

---

# ***T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X***

David J. Scott

`d.scott@auckland.ac.nz`

Department of Statistics, University of Auckland

# Outline

---

- What is T<sub>E</sub>X?
- Some examples
- Document Structure
- Environments
- Typesetting text
- Typesetting mathematics
- Tables and arrays
- Packages
- Graphics
- Customising L<sup>A</sup>T<sub>E</sub>X
- Slides using BEAMER

# Resources

---

- CTAN (Comprehensive T<sub>E</sub>X Archive Network), local mirror  
<http://elena.aut.ac.nz/ctan/>
- The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>  
[www.stat.auckland.ac.nz/~dscott/782/lshort.pdf](http://www.stat.auckland.ac.nz/~dscott/782/lshort.pdf)
- Short Math Guide for L<sup>A</sup>T<sub>E</sub>X  
[www.stat.auckland.ac.nz/~dscott/782/short-math-guide.pdf](http://www.stat.auckland.ac.nz/~dscott/782/short-math-guide.pdf)
- Tutorials and slides by Andrew Roberts  
[www.comp.leeds.ac.uk/andyr/misc/latex/index.html](http://www.comp.leeds.ac.uk/andyr/misc/latex/index.html)
- AUCTeX Manual  
[www.stat.auckland.ac.nz/~dscott/782/auctex.pdf](http://www.stat.auckland.ac.nz/~dscott/782/auctex.pdf)
- AUCTeX Reference Card  
[www.stat.auckland.ac.nz/~dscott/782/tex-ref.pdf](http://www.stat.auckland.ac.nz/~dscott/782/tex-ref.pdf)

# Resources

---

- Packages in the 'graphics' bundle (*grfguide.pdf*)  
[www.stat.auckland.ac.nz/~dscott/782/grfguide.pdf](http://www.stat.auckland.ac.nz/~dscott/782/grfguide.pdf)
- Using EPS Graphics in  $\text{\LaTeX} 2_{\epsilon}$   
[www.stat.auckland.ac.nz/~dscott/782/epslatex.pdf](http://www.stat.auckland.ac.nz/~dscott/782/epslatex.pdf)
- Manual for the Prosper Class  
[www.stat.auckland.ac.nz/~dscott/782/prosper-doc.pdf](http://www.stat.auckland.ac.nz/~dscott/782/prosper-doc.pdf)
- HA-prosper package: Documentation  
[www.stat.auckland.ac.nz/~dscott/782/HA-prosper.pdf](http://www.stat.auckland.ac.nz/~dscott/782/HA-prosper.pdf)
- BEAMER Homepage  
<http://latex-beamer.sourceforge.net/>
- Most of what is needed is contained in:  
*The Not So Short Introduction to  $\text{\LaTeX} 2_{\epsilon}$*

# What are $T_{E}X$ and $L_{A}T_{E}X$ ?

- $T_{E}X$  is a program for computerised typesetting of documents
- $L_{A}T_{E}X$  is a macro package which makes many tasks easier and which uses  $T_{E}X$  to do the actual typesetting
- $L_{A}T_{E}X 2_{\epsilon}$  is the current version of  $L_{A}T_{E}X$ ,  $L_{A}T_{E}X 3$  is the version currently being developed.
- Different to word processors which are WYSIWYG (or almost WYSIWYG)
- A  $T_{E}X$  file is ASCII: it is processed by  $T_{E}X$  to produce a dvi file
- A dvi (device independent) file is an electronic version of a typeset document
- A dvi file can be displayed or printed (e.g. with xdvi) or processed further to postscript or pdf

---

# *Examples of L<sup>A</sup>T<sub>E</sub>X Files*

# A Sample File

- Consider the file `hello.tex`

```
% hello.tex - Our first Latex example!  
\documentclass{article}  
\begin{document}  
Hello World!  
\end{document}
```

- If AUCT<sub>E</sub>X is loaded in XEmacs, there will be two extra menus:  
**LaTeX** and **Command**
- To ensure AUCT<sub>E</sub>X is loaded automatically it may be necessary to edit the file `init.el` in the directory `.xemacs`  
Add `(require 'tex-site)` to the end of this file

# Making a .dvi and a .ps File

---

- Turn into a dvi file with either
  - On the command line `latex hello.tex` or `latex hello`
  - In XEmacs `C-c C-c`
- I usually convert it to postscript using a binary file or a Makefile
- View with
  - On the command line `xdvi hello&` or
  - In XEmacs `C-c C-c` again
- To turn into postscript and view either
  - On the command line  
`dvips -o hello.ps hello.dvi;gv hello.ps&`
  - In XEmacs `C-c C-c File` followed by `gv hello.ps&` on the command line
  - I have commands added to the **Command** menu



# Some More Detail

- A more complex file

```
\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Some Interesting Words}
Well, and here begins my lovely article.
\section{Good Bye World}
\ldots{} and here it ends.
\end{document}
```

# *Even More Detail*

---

- Even more detail is found in the sample file from Rob Hyndman  
`RHsample.txt`
- Specially created to illustrate many aspects of  $\text{\LaTeX}$
- No tables or graphs however
- Examine the typeset result and compare with the tex file

---

# ***Basic Document Layout***

# Basic Document Layout

The basic structure of a  $\text{\LaTeX}$  document is always:

```
\documentclass[ options ]{ format }  
  document preamble  
\begin{document}  
  document body  
\end{document}
```

- Most common *format* specification is “article”  
Other possible options are “report,” “book” and “letter.”
- The *options* specification consists of a comma-separated list of options.
- The most common of these are the page size (usually “a4paper” or “letterpaper”) and the font size (one of “10pt,” “11pt” or “12pt”)

# Document Preamble

---

- The preamble contains declarations affecting structure and appearance.
- It usually contains a number of extension package inclusions of the form:

```
\usepackage{ package-name }
```

- For example, the statement

```
\usepackage{times}
```

causes the document to be typeset using a *Times-Roman* font, rather than the default *Computer-Modern* font.

# *Document Body*

---

- The region between `\begin{document}` and `\end{document}` is called the document body
- It contains the text to be typeset together with the  $\text{\LaTeX}$  markup instructions

# *XEmacs Shortcuts*

---

- If you open a new  $\text{\LaTeX}$  document with XEmacs, you can create the basic document structure by typing the keystrokes `C-c C-e` (insert environment)
- The minibuffer at the bottom of the frame will then contain the line:  
`Environment type: (default document)`  
which is asking you to verify that the environment to be inserted is a the basic template
- Confirm that this is what you want by pressing the enter key.
- You will then be asked to specify your desired format (the default is `article`) and a list of options.

# Sectioning

---

- An article consists of a number of sections (and subsections), each preceded by a header
- Sections are declared with the  $\text{\LaTeX}$  declaration  
`\section{ section-name }`
- Subsections are declared with a `subsection` declaration
- These declarations result in numbered sections and subsections.
- There are also `*`-variants which can be used to produce unnumbered sections and subsections.

`\section*{ section-name }`



# Section Labelling

- Numbered sections and subsections can be labelled so that they can be referenced from elsewhere in the document
- Follow the section or subsection declaration by a `label` declaration
- Labelled sections can be referenced from anywhere in the document using their label
- When the document is typeset, the label is replaced by the section number

```
\section{An Important Section}
```

```
\label{sec:important}
```

```
...
```

```
In section \ref{sec:important} it was  
proposed ...
```

# ***XEmacs Shortcuts***

---

- In XEmacs, you can insert a section or subsection by typing  
C-c C-s
- You will then be asked what level of sectioning you desire (section, subsection, subsection) and for the section title and label
- If you don't want a label, just type the Enter key when asked

# Font Changes

---

- L<sup>A</sup>T<sub>E</sub>X provides good control over font changes
- There are declarations which will emphasize text, set it in italics or bold-face or switch to sans-serif or typewriter typeface
- The most common declaration is emphasis  
`This results in the \emph{emphasized} word ...`
- This results in the *emphasized* word being typeset in italics.

# Available Typefaces

---

Other font options are:

plain(roman)text      `\textrm{ ... }`

**boldface text**      `\textbf{ ... }`

*italic text*      `\textit{ ... }`

*slanted text*      `\textsl{ ... }`

SMALL CAPITALS      `\textsc{ ... }`

sans serif text      `\textsf{ ... }`

typewriter text      `\texttt{ ... }`

# Flexibility

---

- These font declarations can be nested, so that it is possible to obtain a switch to ***bold-italic text*** with the declaration  
`\textbf{\textit{bold-italic text}}`
- Be warned, however, that it is bad typesetting practice use lots of font changes
- Font changes are distracting and make your documents harder to read

# ***XEmacs Shortcuts***

- XEmacs makes it very easy to change fonts.

