LaTeX and Linux practical StatML CDT 2019

Susan Hutchinson
Department of Statistics,
University of Oxford

October 2019

Contents

List of Figures

1 Introduction

This document introduces the Linux command line with exercises to explore the features described. Following this, there are some exercises to create and print LaTeX documents using the Texmaker editor.

This file available at **http://www.stats.ox.ac.uk/pub/susan/statml/Exercises.pdf**. I recommend that you open a copy as many of the links are clickable and it will be easy to copy Linux commands and LaTeX text in the later exercises.

2 The Linux command line

You will need a Linux terminal for these exercises. There are many ways of accessing this:

- On a local Linux system.
- On a remote Linux system. To access a remote system use ssh (see later) or use PuTTy on Windows.
- The Mac terminal utility.
- For Windows users, the Linux Bash Shell is available in the latest version.

For later exercises, access to a remote server is needed.

Download some files

Download some files and directories which will be used during these exercises. Although it is possible to use a browser to download this file you can also do this from the command line.

In a terminal window type in the following commands exactly as they appear. The commands will all be explained later.

```
cd
curl http://www.stats.ox.ac.uk/pub/susan/linux/StatML.tgz >StatML.tgz
```

to download the files and then

```
tar -xvzf StatML.tgz
```

to unpack them.

> In the following sections Linux commands are introduced and explained, each followed by examples and exercises to illustrate some commonly used features. Depending on how you are using Linux, some details, particularly file locations, may differ.

2.1 Where am I?

This exercise explores Linux file system navigation.

Note that all commands are typed in lower case. There are very few Linux commands which have any uppercase (CAPITAL) letters.

You will be using these commands:

Command	**Purpose**
`cd`	Change directory. In other words, change my current location.
`pwd`	Print working directory. In other words, "where am I?"
`ls [options]` *`directory`*	List files. If used on its own, it lists everything in the current working directory (where you are currently located).
`file` *`filename`*	Tells you what sort of file the file named *filename* (for example) is.

Table 1: Navigation and file query commands

Right away we will see how quiet Linux commands are by default. Try typing in

```
cd
```

at the prompt and you will get no output at all. This does not mean that anything has gone wrong. For many commands, no output means successful completion.

A digression on prompts:
You can customise your prompt to look however you like. We won't do that now, but you may notice that it changes as you move around the file system.

Not all commands are silent. Try

```
pwd
```

You should get a response like: `/homes/`*`user`*. In all examples *`user`* refers to your username. Now try

```
ls
```

You should now see a listing of all the files in the directory `/homes/`*`user`*. Let's find out about some of the contents of `/homes/`*`user`* using the `file` command.

```
file TestDir
```

The shell tells you that this isn't a regular file, it's a directory. In other words it's a special file which acts as a holder for yet more files (like a folder in Windows).

```
file longfile.txt
longfile.txt:  ISO-8859 text
```

so this file is a simple text file.

2.1.1 Absolute and relative paths

We will make use of the following character shortcuts.

Special character	Description	Purpose
.	One dot	The current directory.
..	Two dots	The directory above the current directory.
/	A slash or forward slash	The root or top directory.
~	The tilde character	Your home directory.

Table 2: Symbols used by `cd`

Let's explore the idea of relative and absolute path names using the cd command. Change into the directory `TestDir`.

Note that if you can't find `TestDir` then make sure you have downloaded and unpacked the files as described on page 1 in the **Download some files** section.

```
cd ~/TestDir
pwd
cd ..
pwd
cd ..
pwd
```

and so on until you can't go any further (you won't see an error, you just stop going anywhere). `..` is a special location which means up one level. All directories contain a `..` so you can go up a level. The exception is called / or sometimes "the root" or just "slash". You can't go any higher than / so `..` doesn't take you anywhere. Note that there is another special directory called `.` (a single dot) which means "current location".

During the above task you went up the directories one level at a time. Now let's reverse the process and go back to the Desktop directory one level at a time. You should be in "/". You don't need the pwds but it may help you see what is going on.

```
cd homes
pwd
cd user
pwd
cd TestDir
pwd
```

Remember always to replace *`user`* by your own username.

We're now going to make use of two things, the `ls` command and the knowledge that the file called `/usr/share/misc` is a directory, to further illustrate the concepts of absolute and relative pathnames.

```
cd
cd ../../usr/share
ls misc
```

and you will get a listing of the contents of the directory.

```
ls /usr/share/misc
```

and you should get the same list of files.

The absolute (i.e. complete) location of the misc directory is `/usr/share/misc`. We have just asked to see what is kept inside it in two different ways. The first is a relative pathname while the second is the full or absolute path. Imagine the `misc` directory is a particular house, say 42, High Street, Abingdon and I ask you to deliver a letter there. I could tell you to deliver the letter to "42 High Street, Abingdon": the full/absolute address. No matter where you are in the UK, that's enough information. However, if you were already in Abingdon I could tell you to deliver the letter to the relative address of " 42, High Street" or even better, if you were standing on the high street just "number 42" would be enough. The `ls misc` command worked because you were already in the `/usr/share/` directory. It wouldn't work from another location. The command `ls /usr/share/misc` command will work from anywhere (although it's more long winded). Let's prove it by changing our current location using the cd command.

```
cd
cd TestDir
pwd
```

you should get `/homes/`*`user`*`/TestDir` i.e. you have moved into the TestDir directory.

```
ls /homes/user/TestDir
```

should give you a list of files in that directory. You could use `ls` on its own without the name of the directory because you have already moved there with `cd`. Let's see what happens when we make a mistake:

```
ls TestDir
```

should give you an error saying there is `No such file or directory` which is correct. The command fails because `TestDir` on its own is a relative path and you've started from the wrong place.

Try to answer/do the following:

1. Were you just using absolute or relative paths in the last example?
2. Now try to get back to the root (or `/`) directory with one command only using an absolute path.
3. Now get back to the `/homes/`*`user`*`/TestDir` directory using one command only.
4. What are the contents of the `/` directory? From your home directory use only one command to find out.
5. Make sure you can see your data directory: `/data/`*`host`*`/`*`user`*.

See Section 5 on page 26 for answers.

Finally, a short cut! In order to change to a directory in your home directory, use the ~ character which represents your home directory to move to `TestDir` from any location.

```
cd ~/TestDir
```

2.2 File and directory manipulation

Now we're going to create a directory and put some files there.

Command	**Purpose**
`cd`	Change directory – or change my current location.
`mkdir` *`directoryname`*	Create a directory called *`directoryname`*.
`touch` *`file1 file2`*	Create one or more empty file(s) called *`file1`*, *`file2`* ...
`cp` *`file1 file2`*	Copy *`file1`* to *`file2`*. Also used to copy directories.
`ls`	List files.
`rm` *`file1`*	Remove (or delete) a file *`file1`*. Also remove directories.

Table 3: File and directory manipulation commands

Now execute these commands:

```
cd
mkdir directory1
cd directory1
touch file1 file2 file3 file4
```

Words in italic should be replaced by names that you have chosen. Experiment to see what happens if you are not in your home directory. What happens if you try to create a directory in `/usr/bin`? Is there anywhere outside your home directory where you **are** allowed to create directories? [Hint: look at the top level directory – you should be able to create a files and directories in one of those. The name of the directory might also be a clue.]

Use the `cp` command to copy one file to another and then use `ls` to check that you have done what you want. Then delete a file using

```
rm file1
```

Now we are going to copy one directory to another. The commands you need are

```
cd
cp -r directory1 directory2
```

Use ls to make sure you have done what you want. The new directory should contain exactly the same files as the old one. Note use of the -r option. This makes cp copy the contents of a directory — this is known as a recursive copy. Finally remove the new directory with

```
rm -rf directory2
```

Note that this is a dangerous command and should be used with care!

Use `ls` to check that this has worked.

You should now be familiar with these simple file manipulation commands. Remember that in Linux the `rm` command really does delete files. There is no Recycle Bin to retrieve files that were deleted by mistake.

2.3 Viewing files

In these exercises we explore commands to view the contents of files.

Command	**Purpose**
`cat` *`file`*	Show the while contents of a file called *`file`*.
`less` *`file`*	Display the contents of *`file`* a screenful at a time, but with more options. For example, after starting `less` enter `G` to go straight to the end of a file and then move backwards.

Table 4: Viewing the contents of files

Use the following commands to look at the contents of the file `google.txt`.

```
cd
cd Files
cat google.txt
```

This is not very useful if the file is more than a screenful.

```
less google.txt
```

Note that <space> takes you to the next page and `q` will quit before the end of the file.

See if you can get to the end of the file with a single command. Use google to help if you like. Then use `q` to exit.

Speed things up

By now you may be fed up with all this typing! When using the command line in a terminal window, there are ways to make life easier for you:

Filename and command completion

- <tab> key completes commands and filenames

Arrow keys allow us to:

- recall previous commands
- change previous commands

2.4 Help commands

These exercises explore various ways to get help with and about commands. If you know what command you need, you can use the `man` command to find out the details of that command. Try it with a few of the commands you have used already.

Command	**Purpose**
`man` *`command`*	Read the manual page for a command. So `man ls` would give details of the `ls` command and `man less` of the `less` command.
`apropos` *`word`*	Search the manual pages for names and descriptions which contain *`word`*. So `apropos copy` would list all the commands that have the word `copy` in the description.
`which` *`command`*	Display the location of the command being used.
`whatis` *`command`*	Gives a brief description of a command.
`command` `--help`	Gives a brief description of a command then displays the options.

Table 5: Linux help commands

```
man ls
```

to find out details of the `ls` command.

i. What option is used to display modification time?
ii. What option is used to display the size of a file?
iii. How can you reverse the order of the sort so that the largest/most recently changed file is at the bottom of the list?

Sometimes you might not be sure exactly what command you need. In that case you can use the `apropos` command which finds all command descriptions which match a given word. So to find out what commands there are to manipulate files are available use

```
apropos file
```

Note that the output from this command is very long! In a following exercise we explore commands which make this more useful.

Sometimes you need to know where Linux stores a command. Use `which` to display the location of the file. Try it with `less`, `cp`, `apropos`:

```
which less
which cp
```

Finally you may have seen a command and want to know briefly what it does. Use the `whatis` command to find out. Try this on some commands you have already used.

2.5 Looking at parts of a file

We have already used `cat` and `less` to look at the contents of files. But what if we just want to look at parts of a file, or find out some characteristics of a file such as the number of lines or words?

Command	**Description**
`head` *`file`*	Display the first 10 lines of a file.
`tail` *`file`*	Display the last 10 lines of a file.
`wc` *`file`*	Counts the number of characters, words and lines in a file.

Table 6: Displaying parts of a file

Using

```
cd
cd Advanced
cat longfile.txt
```

you see all of `longfile.txt`; using

```
less longfile.txt
```

gives you a screenful at a time. Use the spacebar to move on a screenful, G to go to the end of the file and q to exit. Now use

```
head longfile.txt
tail longfile.txt
```

to look at the first 10 lines with `head` and last 10 lines with `tail`.

Now use the `wc` command to find out how many lines, words and characters in `longfile.txt`.

Remember that most commands have a `--help` option which displays helpful information.

