



Learning Latex before it's too late!

Session 3

Andrew Roberts

andyr@comp.leeds.ac.uk



Overview

- Tables
 - Structure
 - Formatting
 - Spanning rows and columns
- Images
 - Importing
 - Simple operations
- Floats
 - Dynamically positioning tables and images
 - Captions



Tables – Basics

- Example:

```
\begin{tabular}{ c | c }  
  \hline  
  Item 1 & Item 2 \\  
  Item 3 & Item 4  
\end{tabular}
```



Tables – Basics (2)

- Use the *tabular* environment.
- Argument specifies alignment of columns.
- Vertical bars in between column declaration for vertical lines.
- `\\` (new line) – ends the current row.
- `\hline` – draws an horizontal line.



Tables – Wrapping Text

- Latex doesn't automatically wrap text if it overruns the page!
- Use paragraph column types:
 - `p{width}` – top aligned
 - `m{width}` – middle aligned
 - `b{width}` – bottom aligned... on columns that require it.



Tables – Spanning Columns

- `\multicolumn{num_cols}{alignment}{contents}`
 - `num_cols` – number of subsequent columns to span
 - `alignment` – pretty obvious!
 - `contents` – the data



Tables – Spanning Rows

- Use the *multirow* package.
- `\multirow{num_rows}{width}{contents}`
 - `num_rows` – number of subsequent rows to span.
 - `width` – use `*` for natural width, otherwise specify a length.
 - `contents` – the data.



Images

- Many ways to do this. The *graphicx* package is my preferred method.
- Images should be in Encapsulated Postscript (EPS) format.
 - Can easily convert to from majority of graphics utilities, eg Paint Shop Pro, Photoshop, Corel Draw, etc.



Importing Images

- `\includegraphics[options]{imagefile}`
 - Can omit the file extension.
- Options:
 - `width=xx` (e.g., 5cm)
 - `height=xx` (e.g., 2.5in)
 - `keepaspectratio=true|false`
 - `scale=xx` (e.g., 0.5)
 - `angle=xx` (e.g., 90)
 - `trim=l b r t`
 - `crop=true|false`



Floats

- Floats are elements that cannot be broken over a page (e.g., tables and figures).
- Floats are positioned by Latex at the most suitable spot.
- Will rearrange text accordingly.



Floating Tables and Figures

- Simply put your table (tabular) into a *table* environment.
- Similarly, to float material that constitutes a figure, embed in the *figure* environment.



Placement Specifiers

- Options that give user control of where floats are placed.

Specifier	Permission
h	Place the float <i>here</i> .
t	Position at top of page.
b	Position at bottom of page.
p	Place on a special page of floats only.
!	Override internal parameters for 'good' positioning.



Captions

- Use the `\caption{text}` command to add a caption to your floats.
- To ensure it appears after the float, use the command after the element being floated.
- Latex will take care of the counting.
 - Each float type has its own independent counter.



Labels and Cross-refs

- To cross reference (eg figure, section, etc) you must first add a label within the thing to be referenced:

```
\label{marker}
```

- To then reference:

```
\ref{marker}
```

(`\pageref{marker}` is also useful)



Other tips

- Quotes:

`single quotes' ‘single quotes’

` `double quotes” “double quotes”

- Dashes:

- inter-word

-- page range

--- punctuation dash



Accents

Command	Output	Command	Output
<code>\' {o}</code>	ò	<code>\' {o}</code>	ó
<code>\" {o}</code>	ö	<code>\H {o}</code>	õ
<code>\^ {o}</code>	ô	<code>\~ {o}</code>	õ
<code>\v {o}</code>	õ	<code>\= {o}</code>	ō
<code>\b {o}</code>	ò	<code>\. {o}</code>	ò
<code>\d {o}</code>	ò	<code>\c {o}</code>	ç
<code>\r {o}</code>	ö	<code>\t {oo}</code>	ö
<code>\i</code>	ı		



Babel

- The babel package is responsible for Latex language support (US is default), including date format.

```
\documentclass[english]{article}  
\usepackage{babel}
```

Will format auto-generated dates in day/month/year order.



More...

- Goto:

<http://www.comp.leeds.ac.uk/andyr/misc/latex>

and

<http://www.comp.leeds.ac.uk/andyr/misc>