*emphasized text*

C-c C-f C-e

plain(roman)text

C-c C-f C-r

**boldface text**

C-c C-f C-b

*italic text*

C-c C-f C-i

*slanted text*

C-c C-f C-s

SMALL CAPITALS

C-c C-f C-c

sans serif text

C-c C-f C-f

typewriter text

C-c C-f C-t

- You can switch fonts before you type the text
- Or you can change existing text by selecting it with the mouse before typing the keystrokes above

---

# ***Environments***

# *Environments*

---

- $\text{\LaTeX}$  *environments* are used to provide many kinds of displays. The most common environments are:
  - Numbered lists
  - Bullet lists
  - Centered and justified displays
  - Quotations
  - Tables and arrays
  - Equations
- Users can also define their own environments



# *Bullet Point Lists*

---

A bullet point list is provided by the `itemize` environment.

```
\begin{itemize}
\item The first item,
\item the second item, and
\item the third item.
\end{itemize}
```

This results in

- The first item,
- the second item, and
- the third item.

# Numbered Lists

---

Numbered lists are provided by the `enumerate` environment.

```
\begin{enumerate}  
\item The first item,  
\item the second item, and  
\item the third item.  
\end{enumerate}
```

This results in

1. The first item,
2. the second item, and
3. the third item.

# *Nested Environments*

- Environments can be nested within each other
- It is possible to have enumerated lists nested within enumerated lists
- A different style of numbering is used within each list

```
\begin{enumerate}
\item
  \begin{enumerate}
    \item The first item,
    \item the second item, and
    \item the third item.
  \end{enumerate}
\item the second item, and
\item the third item.
\end{enumerate}
```

# *Nested Environments*

---

This will produce

1. (a) The first item,  
    (b) the second item, and  
    (c) the third item.
2. the second item, and
3. the third item.

# ***XEmacs Shortcuts***

---

- In XEmacs, environments are inserted by typing `C-c C-e`
- This results in you being queried for the type of environment
- You should respond by typing either `itemize` or `enumerate`
- You can then type the items
- To introduce an item type `M-RET` (the Alt and Return keys pressed together) or `ESC RET` (Escape followed by Return)

# Centered Displays

- Centered displays are created with the *center* environment
- Lines within the display are individually centered

```
\begin{center}
  Here is the first line to be centered, \\
  here is the second \\
  and here is the last.
\end{center}
```

- This results in the  $\text{\LaTeX}$  output:

Here is the first line to be centered,  
here is the second  
and here is the last.

# *Flushleft and Flushright Displays*

- These two display types are similar to the center environment
- They result in the lines they contain being aligned with the left and right margins

```
\begin{flushleft}
```

```
lines to be \\ aligned with the left margin
```

```
\end{flushleft}
```

```
\begin{flushright}
```

```
lines to be \\ aligned with the right margin
```

```
\end{flushright}
```

- This results in the following output:

lines to be

aligned with the left margin

lines to be  
aligned with the right margin

# Quotations

- The *quote* environment provides quoted paragraphs
- The material in the environment is typeset with slightly indented margins and offset from the surrounding text

```
\begin{quote}
```

```
Short quotes can be included between  
quotation marks, but longer quotes  
should be included within a ``quote''  
environment.
```

```
\end{quote}
```

- This results in the output:

Short quotes can be included between quotation marks, but longer quotes should be included within a “quote” environment.



# Other Quotation Environments

- There are two other environments for providing quotation: *quotation*, which indents paragraphs (quote does not) and *verse*, which can be used for poetry.

```
\begin{verse}
```

```
A person who eats meat wants to  
get his teeth into something. \\
```

```
A person who does not eat meat wants to  
get his teeth into something else.
```

```
\end{verse}
```

- This produces line wrapping which preserves the structure.

A person who eats meat wants to get his teeth into  
something.

A person who does not eat meat wants to get his teeth  
into something else.

# Verbatim

- The *verbatim* environment provides a way of presenting verbatim text
- All characters are treated literally and are presented in a typewriter font. This is useful for presenting the text of computer programs or for showing the output they produce.
- The output of verbatim environments is printed flush against the left-hand margin. It can be useful to embed a verbatim environment within a quote environment to make it stand out a little more.

Here is some verbatim text.

Note that `\cmd{}`LaTeX control directives (like `\cmd{}`this) or `$ this $ & % this %` are ignored.

# Tables

- The *tabular* environment provides the simplest way to produce a table.

```
\begin{center}
\begin{tabular}{lrrr}
& Won & Drawn & Lost \\
New Zealand & 0 & 0 & 1 \\
England & 1 & 0 & 0 \\
\end{tabular}
\end{center}
```

- This describes a table with four columns.

|             | Won | Drawn | Lost |
|-------------|-----|-------|------|
| New Zealand | 0   | 0     | 1    |
| England     | 1   | 0     | 0    |

# Tables

---

- Novice users tend to overdo the use of lines in tables
- If you examine a well-typeset book you will find that tables generally use only a minimal number of lines
- There are a number of extension packages which will provide additional table capabilities.

`array` – increased flexibility for the basic tabular macro

`dcolumn` – alignment of numbers on decimal points

`tabularx` – automatic calculation of column widths

`supertab` – multipage tables

`longtable` – sophisticated multipage tables

---

# ***Typesetting Mathematics***

# *Mathematical Typesetting*

---

$\text{\LaTeX}$  is the best mathematical typesetting system available

- It provides an easy way to specify how even very complex mathematics should be typeset
- The typesetting results are as good as those obtained by the professional typesetters who work for publishing companies

# An Example

- The equation:

$$F_{XX}(\lambda) = \lim_{T \rightarrow \infty} \frac{1}{2\pi} \sum_{u=-T}^T c_{XX}(u) \frac{e^{-i\lambda u} - 1}{-iu}$$

is produced by the  $\text{\LaTeX}$  input:

```
\[
F_{XX}(\lambda) = \lim_{T \rightarrow \infty}
\frac{1}{2\pi} \sum_{u=-T}^T
c_{XX}(u) \frac{e^{-i\lambda u} - 1}{-iu}
\]
```

# Mathematical Text

- Mathematics can appear in documents as:
  - Inline Text
  - Displayed Equations
- Displayed equations can either be *numbered* or *unnumbered*
- Mathematics is typeset differently depending on whether it is being displayed or exhibited as inline text

*Displayed*

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

*Inlined*

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$



# *Inline Mathematics*

---

- Inline mathematics is delimited with dollar signs

$\$ c^2 = a^2 + b^2 \$$

- Or with math delimiters

$\( c^2 = a^2 + b^2 \)$

- Or by enclosing it a *math* environment

$\begin{math} c^2 = a^2 + b^2 \end{math}$

- These are completely equivalent, but delimiting dollar signs may be easiest to type.

# Displayed Mathematics

- Displayed mathematics can be produced in a number of ways
- To produce an unnumbered display, the material can be delimited like this:

```
$$ c^2 = a^2 + b^2 $$
```

- Or this:

```
\[ c^2 = a^2 + b^2 \]
```

- Or this:

```
\begin{displaymath}  
c^2 = a^2 + b^2  
\end{displaymath}
```

# Numbered Equations

- To produce a numbered display, delimit the material as follows:

```
\begin{equation}  
c^2 = a^2 + b^2  
\end{equation}
```

- This will produce a display of the form

$$c^2 = a^2 + b^2 \tag{1}$$

- The assigned number depends on the type of document specified in the `\documentclass` declaration

# Cross Referencing

- Numbered equations can be labelled with a `\label` directive and referenced with `\ref`
- For example

```
\begin{equation}
c^2 = a^2 + b^2
\label{eq:pythag}
\end{equation}
```

According to equation (`\ref{eq:pythag}`) ...  
results in the following output:

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

According to equation (2) ...

# Simple Mathematics

- The basic operations in mathematics are: addition, subtraction, multiplication and division

- The specification of addition and subtraction are easy

`\begin{math} a + b - c \end{math}`

- Multiplication is not written explicitly, but is denoted by juxtaposition of symbols

`\begin{math} a ( b + c ) \end{math}`

- Division is denoted by the formation of quotients

`\begin{math} ( a + b ) / c \end{math}`

# Fractions

- In displayed mathematics it is sometimes useful to have quotients in the form of fractions

```
\begin{displaymath}  
  \frac{2a}{b + c}  
\end{displaymath}
```

- This is typeset as follows:

$$\frac{2a}{b + c}$$

# Using Fractions

- It is not wise to have nested uses of `\frac`. It is better to use `/` instead

- For example

```
\begin{displaymath}
  \frac{a}{\frac{b}{2} + c} + \frac{a}{b/2 + c}
\end{displaymath}
```

produces:

$$\frac{a}{\frac{b}{2} + c} + \frac{a}{b/2 + c}$$

- The denominator of the first term does not look correct

# Numerical Fractions

- When setting numerical fractions like one half or one tenth, it can be useful to manipulate the rules used by L<sup>A</sup>T<sub>E</sub>X for laying out fractions
- Compare

$$\frac{1}{2} \quad \text{and} \quad \frac{1}{2} \quad \text{and} \quad \frac{1}{2}.$$

- This is done as follows (`\quad` inserts a large space)

```
\[
\frac{1}{2} \quad \text{and} \quad \quad
{\textstyle
\frac{\displaystyle 1}{\displaystyle 2}}
\quad \text{and} \quad \quad
{\textstyle \frac{1}{2}} .
\]
```