2.6 Using wildcards to match file names

Special character	**What it matches**
`*`	The * (asterisk or star) matches any number of characters or none.
`?`	The ? (question mark) matches exactly one occurrence of any character.
`[ ]`	Matches any characters in a given range.

Table 7: Wildcard special characters

File globbing or wildcard expansion allows you to use special characters to match more than one file or directory name.

Change to the directory called `WildCards`.

```
cd ~/WildCards
```

Now experiment with wild card characters. What do the following match?

```
ls foo?
ls foo2*
ls foo[1-2]
```

What command would you need to match only the files `foo20` and `foo2bar`? [Hint: you might need to use more than one wild card character.] The answers are at in Section 5 on page 26. Now use

```
wc -l *
```

to find the length of each file. Note that the -l is a hyphen (-) followed by the lowercase letter `l`. You should see output like this:

```
  1 foo
  2 foo1
  2 foo10
  1 foo2
  1 foo20
  3 foo2bar
 10 total
```

Now see if you can create a match so that `wc -l` just shows the files with a 1 (the number one) in their name. Again a possible answer is at the end.

2.7 Pipes and redirection

In this exercise we are going to explore two very powerful command-line features which increase the flexibility and range enormously. We will use these commands:

Command	**Description**
`du -sk`	Display the sizes in Kilobytes of all the files in a directory.
`grep`	Search for the characters with a given pattern in file.
`sort`	Sorts the contents of one or more files.
`tail`	View the last few lines of a file.
`wc`	Counts the number of characters, words and lines in a file.

Table 8: Commands used in our pipe

and the following characters:

Special character	**Purpose**
>	Sends the output from a command to the named file. If the file already exists the previous contents will be lost. If the file doesn't exist it will be created.
≫	Appends the output from a command to the named file. If the file doesn't exist it will be created.
<	Reads input from the named file. NB This option is rarely used.
|	Uses the output from one command as the input to the next.

Table 9: Characters used for pipes and redirection

Almost all Unix/Linux commands use standard input for receiving instructions and standard output for displaying the results. Commands make use of special data streams to move input and output to and from the command. STDIN can be though of as a gateway into the code, STDOUT is a gateway for output. Note that there is a third standard data stream called STDERR (standard error) which commands often use to print error messages and warnings. We won't mention STDERR again today. By default STDOUT gets routed to the screen display. It is also easy to connect a commands STDIN to the keyboard device for example. The STDIN allows a program to ask you questions and you can type responses.

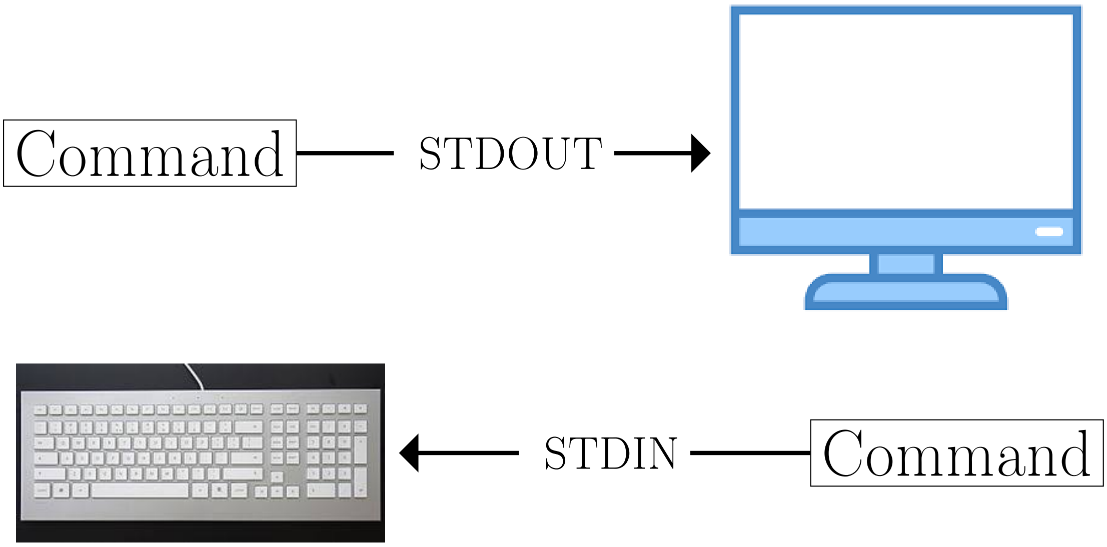


Figure 1: Standard output and standard input

Redirection is a way of "grabbing" STDOUT or STDIN and forcing it to go somewhere other than the default. The most common use is to redirect STDOUT into a new file. This is extremely useful because it means you can run a command and save the results automatically. To redirect STDOUT use the > symbol followed by the target so *command* > file.txt redirects the output from *command* into a new file called file.txt. To redirect STDIN into a command use < like this: *command* < file.txt.

Pipes are an even more useful feature that uses STDIN and STDOUT. It is possible to connect commands together; the output from one command becomes the input to the next command.

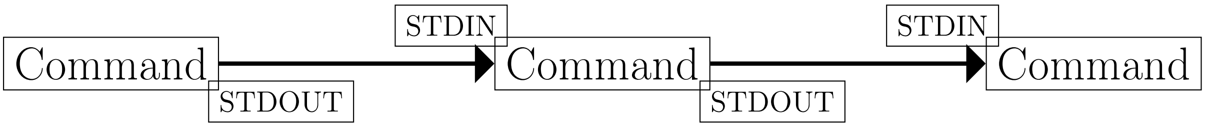


Figure 2: Using pipes

Connecting commands together with pipes is one of the most powerful features of Linux. Linux does not have a command to count the number of files in a directory but it does have one command to list the files (ls) and a second command (wc) to count the number of lines in a list.

You could therefore use a pipe (`|`) to glue `ls` and `wc` together:

```
ls | wc -l
```

Note that we are using the -l (that is a hyphen followed by the lower case letter l) option for wc. To get information about any command use

`man` *`command`* or *`command`* `--help`

eg

```
man wc
wc --help
```

2.7.1 Examples using redirection and pipes

In the directory `Searching` there are three small text files. Check their contents using `cat`. If we wanted to search for all occurrences of “green” in the files `fruit` and `veg` we would use

```
grep green fruit veg
```

The output is displayed on the screen. To store the output from the command in a file use

```
grep green fruit veg > out.txt
```

which would store the output of the `grep` command in a file called `out.txt`. Check this with

```
cat out.txt
```

If you run the command again, this time using

```
grep green fruit veg ≫ out.txt
```

you should see that there are now two copies of the output in the file. It is also possible to use < redirect the input from the keyboard to a file. For example

```
cat <out.txt
```

also works.

In this short example we could just count how many times “green” appears, but when there are many matches it would be useful to use `wc` to find out. We can redirect the output from the command into a file and then check the length. Note that in all cases the `-l` option to `wc` is a minus sign followed by the lower case letter `l`.

```
grep green fruit veg >out.txt
```

and then run `wc` on the file

```
wc -l out.txt
```

but it would be much more efficient to join the two commands together with a pipe. Use

```
grep green fruit veg | wc -l
```

Here the two commands `grep green fruit veg` and `wc -l` are joined together by a special symbol called a pipe.

As a final example, we’re going build a longer command which will find the 5 largest files in a directory. When building pipes of commands it often helps if you make sure each link in the pipe works before adding the next. First display the size of all the files in the `/usr/bin` directory.

```
du -sk /usr/bin/*
```

Now sort this output by size, so that the largest are first.

```
du -sk /usr/bin/* | sort -nr
```

Now display the final 5 lines which will be the 5 largest files.

```
du -sk /usr/bin/* | sort -nr | head -5
```

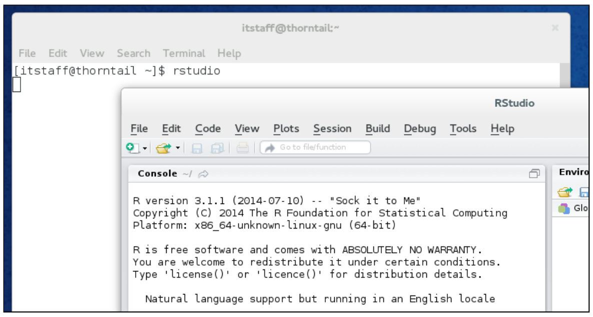
Now how you would you find the 5 largest files in the `/usr/bin` directory beginning with the letter ’s’? See section 5 on page 26 for a solution.

3 Linux process control

This section explores how to manage and control processes. Running any command creates a process; this process can then be monitored and managed. The sections below explore frequently used ways of doing this.

3.1 Running commands in the background

When you start a command such as `rstudio` from a terminal window it is good practice to add an & (ampersand) character after the command to keep access to the command line. Compare Figure 3 and Figure 4.

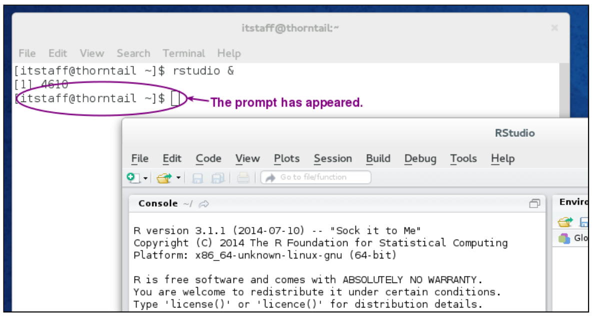


itstaff@thorntail:~
File Edit View Search Terminal Help
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Console ~/
Enviro
Glo

```
[itstaff@thorntail ~]$ rstudio
```

```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-unknown-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale
```

Figure 3: Starting `rstudio` without &.



itstaff@thorntail:~
File Edit View Search Terminal Help
The prompt has appeared.
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Console ~/
Enviro
Glo

```
[itstaff@thorntail ~]$ rstudio &
[1] 4610
[itstaff@thorntail ~]$
```

```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-unknown-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

Figure 4: Starting `rstudio` with &.

3.2 Finding process information

Command	**Purpose**
`ps`	Display a snapshot of current processes.
`sleep`	Do nothing for a specified time.
`pgrep`	Search for a particular process.

Table 10: Finding processes

Start a sleep process in the background to run for 5 minutes.

