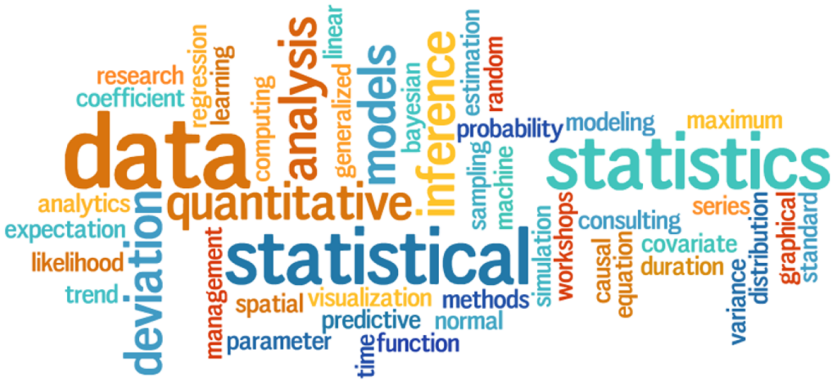
Introduction to LateX II

Intermediate LaTeX Uses

UVA Library StatLab
University of Virginia
523 Alderman Library

February 9, 2014



1. Displayed Text
   - Font Size
   - Text Font Attributes
   - Font Alignment
2. Spacing
   - Arbitrary Spacing
   - Paragraph Spacing
   - Paragraph Indentation
   - Page Breaks
3. Tables
4. Lists
5. Creating Shortcuts
6. More Math Formats
   - Math Font Attributes
   - Matrices
   - Theorem-like Environments
7. Including R Code

Introduction

In this presentation, two LaTeX instructions will be used

command an instruction to LaTeX to do something special, e.g. change the current typeface

1. single character, e.g. `#, $, &`
2. bachslash plus a non-letter character, e.g. `\=`
3. backslash plus a sequence of letters, e.g. `\large`

environment has effect that the text within it is treated differently according to environment parameters, e.g.`\begin{environment} text \end{environment}`

Choice of font size

The following declarations are available for changing the font size. These commands can be declared by themselves without `{text}`, such as `\small` but then all displayed text after the command will be of that size until another size command is declared.

		size relative to font of		
command	actual size	[10pt]	[11pt]	[12pt]
`\normalsize{text}`	normal	10	11	12
`\small{text}`	small	9.2	10	11
`\footnotesize{text}`	smaller	8.5	9.2	10
`\scriptsize{text}`	very small	8	8.5	8.5
`\tiny{text}`	smallest	6.8	7.3	7.3

Choice of font size

command	actual size	size relative to font of [10pt]	[11pt]	[12pt]
`\normalsize{text}`	normal	10	11	12
`\large{text}`	large	11.7	11.7	14.1
`\Large{text}`	larger	14.1	14.1	15.8
`\LARGE{text}`	even larger	15.8	15.8	19.0
`\huge{text}`	still larger	19.0	19.0	22.8
`\Huge{text}`	largest	22.8	22.8	22.8

You can also use any of these commands as environments, for instance

```
\begin{small}
    text
\end{small}
```

Text Font attributes

The size of the font is only one of several attributes that may be used to change/describe text.

Family Style describes the overall style of the font.

Family Style		
`\textrm{text}` or `\rm{text}`	Roman font	Text Sample
`\texttt{text}` or `\tt{text}`	Typewriter font	`Text Sample`
`\textsf{text}` or `\sf{text}`	Sans Serif font	Text Sample

Shape describes the form of the font.

Shape Form		
`\textup{text}`	Upright font	Text Sample
`\textit{text}` or `\it{text}` or `\emph{text}`	Italic shape	*Text Sample*
`\textsc{text}` or `\sc{text}`	All Caps	TEXT SAMPLE

Shape describes the form of the font.

