FALL 2013
Stat 8003: Statistical Methods I
Lecture 2

Jichun Xie

1 Introduction to LaTeX

References:

- LaTeX-project: `http://www.latex-project.org/`
- LaTeX wikibook: `http://en.wikibooks.org/wiki/LaTeX`

1.1 Basics

1.1.1 Global Structures

```
\documentclass{...}
\usepackage{...}

\begin{document}
...
\end{document}
```

- The preamble: the area between `\documentclass{...}` and `\begin{document}`.
- document text: the area between `\begin{document}` and `\end{doucment}`.

1.1.2 Text and Paragraph Formatting

```
  \emph{emphasis}, \textbf{bold}, \textit{italic}
```

emphasis, **bold**, *italic*

```
\begin{verbatim}
 The verbatim environment
   simply reproduces every
  character you input,
 including all  s p a c e s!
\end{verbatim}
```

```
The verbatim environment
  simply reproduces every
 character you input,
including all  s p a c e s!
```

```
\begin{doublespace}
  This paragraph has \\ double \\ line spacing.
\end{doublespace}
```

This paragraph has

double

line spacing.

```
\begin{spacing}{2.5}
  This paragraph has \\ huge gaps \\ between lines.
\end{spacing}
```

This paragraph has

huge gaps

between lines.

1.1.3 Labels and Cross Referencing

Another good point of LaTeX is that you can easily reference almost anything that is numbered (sections, figures, formulas), and LaTeX will take care of numbering, updating it whenever necessary. The commands to be used do not depend on what you are referencing, and they are:

```
\label{marker}
```

You give the object you want to reference a marker, you can see it like a name.

```
\ref{marker}
```

You can reference the object you have marked before. This prints the number that was assigned to the object.

```
\pageref{marker}
```

It will print the number of the page where the object is.

1.2 Mathematics

One of the greatest motivating forces for Donald Knuth when he began developing the original TeX system was to create something that allowed simple construction of mathematical formulas, whilst looking professional when printed. The fact that he succeeded was most probably why TeX (and later on, LaTeX) became so popular within the scientific community. Typesetting mathematics is one of LaTeX's greatest strengths. It is also a large topic due to the existence of so much mathematical notation.

If your document requires only a few simple mathematical formulas, plain LaTeX has most of the tools that you will need. If you are writing a scientific document that contains numerous complicated formulas, you might need the `amsmath` package and the `mathtools` packages. Include

```
\usepackage{amsmath,mathtools}
```

in the preamble.

1.2.1 Mathematics Environments

Text:

```
x in the math text mode: $x$, \(x\)
```

x in the math text mode: x, x

Displayed:

```
There are Three ways to display a math equation:
$$ x + y = z $$
\[x + y = z\]
\begin{equation*}
  x + y = z
\end{equation*}
```

There are Three ways to display a math equation:

$$x + y = z$$

$$x + y = z$$

$$x + y = z$$

Displayed with equation number:

```
\begin{equation}\label{eq:xyz}
  x + y = z
\end{equation}
```

$$x + y = z \tag{1}$$

```
Equation (\ref{eq:xyz}) on page \pageref{eq:xyz} describes a
  relationship between $x$, $y$ and $z$.
```

Equation (1) on page 4 describes a relationship between x, y and z.

Multi-equation display:

```
\begin{align*}
    x + y +z & =  6\\
    x - y & = 2\\
    x - z & = 4
\end{align*}
```

$$\begin{aligned} x + y + z &= 6 \\ x - y &= 2 \\ x - z &= 4 \end{aligned}$$

1.2.2 Symbols

Mathematics has many symbols! One of the most difficult aspects of learning LaTeX is remembering how to produce symbols. There are of course a set of symbols that can be accessed directly from the keyboard:

```
+ - = ! / ( ) [ ] < > | ’ :
```

Beyond those listed above, distinct commands must be issued in order to display the desired symbols. There are a great deal of examples such as Greek letters, set and relations symbols, arrows, binary operators, etc. For example,

```
  \forall x \in X, \quad \exists y \leq \epsilon
```

$$\forall x \in X, \quad \exists y \leq \epsilon$$

Fortunately, there’s a tool that can greatly simplify the search for the command for a specific symbol. Search `Detexify` and you will find a website that allows you search for the command by inputing handwriting symbols. There are also apps for iPhone and Android phones. Some softwares such as WinEdt incorporates a list of basic symbols in the toolbox. Another option would be to look in the “The Comprehensive LaTeX Symbol List" available at:

`http://www.ctan.org/tex-archive/info/symbols/comprehensive`

Now we introduce some basic symbols that are commonly used.