```
sleep 300 &
```

We will now use `ps` find the **P**rocess **ID**entifier or **PID**. First run

```
ps
```

which shows all the processes you own. If there are many processes, then it is helpful to find a particular one:

```
ps | grep sleep
```

which should display something like:

```
31481 pts/8    00:00:00 sleep
```

The number in the first column is the PID. Using options to `ps` gives more information about the process. There are many options to `ps`. Try

```
ps -A
ps -efl
```

and see if you can work out what you are seeing. `ps --help all` may help.

If you know the command you are looking for, then

```
pgrep sleep
```

also works.

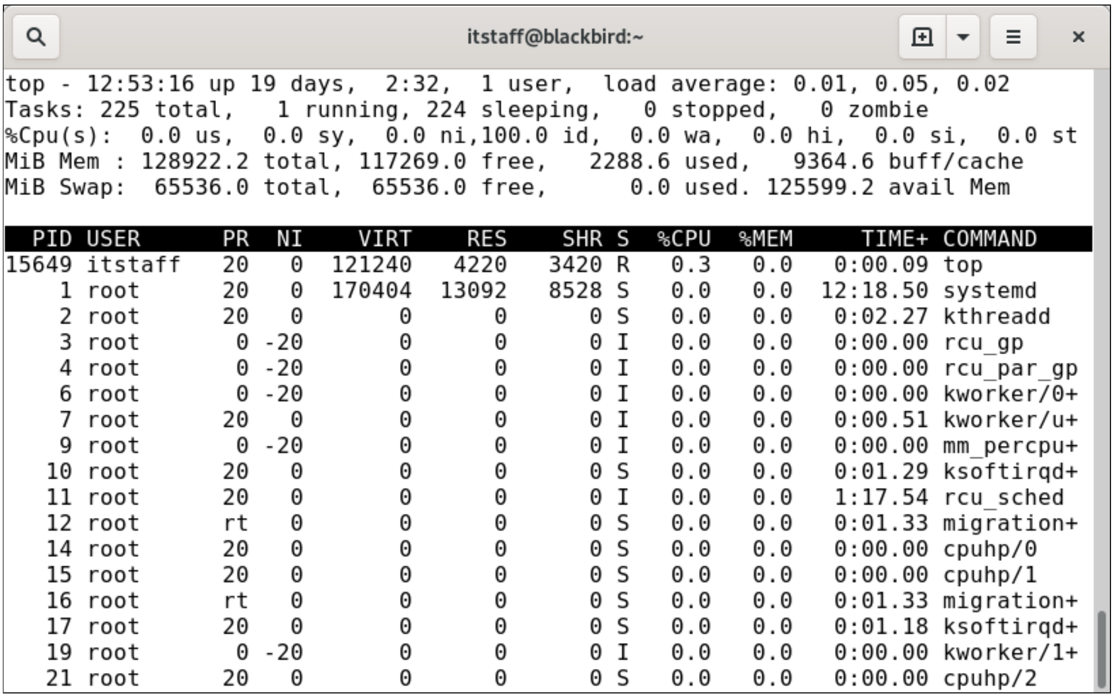
3.3 Looking at process resources

Command	Purpose
`top`	Display processes.
`htop`	Interactive process viewer.

Table 11: Monitoring processes

The `top` command shows how much of the systems resources processes are using. It runs until you tell it to quit, and updates every second.

```
top
```



```
top - 12:53:16 up 19 days,  2:32,  1 user,  load average: 0.01, 0.05, 0.02
Tasks: 225 total,   1 running, 224 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 128922.2 total, 117269.0 free,   2288.6 used,   9364.6 buff/cache
MiB Swap:  65536.0 total,  65536.0 free,      0.0 used. 125599.2 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
15649 itstaff   20   0  121240   4220   3420 R   0.3   0.0   0:00.09 top
    1 root      20   0  170404  13092   8528 S   0.0   0.0  12:18.50 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:02.27 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0+
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.51 kworker/u+
    9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu+
   10 root      20   0       0      0      0 S   0.0   0.0   0:01.29 ksoftirqd+
   11 root      20   0       0      0      0 I   0.0   0.0   1:17.54 rcu_sched
   12 root      rt   0       0      0      0 S   0.0   0.0   0:01.33 migration+
   14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   16 root      rt   0       0      0      0 S   0.0   0.0   0:01.33 migration+
   17 root      20   0       0      0      0 S   0.0   0.0   0:01.18 ksoftirqd+
   19 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1+
   21 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
```

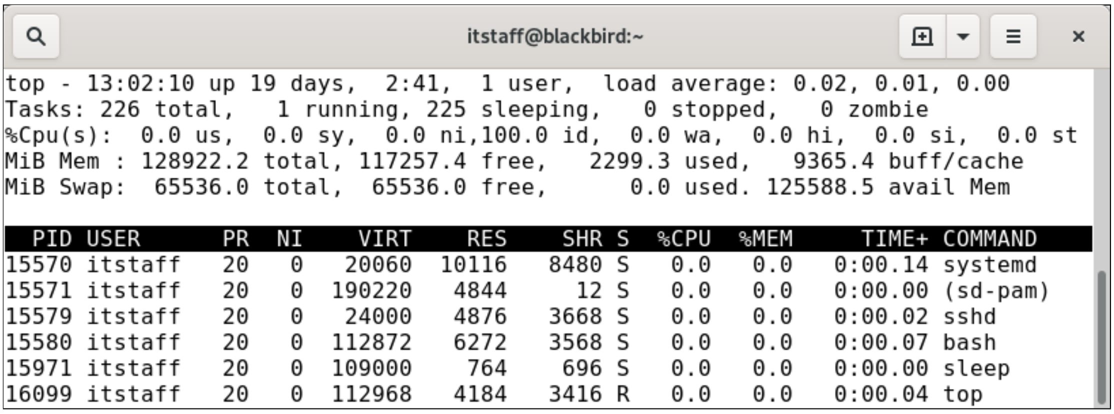
Figure 5: top on a quiet system

Use `q` to leave top.

Note that, even though there is a sleep process running it doesn't appear, as it is using no resources. To restrict top processes to just ones you own use

```
top -u USER
```

replacing *USER* by your own username.



```
top - 13:02:10 up 19 days,  2:41,  1 user,  load average: 0.02, 0.01, 0.00
Tasks: 226 total,   1 running, 225 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 128922.2 total, 117257.4 free,   2299.3 used,   9365.4 buff/cache
MiB Swap:  65536.0 total,  65536.0 free,      0.0 used. 125588.5 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
15570 itstaff   20   0   20060  10116   8480 S   0.0   0.0   0:00.14 systemd
15571 itstaff   20   0  190220   4844     12 S   0.0   0.0   0:00.00 (sd-pam)
15579 itstaff   20   0   24000   4876   3668 S   0.0   0.0   0:00.02 sshd
15580 itstaff   20   0  112872   6272   3568 S   0.0   0.0   0:00.07 bash
15971 itstaff   20   0  109000    764    696 S   0.0   0.0   0:00.00 sleep
16099 itstaff   20   0  112968   4184   3416 R   0.0   0.0   0:00.04 top
```

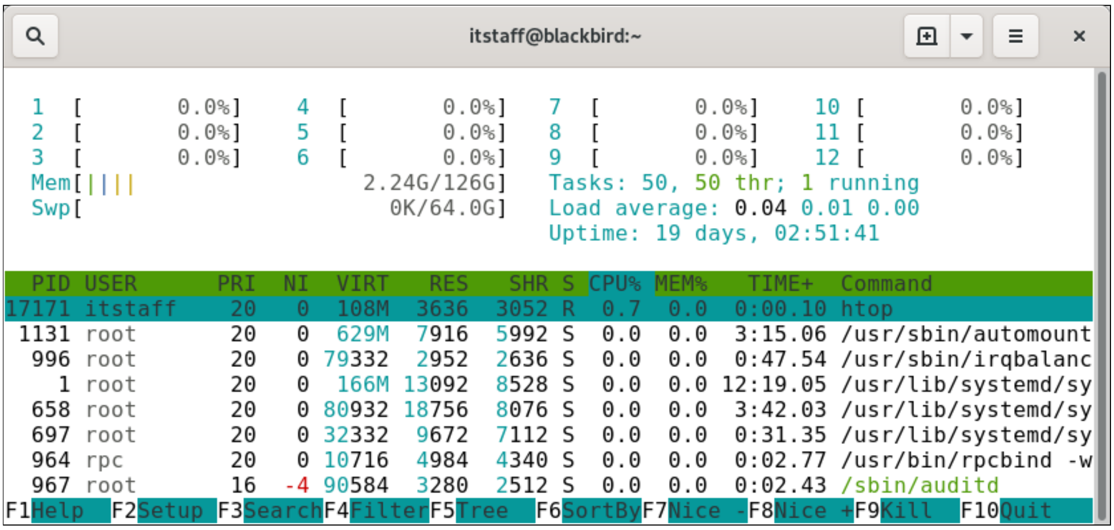
Figure 6: top showing just one user's processes

I am not going to explain the details of what top displays now, but use `man top` to find out more.

Finally, if you are running jobs on a shared server which will have many processors, use

```
htop
```

to display processor usage. Using `htop -u USER` will show information about all processors, but only your processes.



```
  1  [          0.0%]   4  [           0.0%]   7  [           0.0%]   10 [           0.0%]
  2  [          0.0%]   5  [           0.0%]   8  [           0.0%]   11 [           0.0%]
  3  [          0.0%]   6  [           0.0%]   9  [           0.0%]   12 [           0.0%]
  Mem[||||                      2.24G/126G]   Tasks: 50, 50 thr; 1 running
  Swp[                           0K/64.0G]   Load average: 0.04 0.01 0.00
                                             Uptime: 19 days, 02:51:41

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
17171 itstaff    20   0  108M  3636  3052 R  0.7  0.0  0:00.10 htop
 1131 root       20   0  629M  7916  5992 S  0.0  0.0  3:15.06 /usr/sbin/automount
  996 root       20   0 79332  2952  2636 S  0.0  0.0  0:47.54 /usr/sbin/irqbalanc
    1 root       20   0  166M 13092  8528 S  0.0  0.0 12:19.05 /usr/lib/systemd/sy
  658 root       20   0 80932 18756  8076 S  0.0  0.0  3:42.03 /usr/lib/systemd/sy
  697 root       20   0 32332  9672  7112 S  0.0  0.0  0:31.35 /usr/lib/systemd/sy
  964 rpc        20   0 10716  4984  4340 S  0.0  0.0  0:02.77 /usr/bin/rpcbind -w
  967 root       16  -4 90584  3280  2512 S  0.0  0.0  0:02.43 /sbin/auditd
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Figure 7: htop on a quiet system

In this example, no processors are being used.

3.4 Killing, suspending and restarting processes

Command	Purpose
`kill`	Send a signal to process.
`jobs`	Show status current jobs.

Table 12: Controlling processes

Jobs can run out of control, or too many processes may be running. To terminate a job, use

```
kill PID
```

where *PID* is replaced by the Process IDentifier, as described above. If that doesn't kill the job, use

```
kill -9 PID
```

Try this by starting a sleep process, then killing it.

It is not always necessary to kill a process, just stop it while other processes complete. To stop a job use

```
kill -STOP PID
```

and then to resume it use

```
kill -CONT PID
```

If you have many jobs running it can be difficult to remember the status of each job. Use

```
jobs
```

to list jobs.

3.5 Control sequences

Another way to manage processes is with control sequences. Technically, these control sequences send a particular signal to a process. Here is a brief summary of some frequently used command line control sequences.

CTRL-C interrupt a running program.

CTRL-D send an end of file, ending text input for most Linux/Unix programs

CTRL-Q unfreezes the screen if CTRL-S has been used to freeze it.

CTRL-U deletes the last line typed

CTRL-Z suspends a running program. Use `bg` to continue running the program in the background or `fg` to continue running the program as it was.

It is also possible to move a process from the foreground to the background. If you have forgotten to include `&` at the end of a line, use

```
CTRL-Z
```

to suspend the process. You should then see the command prompt, then use

```
bg
```

to restart the process in the background.