Shape Form		
`\textup{text}`	Upright font	Text Sample
`\textit{text}` or `\it{text}` or `\emph{text}`	Italic shape	*Text Sample*
`\textsc{text}` or `\sc{text}`	All Caps	TEXT SAMPLE

Series describes the width and/or boldness of the font

Series Width		
`\textmd{text}`	medium	Text Sample
`\textbf{text}` or `\bf{text}`	bold face	**Text Sample**

Shape describes the form of the font.

Shape Form		
`\textup{text}`	Upright font	Text Sample
`\textit{text}` or `\it{text}` or `\emph{text}`	Italic shape	*Text Sample*
`\textsc{text}` or `\sc{text}`	All Caps	TEXT SAMPLE

Series describes the width and/or boldness of the font

Series Width		
`\textmd{text}`	medium	Text Sample
`\textbf{text}` or `\bf{text}`	bold face	**Text Sample**

Besides reseting the changed attribute to the defaults, to change the font back use `\textnormal{text}`

Text Alignment

Now that you knowing how to change the text characteristics, we can switch over to use of empty space in a document.

```
\begin{center} line 1 \\  ... \\ line n \end{center}
\begin{flushleft} line 1 \\ ... \\ line n \end{flushright}
\begin{flushright} line 1 \\ ... \\ line n \end{flushright}
```

Spacing

Any of the following commands insert an arbitrary amount of blank space within a document at the point in the text where it appears

Command	Description
`\hspace{space}`	creates horiz. empty space of length `space`
`\quad`	creates 10pt typeface of horiz. empty space
`\qquad`	creates 20pt typeface of horiz. empty space

Spacing

Any of the following commands insert an arbitrary amount of blank space within a document at the point in the text where it appears

Command	Description
`\hspace{space}`	creates horiz. empty space of length space
`\quad`	creates 10pt typeface of horiz. empty space
`\qquad`	creates 20pt typeface of horiz. empty space
`\hspace{\fill}` or `\hfill`	Inserts enough space to force the text on either side to be pushed over to the left and right margins
multiple `\hfill` in one line	inserts the same amount of spacing so that the line becomes left and right justified

Spacing

Any of the following commands insert an arbitrary amount of blank space within a document at the point in the text where it appears

Command	Description
`\hspace{space}`	creates horiz. empty space of length space
`\quad`	creates 10pt typeface of horiz. empty space
`\qquad`	creates 20pt typeface of horiz. empty space
`\hspace{\fill}` or `\hfill`	Inserts enough space to force the text on either side to be pushed over to the left and right margins
multiple `\hfill` in one line	inserts the same amount of spacing so that the line becomes left and right justified
`\dotfill`	inserts .....
`\hrulefill`	inserts _____

Paragraph spacing

The default spacing between paragraphs can be changed in the preamble by using

```
\setlength{\parskip}{space}
```

Paragraph spacing

The default spacing between paragraphs can be changed in the preamble by using

```
\setlength{\parskip}{space}
```

Extra vertical spacing can also be added individually between paragraphs by using

```
\vspace{space}
```

Using `\vspace*{space}` will add the extra space even when a new page occurs or when the command appears at the top of a new page. Note same *-form can be used on `\hspace{space}`.

Paragraph Indentation

While the command \\ creates a newline or paragraph, no indentation occurs. If indentation is desired you can either

- add an extra blank line between the paragraphs in the code
- force an indentation by inserting `\indent` at the beginning of the paragraph
  - To suppress an indentation for a paragraph just have `\noindent` at the beginning of the paragraph

Paragraph Indentation

While the command \\ creates a newline or paragraph, no indentation occurs. If indentation is desired you can either

- add an extra blank line between the paragraphs in the code
- force an indentation by inserting `\indent` at the beginning of the paragraph
  - To suppress an indentation for a paragraph just have `\noindent` at the beginning of the paragraph

Similar to the default spacing between paragraphs you can also change the indentation spacing in the preamble by using

```
\setlength{\parindent}{space}
```

