Exercises for Further LaTeX Level 4

Susan Hutchinson
Department of Statistics,
University of Oxford

June 2009

Table of Contents

1 Before you start 2
2 Introduction 4
3 Long documents 5
4 Customising LaTeX 7
5 Creating a slide show 9
6 Exploring packages 10
7 Conclusion 14
8 Answers 15
Glossary 16

1 Before you start

We will be using OSS Watch Ubuntu Linux today for the exercises. The CD runs a 'live' version of Linux which does not change the hard disk. The OSS Watch version of Ubuntu Linux contains most of the commonly used features of LaTeX although some of the more obscure mathematical symbols may not be available. TeX and LaTeX work equally well on Windows, Macintosh and many other operating systems. These exercises were prepared using the OSS Watch Live Ubuntu CD and a memory stick. I made frequent backups!

More information about Ubuntu can be found at `http://www.ubuntu.com/` and information about OSS Watch can be found at `http://www.oss-watch.ac.uk/`.

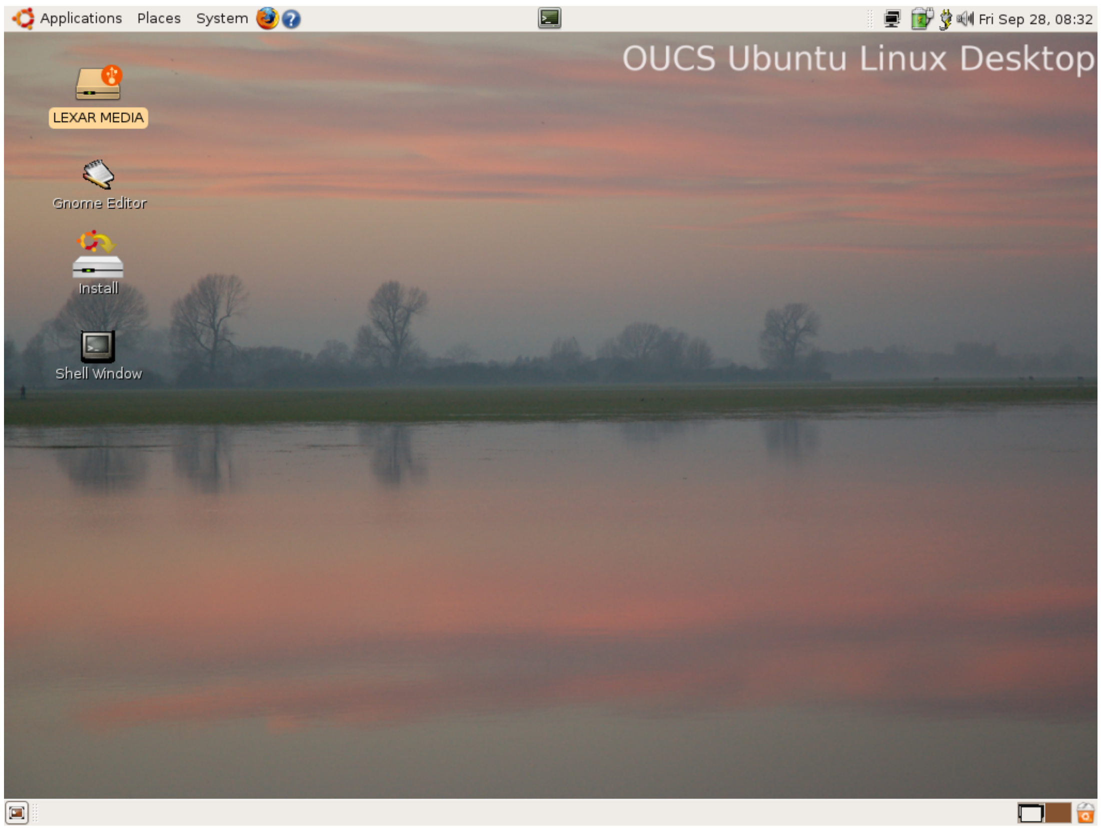

Figure 1: The OSS Watch Ubuntu start-up screen

If Ubuntu is not already running, put the CD into the CD drive of your PC and reboot the computer. A screen like Figure 1 should appear.

If a browser window opens describing the work of OSS Watch close it.