Greek letters. Greek letters are commonly used in mathematics, and they are very easy to type in *math mode*. You just have to type the name of the letter after a backslash: if the first letter is lowercase, you will get a lowercase Greek letter, if the first letter is uppercase (and only the first letter), then you will get an uppercase letter.

```
 \alpha, \beta, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \Phi
```

$$\alpha, \beta, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \Phi$$

Operators. An operator is a function that is written as a word: e.g. trigonometric functions (sin, cos, tan), logarithms and exponentials (log, exp). LaTeX has many of these defined as commands:

```
  \cos (2\theta) = \cos^2 \theta - \sin^2 \theta
```

$$\cos(2\theta) = \cos^2\theta - \sin^2\theta$$

For certain operators such as limits, the subscript is placed underneath the operator:

```
\lim_{x \to \infty} \exp(-x) = 0
```

$$\lim_{x \to \infty} \exp(-x) = 0$$

For the modular operator there are two commands: `\bmod` and `\pmod`

```
a \bmod b, \quad x \equiv a \pmod b
```

$$a \bmod b, \quad x \equiv a \pmod b$$

Powers and indices. Powers and indices are equivalent to superscripts and subscripts in normal text mode. The caret (^) character is used to raise something, and the underscore (_) is for lowering. If more than one expression is raised or lowered, they should be grouped using curly braces ({ and }).

```
  k_{n+1} = n^2 + k_n^2 - k_{n-1}
```

$$k_{n+1} = n^2 + k_n^2 - k_{n-1}$$

An underscore (_) can be used with a vertical bar (|) to denote evaluation using subscript notation in mathematics:

```
f(n) = n^5 + 4n^2 + 2 |_{n=17}
```

$$f(n) = n^5 + 4n^2 + 2 |_{n=17}$$

Fractions and binomials. A fraction is created using the `\frac{numerator}{denominator}` command. (For those who need their memories refreshed, that's the top and bottom respectively!). Likewise, the binomial coefficient (aka the Choose function) may be written using the `\binom` command:

```
\frac{n!}{k!(n-k)!} = \binom{n}{k}
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

For relatively simple fractions, it may be more aesthetically pleasing to use powers and indices:

```
^3/_7, \quad 3/7
```

$$^3/_7, \quad 3/7$$

Roots. The `\sqrt` command creates a square root surrounding an expression. It accepts an optional argument specified in square brackets ([and]) to change magnitude:

```
\sqrt{\frac{a}{b}}, \quad \sqrt[n]{1+x+x^2+x^3+\ldots}
```

$$\sqrt{\frac{a}{b}}, \quad \sqrt[n]{1+x+x^2+x^3+\ldots}$$

Sums and integrals. The `\sum` and `\int` commands insert the sum and integral symbols respectively, with limits specified using the caret (^) and underscore (_). The typical notation for sums is:

```
\sum_{i=1}^{10} t_i
```

$$\sum_{i=1}^{10} t_i$$

The limits for the integrals follow the same notation. It's also important to represent the integration variables with an upright d, which in math mode is obtained through the `\mathrm{}` command, and with a small space separating it from the integrand, which is attained with the `\,` command.

```
\int_0^\infty \mathrm{e}^{-x}\,\mathrm{d}x
```

$$\int_0^\infty \mathrm{e}^{-x}\,\mathrm{d}x$$

Automatic sizing. Very often mathematical features will differ in size, in which case the delimiters surrounding the expression should vary accordingly. This can be done automatically using the `\left`,`\right`, and `\middle` commands. Any of the previous delimiters may be used in combination with these:

```
\left(\frac{x^2}{y^3}\right)
```

$$\left(\frac{x^2}{y^3}\right)$$

```
P\left(A=2\middle|\frac{A^2}{B}>4\right)
```

$$P\left(A=2\middle|\frac{A^2}{B}>4\right)$$

Curly braces are defined differently by using `\left\{` and `\right\}`,

```
\left\{\frac{x^2}{y^3}\right\}
```

$$\left\{\frac{x^2}{y^3}\right\}$$

If a delimiter on only one side of an expression is required, then an invisible delimiter on the other side may be denoted using a period (`.`).