3.6 Low priority jobs

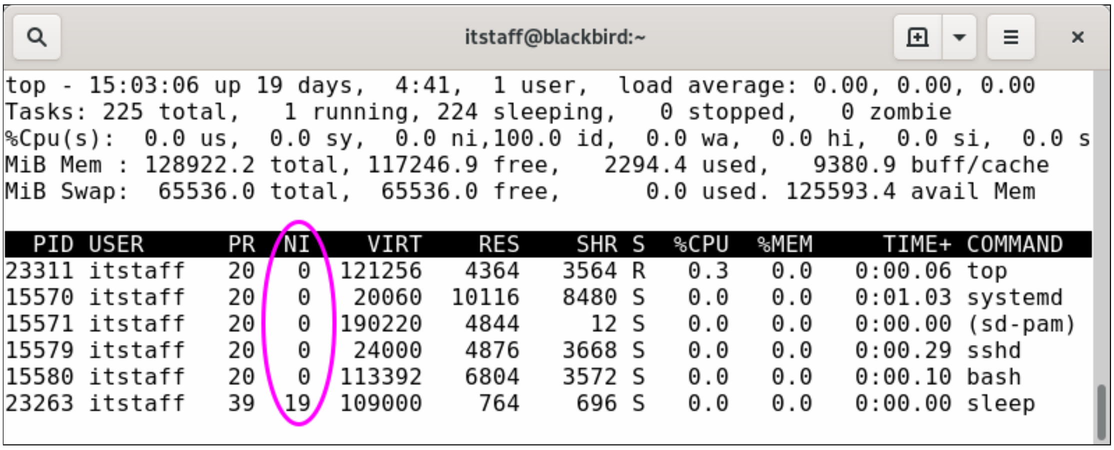
Command	Purpose
`nice`	Start a process with a low priority.
`renice`	Change a job to a different priority.

Table 13: Run jobs at a low priority

It is good practice to run jobs with low priority. This is particularly important on servers where jobs are competing for resources. Using a low priority means that cpu-intensive jobs will not interfere with processes which manage the system. The priority can either be set when the job is started, or reset once the job has started.

To start a sleep job at the lowest priority use

```
nice -19 sleep 300 &
```



```
top - 15:03:06 up 19 days,  4:41,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 225 total,   1 running, 224 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 s
MiB Mem : 128922.2 total, 117246.9 free,   2294.4 used,   9380.9 buff/cache
MiB Swap:  65536.0 total,  65536.0 free,      0.0 used. 125593.4 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
23311 itstaff   20   0  121256   4364   3564 R   0.3   0.0   0:00.06 top
15570 itstaff   20   0   20060  10116   8480 S   0.0   0.0   0:01.03 systemd
15571 itstaff   20   0  190220   4844     12 S   0.0   0.0   0:00.00 (sd-pam)
15579 itstaff   20   0   24000   4876   3668 S   0.0   0.0   0:00.29 sshd
15580 itstaff   20   0  113392   6804   3572 S   0.0   0.0   0:00.10 bash
23263 itstaff   39  19  109000    764    696 S   0.0   0.0   0:00.00 sleep
```

Figure 8: Using top to view a job's priority

It is also possible change the priority of a running process with

```
renice 19 PID
```

where *`PID`* is the Process ID of the running process.

3.7 Checkpointing

We recommend that checkpointing is used to protect running jobs from unexpected events such as system reboot, power failure, server crash, human error. Checkpointing allows jobs to restart from the last checkpoint rather than the beginning.

`dmtcp` will checkpoint most binary programs including

- R
- Python

- Matlab
- Perl
- ... and many more

There is more about `dmtcp`, including an FAQ on the DMTCP website here:`http://dmtcp.sourceforge.net/FAQ.html`.

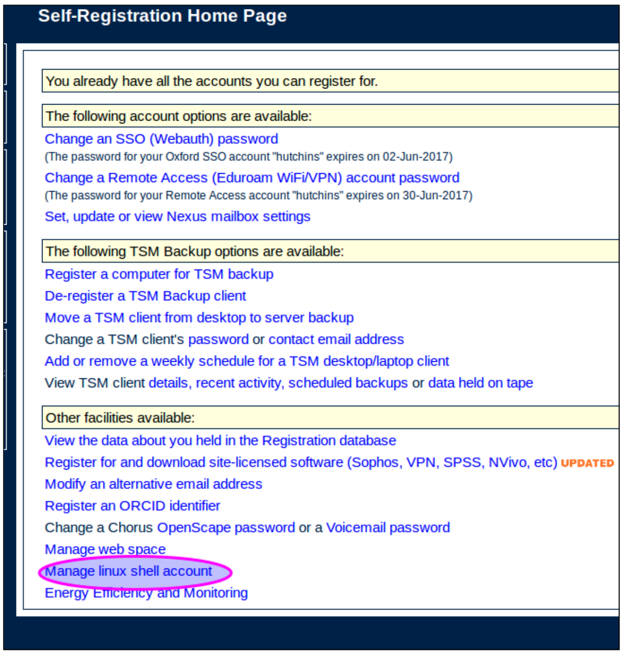
4 Linux remote working

Most of the work you do will likely be on remote systems as these will be more powerful and better suited to long-running, computationally intensive jobs.

For this exercise, we will be using the University's Linux service to explore ways of accessing remote systems. The service should not be used for your studies.

4.1 Setting up access to the University's Linux service

Go to `https://help.it.ox.ac.uk/` and click on **Manage passwords and accounts** on the left side of the bar at the top of the page. You will need to use your Single Sign-On (SSO) account to access this page. A page displaying accounts you can register for and services you can use will appear. In the **Other facilities available** section, you should see **Manage linux shell account**. Click on this link and you should now see the screen to activate the account. If you are asked to choose a shell, make sure you choose **/bin/bash**.



Self-Registration Home Page

You already have all the accounts you can register for.

The following account options are available:

Change an SSO (Webauth) password

(The password for your Oxford SSO account "hutchins" expires on 02-Jun-2017)

Change a Remote Access (Eduroam WiFi/VPN) account password

(The password for your Remote Access account "hutchins" expires on 30-Jun-2017)

Set, update or view Nexus mailbox settings

The following TSM Backup options are available:

Register a computer for TSM backup

De-register a TSM Backup client

Move a TSM client from desktop to server backup

Change a TSM client's password or contact email address

Add or remove a weekly schedule for a TSM desktop/laptop client

View TSM client details, recent activity, scheduled backups or data held on tape

Other facilities available:

View the data about you held in the Registration database

Register for and download site-licensed software (Sophos, VPN, SPSS, NVivo, etc) UPDATED

Modify an alternative email address

Register an ORCID identifier

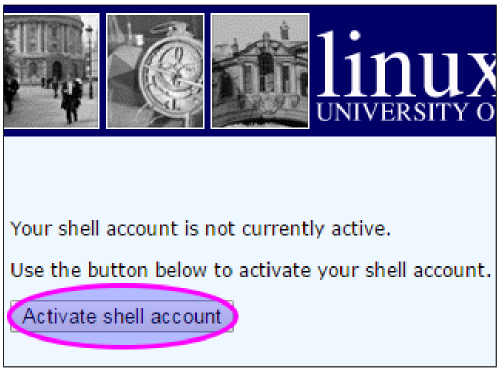
Change a Chorus OpenScape password or a Voicemail password

Manage web space

Manage linux shell account

Energy Efficiency and Monitoring

Figure 9: Self-Registration Home Page



Your shell account is not currently active.

Use the button below to activate your shell account.

Figure 10: Activate shell account

Now the account is enabled you can log on.

4.2 Accessing the service

Command	Purpose
`ssh user@linux.ox.ac.uk`	Log on to a remote system.

Table 14: Logging on to a remote system

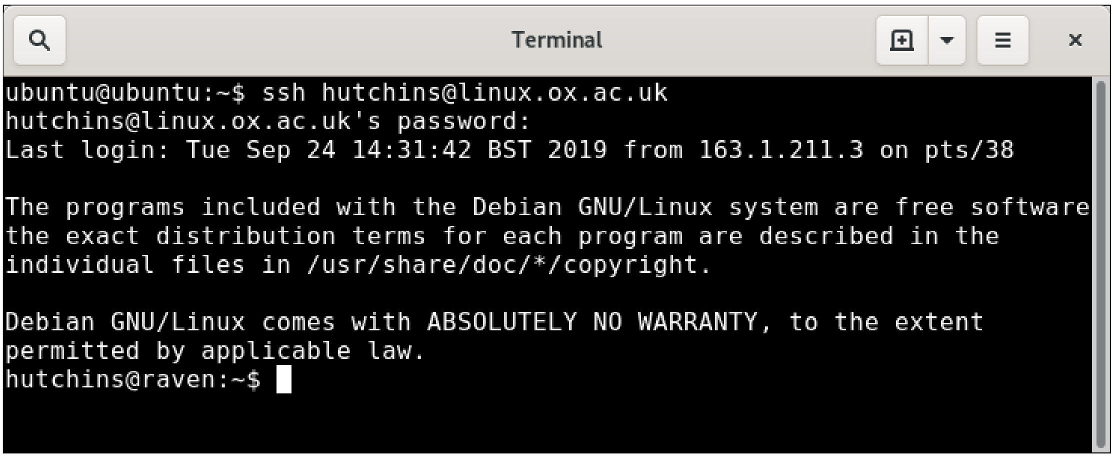
From a terminal window type in

```
ssh SSO@linux.ox.ac.uk
```

then at the login prompt enter your Single Sign-On username and password. For example, if your SSO is `coll1234` you would type in

```
ssh coll1234@linux.ox.ac.uk
```

A screen like Figure 11 should appear.



```
ubuntu@ubuntu:~$ ssh hutchins@linux.ox.ac.uk
hutchins@linux.ox.ac.uk's password:
Last login: Tue Sep 24 14:31:42 BST 2019 from 163.1.211.3 on pts/38

The programs included with the Debian GNU/Linux system are free software
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
hutchins@raven:~$ 
```

Figure 11: Log on to linux.ox.ac.uk

4.3 Using the Linux service

The following assumes that you have downloaded the files as described in Section 2 on page 2. All the commands that you have used in earlier exercises will work on this linux system too. For example, repeat the commands you used at the start.

```
cd
curl http://www.stats.ox.ac.uk/pub/susan/linux/StatML.tgz >StatML.tgz
```

to download the files and then

```
tar -xvzf StatML.tgz
```

to unpack them. Do try other commands if you have time from the exercises in Section 2 on page 2.

4.4 Using graphical applications remotely

It is possible — and sometimes necessary — to open a graphical application on the remote system and have the window displayed locally.

To do this logout of linux.ox.ac.uk

```
exit
```

and log back in again

```
ssh -X SSO@linux.ox.ac.uk
```

Note the use of the `-X` option which enables X11 forwarding, or the opening of remote windows. If you are using a Mac terminal you will need to install XQuartz. Type in the command

```
xclock &
```

and you should see a clock like Figure 12.

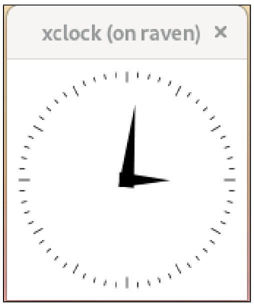


Figure 12: Remote clock

4.5 Downloading and copying files to remote systems

As well as logging on to remote systems, you may need to copy files and directories from one system to another. These commands will be used.