Follow these commands to download sample files to help with later exercises.

1. Create a new directory and use `cd` to change to the new directory.
2. Use

```
wget http://www.stats.ox.ac.uk/pub/susan/latex/samples.tgz
```

to download a bundle of useful files that you will need today. I have also included a copy of these exercises as some of the features are only obvious when viewed online.

3. Use **`tar -xzvf samples.tgz`** to unpack the files.

We will be using **`pdflatex`** to compile **`.tex`** files today. The **`pdflatex`** command offers a richer environment for graphics.

The exercises in this document are only suggestions. If there are other packages and techniques that you want to explore then please do so and we will do our best to help but priority will be given to those needing help with these exercises.

Most sets of exercises include a "If you have time" section. These are designed to develop a particular technique. Not all the information needed is necessarily included – you may need to search online or read the documentation.

Typesetting conventions

Distinct fonts and structures have been used to distinguish between various LaTeX features.

Text that needs to be typeset exactly as it appears in the exercises looks like this:

```
\begin{document}
\usepackage{parskip}
\begin{document}
Hello.
\end{document}
```

[For the curious this is achieved adding the `\usepackage{Verbatim}` to the preamble and enclosing the text in a `Verbatim` environment.]

Commands that are entered in the command tool window appear in a typewriter font **`pdflatex test.tex`**.

Finally whenever I need to display the output from a LaTeX compilation I have enclosed it between two lines.

Here's a reference to Figure 2.1 in Chapter 2.

---

2 Introduction

The first exercise is a refresher. Using your preferred editor (emacs, vi and gedit are all available) create a new file called **`first.tex`**.

▷ Exercise 1 Creating a sample file

Insert the following lines into the file

```
\documentclass[a4paper,12pt]{report}
\begin{document}
\title{My sample document}
\author{My Name}
\date{November 2007}
\maketitle
\pagenumbering{roman}
\tableofcontents
\chapter*{Acknowledgements}
\pagenumbering{arabic}
\chapter{Introduction}
\chapter{Background}
\chapter{Recent developments}
\chapter{Areas for study}
\chapter{Conclusion}
\end{document}
```

Most of the markup in the document should be familiar but please ask if any of the commands are not clear.

Compile your LaTeX file with **`pdflatex first.tex`** and preview it using **`xpdf first.pdf`**. There should be no errors, but there's not a lot of useful information here, either. We will develop this document as the course progresses.

From now on, I won't be giving exact commands like '**`pdflatex first.tex`**' but just say recompile your LaTeX file.

▷ Exercise 2 Changing the document

1. Add your name and a title to the title page.
2. Add some text to the 'First document' chapter

When you've saved the changes, recompile the file. You do not need to re-open the previewer, just reload it with '**`R`**'.

Before we go on to the next set of exercises make sure you are confident with the edit → compile → preview cycle. You will be doing this many times during the course.

3 Long documents

It is easier to manage longer documents like a thesis or dissertation if they are organised in separate files. An advantage of this approach is that certain chapters only can be selected for compilation. For example, a chapter with lots of pictures can be omitted to speed up the compile → edit → preview cycle while writing other parts of the document. If this file containing the chapter has previously been compiled then references to figures in that chapter will still work.

▷ Exercise 3 Splitting up a large document

Using the **`cp`** command copy **`first.tex`** to **`second.tex`**. Load **`second.tex`** into an editor. Now we are going to remove all the text associated with the first chapter and put it in a separate file. Of course the words you added in the previous exercise may well be different from mine!

1. Copy the lines

   ```
   \chapter{Introduction}
   Here are some words for the first chapter.
   There's not a lot but it's a start.
   ```

   into a new file called **`intro.tex`** and save.

2. Delete these lines from **`second.tex`**.
3. Add the line `\include{intro}` to **`second.tex`**.
4. Recompile **`second.tex`** and preview it.

It should look exactly the same as **`first.tex`**.

Now replace `\chapter{Background}` with `\include{background}` and create a new file called **`background.tex`** with the first line `\chapter{Background}` and recompile. Again, it should look exactly the same.