```
\left.\frac{x^3}{3}\right|_0^1
```

$$\left.\frac{x^3}{3}\right|_0^1$$

Matrices and arrays. A basic matrix may be created using the matrix environment[3]: in common with other table-like structures, entries are specified by row, with columns separated using an ampersand (&) and a new rows separated with a double backslash (\\)

```
\begin{matrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{matrix}
```

$$\begin{matrix}
a & b & c \\
d & e & f \\
g & h & i
\end{matrix}$$

When writing down arbitrary sized matrices, it is common to use horizontal, vertical and diagonal triplets of dots (known as ellipses) to fill in certain columns and rows. These can be specified using the `\cdots`, `\vdots` and `\ddots` respectively:

```
A_{m,n} =
\begin{pmatrix}
  a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
  a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
  \vdots  & \vdots  & \ddots & \vdots  \\
  a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}
```

$$A_{m,n} =
\begin{pmatrix}
a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}$$

1.3 Tables

1.3.1 The Tabular Environment

The tabular environment can be used to typeset tables with optional horizontal and vertical lines. LaTeX determines the width of the columns automatically. The first line of the environment has the form:

```
  \begin{tabular}[pos]{table spec}
```

The *table spec* argument tells LaTeX the alignment to be used in each column and the vertical lines to insert. The number of columns does not need to be specified as it is inferred by looking at the number of arguments provided. It is also possible to add vertical lines between the columns here. The following symbols are available to describe the table columns (some of them require that the package array has been loaded):

<table>
<tr><td>l</td><td>left-justified column</td></tr>
<tr><td>c</td><td>centered column</td></tr>
<tr><td>r</td><td>right-justified column</td></tr>
<tr><td>p’width’</td><td>paragraph column with text vertically aligned at the top</td></tr>
<tr><td>m’width’</td><td>paragraph column with text vertically aligned in the middle (requires array package)</td></tr>
<tr><td>b’width’</td><td>paragraph column with text vertically aligned at the bottom (requires array package)</td></tr>
<tr><td>|</td><td>vertical line</td></tr>
<tr><td>||</td><td>double vertical line</td></tr>
</table>

The optional parameter pos can be used to specify the vertical position of the table relative to the baseline of the surrounding text. In most cases, you will not need this option. It becomes relevant only if your table is not in a paragraph of its own. You can use the following letters:

b	bottom
c	center (default)
p	top

In the first line you have pointed out how many columns you want, their alignment and the vertical lines to separate them. Once in the environment, you have to introduce the text you want, separating between cells and introducing new lines. The commands you have to use are the following:

&	column separator
\\	start new row (additional space may be specified after \\ using square brackets, such as \\[6pt])
\hline	horizontal line

1.3.2 Basic Examples

This example shows how to create a simple table in LaTeX. It is a three-by-three table, but without any lines.

```
\begin{tabular}{ l c r }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

1	2	3
4	5	6
7	8	9

Expanding upon that by including some vertical lines:

```
\begin{tabular}{ l | c || r }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

1	2	3
4	5	6
7	8	9

To add horizontal lines to the very top and bottom edges of the table:

```
\begin{tabular}{ l | c || r }
  \hline
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
  \hline
\end{tabular}
```

1	2	3
4	5	6
7	8	9

And finally, to add lines between all rows, as well as centering (notice the use of the center environment – of course, the result of this is not obvious from the preview on this web page):

```
\begin{center}
  \begin{tabular}{ l | c || r }
    \hline
    1 & 2 & 3 \\ \hline
    4 & 5 & 6 \\ \hline
    7 & 8 & 9 \\
    \hline
  \end{tabular}
\end{center}
```

1	2	3
4	5	6
7	8	9

1.3.3 Floating with Table

To tell LaTeX we want to use our table as a float, we need to place a tabular environment in a table environment, which is able to float and add a label and caption.

The table environment is a type of floats just as figure is. In fact, they bear a lot of similarities (positionning, captions, *etc.*).

```
\begin{table}[position specifier]
  \centering
  \begin{tabular}{table spec}
    ... your table ...
  \end{tabular}
  \caption{This table shows some data}
  \label{tab:myfirsttable}
\end{table}
```

You can set the optional parameter position specifier to define the position of the table, where it should be placed. The following characters are all possible placements. Using sequences of it define your "wishlist" to LaTeX.

h	where the table is declared (here)
t	at the top of the page
b	at the bottom of the page
p	on a dedicated page of floats
!	override the default float restrictions. E.g., the maximum size allowed of a “b” float is normally quite small; if you want a large one, you need this ! parameter as well.

Table 1 is an example of floating table.

```
\begin{table}[h]
  \centering
  \begin{tabular}{ l | c || r }
    \hline
    1 & 2 & 3 \\ \hline
    4 & 5 & 6 \\ \hline
    7 & 8 & 9 \\
    \hline
  \end{tabular}
  \caption{This table shows some data}
  \label{tab:myfirsttable}
\end{table}
```

1	2	3
4	5	6
7	8	9

Table 1: This table shows some data

1.4 Floats, Figures and Captions

1.4.1 Figures

To create a figure that floats, use the figure environment.