Command	**Purpose**
`scp` *`file1`* *`user`*`@linux.ox.ac.uk:`*`location/file2`*	Copy a single file *`file1`*, from the current directory on the local system to *`location`* on the remote system, with name *`file2`*.
`scp -r` *`directory`* *`user`*`@linux.stats.ox.ac.uk:`*`location`*	Copy a directory, *directory*, to location, *location* on the remote system using the same name.

Table 15: The `scp` command

These exercises are best done with two open terminal windows.

4.5.1 Copying files between systems

On `linux.ox.ac.uk`, create a new directory to store the files and directories you will be copying.

```
cd
mkdir Copies
```

Use `ls Copies` to make sure the new directory is empty.

Now in the local terminal (ie on the computer in front of you) type in

```
scp longfile.txt SSO@linux.ox.ac.uk:~/Copies/.
```

Make sure you type the command exactly as it appears.

The command is quite complex so each component is explained in detail here:

Command	**Purpose**
`scp`	Copy files and directories between systems.
`longfile.txt`	Name of file to be copied.
`SSO`	Username on remote system. Replace by your single sign-on.
`@`	Location indicator.
`linux.ox.ac.uk`	Name of remote system.
`:`	Separator between remote system name and location of file.
`~`	A useful shortcut which refers to your home directory. See ...
`/Copies/`	The directory where the file will be stored
`.`	"." indicates that the file should have the same name as the original.

Table 16: Copying files using the `scp` command

Now use `ls Copies` on `linux.ox.ac.uk` to make sure the files have copied successfully.

See if you can do the following:

- Copy longfile.txt from your local system to the remote system, giving a different name on the remote system.
- Use wildcards to copy all the files in the Wildcards directory beginning foo2 to the remote system, using the same name.

See Answers on page 5.

4.5.2 Copying files between systems

On the local system type in

```
cd
scp -r TestDir SSO@linux.ox.ac.uk:~/Copies/.
```

Then, on `linux.ox.ac.uk` check that the directory and contents have been copied.

4.6 Managing sessions on remote systems

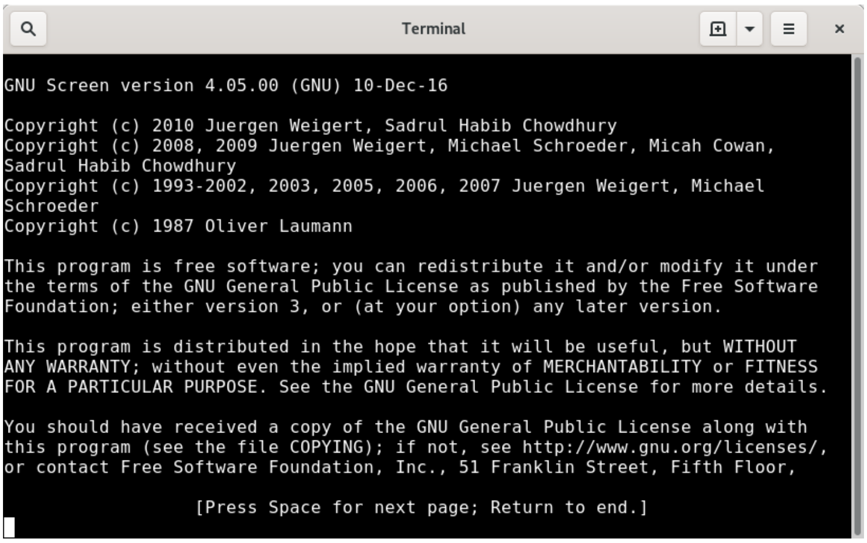
Command	**Purpose**
`screen`	Connect and disconnect from a session, possibly from multiple locations, and allow long-running processes to persist without an active shell session. Options used: `-r` reattach a screen session
`-list` display all screen sessions	

Table 17: Managing remote sessions

Again, it useful to have two terminal windows open. On `linux.ox.ac.uk` type in

```
screen
```

to start the screen session. You should see a window like this:



```
GNU Screen version 4.05.00 (GNU) 10-Dec-16

Copyright (c) 2010 Juergen Weigert, Sadrul Habib Chowdhury
Copyright (c) 2008, 2009 Juergen Weigert, Michael Schroeder, Micah Cowan,
Sadrul Habib Chowdhury
Copyright (c) 1993-2002, 2003, 2005, 2006, 2007 Juergen Weigert, Michael
Schroeder
Copyright (c) 1987 Oliver Laumann

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 3, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with
this program (see the file COPYING); if not, see http://www.gnu.org/licenses/,
or contact Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor,

                  [Press Space for next page; Return to end.]
```

Figure 13: The Linux screen command

Press return to end the message. You should now see the standard prompt. Use

```
cd
curl http://www.stats.ox.ac.uk/pub/susan/linux/my_date.sh >my_date.sh
```

Permanently change the permissions so that the script can be executed.

```
chmod +x my_date.sh
```

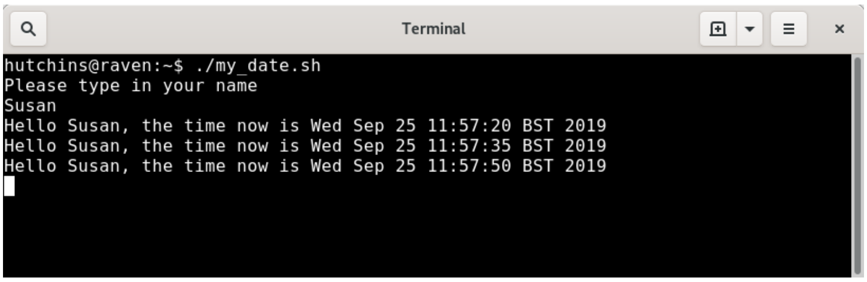
Have a look at the `my_date.sh` script to see what it does. Don't worry if you don't understand every line

```
#!/bin/bash
# Ask for name.
# Repeatedly output the name and date, waiting 15 seconds.
# Terminate after 15 minutes.
echo "Please type in your name"
read name
start=‘date +%s‘
now=$start
while (( (( $start + 900 )) > $now )) ;
do
        echo "Hello $name, the time now is ‘date‘"
        sleep 15
        (( now += 15 ))
done
```

The script displays the date and time every 15 seconds for 15 minutes. Run the script interactively to watch what happens.

```
./my_date.sh
```

Watch the output appearing for a minute or so. You should see something like this:



```
hutchins@raven:~$ ./my_date.sh
Please type in your name
Susan
Hello Susan, the time now is Wed Sep 25 11:57:20 BST 2019
Hello Susan, the time now is Wed Sep 25 11:57:35 BST 2019
Hello Susan, the time now is Wed Sep 25 11:57:50 BST 2019
```

Figure 14: Output from my_date.sh before detaching the session

Now detach from the screen session using

```
^CTRL-a D
```

That is hold down the `CTRL` key and `a` then press `D`. You should see a message

```
[detached from XXXX.pts-XX.raven]
```

where each X is a digit. You can check what screens you have with

```
screen -list
```

```
There is a screen on:
7712.pts-19.raven (09/25/19 11:42:15) (Detached)
1 Socket in /run/screen/S-hutchins.
```

Log out from linux.ox.ac.uk

```
exit
```

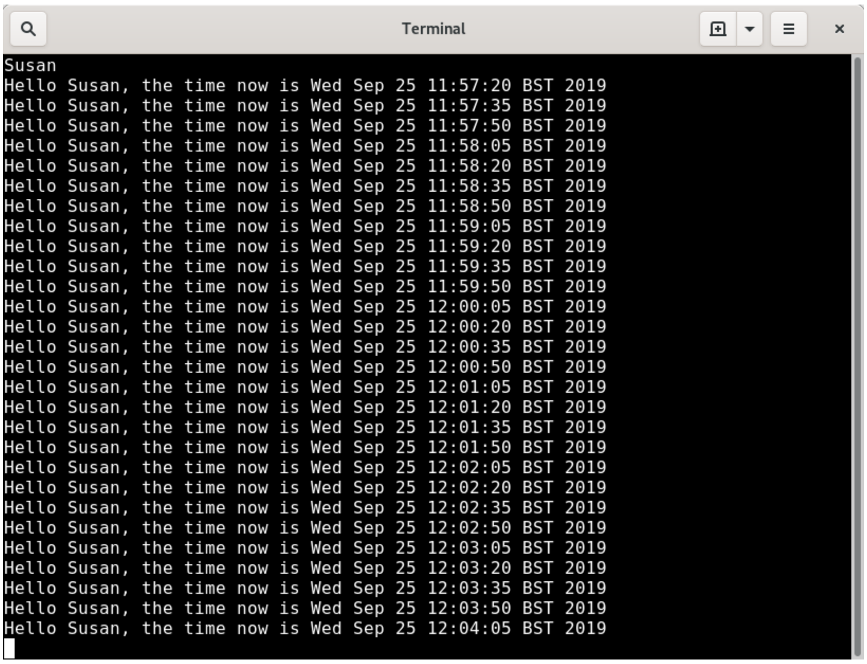
Log on again (and note that you can do this from any terminal, it does not have to be the same system)

```
ssh SSO@linux.ox.ac.uk
```

Reattach the screen

```
screen -r
```

and you should see the list of times with no interruptions.



```
Susan
Hello Susan, the time now is Wed Sep 25 11:57:20 BST 2019
Hello Susan, the time now is Wed Sep 25 11:57:35 BST 2019
Hello Susan, the time now is Wed Sep 25 11:57:50 BST 2019
Hello Susan, the time now is Wed Sep 25 11:58:05 BST 2019
Hello Susan, the time now is Wed Sep 25 11:58:20 BST 2019
Hello Susan, the time now is Wed Sep 25 11:58:35 BST 2019
Hello Susan, the time now is Wed Sep 25 11:58:50 BST 2019
Hello Susan, the time now is Wed Sep 25 11:59:05 BST 2019
Hello Susan, the time now is Wed Sep 25 11:59:20 BST 2019
Hello Susan, the time now is Wed Sep 25 11:59:35 BST 2019
Hello Susan, the time now is Wed Sep 25 11:59:50 BST 2019
Hello Susan, the time now is Wed Sep 25 12:00:05 BST 2019
Hello Susan, the time now is Wed Sep 25 12:00:20 BST 2019
Hello Susan, the time now is Wed Sep 25 12:00:35 BST 2019
Hello Susan, the time now is Wed Sep 25 12:00:50 BST 2019
Hello Susan, the time now is Wed Sep 25 12:01:05 BST 2019
Hello Susan, the time now is Wed Sep 25 12:01:20 BST 2019
Hello Susan, the time now is Wed Sep 25 12:01:35 BST 2019
Hello Susan, the time now is Wed Sep 25 12:01:50 BST 2019
Hello Susan, the time now is Wed Sep 25 12:02:05 BST 2019
Hello Susan, the time now is Wed Sep 25 12:02:20 BST 2019
Hello Susan, the time now is Wed Sep 25 12:02:35 BST 2019
Hello Susan, the time now is Wed Sep 25 12:02:50 BST 2019
Hello Susan, the time now is Wed Sep 25 12:03:05 BST 2019
Hello Susan, the time now is Wed Sep 25 12:03:20 BST 2019
Hello Susan, the time now is Wed Sep 25 12:03:35 BST 2019
Hello Susan, the time now is Wed Sep 25 12:03:50 BST 2019
Hello Susan, the time now is Wed Sep 25 12:04:05 BST 2019
```

Figure 15: Output from my_date.sh after reattaching the session

This is a slightly contrived example. In general output from long-running jobs should be saved in a file rather than displayed on a screen, as there are limits to the amount of data a terminal displays.