▷ Exercise 4 Conditional compilation

We are now going to test that references will work even if we leave out some chapter files. Now make these changes:

1. Add these lines to **`background.tex`**

   ```
   \label{ch:background}
   I'm going to add some words and a picture in the second chapter.
   \begin{figure}[ht]
   \centering
   \includegraphics{ox-crest.png}
   \caption{The University of Oxford crest}
   \label{fig:OxCrest}
   \end{figure}
   ```

2. Add `\usepackage{graphics}` to the preamble in **`second.tex`**.
3. Add these lines to **`intro.tex`**

```
Here’s a reference to Figure~\ref{fig:OxCrest} in
Chapter~\ref{ch:background}.
```

and recompile *twice* and preview the output.

You should see something like this in your preview window.

Here’s a reference to Figure 2.1 in Chapter 2.

Now we are going to run a conditional compilation, including only **`intro.tex`** and excluding **`background.tex`**.

1. Add the line `\includeonly{intro}` to the preamble of **`second.tex`**.
2. Save the file and recompile.

The new document will be missing Chapter 2 but the references will still be correct and there will still be an entry labelled ‘Background’ in the table of contents.

Advanced

If you have time put all the information in the preamble in a separate file. See if you can explain how the `\input{file.tex}` command differs from `\include{file.tex}`. You may find google useful. Would you use `\input{file.tex}` or `\include{file.tex}` for the information in the preamble?

Section 8 contains the answers.

4 Customising LaTeX

Often you will find that you are repeating the same LaTeX markup over and over. The next exercise demonstrates how to create your own commands which can then be reused throughout your document.

For example if we have many occurrences of the phrase 'Oxford University' in our document it is much quicker to add the line `\newcommand{\OU}{Oxford University}` to the preamble. Subsequently, whenever we want to write 'Oxford University' all that is needed is to include the markup `\OU` in the text.

▷ Exercise 5 Creating a new command

Using the **`cp`** command to copy **`second.tex`** to **`third.tex`**. Load **`third.tex`** into an editor. We are now going to add the new command to the preamble **`third.tex`** and make use it in both this document and in **`intro.tex`**.

You may want to remove or comment out (using a %) the `\includeonly...` line.

1. Add `\newcommand{\OU}{Oxford University}` to the preamble.
2. Add a suitable sentence such as 'I am grateful to `\OU` for the support I have received' to the Acknowledgements chapter. You may need to add {} after `\OU` to make sure the spacing is correct.
3. Add another sentence including `\OU` to your **`intro.tex`**.
4. Save all the files and recompile. Open a preview window and check that you can see 'Oxford University' in the text.
5. At this point you realise that it should be 'University of Oxford' not 'Oxford University'. Change the `\newcommand` line in **`third.tex`** and recompile.

This demonstrates the advantage of using `\newcommand` for frequently used phrases or commands. If you need to make a change, it is only necessary to do it in one place.

▷ Exercise 6 Using arguments with new commands

Let's assume our document is going to make reference to several different Universities. Whenever we do that we want to change the font to bold, and make the colour magenta.

1. We need to add `\usepackage{xcolor}` to the preamble.
2. Add a line

   ```
   \newcommand{\Uni}[1]{\textbf {\color{magenta} University of #1}}
   ```

   to the preamble.
3. Now add two or more entries to your text using the `\Uni` command. Perhaps you could have `\Uni{Manchester}` and `\Uni{Cambridge}`.
4. See if you can make the font bold and sans serif and change the colour to blue. You will need the `\textsf` command.

Chapter 8 contains answers where they don't appear in the text.

▷ Exercise 7 Customising existing commands

It is also possible to modify existing commands so that they behave as you want. The classic example of this is the `\today` command which by default produces

---

June 17, 2009

---

which is an American format. Many people prefer a date of the form 17 June 2009.

The standard definition of `\today` is

```
\newcommand{\today}{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}
```

Using `\renewcommand` see if you can change the format of the `\today` command. You will need to add this to the preamble. See chapter 8 for the answer.

Advanced

See if you can make the colour of your new `\Uni` command an argument as well so that the command is `\OU{green}{Leeds}` for example.

5 Creating a slide show

See separate document.

6 Exploring packages

Many packages are available in LaTeX. These add features or change the behaviour of existing packages. We will be looking at just a few commonly used packages in this set of exercises.

Using the **`cp`** command copy **`third.tex`** to **`fourth.tex`**. Load **`fourth.tex`** into an editor.