Page and Line Breaks

While breaking text into pages or lines occurs automatically in LaTeX, you may want to interfere with the program and tell it where a break should take place. To do this use

`\pagebreak`	`\linebreak`
`\nopagebreak`	`\nolinebreak`

Page and Line Breaks

While breaking text into pages or lines occurs automatically in LaTeX, you may want to interfere with the program and tell it where a break should take place. To do this use

`\pagebreak`	`\linebreak`
`\nopagebreak`	`\nolinebreak`

Usage of `\pagebreak`

- If it occurs between two paragraphs, a new page will be forced at that point
- If it occurs within a paragraph, the new page will be implemented after the current line is completed

To properly end a page in the middle and have the remaining of the page filled with empty space use `\newpage`

To properly end a page in the middle and have the remaining of the page filled with empty space use `\newpage`

If the document class is using the option `twocolumn` then

- `\pagebreak` and `\newpage` end the current column and begin a new one
- `\clearpage` terminates the current page, inserting an empty right column if necessary

Tables

Tables can be created using two different environments

```
\begin{array}[pos]{cols}    rows    \end{array}
\begin{tabular}[pos]{cols}  rows   \end{tabular}
```

- Both of these environments' syntax and parameters are the same.
- The array environment can only be used in mathematical mode (e.g. `\[ \]`)

Large tables can be inserted into LaTeX from excel by using an add-on called "excel2latex" (`http://www.ctan.org/tex-archive/support/excel2latex/`). You would still have to mainly change the table aspects in LaTeX however.

Tables

Tables can be created using two different environments

```
\begin{array}[pos]{cols}    rows    \end{array}
\begin{tabular}[pos]{cols}  rows   \end{tabular}
```

- Both of these environments' syntax and parameters are the same.
- The array environment can only be used in mathematical mode (e.g. `\[ \]`)

Large tables can be inserted into LaTeX from excel by using an add-on called "excel2latex" (`http://www.ctan.org/tex-archive/support/excel2latex/`). You would still have to mainly change the table aspects in LaTeX however.

- `pos` is the vertical positioning argument. It can take the values
  - `t` – top line of the talbe is aligned with the baseline of the current external line of text
  - `b` – bottom line of the table is aligned with the external baseline
  - no value defined – table is centered on the external baseline

- `cols` is the column formatting argument, in which there must be an entry for every column (extra entries can be added for the left and right borders or intercolumn spacing). It can take the following values
  - column formatting
    - `l` – column contents are left justified
    - `r` – column contents are right justified
    - `c` – column contents are centered
    - `*{num}{cols}` – column format in `cols` is reproduced `num` times
      (e.g. `*{3}{c}` is same as `c c c`)
  - formatting symbols
    - `|` – draws vertical line
    - `||` – draws two vertical lines next to each other
    - `@{text}` – inserts `text` in every line of the table between the two columns where it appears
      (e.g. `@{\hspace{0.5in}}` adds extra spacing)
    - `>{$}l<{$}` – causes all entries in that column to be math mode (allows you not to have to declare math mode for each entry individually)

- `rows` contain the actual entries in the table, each horizontal row being terminated with \\. Each column in each row is sperated by the & symbol and each row must contain the same number of entries as the number of columns defined in `cols`. (Entries can be empty)

- `rows` contain the actual entries in the table, each horizontal row being terminated with \\. Each column in each row is sperated by the & symbol and each row must contain the same number of entries as the number of columns defined in `cols`. (Entries can be empty)
  - `\hline` – draws a horizontal line the full width of the table below the row that was just ended (\\) or at the top of the table, depending on placement. (Two together creates two horizontal lines with a little space between them)
  - `\cline{n-m}` – draws a horizontal line from the left side of column `n` to the right side of column `m`.
  - `\multicolumn{num}{col}{text}` – combines the following `num` columns into a single column with their total width including intercolumn spacing. Only used at the start of a row or right after a column separation symbol &
  - `\vline` – draws a vertical line with the height of the row at the location where it appears.