5 Linux answers

Absolute and relative paths

1. A relative path – which is why it didn't work.
2. `cd /`
3. `cd /homes/`*`user`*`/Testdir` or cd ~/Testdir.
4. `ls /` will display the contents of the root directory.
5. Use `ls`.

Using wildcards to match file names

`ls foo*` matches

```
foo1 foo2
```

`ls foo2*` matches

```
foo2 foo20 foo2bar
```

`ls foo[1-2]` matches

```
foo1 foo2
```

To match just the files `foo20` and `foo2bar` use

```
ls foo2?*
```

To use `wc -l` just to display the files with a 1 in their name

```
wc -l *1*
```

Sorting and searching – some examples

To find the 5 largest files beginning with 's' in the `/usr/bin` directory use

```
du -sk /usr/bin/s* | sort -n | tail -5
```

Copying files between systems

```
scp longfile.txt SSO@linux.ox.ac.uk:~/Copies/longfile1.txt
```

```
scp foo2 SSO@linux.ox.ac.uk:~/Copies/.
```

6 LaTeX

If you are new to LaTeX then have a look at section 6.3 on page 39 which contains further exercises and their answers. These can be done independently and should provide a useful resource when you come to write more complex documents.

Finally, Section 6.5, 'Helpful Hints', contains several pages of characters and their markup.

There is also have a copy of 'The Not So Short Introduction to LaTeX2e' in each office which will be a useful reference for the independent exercises.

6.1 Texmaker

We will be using Texmaker to create, edit, typeset and view LaTeX documents. To start Texmaker enter `texmaker` in the `Type to search` box as described in Section ?? on page ??. A window like Figure 16 should appear:

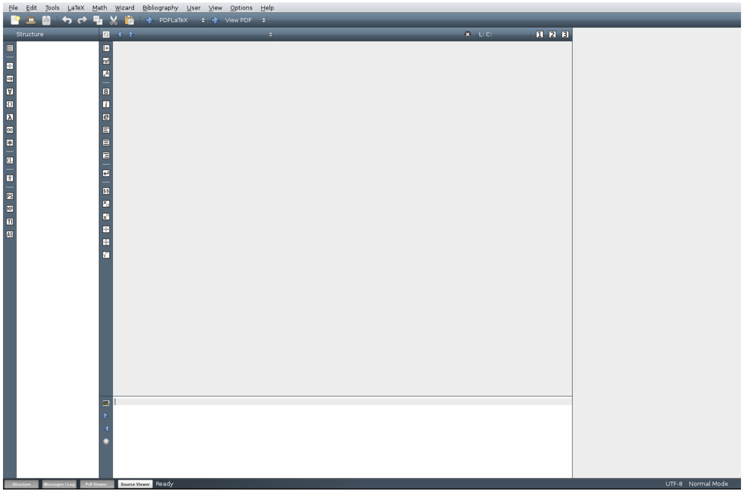


Figure 16: Texmaker: Start–up screen

Find the Run drop-down menu box to the right of PDFLaTeX. See what other options there are.

Find the following buttons:

- View (F7)
- Find (Ctrl+F)
- Replace (Ctrl+R)
- Save (Ctrl+S)

The keyboard shortcuts are included after each command.

These buttons are used for compiling and viewing LaTeX documents. Most of the other buttons are used for typesetting. See if you can find some of these:

- **Bold**
- ***Italic***
- **Left**

- **Part/Chapter/Section**
- **Relation symbols**
- **Greek Letters**

6.2 Building a LaTeX document

6.2.1 Before you start

You will need to download some files before starting.

1. Create a new directory called LaTeX for this work.
2. Download the files `template.tex` and `test.bib` from **http://www.stats.ox.ac.uk/pub/susan/statml** for use during this practical.
3. Find an R plot you have created or download `Hills.pdf` from the above location.

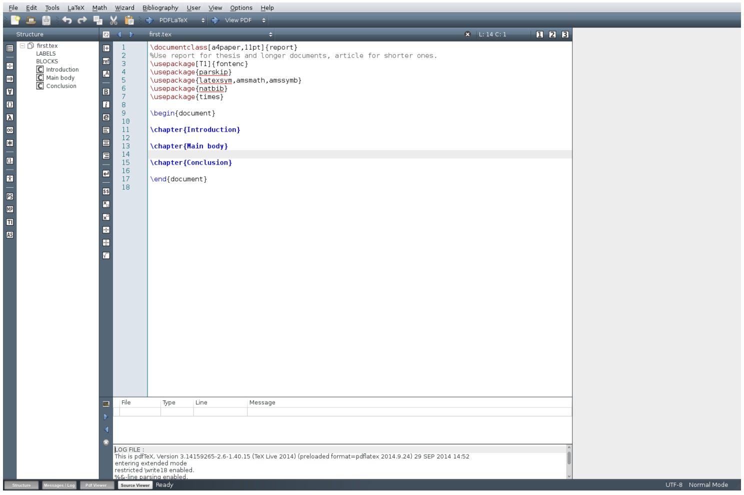
You can download fresh copies of the files whenever you want. Other files here include

`Exercises.pdf` This file.

`final_template.tex` For downloading at the end of the practical session. This file can be used as a template for your practicals.

`small.tex` For the independent exercises.

Open the file `template.tex` from Texmaker and save it as `first.tex`. LaTeX documents must *always* end with `.tex`. Click on the arrow to the left of **PDFLaTeX** to build the document.



```
\documentclass[a4paper,11pt]{report}
%Use report for thesis and longer documents, article for shorter ones.
\usepackage[T1]{fontenc}
\usepackage{parskip}
\usepackage{latexsym,amsmath,amssymb}
\usepackage{natbib}
\usepackage{times}

\begin{document}

\chapter{Introduction}

\chapter{Main body}

\chapter{Conclusion}

\end{document}
```

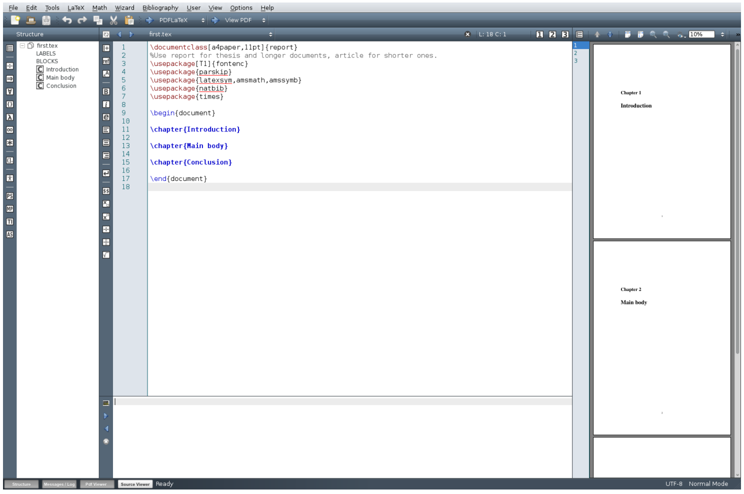
```
LOG FILE :
This is pdfTeX, Version 3.14159265-2.6-1.40.15 (TeX Live 2014) (preloaded format=pdflatex 2014.9.24) 29 SEP 2014 14:52
entering extended mode
restricted \write18 enabled.
%&-line parsing enabled.
```

Figure 17: Building `first.tex`

Each time you finish making changes to `first.tex`, it must be built as in Figure 17.

The large centre screen is the editor window where you add to and change your document. Notice the small window below this. Information about the build will appear here. If there are errors then they will be shown here.

If the build has completed without errors, then click on the arrow to the left of the **View PDF** to preview your document. The preview window on the right shows you how the document will look when it's printed as in Figure 18.



```
\documentclass[a4paper,11pt]{report}
%Use report for thesis and longer documents, article for shorter ones.
\usepackage[T1]{fontenc}
\usepackage{parskip}
\usepackage{latexsym,amsmath,amssymb}
\usepackage{natbib}
\usepackage{times}

\begin{document}

\chapter{Introduction}

\chapter{Main body}

\chapter{Conclusion}

\end{document}
```

Figure 18: Previewing `first.tex`

Note that some of the text in the editing screen (not on this printed page!) is highlighted in different colours. This shows us which are typesetting commands and which is text; it also makes finding errors much easier.

Now change the text `Main body` to `Research`, build the document again and click on the **View PDF** arrow. Do you notice how Chapter 2 is now ‘Research’? as in Figure 19.

6.2.2 How does it work?

Let’s look at the significant lines in `first.tex`.

```
\documentclass[a4paper,11pt]{article}
```

All LATEX documents begin with a `\documentclass` declaration. The two options between [. . .] specify the default font size (`11pt`) and paper size (`a4paper`). The final statement `{article}` describes the document’s class. Conventionally the `article` class is used for shorter documents and the `report` class for longer dissertations. Other options include `book` and `letter`. Note that everything that appears on a line after a % is a comment and is ignored.

The syntax of LATEX commands is consistent — a command begins with a \ (a backslash), is altered by options contained in [. . .] (square brackets) and is defined by what is between {. . . } (curly brackets). See for example the first line of the `first.tex` document:

```
\documentclass[a4paper,11pt]{article}
```

The next line

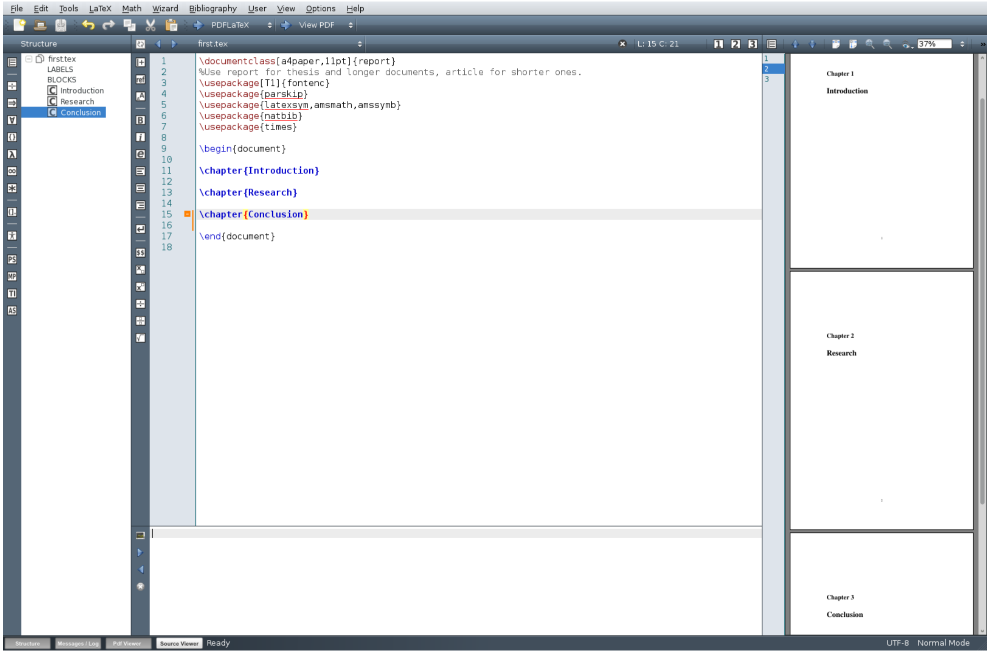
```
%Use article for short documents
```

is a comment. It is ignored when the document is typeset. The following lines beginning

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

add extra features such as additional symbols for maths — `amsmath, amssymb` — or change the default behaviour — `times` which changes the font to Times. You will be adding more packages in later exercises.