# Subscripts and Superscripts

- The symbols `_` and `^` are used to produce subscripts and superscripts
- For example:

```
\begin{displaymath}
X_i^2 = (\mu + \sigma Z_i)^2
      = \mu^2 + 2\mu \sigma Z_i
        + \sigma^2 Z_i^2
\end{displaymath}
```

produces

$$X_i^2 = (\mu + \sigma Z_i)^2 = \mu^2 + 2\mu\sigma Z_i + \sigma^2 Z_i^2$$

# Subscripts and Superscripts

- The subscript and superscript operators only apply to the single following object
- Braces can be used to group symbols into a single object
- For example:

```
\begin{displaymath}  
a^b/2+c \neq a^{b/2+c}  
\end{displaymath}
```

produces

$$a^b/2 + c \neq a^{b/2+c}$$

- Note that quotients in subscripts or superscripts should always be written using /

# Integrals, Sums and Products

- Integrals, sums and products are specified with the `\int`, `\sum` and `\prod` operators

- For example:

```
\begin{displaymath}
\int_{-1}^1 x^3 + x \; dx = 0, \quad \overline{x} = \frac{1}{n} \sum_{i=1}^n x_i
\end{displaymath}
```

yields:

$$\int_{-1}^1 x^3 + x \, dx = 0, \quad \overline{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- The macro `\prod` can be used to produce products

# *Maxima, Minima and Limits*

- Various kinds of limit-like constructions can be produced with the macros `\lim`, `\liminf`, `\limsup`, `\inf`, `\sup`, `\min`, `\max`
- The equation

$$\lim_{n \rightarrow \infty} f_n(x) = f(x)$$

is produced by the  $\text{\LaTeX}$  construction.

```
\lim_{n \rightarrow \infty} f_n(x) = f(x)
```

# Standard Operators

- There are a variety of mathematical operators which can be used
- These include the standard arithmetic operators

$$+ \quad - \quad \times \quad \div$$

- Note that the last two of these (`\times` and `\div`) are only rarely used in mathematical writing
- The usual inequality operators  $<$ ,  $>$ ,  $\leq$  and  $\geq$  are produced by `<`, `>`, `\le` and `\ge`
- The set theory operators  $\cup$ ,  $\cap$  and  $\in$  are produced by `\cup`, `\cap` and `\in`

# Common Functions

- Most common mathematical functions have  $\text{\LaTeX}$  commands which produce the appropriate versions of the function

- For example:

```
\begin{displaymath}
e^{i\theta} = \cos \theta + i \sin \theta
\end{displaymath}
```

produces:

$$e^{i\theta} = \cos \theta + i \sin \theta$$

- The `\sin` and `\cos` macros produce function names set in Roman rather than Italic font
- This is the correct mathematical convention

# Accents

- Statistics uses a number of standard mathematical accents
- The most commonly encountered ones are the *bar*, the *hat* and the *tilde*

$$\bar{X}, \hat{X}, \tilde{X}$$

- There are narrow version of these accents, but it is most common to use the wide versions, which are produced as follows

```
\overline{X}   \widehat{X}, \widetilde{X}
```

- These accents stretch to cover wide expressions
- Narrower forms of these are produced using `\bar X`, `\hat X`, `\tilde X`, to give

$$\bar{X}, \hat{X}, \tilde{X}$$

# *Greek Letters*

- $\text{\LaTeX}$  provides a very rich set of mathematical symbols, including the letters of the Greek alphabet and assorted mathematical symbols
- In statistics, the Greek letters

$$\alpha, \beta, \varepsilon, \mu, \sigma, \omega, \Omega$$

etc. are important

- These are produced by the latex fragments: `\alpha`, `\beta`, `\varepsilon`, `\mu`, `\sigma`, `\omega` and `\Omega`
- Don't confuse the summation sign  $\sum$  with the Greek letter Sigma  $\Sigma$
- They look similar, but are treated differently



# Miscellaneous Symbols

|   |   |  |
|---|---|--|
| ← | → | <code>\leftarrow, \rightarrow</code>             |
| ↻ | ↻ | <code>\circlearrowleft, \circlearrowright</code> |
| ♦ | ♥ | <code>\diamondsuit, \heartsuit</code>            |
| ♣ | ♠ | <code>\clubsuit, \spadesuit</code>               |
| © | ® | <code>\copyright, \textregistered</code>         |
| ♭ | ♯ | <code>\flat, \sharp</code>                       |
| § | ¶ | <code>\S, \P</code>                              |
| ■ | ★ | <code>\blacksquare, \bigstar</code>              |

# Mathematical Fonts

---

$ABCD$      `\mathcal{ABCD}`

$\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{D}$      `\mathscr{ABCD}`

$\mathfrak{ABCD}$      `\mathfrak{ABCD}`

$\mathbb{CRNZQ}$      `\mathbb{CRNZQ}`

$\mathbf{u} + \mathbf{v}$      `\mathbf{u} + \mathbf{v}`

- Note that using `\mathscr` requires loading the package `mathrsfs`

---

# ***Tables and Arrays***

# *Tables and Arrays*

---

- Situations in which column alignment is important
  - Tables in the text of a document
  - Matrices and vectors in equations
  - Program source code
- There are common elements to the first two of these situations
- The third requires special treatment

# *The tabular Environment*

---

- Tables are created using the `tabular` environment

```
\begin{tabular} {format}  
table rows  
\end{tabular}
```

- The `format` description describes how the columns the table are to be laid out
- The `table rows` give the data in the table and formatting information which is specific to particular rows
- The rows are described by giving the elements of the row separated by `&` and with the end of the row indicated by `\\`

# Column Format Specifications

- The simplest column format specifications are:

|                           |  |
|---------------------------|--|
| <code>l</code>            | a left justified column                                    |
| <code>c</code>            | a centered column  |
| <code>r</code>            | a right justified column                                   |
| <code> </code>            | a vertical line at this position                           |
| <code>@{text}</code>      | material to be inserted at this position in each row       |
| <code>p{wd}</code>        | a paragraph of width <code>wd</code> (e.g. 4cm)            |
| <code>*{num}{cols}</code> | equivalent to <code>num</code> copies of <code>cols</code> |

- The add-on package `array` adds additional column formats as well as the ability to define new ones

# A Simple Example

- The table specification:

```
\begin{quote}
\begin{tabular}{rrr}
\emph{Var. 1} & \emph{Var. 2} & \emph{Var. 3} \\
  \\\
  \$-0.4650552\$ & \$-0.2738921\$ & \$-0.8044528\$ \\
  \\\
  \$-0.4750023\$ & \$-0.2240469\$ & \$ 0.6646691\$ \\
  \\\
\end{tabular}
\end{quote}
```

produces:

| <i>Var. 1</i> | <i>Var. 2</i> | <i>Var. 3</i> |
|---------------|---------------|---------------|
| -0.4650552    | -0.2738921    | -0.8044528    |
| -0.4750023    | -0.2240469    | 0.6646691     |

# Spanning Multiple Columns

- Within each data row it is possible to provide a data element which spans several columns of the table using a `\multicolumn` directive  
`\multicolumn{cols}{align}{text}`
- This indicates that the next `cols` columns should be spanned and that the alignment should be `align`
- The data to appear in the combined “cell” is given by `text`
- `\multicolumn` is most commonly used for placing titling information in a table



# A Tricky Example

- Uses `@{ }` to align on a decimal point and `\multicolumn` to provide a column heading

```
\begin{tabular}{c r @{.} l}  
Pi expression & & \\  
\multicolumn{2}{c}{Value} \\ \hline  
$\pi$ & 3&1416 \\  
$\pi^{\pi}$ & 36&46 \\  
$(\pi^{\pi})^{\pi}$ & 80662&7 \\  
\end{tabular}
```

| Pi expression       | Value   |
|---------------------|---------|
| $\pi$               | 3.1416  |
| $\pi^{\pi}$         | 36.46   |
| $(\pi^{\pi})^{\pi}$ | 80662.7 |

# *(Mis-)Using Lines In Tables*

- Here is an example which uses lines within a table

```
\begin{tabular}{|r|l|} \hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline \end{tabular}
```

|             |             |
|-------------|-------------|
| 7C0         | hexadecimal |
| 3700        | octal       |
| 11111000000 | binary      |
| 1984        | decimal     |

# *Inserting Tables Into Documents*

---

- The basic `tabular` environment creates an item which is inserted in-line into to text of a document
- Additional steps must be taken to place tables correctly, for example, using a `center` environment

```
\begin{center}  
  \begin{tabular}{ ... }  
    ...  
  \end{tabular}  
\end{center}
```

- The `flushleft`, `flushright` and `quote` environments can also be used

# Floating Tables

- To avoid problems with tables spanning page boundaries it is common allow tables to *float* to a nearby page

```
\begin{table}[tbp]
  \centering
  \label{tab:example}
  \caption{An Example Table}
  \begin{tabular}{...}
    ...
  \end{tabular}
\end{table}
In table \ref{tab:example} ...
```

- This `table` environment allows the table to float to the top, or bottom of a page, or to occupy a page on its own

# *Captions and Floats*

---

- The standard placement of captions on floats in documents is as follows
  - Captions on tables appear at the top of the table
  - Captions on figures appear at the bottom of the figure
- Unfortunately, XEmacs places all captions at the bottom so you may need to do some rearrangement to achieve the standard effect