▷ Exercise 8 The **`fancyhdr`** package

The basic headers provided by LaTeX are rather limited. The **`fancyhdr`** packages adds more information in the header and footer and extends the customistations you can make, allowing you, for example, to add your name to the footer of each page.

1. Add `\usepackage{fancyhdr}` to the preamble.
2. Add `\pagestyle{fancy}` just after the `\maketitle`.
3. Compile **`fourth.tex`** and preview it. Do you notice any difference?

It could be that you can't see any difference to your document. If each chapter is only a page long then nothing will have changed. This is because header information is only added to the second and subsequent pages of any chapter. Add something like this to a chapter

```
\newpage
With a second page. It may be that headings
only appear on the second
page of a chapter.
\section{My big idea}
Here's the second page.

With a second paragraph.
\newpage
Again we need a new page to see what the
default headings look like.
\subsection{More detail}
Why this is an excellent idea.
\newpage
More information about my very good idea.
```

and recompile (possibly a couple of times if you want references to be correct). Now look at the document. You should notice

CHAPTER 1. INTRODUCTION

at the top of the second page. If you have included sections as in the example above you should see something like this.

1.1 MY BIG IDEA *CHAPTER 1. INTRODUCTION*

Notice how the titles and numbers of the chapter and section now appear in the header. We are now going to customise the fancyhdr package by including our name in the footer.

▷ Exercise 9 Using the **`texdoc`** command

LaTeX packages all come with documentation which can be read using the **`texdoc`** command. So to find out more about the **`fancyhdr`** package use **`texdoc fancyhdr`** in a command window.

From this document see if you can find out how to make these changes to the footer:

1. Add your name to the lefthand side of the footer.
2. Move the pagenumber from the centre to the righthand side.

▷ Exercise 10 Creating a glossary

I suggest you use **`finalglos.dvi`** to see how your final version will look. I'm going to write a paragraph to demonstrate how glossaries work. Glossaries usually appear at the end of documents, but can be referred to at any time. In particular, I will describe how glossaries work in LaTeX.

How I made it work

1. First I read the documentation using **`texdoc glossary`** . I found it quite difficult to understand as there was a *lot* of information. Try it!
2. Googled. When I searched for 'glossary package latex' the first link to `http://theoval.sys.uea.ac.uk/~nlct/latex/thesis/node25.html` which contained a useful summary. Note that this page was written by the author of the glossary package and so is likely to be authoratative.

Cut and paste this into a separate file called glos.tex starting here:

```
\documentclass[12pt,a4paper]{article}
\usepackage{glossary}
\makeglossary
\begin{document}
I'm going to write a paragraph
\glossary{name=paragraph,description=sentences
on a linked theme}
to demonstrate how glossaries \glossary{name=glossary,description=an
explanation of terms}
work. Glossaries usually appear at the end of documents, but can be
referred to at any time. In particular, I will describe how glossaries
work in  \LaTeX \glossary{name=latex,description=a mathematical typesetting  lan
\printglossary
\end{document}
```

We will be using **`latex`** rather than **`pdflatex`** for this exercise as the markup is slightly simpler for xdvi files.

```
latex glos.tex
makeindex -t glos.glg -o glos.gls -s glos.ist glos.glo
latex glos.tex
latex glos.tex
```

[*Note that the* `makeglos.pl` *command doesn't work on this live system.*]

Now preview your file to see what it looks like with `xdvi glos.dvi`.

Here's my version:

I'm going to write a paragraph to demonstrate how glossaries work. Glossaries usually appear at the end of documents, but can be referred to at any time. In particular, I will describe how glossaries work in LaTeX .

Let's see how we can make this better. I would like the words in the text to be linked to entries in the glossary. Reading the documentation it appears that `xglossary` would work as long as the `\hyperref` package is used. So we now have

```
\documentclass[12pt,a4paper]{article}
\usepackage{hyperref}
\usepackage{glossary}
\makeglossary
\begin{document}
I'm going to write a paragraph \xglossary{name=paragraph,
description=sentences on a linked theme}
to demonstrate how glossaries \xglossary{name=glossary,
description=an explanation of terms}
work. Glossaries usually appear at the end of
documents, but can be referred to at any time.
In particular, I will describe how glossaries work
in \LaTeX\ \xglossary{name=latex,description=a mathematical
typesetting language}. I will also describe amendments
\xglossary{name=amendments,description=a set of
changes or improvements} that can be made.
\end{document}
```

And my version:

I'm going to write a paragraph to demonstrate how glossaries work. Glossaries usually appear at the end of documents, but can be referred to at any time. In particular, I will describe how glossaries work in LaTeX for example. I will also describe amendments that can be made.