This part of the document is known as the preamble.



```
\documentclass[a4paper,11pt]{report}
%Use report for thesis and longer documents, article for shorter ones.
\usepackage[T1]{fontenc}
\usepackage{parskip}
\usepackage{latexsym,amsmath,amssymb}
\usepackage{natbib}
\usepackage{times}

\begin{document}

\chapter{Introduction}

\chapter{Research}

\chapter{Conclusion}

\end{document}
```

Figure 19: Typesetting with a new chapter heading

The main body of a document – that is the information that you want people to read – begins with

```
\begin{document}
```

and is always ended by

```
\end{document}
```

The information within the main document has been divided into sections with the commands

```
\chapter{Introduction}
```

and

```
\chapter{Research}
```

and so on. As this is part of the main document — the part people will read — the text appears in the preview window, after the document has been built. The chapters, sections and subsections are numbered automatically.

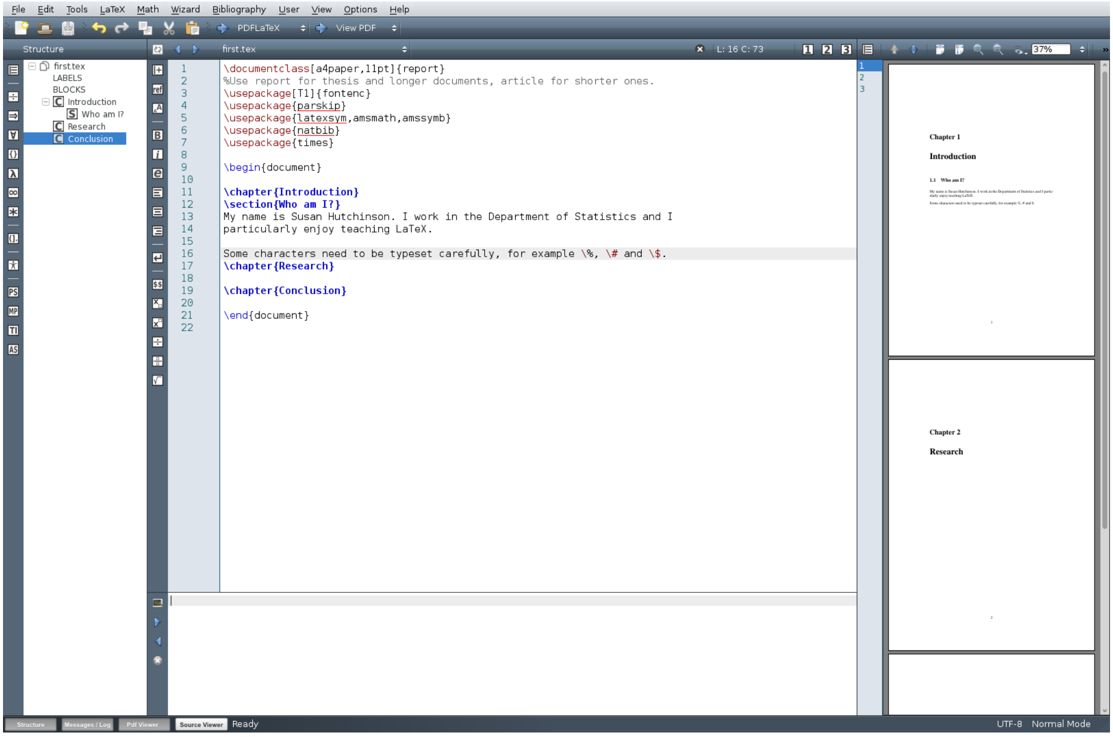
6.2.3 Adding some text

Now add some text after `\Chapter{Introduction}` and build the document again. Perhaps you could write a little about yourself.

In these exercises, all text to be typed in will appear in a font `like this` with a line above and below. Here's the text I typed into `first.tex`.

```
\section{Who am I?}
My name is Susan Hutchinson. I work in the Department of Statistics and I
particularly enjoy teaching LaTeX.

Some characters need to be typeset carefully, for example \%, \# and \$.
```

When you've built your document again and clicked on the **View PDF**, the text you entered should appear in the right hand preview window as in Figure 20.



```
\documentclass[a4paper,11pt]{report}
%Use report for thesis and longer documents, article for shorter ones.
\usepackage[T1]{fontenc}
\usepackage{parskip}
\usepackage{latexsym,amsmath,amssymb}
\usepackage{natbib}
\usepackage{times}

\begin{document}

\chapter{Introduction}
\section{Who am I?}
My name is Susan Hutchinson. I work in the Department of Statistics and I
particularly enjoy teaching LaTeX.

Some characters need to be typeset carefully, for example \%, \# and \$.
\chapter{Research}

\chapter{Conclusion}

\end{document}
```

Figure 20: Adding some text

When writing text in a LATEX document some characters have a special meaning. The following seven characters are printed by typing a backslash in front of them: $ & # % _ { and } so you need to type \$, \& and so on.

Most of the time you will be writing text for reports or dissertations as you have done here. For more about entering text see the first exercise in Section 6.3.1. The exercises in that section can be done independently later. Here you will find out to change the format of text using commands that are embedded within the text, for example `\emph` is used to *italicise* text and `\textbf` for **bold** text.

The exercises in this section demonstrate other skills that you will need when writing LATEX documents. I have also suggested some further exercises for you to try on your own.

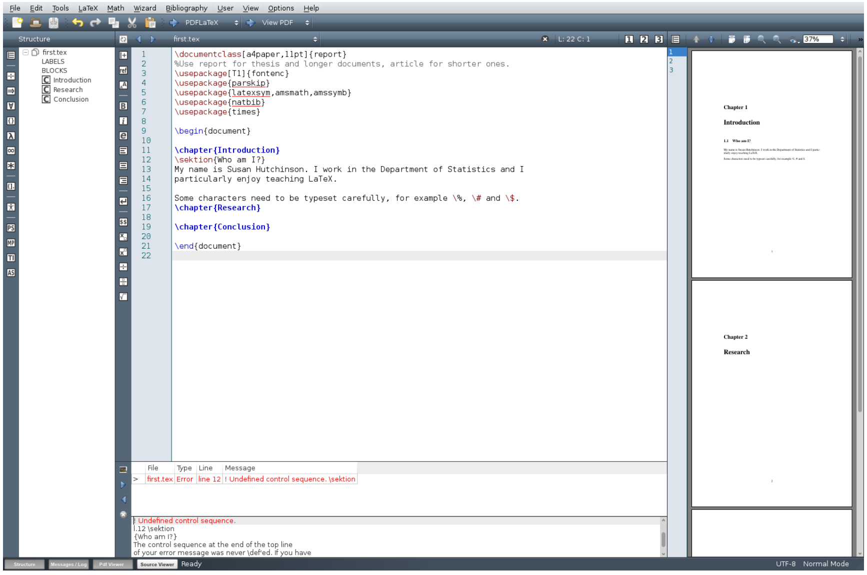
6.2.4 Fixing errors

It often happens that you make a mistake in your document. One important skill you will need is finding the error and fixing it.

Change the document so that it contains a mistake – perhaps by replacing 'section' by 'sektion' or leaving out the closing } after `\begin{document`.

Now typeset the document again and see what happens. Depending on the error you have introduced you may see something like Figure 21 appear in a new screen at the bottom of your Texmaker editing window. The line number where the error appears is displayed here.

Now fix the error and typeset the document again.



```
\documentclass[a4paper,11pt]{report}
%Use report for thesis and longer documents, article for shorter ones.
\usepackage[T1]{fontenc}
\usepackage{parskip}
\usepackage{latexsym,amsmath,amssymb}
\usepackage{natbib}
\usepackage{times}

\begin{document}

\chapter{Introduction}
\sektion{Who am I?}
My name is Susan Hutchinson. I work in the Department of Statistics and I
particularly enjoy teaching LaTeX.

Some characters need to be typeset carefully, for example \%, \# and \$.
\chapter{Research}

\chapter{Conclusion}

\end{document}
```

	File	Type	Line	Message
>	first.tex	Error	line 12	! Undefined control sequence. \sektion

```
! Undefined control sequence.
l.12 \sektion
{Who am I?}
The control sequence at the end of the top line
of your error message was never \def'ed. If you have
```

Figure 21: Texmaker: Locating errors

6.2.5 Typesetting mathematics

There are several different ways of typesetting formulae and equations. They can appear ‘inline’ – within a paragraph – like this: $C(n,r) = n!/(r!\,(n-r)$ or separately as follows so the paragraph is broken up. This is known as ‘display math’.

$$C(n,r) = n!/(r!\,(n-r)!)$$

Now add a subsection`\section{Maths}` to the Research chapter of document and then enter the following to typeset two equations.

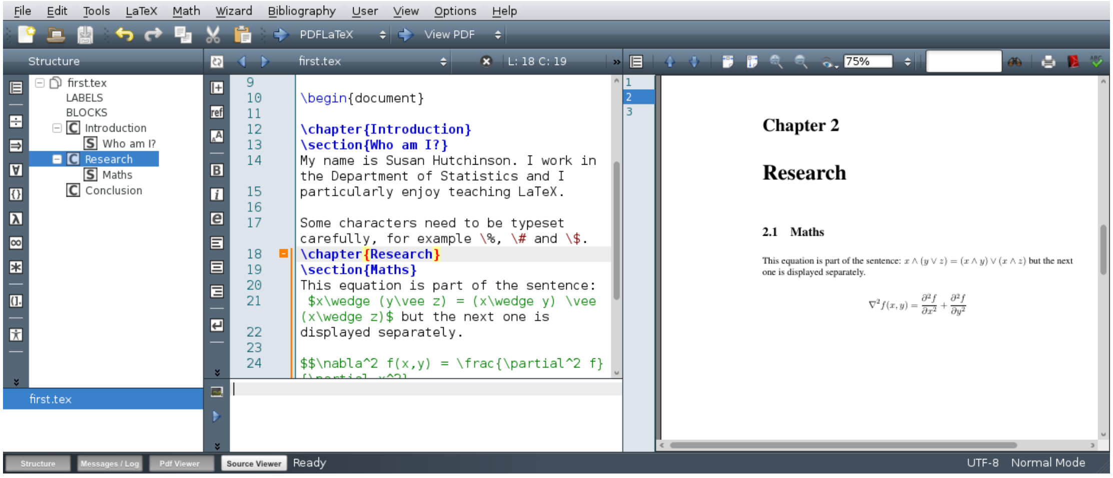
```
\section{Maths}
This equation is part of the sentence:
 $x\wedge (y\vee z) = (x\wedge y) \vee (x\wedge z)$ but the next one is
displayed separately.

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2}
+ \frac{\partial^2 f}{\partial y^2}$$
```

Your editor and preview windows should look like Figure 22. Note how the \$ symbol is used at the start and end of each equation. To typeset the equation centred on a line on its own in display math mode use \$\$ at the beginning and end of the equation.