# Alignment in Mathematics

- There are a number of places in mathematics where column alignments are used
- Such alignments are provided by `array` environments

```
\[
|x| = \left\{
\begin{array}{cl}
x & \text{\mbox{if}}\ x \ge 0, \\
-x & \text{\mbox{if}}\ x < 0.
\end{array}
\right.
\]
```

- This produces:

$$|x| = \begin{cases} x & \text{if } x \geq 0, \\ -x & \text{if } x < 0. \end{cases}$$

# *The cases Environment*

- The previous kind of environment is quite common
- The `amsmath` package provides a special construction for it

```
\[
|x| =
\begin{cases}
x & \mathrm{if} \ x \ge 0, \\
-x & \mathrm{if} \ x < 0.
\end{cases}
\]
```

- This environment can have any number of lines within it

# Matrices and Vectors

- Statistics makes extensive use of matrices and vectors
- For example:

$$\mathbf{x} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$$

- This is typeset as follows:

```
\[  
\mathbf{x} = \left( \begin{array}{ccc}  
x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{array} \right)  
\]
```



# Aligned Equations

- In base  $\text{\LaTeX}$  aligned equations are produced by the `eqnarray` and `eqnarray*` environments
- The `eqnarray` environment produces numbered lines and the `eqnarray*` one does not
- The following example was produced with `eqnarray*`

$$\begin{aligned} E(\hat{\theta} - \theta_0)^2 &= E(\hat{\theta} - E\hat{\theta} + E\hat{\theta} - \theta_0)^2 \\ &= E(\hat{\theta} - E\hat{\theta})^2 + (E\hat{\theta} - \theta_0)^2 \\ &= \text{variance}(\hat{\theta}) + \text{bias}(\hat{\theta})^2 \end{aligned}$$

- Unfortunately, these arrays produce rather too much space around the = sign
- This means that they need some tuning to produce perfect looking equations

# The `eqnarray*` Environment

- The previous example was produced by:

```
\begin{eqnarray*}
E(\hat{\theta} - \theta_0)^2
& = & E(\hat{\theta} - E\hat{\theta})
      + E\hat{\theta} - \theta_0)^2 \\
& = & E(\hat{\theta} - E\hat{\theta})^2
      + (E\hat{\theta} - \theta_0)^2 \\
& = & \mbox{variance}(\hat{\theta})
      + \mbox{bias}(\hat{\theta})^2
\end{eqnarray*}
```

# Spacing in Aligned Equations

- To improve the appearance of aligned equations, the  $\text{\LaTeX}$  fragment

```
\setlength\arraycolsep{2pt}
```

should be inserted as follows

```
{\setlength\arraycolsep{2pt}
```

```
\begin{eqnarray*}
```

```
...
```

```
\end{eqnarray*}}
```

- This produce a slightly neater looking result

$$\begin{aligned} E(\hat{\theta} - \theta_0)^2 &= E(\hat{\theta} - E\hat{\theta} + E\hat{\theta} - \theta_0)^2 \\ &= E(\hat{\theta} - E\hat{\theta})^2 + (E\hat{\theta} - \theta_0)^2 \\ &= \text{variance}(\hat{\theta}) + \text{bias}(\hat{\theta})^2 \end{aligned}$$

# Numbering Aligned Equations

- When using `eqnarray` every line is numbered unless the number is numbering is turned off with `\nonumber`

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
E(\hat{\theta} - \theta_0)^2
& = & E(\hat{\theta} - E\hat{\theta})
& + E\hat{\theta} - \theta_0)^2 \nonumber \\
& = & E(\hat{\theta} - E\hat{\theta})^2
& + (E\hat{\theta} - \theta_0)^2 \\
& = & \mbox{variance}(\hat{\theta}) +
& \mbox{bias}(\hat{\theta})^2 \nonumber
\end{eqnarray}}
```

- Here the numbering is turned off for the first and last lines

# Numbering Aligned Equations – Example

---

- The result of the previous example are as follows

$$\begin{aligned} E(\hat{\theta} - \theta_0)^2 &= E(\hat{\theta} - E\hat{\theta} + E\hat{\theta} - \theta_0)^2 \\ &= E(\hat{\theta} - E\hat{\theta})^2 + (E\hat{\theta} - \theta_0)^2 \\ &= \text{variance}(\hat{\theta}) + \text{bias}(\hat{\theta})^2 \end{aligned} \tag{3}$$

# *Alternatives from `amsmath`*

---

- The American Mathematical Society has commissioned extensions to  $\text{\LaTeX}$  which provide a wide of mathematical notation
- These extensions are available in the extension package `amsmath` which can be loaded by placing the command `\usepackage{amsmath}` in the document preamble
- The `amsmath` package provides a number of environments for aligned equations
- The most useful of these are `align` and `align*`

# The *align* and *align\** Environments

- The `align` environment produces numbered equations
- Numbering can be inhibited by using `\nonumber`
- The fragment:

```
\begin{align}
  x^2 + y^2 &= 1 \\
  x &= \sqrt{1-y^2}
\end{align}
```

produces:

$$x^2 + y^2 = 1 \tag{4}$$

$$x = \sqrt{1 - y^2} \tag{5}$$

- The `align*` environment produces the same result, but without the numbering

# The *gather* and *gather\** Environments

- The `gather` environment is useful to present a series of equations without alignment The fragment:

```
\begin{gather}
(a + b)^2 = a^2 + 2ab + b^2 \\
(a + b)(a - b) = a^2 - b^2
\end{gather}
```

produces

$$(a + b)^2 = a^2 + 2ab + b^2 \tag{6}$$

$$(a + b)(a - b) = a^2 - b^2 \tag{7}$$

- Again, the `*` form produces unnumbered equations, or you can use `\nonumber` to inhibit some numbers



# *Text Within Aligned Equations*

- The `\intertext` directive can be used to include a line of text within a series of aligned equations

```
\begin{align*}
1 + 1 + 1 + 1 &= 4 \\
4 &= 1 + 1 + 1 + 1
\end{align*}
```

produces the output:

$$1 + 1 + 1 + 1 = 4$$

and

$$4 = 1 + 1 + 1 + 1$$

# Extra Vertical Space in Equation Groups

- Equations within an aligned group can be separated by additional space by including the amount of space as an optional argument to the end-of-line command `\\`
- The amount of space should not be absolute, but should specify an amount relative to the current font size
- For example the specification  
`\\[0.5em]`  
inserts an extra amount of space which is half the width of the letter “m” in the current font
- *Don't overuse this facility. Most of the time  $\LaTeX$  does the right thing*

# *The tabbing Environment*

- The tabbing environment provides a way of aligning text at “tab stops”
- This is often used in laying out program text

```
{\it \begin{tabbing}
If \= it's raining \\
    \> then \= put on boots, \\
    \>         \> take hat; \\
    \> else \\
    \>         \> smile.
\end{tabbing}}
```

- The command `\=` sets the next logical tab position and `\>` spaces to the next tab position
- Using `\kill` in place of `\\` produces no output but can be used for setting tab stops

# A *Tabbing Example*

---

- The example on the previous slide produces:

```
If it's raining  
  then put on boots,  
    take hat;  
else  
  smile.
```

- The `tabbing` environment can be used to produce rather more elegant program layout than the blunt instrument provided by the `verbatim` environment

---

# ***Typesetting Text***

# Paragraphs

---

- Paragraphs are the most important text unit
  - should contain one major idea
  - usually comprised of a number of sentences
- In LaTeX, the end of a paragraph is indicated by the entering an empty line
- Paragraphs are grouped into higher logical units making subsections, sections and chapters
- I don't like indented paragraphs and prefer  
`\parindent=0cm \parskip=12pt`  
in the preamble

# Page Structure

---

- In books, paragraphs are commonly *fully-justified*
- Each line is of exactly the same length so that the left and right sides of the paragraph are each vertically aligned
- This involves
  - choice of the best location for a line break
  - subtle expansion or contraction of inter-word spaces
  - use of hyphenation

# Page Structure

---

- T<sub>E</sub>X will choose line breaks, spread out words, and hyphenate words automatically
- Line breaks can be forced with `\`, or suggested more or less strongly with `\linebreak[n]`
- Hyphenation of words can be specified with `\hyphenation{word list}`, for example:  
`\hyphenation{Bayes, Auck-land, sto-chas-tic}`
- Hyphenation is only permitted where a hyphen appears in the word



# Page Structure

- Words can be kept together on the one line using `\mbox`  
This is a test sentence which has a number  
`\mbox{+64 (9) 373 7599 Ext 86830}`  
which will cause problems  
  
This is a test sentence which has a number  
`+64 (9) 373 7599 Ext 86830` which will cause problems
- Words can be kept together using `~` also
- For example `Ext~86830`
- `~` or `\` can be used to ensure correct spacing after a period, in  
for example `Ann.\ Statist.\`

# Typesetting Details

- Special strings  
`\today`, `\TeX`, `\LaTeX`, `\LaTeXe`  
produce  $\text{T}_{\text{E}}\text{X}$ ,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2_{\epsilon}+}$
- There are four kinds of dashes: the ordinary dash or hyphen (-), the en-dash (–), the em-dash (—) and the minus sign (−), produced by  
`-`, `--`, `---`, and `$-$`
- Usage is
  - the hyphen for ordinary hyphenation
  - the en-dash for a range (pp. 21–63)
  - the em-dash for punctuation (my first thought—which I then qualify)
  - the minus sign in mathematics

# Ligatures

- Some letter combinations are typeset using special symbols, called ligatures
- Particular examples are:  
ff fi fl ffi  
which may be compared to  
ff fi fl ffi
- Separation is achieved by using `f { } f` or `f \mbox{ } f`
- Separation may be needed for some compound words

# Accents

- All sorts of accents needed for foreign languages are available

- For example