Lists

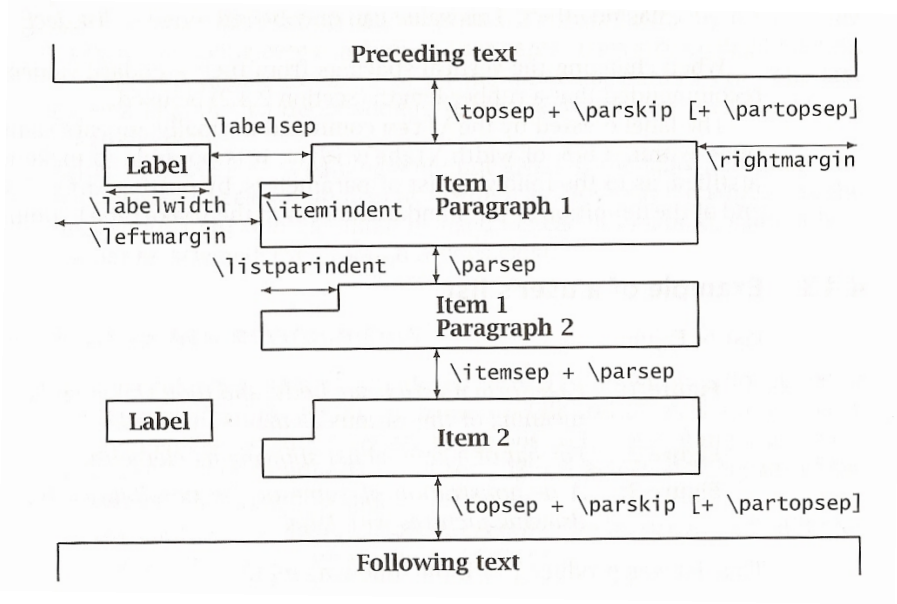
There are 3 environments available for producing formatted lists

1. `\begin{itemize} list text  \end{itemize}`
2. `\begin{enumerate} list text  \end{enumerate}`
3. `\begin{description} list text  \end{description}`

In which, each new entry should begin it's line with `\item` for itemize and enumerate environments and `\item[label]` for description environment.

Any of the above lists can be nested within one another, consisting of either mixed lists or of one type, to the depth of four levels.

Just like all other environments, lists have `[options]` one can list after the beginning declaration to format their appearance.



The most commonly used are

- `\labelsep`
- `\itemindent`
- `\parsep`
- `\topsep`

Creating Your Own Commands

Often it is helpful to create your own commands that either are entirely new or is a shorthand notation for a commonly used command/environment. The general setup is

```
\newcommand{name}{definition}
```

which needs to be created in the preamble of the text file.

- `name` is the code you will declare in the text to call the command
- `definition` is what will be outputted by calling the command or the original environment that was shorthanded

Math Font Attributes

Using the `amsfonts` package, one can use the following environments on any letters, numbers, and uppercase Greek symbols in math mode.

Command	Sample	Description
`\mathnormal{...}`	$ABCDabcd1234$	default math font
`\mathrm{...}`	$\mathrm{ABCDabcd1234}$	default normal font
`\mathit{...}`	$\mathit{ABCDabcd1234}$	italicised font
`\mathbf{...}`	$\mathbf{ABCDabcd1234}$	bold font
`\mathsf{...}`	$\mathsf{ABCDabcd1234}$	Sans-serif font
`\mathtt{...}`	$\mathtt{ABCDabcd1234}$	Monospace (fixed width) font
`\mathcal{...}`	$\mathcal{ABCDEF}$	Calligraphy (uppercase only)
`\mathfrak{...}`	$\mathfrak{ABCDabcd1234}$	Fraktur font
`\mathbb{...}`	$\mathbb{ABCDEF}$	Blackboard bold

To bold lowercase Greek and other symbols use `\boldsymbol`.

