Styles

In the itemized section (the two bulleted items), the items are printed in a slanted font, because of the `\sl` command just before the `\begin{itemize}`. Instead of slanted, other choices include `\it` (*italic*), `\em` (*emphasis*), and `\sc` (SMALL CAPS). The style is returned to normal with `\rm`, which stands for Roman font family. Other families include `\sf` (sans serif) and `\tt` (`typewriter text`).

Often a better way to change styles is to enclose the whole affected area by curly braces, as was done for the text *margin of error* in the subsequent paragraph. The curly braces clearly denote the section of text affected by the change in style.

Numbered List

The list of points has bullets. You can get LaTeX to automatically number the items by using `\begin{enumerate}` instead of `\begin{itemize}`. Try changing this, and note that you also need to change the `\end{itemize}`. You can also specify your own index scheme or symbols for the items by explicitly including the label in square brackets after the `\item`, e.g., `\item[(i)]`.

Comments

Comments are started by the percent sign (%). Anything that appears on a line after a % is ignored by LaTeX. This can be helpful when making changes to a document. But this is why you must always remember to put a backslash before a percent sign when you want a percent sign in your text, i.e., `\%`. Try removing the backslash in the example file before the percent sign of `4\% margin of error`, recompile, and see what happens.

Fractions

At the end of Section 1.2 are two fractions, 19/200 and 9/200. Normally this method of displaying fractions (i.e., with a solidus (the slash) instead of a horizontal bar) is preferred in text, since it doesn't mess with the spacing of the lines. However, sometimes you might prefer to express the fractions with a horizontal bar, e.g., $\frac{19}{200}$. To do this, you need to invoke math mode with a single dollar sign and then use the `\frac{}{}` command, which takes two arguments, the numerator and the denominator. Try changing 19/200 to `$\frac{19}{200}$` and see how it changes the fraction. In this case, it may not have much effect on the spacing between that line and the previous line (it's at the end of a paragraph), but a more complicated fraction (like `$\frac{1+\frac{1}{1+\frac{1}{19}}}{1+\frac{1}{200}}$`) may have a more noticeable effect.

Adding Space

In Section 1.3, just before the table, you will see a `\medskip` command. This adds a medium amount of vertical space before the table. Try commenting it out and recompiling to see what happens.

You will probably not notice any difference after commenting out the `\medskip` command. Why does this happen? In this case, it is probably because LaTeX automatically stretches spaces on the page so the that the top and bottom lines of the page are in the same place on every page (vertical justification). So it ends up stretching the space above the table by about the same amount as a `\medskip` would do, and the `\medskip` is unnecessary here. However, if you were to make some other changes to the document, it might affect the rest of the spacing and the `\medskip` might become important again.

Other shortcut space-adding commands are `\smallskip` and `\bigskip`. Instead of such shortcuts, you can specify the amount of space directly with `\vspace{}`. The argument for `\vspace` is the amount of space, either in inches (in), centimeters (cm), or points (pt), e.g. `\vspace{1.0in}`. Try adding a full inch of space instead of `\medskip` and see how it affects the document.

Horizontal space can be added with `\hspace{}`. `\quad` adds an amount of space equal to the current font size (e.g. 10 points of space for a 10-point font size), as demonstrated in the table.

Large Delimiters

In Section 2.1 there is the equation

$$p(X|\theta) = p(x_1, \ldots, x_n|\theta) = \prod_{i=1}^{n} p(x_i|\theta) = \theta^y(1-\theta)^{n-y}$$

Suppose we wanted to put large square brackets around the whole expression being multiplied, i.e.,

$$p(X|\theta) = p(x_1, \ldots, x_n|\theta) = \prod_{i=1}^{n} [p(x_i|\theta) = \theta^y(1-\theta)^{n-y}]$$

Note that if you just use `[` and `]`, they do not stretch vertically to fit the whole expression. Try it, and see that the final $n-y$ exponent doesn't really fit inside the `]`. The fix is to use `\left[` and `\right]`. This tells LaTeX to stretch the delimiters to encompass the expression. This can also be used with `\left(` and `\left\{`. Note that to get a curly brace to print, you must use `\{`, since the curly brace is used within LaTeX to bound action areas. You can mix and match your `\left` and `\right`, as long as you have one of each. You can also use `\left.` to match any `\right` but to not print anything (since they must be balanced). For example, you can re-write the previous equation without the `\begin{cases}` command as:

```
p(x_i|\theta) = \theta^{x_i}(1-\theta)^{1-x_i} = \left\{ \begin{array}{ll}
                                                 \theta& \mbox{if $x_i=1$,} \\
                          1-\theta& \mbox{if $x_i=0$.} \end{array} \right.
```

Note that and `array` is just like a `tabular`, except that it must be used in math mode, and everything inside is automatically in math mode. Also note that `\mbox{}` is a method of including regular text (not in math mode) inside of math mode. Try the above expression so you can see all of the parts in action.

Numbered Equations

None of the equations in this section are numbered. The ones which appear on a line by themselves are typeset with `$$` in this example. You could equally use `\[` and `\]`, or `\begin{equation*}` and `\end{equation*}`. If you wanted the equations to be numbered, use `\begin{equation}` and `\end{equation}`. This is analogous to the difference between `\begin{section}` and `\begin{section*}`. Try replacing several of the `$$` (replace in pairs, since everything must match in LaTeX) and see how LaTeX numbers the equations. This can be useful for referring to the later on. You can get LaTeX to make an automatic reference with the `\label{}` and `\ref{}` commands.

Printing

Printing is done with `dvips`. This unix command will create a postscript (`.ps`) file from the `.dvi` file. On some systems, it will send this file directly to the printer, and not save anything to disk unless you explicitly tell it to with the `-o` switch. In ISDS, if you print from Emacs, it should send it directly to a printer, which one will depend on your setup files. In ISDS if you use dvips from the unix prompt, it will only create a postscript file, and not send anything to the printer. You can then print by using `lpr` on the `.ps` file.

You can print two pages in smaller print one one piece of paper (sideways) with the `psnup` command. One way to do this at the unix prompt is:

```
dvips example.dvi -f | psnup -n 2 | lpr -P211
```

where the printer name here is `211`. From within Emacs, you can edit the default print line to match the above.

Note that `dvips` has many options. Important ones include those that specify a print range, such as `-p`, `-n`, and `-f`. For more info, see the `man` page for `dvips`.

Finishing Up

When you are done, make sure to close any `xdvi` windows and exit emacs before logging off. Failure to do so could spawn runaway processes.