```
H\^otel, na\"i ve, \'el\'eve, \\
sm\o rrebr\o d, !'Se\~{n}orita!, \\
Sch\"onbrunner Schlo\ss{ }
Stra\ss e
```

gives

Hôtel, naïve, élève,  
smørrebrød, ¡Señorita!,  
Schönbrunner Schloß Straße

- A table is given in Oetiker *et al* of further possibilities
- Additional foreign language support is provided by the **babel** package

# Typesetting Details

- The ellipsis (...) is spaced differently to just three dots, and has its own control sequence `\ldots`
- The tilde has a special meaning as seen before
  - `\verb@~@` is too high and too small: ~
  - `$_sim$` is too large: ~
  - `\raisebox{0.3ex}{\tiny$_sim$}` is about right: ~
  - I define the control sequence `\mytilde` in the preamble as `\newcommand{\mytilde}{\raisebox{0.3ex}{\tiny$_sim$}}`
- The degree symbol ( $^{\circ}$ ) is produced using `$_,\^{\circ}$`
- `\textcelsius` from the package `textcomp` will also produce the degree symbol

# *The `verb` Environment*

- The `\verb` environment allows the display of special characters or control sequences in a document
- It gives an in-line environment as opposed to the display environment provided by `\verbatim`
- The syntax is to follow the `\verb` control sequence by the text required enclosed in any character except letters `*` or a space
- For example, to produce `}%$`, possible choices include `\verb|}%$|` or `\verb@}%$@`

# Spacing

- T<sub>E</sub>X determines all spacing itself, except in verbatim-like environments
- There are control sequences to insert spaces of various sizes in mathematics
  - `\,` for  $\frac{3}{18}$  quad (␣)
  - `\:` for  $\frac{4}{18}$  quad (␣)
  - `\;` for  $\frac{5}{18}$  quad (␣)
  - `\_` gives a medium space
  - `\quad` gives an “m”-sized space (␣)
  - `\qquad` gives a “2m”-sized space (␣)
  - `\!` gives a negative space of  $\frac{3}{18}$  quad
- A typical use is `\int e^{-x} \, dx` to give  $\int e^{-x} dx$

# Spacing

- In text, space can be inserted explicitly using `\hspace` for horizontal space, and `\vspace` for vertical space
- These take a measurement in braces
- Here is the effect of respectively, `\vspace{0.7cm}`, and `\hspace{3em}` and even `\hspace{-6pt}`

A small amount of space

and some negative space here



# Line Spacing

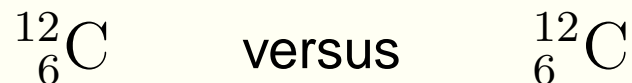
- `\linespread` recommendation is strange
- Usually `\baselinestretch` is used
- For “one and a half” line spacing  
`\renewcommand{\baselinestretch}{1.5}`
- `\baselineskip` is distance between baselines, depends on font, font size plus a bit
- true interline spacing is  
`\baselinestretch\times\baselineskip`
- Change doesn't take place until next change in font size: use `\small\normalsize` to change immediately
- Use package `setspace` to change spacings more intelligently

# Phantoms

- To align text, sometimes a phantom is useful
- This leaves the space needed for the text it specifies, but doesn't actually show the text