Regular text can also be added in a math environment by use `\text{...}` from the `amsmath` package.

Matrices

Matrices can be made using one of many `matrix` environment (in the `amsmath` package) inside the math environment (e.g. `\[ \]`)

This environment is very similar to the `tabular` environment, where entries are specified by row, in which columns are separated using an ampersand (`&`) and rows are seperated with a double backslash (`\\`)

Environment	Surrounding Delimiter
`matrix`	
`pmatrix`	`( )`
`bmatrix`	`[ ]`
`Bmatrix`	`{ }`
`vmatrix`	`| |`
`Vmatrix`	`|| ||`

The default centers columns by default, using the *-form allows you to sepcify alignment of columns

Theorem-like Environments

There are several theorem-like environments that you can create in LaTeX (such as Theorem, Definition, etc.). Use the following command in the preamble

```
\newthorem{env_name}[numbered_like]{caption}[within]
```

- `env_name` is the name of the environment to be defined
- `numbered_like` is is the name of a defined theorem-like environment (used if want two different environments be numbered together)
- `caption` is the caption or text to be printed before the number, e.g. Theorem, Lemma, etc.
- `within` is the name of an already defined counter (e.g. section)

Theorem-like Environments

There are several theorem-like environments that you can create in LaTeX (such as Theorem, Definition, etc.). Use the following command in the preamble

```
\newthorem{env_name}[numbered_like]{caption}[within]
```

- `env_name` is the name of the environment to be defined
- `numbered_like` is is the name of a defined theorem-like environment (used if want two different environments be numbered together)
- `caption` is the caption or text to be printed before the number, e.g. Theorem, Lemma, etc.
- `within` is the name of an already defined counter (e.g. section)

Then in the body you can define the environment by

```
\begin{env_name}[additional_caption]
    text
\end{env_name}
```

Furthermore, one can use the `amsthm` package and change the style of a theorem-like environment from the default setting by using

```
\theoremstyle{style}
```

before defining a theorem-like environment.

This package has 3 predefined styles

- `definition` – fat title, roman body
- `plain` – fat title, italic body
- `remark` – italic title, roman body

If the default style of a theorem-like environment isn't desired you can define your own style, using the `amsthm` package and `\newtheoremstyle` command

```
\newtheoremstyle{stylename}% name of the style to be used
  {spaceabove}% measure of space to leave above the theorem
  {spacebelow}% measure of space to leave below the theorem.
  {bodyfont}% name of font to use in the body of the theorem
  {indent}% measure of space to indent
  {headfont}% name of head font
  {headpunctuation}% punctuation between head and body
  {headspace}% space after theorem head;
  {headspec}% Manually specify head
```

Inputting R code

There are several different ways in which one can input R code (or any other computer program code) into the document.

Depending on how much functionality you actually need, one of the simpliest ways is to use the verbatim environment.

- ```
  \begin{verbatim}
      Code Here
  \end{verbatim}
  ```
- `\verb+code+`
- `\verbatiminput{Filename.R}`

In any of these cases, anything in the verbatim environment is not processed in the usual manner but is outputted directly as typed into the pdf in Monospace font.
  ```

Other useful packages

One more way to integrating R code and LaTeX is to use

- R Sweave
- knitr package in R
- listing package

Other useful things to look into dealing with LaTeX is

- beamer (used to create slides for a presentation)
- bibtex (used to create a bibliography)

Kopka, Helmut, and Patrick W. Daly. A Guide to LATEX: Document Preparation for Beginners and Advanced Users. Harlow, England: Addison-Wesley, 1999. Print.

"LaTeX." - Wikibooks, Open Books for an Open World. Wikibooks, 25 Oct. 2013. Web. 16 Jan. 2014. <http://en.wikibooks.org/wiki/LaTeX>.