I noticed that when I viewed the file the links were to the first letter of the word *after* the glossary not to the word I intended. So I moved the words to after the glossary entry and surrounded them by {}. The paragraph now looks like this:

```
I'm going to write a \xglossary{name=paragraph,
description=sentences on a linked theme}{paragraph}
to demonstrate how \xglossary{name=glossary,
description=an explanation of terms}{glossaries}
work. Glossaries usually appear at the end of
```

```
documents, but can be referred to at any time.
In particular, I will describe how glossaries work
in \xglossary{name=latex,description=a mathematical
typesetting language}{\LaTeX}. I will also describe
\xglossary{name=amendments,description=a set of
changes or improvements}{amendments} that can be made.
```

And again:

I'm going to write a paragraph to demonstrate how glossaries work. Glossaries usually appear at the end of documents, but can be referred to at any time. In particular, I will describe how glossaries work in LaTeX. I will also describe amendments that can be made.

What happens when a new entry is added? Run the command `makeindex -t glos.glg ...` again.

Finally what happens if I want to reuse entries. It makes sense to create definitions separately and then use a shortcut when linking text to the glossary. So my final version is:

```
\storeglosentry{glos:P}{name=paragraph,
description=sentences on a linked theme}
\storeglosentry{glos:G}{name=glossary,
description=an explanation of terms}
\storeglosentry{glos:L}{name=latex,
description=a mathematical typesetting
language}
\storeglosentry{glos:A}{name=amendments,
description=a set of changes or improvements}

I'm going to write a \useGlosentry{glos:P}
{paragraph} to demonstrate how \useGlosentry{glos:G}
{glossaries} work. \useGlosentry{glos:G}{Glossaries}
usually appear at the end of documents, but can be
referred to at any time. In particular, I will
describe how glossaries work in \useGlosentry{glos:L}
{\LaTeX}. I will also describe \useGlosentry{glos:A}
{amendments} that can be made.
```

And finally...

I'm going to write a paragraph to demonstrate how glossaries work. Glossaries usually appear at the end of documents, but can be referred to at any time. In particular, I will describe how glossaries work in LaTeX. I will also describe amendments that can be made.

7 Conclusion

This course has only touched on more advanced features of LaTeX.

8 Answers

\include or \input?

See page 6. I think it makes more sense to use `\input{file.tex}` if you store the information from the preamble in a separate file. This information is always needed. It is also useful to do this so that it can be reused whenever you start a new LaTeX document.

Using arguments with new commands

See page 7.

```
\newcommand{\Uni}[1]{\textsf {\textbf {\color{blue} University of #1}}}
```

Customising existing commands

See page 8.

```
\renewcommand{\today}{\number\day\space \ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
   \space\number\year}
```

Advanced

```
\newcommand{\Uni}[2]{\textsf {\textbf {\color{#1} University of #2}}}
```

Exploring packages

Reading this sort of documentation is not always easy, I find. There is often a lot of detail, particularly at the beginning and there is rarely a simple 'Getting started' section or instructions for frequently-used customisations. I must admit that I struggled to find the answer here and had to resort to Wikipedia! I added the following the preamble beneath `\usepackage{fancyhdr}`.

```
\fancyfoot{}
\lfoot{Susan Hutchinson}
\cfoot{}
\rfoot{\thepage}
```

This was quite easy to find out; the problem I had was that the page number was still centered on the first page of each chapter and no name appeared. This is perfectly acceptable, but I wanted my footer on *all* the pages. The solution was to change `\pagestyle{fancy}}` to `\pagestyle{fancyplain}}`

Glossary

amendments	a set of changes or improvements	12, 13
glossary	an explanation of terms	11, 12, 13
latex	a mathematical typesetting language	11, 12, 13
paragraph	sentences on a linked theme	11, 12, 13