```
\begin{displaymath}
{}^{\{12\}}_{\{\phantom{1}\}6} \mathrm{C}
\quad \text{versus} \quad
{}^{\{12\}}_{\{6\}} \mathrm{C}
\end{displaymath}
```

produces



- Phantoms are useful in aligning text in tables

# ***Floating Bodies***

---

- Figures or tables
- Position specification
  - h        here, where the text is
  - t        at the top of a page
  - b        at the bottom of a page
  - p        on a special page of floats
  - !        ignore internal parameters concerning floats
- The *float* package with the *H* option gives better control of floats
- This attempts to position the figure or table exactly where it is located in the text, otherwise it puts it at the top of the next page

# *XEmacs*

- AUCT<sub>E</sub>X adds menus and functionality when working with T<sub>E</sub>X
- Menus allow creation of environments, processing of documents, insertion of special symbols etc
- Key bindings are given on the menus
- C-c C-e inserts an environment
- C-c ] closes an environment
- C-c C-j or M-`<Enter>` inserts an item
- M-q fills and aligns text

# *Files Used in L<sup>A</sup>T<sub>E</sub>X*

---

| <b>Extension</b>  | <b>File type</b>   |
|-------------------|--|
| <code>.tex</code> | T <sub>E</sub> X or L <sup>A</sup> T <sub>E</sub> X input file                     |
| <code>.sty</code> | L <sup>A</sup> T <sub>E</sub> X macro package loaded with <code>\usepackage</code> |
| <code>.cls</code> | class file used by <code>\documentclass</code>                                     |
| <code>.dvi</code> | device independent file produced by L <sup>A</sup> T <sub>E</sub> X                |
| <code>.log</code> | log file describing last compiler run  |
| <code>.aux</code> | auxiliary file for use in cross-references   |
| <code>.bib</code> | bibliography in special format   |

# Page Style

- L<sup>A</sup>T<sub>E</sub>X has three predefined page styles
  - `plain` prints the page numbers on the bottom of the page, in the middle of the footer. This is the default page style
  - `headings` prints the current chapter heading and the page number in the header on each page, while the footer remains empty
  - `empty` sets both the header and the footer to be empty
- Select using for example `\pagestyle{empty}`
- More complex page styles may be produced by using the package `fancyhdr`

---

# ***Graphics***

# *Vector Graphics*

---

- Vector graphics are composed of geometric objects described in a two dimensional cartesian coordinate system
- Graphics can be magnified to any extent without loss of quality (scalability)
- POSTSCRIPT and SVG are the most common cross-platform vector graphics standards
- XFIG diagrams are common under LINUX and UNIX
- Vector graphics are produced by drawing programs such as *xfig*, *dia* or *illustrator*



# ***Raster Graphics***

---

- Graphics consist of a two-way array of coloured or shaded dots called pixels (picture elements)
- Image quality degrades with magnification because the pixels become more apparent
- Some objects do not look good when drawn in a raster system because of the problem of *aliasing*
- There are many raster graphics file formats. There is no single dominant format, but it is easy to convert between formats.
- Raster graphics are produced by painting programs such as *gimp* and *photoshop*

# Comparing Vector and Raster Graphics

---

## *Vector Graphics:*

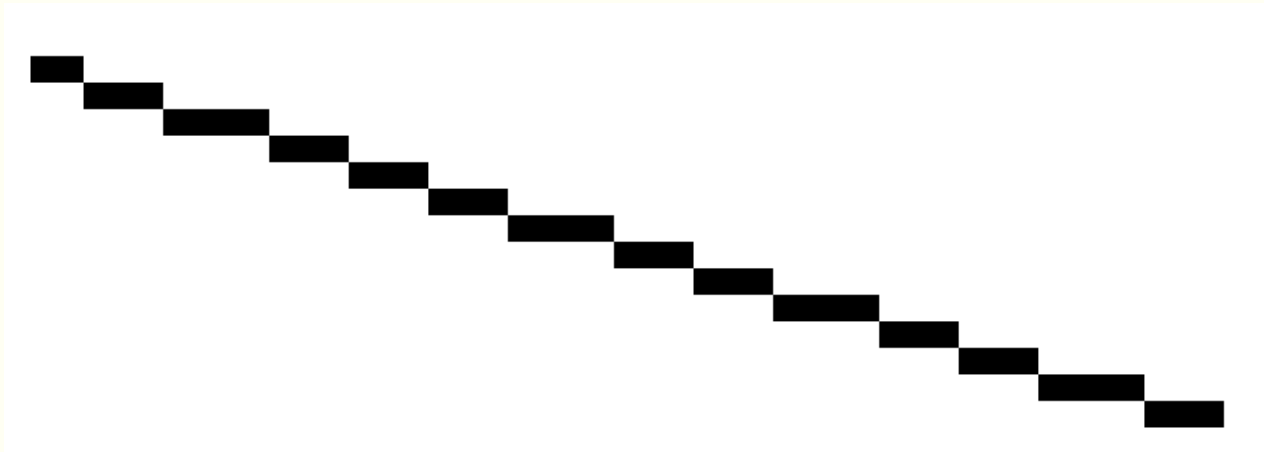
- Useful when there are only clean lines and sharp boundaries to be drawn
- Useful when the same figure must be reproduced at a variety of resolutions

## *Raster Graphics:*

- Useful for images such as photographs where there are many colours and fewer sharp boundaries
- Can be problematic when the image must be scaled for a variety of different resolutions

# *Aliasing*

- When objects such straight lines are drawn in a in a raster graphics system, the raster nature of the display is often visible
- This is a particular problem with lines which are nearly vertical or horizontal



# *Antialiasing*

- A technique called *antialiasing* can be used to alleviate some of the worst problems with aliasing
- Antialiasing adds a small amount of blurring to features like straight lines
- As a side effect of the human vision system removing the blur, the apparent jaggedness of the line is reduced



# *Graphics Formats: JPEG (.jpg)*

---

The Joint Photographic Experts Group format

- A 24-bit format
- Very commonly used on the Web
- The format used by most digital cameras
- Uses a very effective (lossy) compression algorithm (5:1 – 15:1 compression)
- Colors not visible to humans are set to black
- Supported by  $\text{\LaTeX}$

# *Graphics Formats: GIF (.gif)*

---

## The Graphics Interchange Format

- An 8-bit format, so there is a limit of 256 colours per image
- *Was* the most commonly used graphics format
- There were patent problems with its data compression method which has decreased its popularity
- Includes extensions for transparency, incremental loading and animation
- Not supported by  $\text{\LaTeX}$

# *Graphics Formats: PNG (.png)*

---

Portable Network Graphics format

- A 24-bit format
- Designed to replace GIF and avoid patent problems
- Uses an alternative compression algorithm
- Supported by  $\text{\LaTeX}$

# *Graphics Formats: BMP (.bmp)*

---

The standard MS-Windows raster format

- Limited, fixed color palette
- Limited compression options
- Only good for small images
- Not supported by  $\text{\LaTeX}$



# *Graphics Formats: TIFF (.tif)*

---

## The Tagged Image File Format

- A 24-bit format
- Non-lossy compression
- High-quality, but tiff files tend to be very large
- Not supported by  $\text{\LaTeX}$

# *PBM/PGM/PPM (.pbm .pgm .ppm)*

---

Portable Bitmap, Graymap and Pixmap formats

- Text formats
- Often used as an intermediate in format conversions
- Not supported by  $\text{\LaTeX}$

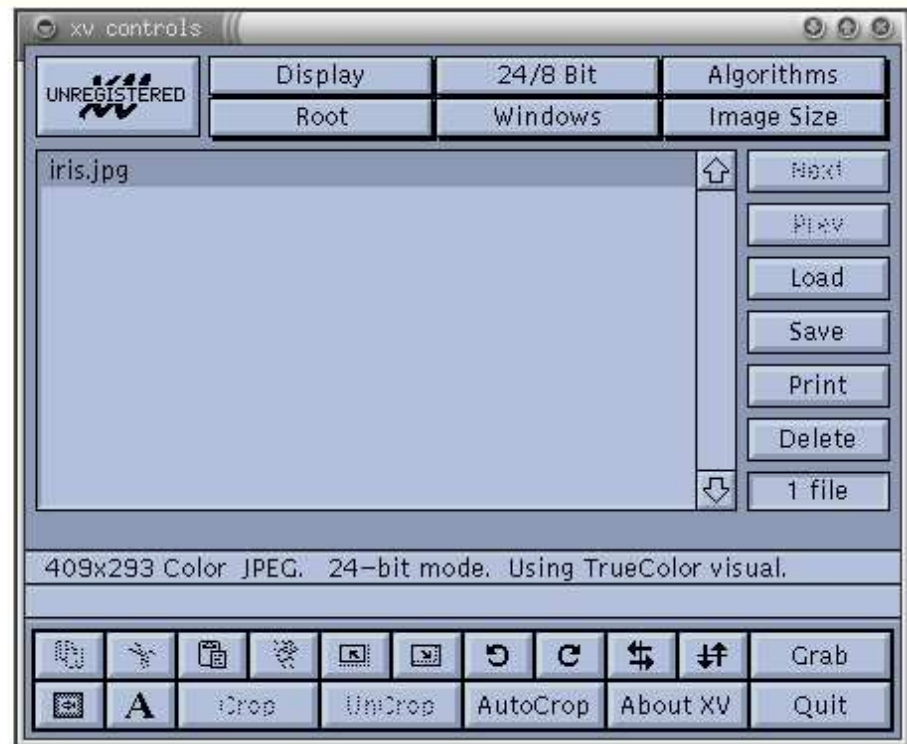
# *Xv—The X Image Viewer*

---

*xv* is able to:

- view images written in a wide variety of raster formats;
- convert images from one format to another;
- take snapshots of on-screen window;
- perform some useful image processing operations:
  - blurring and sharpening,
  - pixelization,
  - image resizing and rotation,
  - colour adjustment and removal

# Xv On Screen



To bring up the dialog box to the right, click the right mouse button in the image window

# Screenshots With Xv

---

xv is a good general purpose tool. It is very useful for taking “screenshots” of (part of) a workstation screen. Do this as follows:

1. Push the *Grab* button to obtain the screenshot dialog
2. Push the *Grab* button in dialog and move the cursor to the window you want the screenshot for. Then press the first mouse button. To capture the entire screen click in the background
3. Alternatively, you can set a delay and press the *AutoGrab* button. After the delay the image will be taken. This is useful if you want to show the contents of menus

---

# ***Specialities***

# Inserting Graphs

- Use the *graphicx* package
- Can resize, rotate, view only a portion of the graphic
- More detail in [grfguide.pdf](#) and [epslatex.pdf](#)
- One point to note is that rotation is done before sizing and this results in the size being reduced from what you might think
- Measurements:

| Abbreviation | Size  |
|--------------|---|
| mm           | millimetres   |
| in           | inches  |
| pt           | “points”, usually 1/72nd inch, in T <sub>E</sub> X 1/72.27 inch |
| bp           | “big points”, 1/72nd inch                                       |
| em           | “em”, width of an 'M' in current font                           |
| ex           | “ex” height of an 'x' in the current font                       |

# Bibliography

- Bibliography entries can be in the `.tex` file
  - Inside the `thebibliography` environment, (like a list)
  - Each one starting with `\bibitem`
- Otherwise in a separate `.bib` file, and only references needed are brought in
- Then need to `latex`, `bibtex`, `latex`, `latex` the  $\text{T}_{\text{E}}\text{X}$  file
- Many style files for formatting bibliographies and citations:  
*natbib.bst*, *chicago.bst*
- Package *custom-bib* will create all sorts of bibliographic styles
- XEmacs can be used to create and maintain  $\text{BIB}_{\text{T}_{\text{E}}\text{X}}$  files



# ***Bibliography Example***

---

```
\begin{thebibliography}{99}
\bibitem{AsJo86} Ashby, M.F., Jones, R.H.,
\it Engineering materials 1, 2nd Ed.},
(Butterworth-Heinemann, Oxford, 1986).
.
.
.
\bibitem{zhou} Zhou, S.X.,
``An integrated model for hot rolling of
steel strips'',
{\it Journal of Materials Processing Technology},
{\bf 134} (2003), 338--351.
\end{thebibliography}
```

# Installing Packages

- On the Linux machines in the grad room, package files are in:  
`/usr/share/texmf/tex/latex`
- Simply add a new package to that directory if you have root access
- If you don't have root access, create a directory and set the environment variable `TEXINPUTS`
- For example, I have  
`.: /home/staff/dscott/Computing/TeX/TeX-Inputs// :`
- On Windows, MikTeX has a package management system, files are in the same sort of tree under `C:`

# Fonts

---

- Fonts are needed for:
  - reading of documents
  - printing of documents
  - web display of documents
- Fonts are needed in a range of sizes from 5 or 6 points to possibly 50 points
- Letters can't simply be scaled up
- Like graphics, there are bitmapped and vector fonts
- Special fonts are needed for mathematics

# Fonts

---

- Some fonts in some applications are free: best known are the POSTSCRIPT fonts from Adobe (Times-Roman, Courier, Helvetica etc.)
- There are 35 POSTSCRIPT fonts which were originally those supplied in the Apple Laser Writer
- Other fonts must be purchased (Lucida, MathTime)
- Standard T<sub>E</sub>X fonts are the Computer Modern fonts which are POSTSCRIPT Type 1 and display well in Acrobat Reader

---

# ***Mathematical Graphics***

# Producing Graphs using T<sub>E</sub>X

- Why?
  - All done in T<sub>E</sub>X
  - No extra files
  - No big graphics files
  - Access to all mathematical typesetting machinery
- Many options
  - The `picture` environment
  - Extensions to `picture`: `epic`, `eepic`
  - `xy-pic`
  - `picTEX` (manual must be purchased)
  - `pstricks`

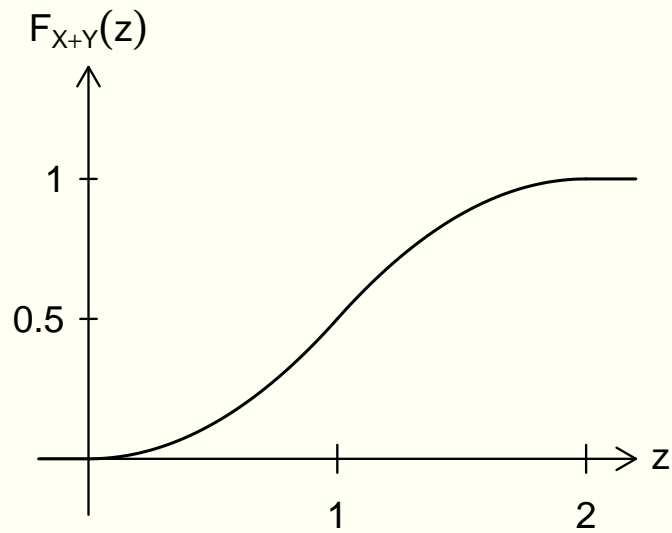
# *Mathematics in Graphs from R*

---

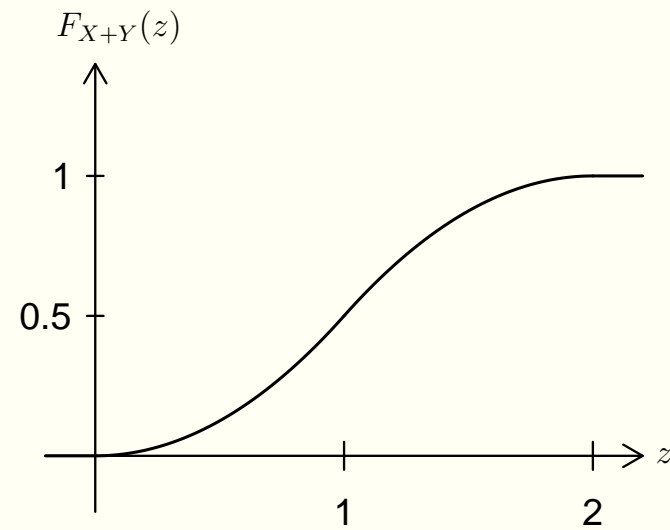
- Two approaches: use annotation from within **R**, or use psfrag
- See later for the **R** approach
- T<sub>E</sub>X approach only works for T<sub>E</sub>X documents, but produces a better result
- Next graph shows respectively the results using **R** alone, and using psfrag

# *psfrag Example*

**Graph of  $F_{X+Y}(z)$**



**Graph of  $F_{X+Y}(z) = 2z - z^2/2 - 1$**





# How is it Done?

- In the **R** commands to produce the graph