```
\begin{figure}[placement specifier]
... figure contents ...
\end{figure}
```

The placement specifier parameter exists as a compromise, and its purpose is to give the author a greater degree of control over where certain floats are placed. Specifier Permission

h	Place the float here, i.e., approximately at the same point it occurs in the source text (however, not exactly at the spot)
t	Position at the top of the page.
b	Position at the bottom of the page.
p	Put on a special page for floats only.
!	Override internal parameters LaTeX uses for determining "good" float positions.
H	Places the float at precisely the location in the LaTeX code. Requires the float package, *e.g.*, `\usepackage{float}`. This is somewhat equivalent to `h!`.

1.4.2 Captions

It is always good practice to add a caption to any figure or table. Fortunately, this is very simple in LaTeX. All you need to do is use the `\caption{’’text’’}` command within the float environment. Because of how LaTeX deals sensibly with logical structure, it will automatically keep track of the numbering of figures, so you do not need to include this within the caption text.

The location of the caption is traditionally underneath the float. However, it is up to you to therefore insert the caption command after the actual contents of the float (but still within the environment). If you place it before, then the caption will appear above the float.

Figure 1 and Figure 2 are two examples.

```
\begin{figure}[h!]
  \caption{A picture of a dog.}
  \centering
    \includegraphics[width=0.5\textwidth,
     natwidth=1024,natheight=768]{dog.jpg}\label{fig:dog}
\end{figure}
```

```
\begin{figure}[h!]
  \centering
    \reflectbox{ \includegraphics[width=0.5\textwidth,natwidth=1920,
     natheight=1080]{dog.jpg} }
  \caption{A picture of the same dog
           looking the other way!}
\end{figure}
```

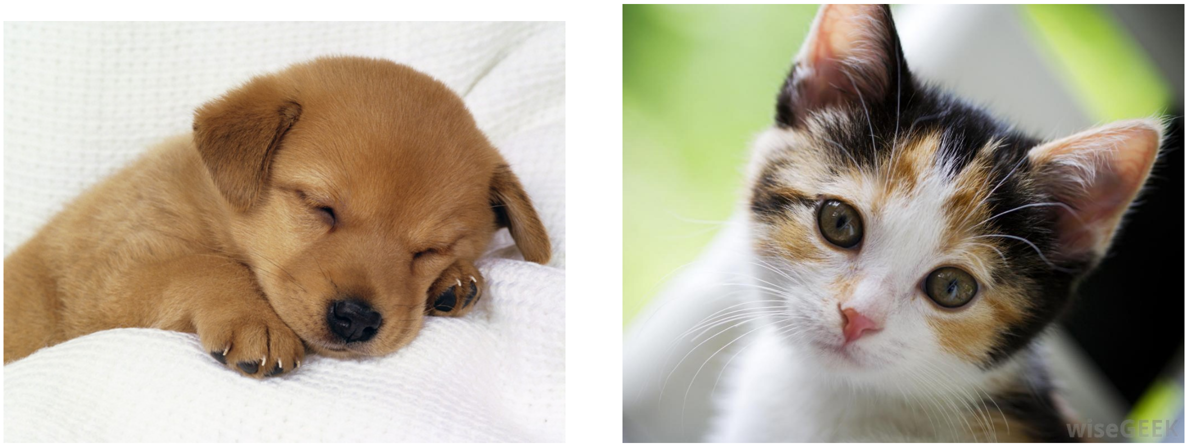
Figure 1: A picture of a dog.

Figure 2: A picture of the same dog looking the other way!

1.4.3 Multiple Figures in One Float

A useful extension is the `subcaption` package, which uses subfloats within a single float. This gives the author the ability to have subfigures within figures,or subtables within table floats. Subfloats have their own caption, and an optional global caption. An example will best illustrate the usage of this package:

```
\begin{figure}[t]
        \centering
        \begin{subfigure}[b]{0.45\textwidth}
                \centering
                \includegraphics[width=\textwidth,natwidth=1024,
```



(a) A dog

(b) A cat

Figure 3: Pictures of animals

```
                     natheight=768]{dog.jpg}
                \caption{A dog}
                \label{fig:dog2}
        \end{subfigure}%
         \qquad %add desired spacing between images, e. g. ~, \quad, \
           qquad etc.
          %(or a blank line to force the subfigure onto a new line)
        \begin{subfigure}[b]{0.45\textwidth}
                \centering
                \includegraphics[width=\textwidth,natwidth=1000,
                     natheight=781]{cat.jpg}
                \caption{A cat}
                \label{fig:cat}
        \end{subfigure}
        \caption{Pictures of animals}\label{fig:animals}
\end{figure}
```