```
text(2.2+2*xtick,0,"exp1",cex=1)
text(0,1.4+4*ytick,"exp2",cex=1)
title("exp3")
```

- In the **L<sup>A</sup>T<sub>E</sub>X** file

```
\usepackage{psfrag}
```

```
⋮
```

```
\begin{center}
```

```
  \psfrag{exp1}[] [c] {$z$}
```

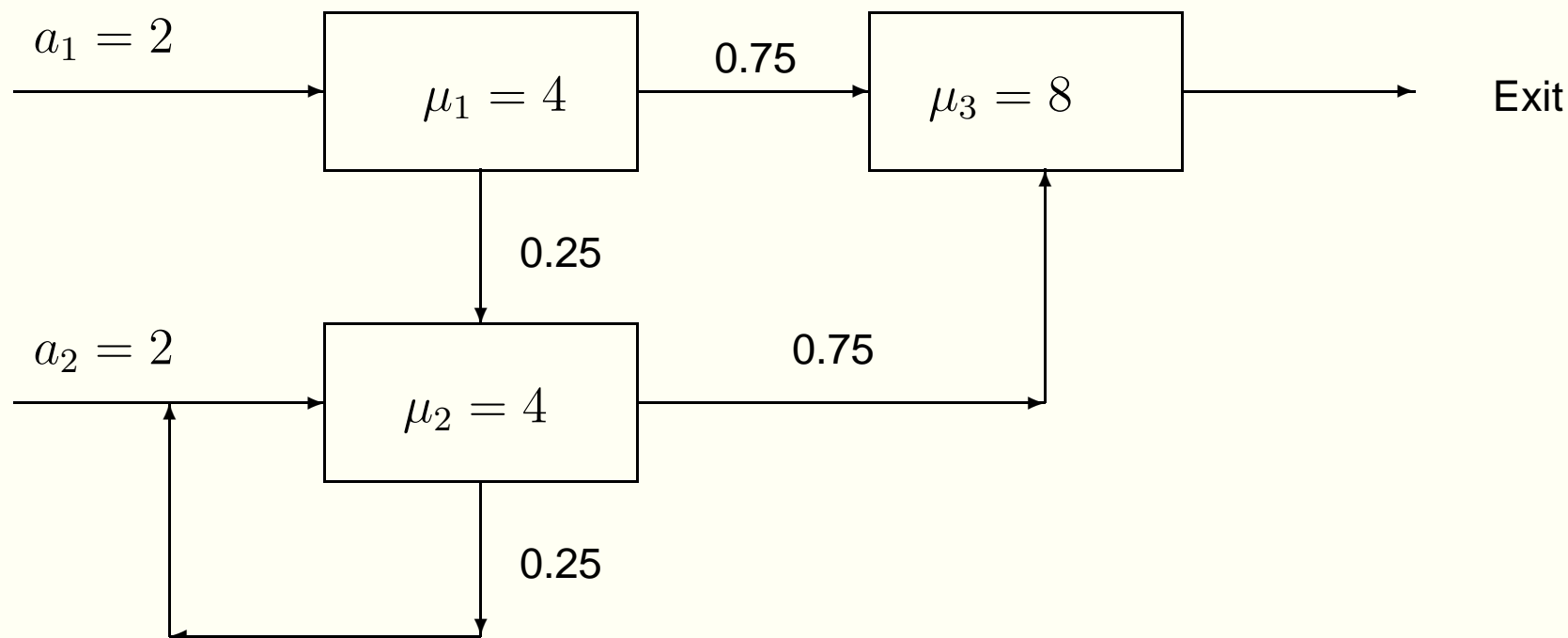
```
  \psfrag{exp2} {$F_{X+Y}(z)$}
```

```
  \psfrag{exp3}[] [c] {\large{\bf Graph of} $F_{X+Y}$}
```

```
  \includegraphics[angle=-90,width=0.8\textwidth,k
```

```
\end{center}
```

# *xfig with L<sup>A</sup>T<sub>E</sub>X*



# How is it Done?

- In `xfig` include  $\text{\LaTeX}$  in text

- Save as  $\text{\LaTeX}$  picture

- In `.tex` file

```
\begin{center}  
  {\resizebox{4in}{!}{\input{diagram.latex}}}  
\end{center}
```

- In `\resizebox`, first set two sets of braces specify horizontal and vertical dimensions

- To specify one dimension only and keep the aspect ratio, replace the other dimension with `!`

- Need to use the package `color`

---

# ***Making pdf's***

- Document formats
  - `dvi`: original T<sub>E</sub>X format, not portable, not suitable for web publishing
  - `.ps`: good for printing, not very portable, not suitable for web publishing
  - `.pdf`: good for printing, highly portable, suitable for web publishing
  - `.html`: good for web publishing, no support for mathematics
- Special attributes of pdf documents: active links (internal and external); thumbnails; bookmarks; direct display in web browsers
- Readers for pdf are free: both Adobe Acrobat Reader and xpdf

# Making pdf Documents

- Use pdf $\text{T}_{\text{E}}\text{X}$  or pdf $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  to skip the `.dvi` format and go directly to `.pdf`
- Use  $\text{T}_{\text{E}}\text{X}$  or  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  then `.dvi` to `.pdf` using for example `dvipdfm` (which is not available on stat12)
- Use  $\text{T}_{\text{E}}\text{X}$  or  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  then `.dvi` to `.ps` using `dvips`, `.ps` to `.pdf` using Ghostscript (`ps2pdf`)
- One important consideration is the format of graphics files: for pdf $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  they can be `.pdf`, `.png`, `.jpg`, but **not** `.ps` or `.eps`
- **R** can produce `.eps`, `.ps`, `.pdf`, and `.png` formats
- Convert `.eps` to `.pdf` using `epstopdf`

# Conversions

- Makefile for pdf $\text{\LaTeX}$ :

```
viewtex: $(FILENAME).tex
    pdflatex $(FILENAME).tex
    acroread $(FILENAME).pdf
```

- Makefile for prosper:

```
viewtex: $(FILENAME).tex
    latex $(FILENAME)
    dvips -Pwww -o $(FILENAME).ps $(FILENAME).dvi
    ps2pdf $(FILENAME).ps $(FILENAME).pdf
    acroread $(FILENAME).pdf&
```

# *The hyperref package*

---

- Allows all the special pdf tricks
- Used in all the slide maker programs
- Allows navigation buttons



# *Making Links: Counters*

---

- $\text{T}_{\text{E}}\text{X}$  uses counters for many items: pages, sections, equations
- counters generally have the same name as the commands that alter them: `chapter`, `section`, `subsection`, `page`, `equation`, `figure`, `table`, `footnote`
- not so clear are `enumi`, `enumii`, `enumiii`, `enumiv`, which count the four possible levels of the `enumerate` environment
- the value of a counter is an integer, usually non-negative

# Making Links: Counters

---

- users may create their own counters with `\newcounter{counter_name} [in_counter]` where *counter\_name* is the name of the counter, and *in\_counter* allows the counter to be nested within another counter
- for example, `subsection` is reset to zero whenever `section` is incremented
- `\newcounter` should go in the preamble
- every counter has an initial value of zero

# Changing Counters

- Commands to change counters are
  - `\setcounter{counter}{num}`
  - `\addtocounter{counter}{num}`
  - `\stepcounter{counter}`  
which increments the counter by one, along with all subcounters
  - `\refstepcounter{counter}`  
which is like `\stepcounter`, but in addition makes the counter the current counter for the cross-referencing command `\label` (see later)
- Counters can be printed in arabic numerals, upper or lower case Roman, and upper or lower case letters using
  - `\arabic{counter}`, `\Roman{counter}`,
  - `\roman{counter}`, `\Alph{counter}`, and
  - `\alph{counter}`

# *Making Links: Using `hyperref`*

---

- `hyperref` can make crossreferences active in a pdf
- It can also be used to create explicit internal and external links
- Easiest with pdf $\LaTeX$
- Using `dvips` and `ps2pdf` care is required
- Consider the `prospcr` example later

---

# *Slides Using T<sub>E</sub>X*

# *The prosper Class*

---

- Some examples of styles and the files used
- Structure of a prosper  $\text{\LaTeX}$  file
- Options for the class
- The preamble

# *The prosper Package*

---

A typical preamble:

```
\documentclass[pdf,  
             ps2pdf,  
             final,  
             slideColor,  
             colorBG,  
             rcore]{prosper}  
% \documentclass[ps,nocolorBG,noaccumulate,rcore]{prosper}  
\usepackage{psfrag,mathrsfs}  
\usepackage{xspace,colortbl,amsmath,amsfonts}
```

# *The prosper Package*

---

Some options (default in boldface):

- **ps** or pdf
- **slideBW** or slideColor
- **colorBG** or nocolorBG
- **final** or draft
- **total** or nototal
- A number of styles are available with the package:  
**troispoints**, azure, lignesbleues, ...



# *The prosper Package*

---

Title page created using:

```
\title{\TeX and \LaTeX}  
\author{David J. Scott}  
\email{d.scott@auckland.ac.nz}  
\institution{Department of Statistics, University of  
  
\begin{document}  
\maketitle
```

# Using hyperref

To create a link to a particular page using the page number:

- Create counter using  
`\newcounter{mypagecount}`
- Increase counter using  
`\stepcounter{mypagecount}`
- Increase counter and create a labelled location using  
`\refstepcounter{mypagecount}\label{titleslide}`
- Create link to labelled location, with counter value at labelled locations showing using  
`\pageref{titleslide}`

# Using hyperref

- To create a link to a particular location
  - Label the link destination  
`\hypertarget{startslide}{Department of Statistics}`
  - Create link with text as given in the second set of braces  
`\hyperlink{startslide}{the first slide}`
- Treatment of links is determined by:  
`\hypersetup{colorlinks,linkcolor=red,urlcolor=red}`
- Links to sections, websites etc should be set up automatically

# *Slides using BEAMER*

---

- BEAMER is a document class to produce slides for display using a pdf reader such as *Acrobat Reader*
- It is a very sophisticated system
- Its progenitor Till Tantau has written a complete graphics engine *pfg* as an alternative to drivers such as *dvips* or *dvipdfm*
- The manual for *pfg* runs to over 400 pages!

# ***BEAMER Features***

---

- A range of predefined themes with different possibilities for colour choice, fonts, environments such as itemize, and slide layout such as the use of sidebars and head and footlines
- Navigation icons
- Overlays to enable parts of a slide to be revealed in sequence—not limited to simply revealing itemized elements one by one
- Inclusion of animations and sounds
- Addition of hyperlinks and buttons
- Easy preparation of notes pages
- Creation of handouts pages
- Preparation of an article using the same file

# *Navigation Symbols*

---

- By default BEAMER displays a set of up to 9 navigation symbols
  - A slide icon—a single rectangle
  - A frame icon—a stack of three slide icons
  - A subsection icon—a highlighted section in a table of contents
  - A section icon—a highlighted section with subsections
  - A presentation icon—a completely highlighted table of contents
  - An appendix icon—highlighted table of contents with only one section
  - Back and forward arrows—curved arrows
  - A search icon—magnifying glass
- Note that a frame is typically a set of slides which are overlays from a complete slide

# *Navigation Symbols*

---

- For the navigation symbols which have arrows next to them
  - clicking the left arrow jumps to the (the last slide of) the previous slide, frame, subsection or section
  - clicking the right arrow jumps to the (the first slide of) the next slide, frame, subsection or section
- For the frame, subsection and section items clicking on the left (right) side of the icon jumps to the first (last) slide in that frame, subsection or section
- Clicking on the slide icon will bring up a box asking for the number of the slide you wish to jump to

# Frames

- A presentation consists of a series of frames
- Each frame consists of a series of slides
- Create a frame with `\frame` or using the `frame` environment
- Overlay specifications can be given

```
\begin{frame} [ <+ - > ]
  \begin{theorem}
    $A = B$
  \end{theorem}
  \begin{proof}
    \begin{itemize}
      \item Clearly $A = C$
      \item As shown earlier, $C = B$
      \item<3-> Thus $A = B$
    \end{itemize}
  \end{proof}
\end{frame}
```



# Frames

---

- The specification [ <+-> ] is a default overlay specification, indicated by the angle brackets within the square brackets
- Options can be given such as `plain` which suppresses headlines, footlines, and sidebars
- Allows frames with different head- and footlines
- Allows frames to contain a large picture which fills the frame

# *Components of a Frame*

---

- A frame may contain
  - a headline and a footline
  - a left and a right sidebar
  - navigation bars
  - navigation symbols
  - a logo
  - a frame title
  - a background
  - some frame contents
- The first three are usually set up by the theme used

# *Navigation Bars*

---

- Typical structure shows sections and subsections with current location highlighted
- Clickable so you can jump to different sections of your talk: e.g for question time
- Show listeners where you are in the talk
- Can take up a lot of room, but can be compressed

# Overlays

- A frame can be split into overlays very simply using `\pause`

```
\begin{frame}
  \begin{itemize}
    \item Shown from first slide on
  \pause
  \item Shown from second slide on
  \item Shown from second slide on
    \begin{itemize}
      \item Shown from second slide on
    \pause
    \item Shown from third slide on
    \end{itemize}
  \end{itemize}
\end{frame}
```

- `\pause` can also take an optional number indicating the slide in the frame from which the following text is to be shown

# Overlays

- Certain commands accept an *overlay specification*
- The overlay specification is given in diagonal brackets

```
\begin{frame}
  \textbf{This line is bold on all three slides}
  \textbf<2>{This line is bold only on the second}
  \textbf<3>{This line is bold only on the third s}
\end{frame}
```

- Other overlay specifications include
  - `\only` for example `\only<2->Text from slide 2 on}`
  - `\uncover`, `\visible`, `\invisible`, `\colorize`
  - Environments such as *itemize* can take overlay specifications

# Overlays

## • Instead of

```
\begin{itemize}
\item<1-> Apple
\item<2-> Peach
\item<3-> Plum
\end{itemize}
```

## BEAMER offers

```
\begin{itemize}
\item<+> Apple
\item<+> Peach
\item<+> Plum
\end{itemize}
```

# *Static Global Structure*

---

- `\titlepage` inserts a title page containing the document title, author(s)'s name, affiliation(s), a title graphic and a date
- The title page can be plain if requested or take its format from the presentation theme being used
- A short title can be given for use in head- and footlines
- `\section` doesn't create a section heading—it creates an entry in the table of contents and for use in navigation bars
- `\subsection` is similar to `\section`
- The start of a section can be treated differently with `\AtBeginSection`
- A talk can be broken up into separate parts each with table of contents, navigation bars etc

# *Static Global Structure*

---

- A table of contents can be inserted with `\tableofcontents`
- This command takes options in square brackets including `currentsection`, `sectionstyle`
- A bibliography can be included
- An appendix can be added



# *Interactive Global Structure*

- Add hyperlinks in three steps
  1. Specify a target with `\hypertarget` or `\label`
  2. Render the button with `\beamerbutton` or a similar command
  3. Put the button inside a `\hyperlink` command

```
\begin{frame}[label=threeitems]  
  \begin{itemize}  
    \item<1-> First item  
    \item<2-> Second item  
    \item<3-> Third item  
  \end{itemize}  
  
  \hyperlink{threeitems<2>  
    {\beamerbutton{Jump to second slide}}  
\end{frame}
```

# Themes

---

## ● Presentation Themes

- all details specified
- includes colours, fonts, bullets, inner and outer themes

## ● Color Themes

## ● Font Themes

## ● Inner Themes

- elements which aren't part of the headline, footline or sidebars
- includes environments such as itemize, enumerate, table of contents

## ● Outer Themes

- specifies location and style of headlines, footlines, sidebars, logo, navigation symbols, frametitle