Many more maths exercises can be found in Section 6.3.2. Again, these can be done independently. Finally, Chapter 3 of *The Not So Short Introduction to LaTeX2e* describes more ways displaying mathematics.

6.2.6 Plots

You will need to include plots from R in your document. Either create a plot from R in PDF format and save it in the same directory as your `.tex` document. Alternatively use the `Hills.pdf` file downloaded at the start of these exercises.



```

\begin{document}

\chapter{Introduction}
\section{Who am I?}
My name is Susan Hutchinson. I work in
the Department of Statistics and I
particularly enjoy teaching LaTeX.

Some characters need to be typeset
carefully, for example \%, \# and \$.
\chapter{Research}
\section{Maths}
This equation is part of the sentence:
 $x\wedge (y\vee z) = (x\wedge y) \vee
(x\wedge z)$ but the next one is
displayed separately.

$$\nabla^2 f(x,y) = \frac{\partial^2 f}
```

Chapter 2

Research

2.1 Maths

This equation is part of the sentence: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ but the next one is displayed separately.

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Figure 22: Some mathematics

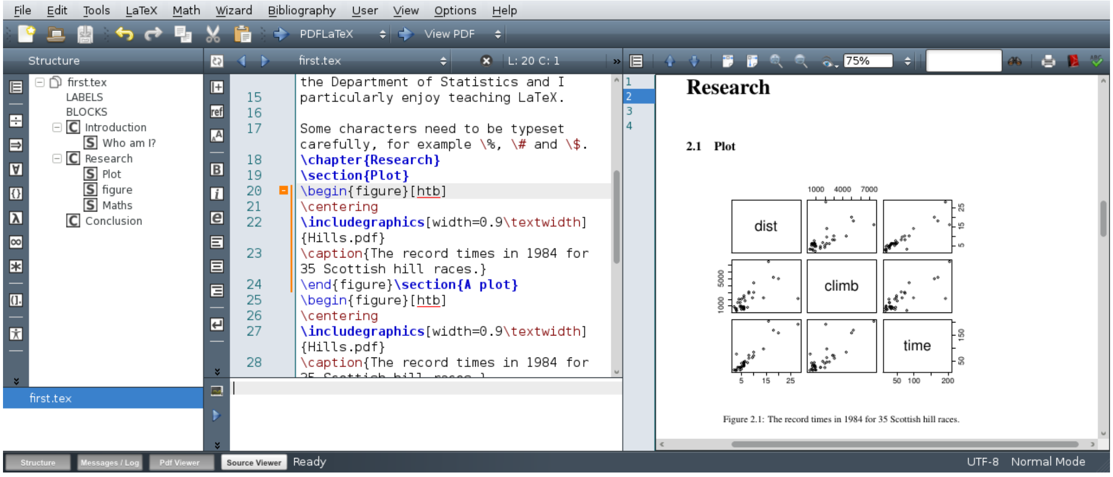
In the preamble (between `\documentclass...` and `\begin{document}`) add this before `\usepackage{times}`.

```
\usepackage{graphicx}
```

Add these lines immediately before `\section{Maths}`

```
\section{A plot}
\begin{figure}[htb]
\centering
\includegraphics[width=0.9\textwidth]{IMAGES/Hills.pdf}
\caption{The record times in 1984 for 35 Scottish hill races.}
\end{figure}
```

After typesetting the output should look like Figure 23.



```
the Department of Statistics and I
particularly enjoy teaching LaTeX.

Some characters need to be typeset
carefully, for example \%, \# and \$.
\chapter{Research}
\section{Plot}
\begin{figure}[htb]
\centering
\includegraphics[width=0.9\textwidth]
{Hills.pdf}
\caption{The record times in 1984 for
35 Scottish hill races.}
\end{figure}\section{A plot}
\begin{figure}[htb]
\centering
\includegraphics[width=0.9\textwidth]
{Hills.pdf}
\caption{The record times in 1984 for
```

Research

2.1 Plot

Figure 2.1: The record times in 1984 for 35 Scottish hill races.

Figure 23: Adding a plot

Let's look more closely at the lines we added.

`\begin{figure} ... \end{figure}`	Starts and ends the figure environment.
`[htb]`	Where to place the plot (here, top, bottom) on the page.
`\centering`	Centre the plot on the page.
`\includegraphics{Hills.pdf}`	Include the plot file.
`[width=0.9\textwidth]`	Make the plot scale to .9 of the width of the page.

In some cases you may want to give a plot an exact height and width in which case you would use

`[width=4.8in, height=3.2in]`	These options give the plot an exact height and width.
`\caption{The record ... races.}`	A caption for the plot.

See the exercises in Section 6.3.4 for ways of typesetting graphs.

6.2.7 Tables from R

Here is a simple table generated in R from the first 10 values in the `hills` data set. Start R, and then execute the following commands which are marked by a >:

```
> library(MASS)
> library(xtable)
> data(hills)
> hills.table <- xtable(hills[1:10,])
> x.ltx <- toLatex(hills.table)
> x.ltx
% latex table generated in R 3.1.1 by xtable 1.7-4 package
% Wed Oct  8 14:00:57 2014
\begin{table}[ht]
\centering
\begin{tabular}{rrrr}
  \hline
 & dist & climb & time \\
  \hline
Greenmantle & 2.50 & 650 & 16.08 \\
  Carnethy & 6.00 & 2500 & 48.35 \\
  Craig Dunain & 6.00 & 900 & 33.65 \\
  Ben Rha & 7.50 & 800 & 45.60 \\
  Ben Lomond & 8.00 & 3070 & 62.27 \\
  Goatfell & 8.00 & 2866 & 73.22 \\
  Bens of Jura & 16.00 & 7500 & 204.62 \\
  Cairnpapple & 6.00 & 800 & 36.37 \\
  Scolty & 5.00 & 800 & 29.75 \\
  Traprain & 6.00 & 650 & 39.75 \\
   \hline
\end{tabular}
\end{table}
```

In a new section, copy the LaTeX table into `first.tex`. Once built, the table should look like Table 6.2.7.

	dist	climb	time
Greenmantle	2.50	650	16.08
Carnethy	6.00	2500	48.35
Craig Dunain	6.00	900	33.65
Ben Rha	7.50	800	45.60
Ben Lomond	8.00	3070	62.27
Goatfell	8.00	2866	73.22
Bens of Jura	16.00	7500	204.62
Cairnpapple	6.00	800	36.37
Scolty	5.00	800	29.75
Traprain	6.00	650	39.75

Let's look at the lines we've entered. There are some similarities with the mark-up for the plot we added earlier. This time we have used the `table` environment not the `figure` environment.

```
\begin{table}[ht]                              Start the table environment.
\centering                                     Centre the table
\begin{tabular}{rrrr}                          Begin the tabular environment
{rrrr}                                         Four right-aligned columns.
  \hline                                       Draw a horizontal line.
 & dist & climb & time \\                      Headers for columns 3, 4 and 5.
  \hline                                       Draw a horizontal line.
Greenmantle & 2.50 & 650 & 16.08 \\           The next 10 lines are the data.
 Carnethy & 6.00 & 2500 & 48.35 \\
  Craig Dunain & 6.00 & 900 & 33.65 \\
  Ben Rha & 7.50 & 800 & 45.60 \\
  Ben Lomond & 8.00 & 3070 & 62.27 \\
  Goatfell & 8.00 & 2866 & 73.22 \\
  Bens of Jura & 16.00 & 7500 & 204.62 \\
  Cairnpapple & 6.00 & 800 & 36.37 \\
  Scolty & 5.00 & 800 & 29.75 \\
  Traprain & 6.00 & 650 & 39.75 \\
   \hline                                      Draw a horizontal line
 \end{tabular}                                 End the tabular environment
 \end{table}                                   End the table environment.
```

Things to note:

`{rrrr}` defines the number and alignment of the columns.

`&` separates columns on each row.

\\ Ends each row.

For further exercises on creating tables see Section 6.3.1.

6.2.8 Cross references

Cross references allow you to refer to other parts of your document. For example, you may want to refer to a plot you have included in another chapter.

We are going to set up a reference to the Hills plot which you added above.

Add these two lines to the plot immediately after the line

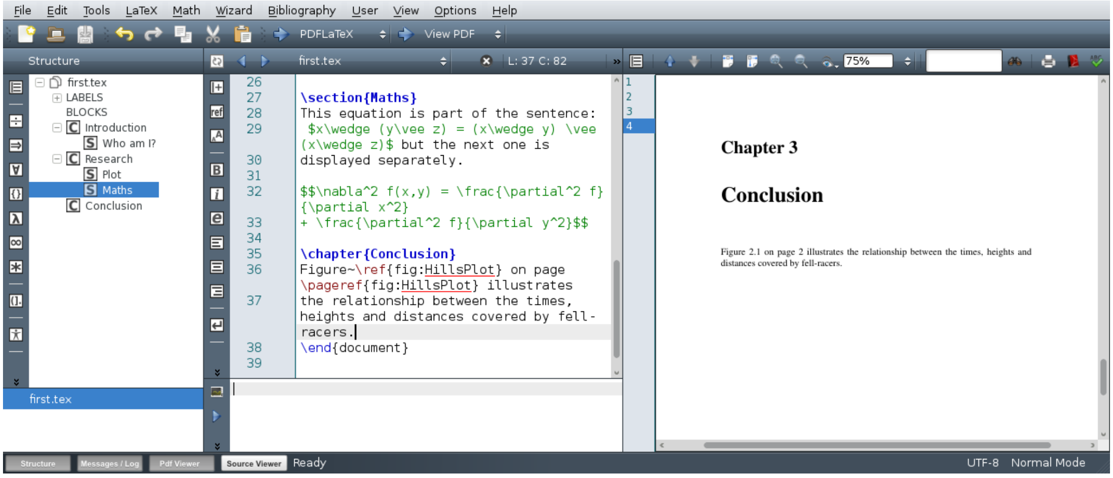
`\includegraphics[width=.0\textwidth]{Hills.pdf}`.

```
\caption{The record times in 1984 for 35 Scottish hill races.}
\label{fig:HillsPlot}
```

Now add the following lines after `\chapter{Conclusion}`.

```
Figure~\ref{fig:HillsPlot} on page~\pageref{fig:HillsPlot} illustrates
the relationship between the times, heights and distances covered by
fell-racers.
```

You will need to run **pdfLaTeX** twice for the references to be resolved. The first time you will see **??** in place of the numbers. See Figure 24 for a final version.



```
\section{Maths}
This equation is part of the sentence:
 $x\wedge (y\vee z) = (x\wedge y) \vee
(x\wedge z)$ but the next one is
displayed separately.

$$\nabla^2 f(x,y) = \frac{\partial^2 f}
{\partial x^2}
+ \frac{\partial^2 f}{\partial y^2}$$

\chapter{Conclusion}
Figure~\ref{fig:HillsPlot} on page
\pageref{fig:HillsPlot} illustrates
the relationship between the times,
heights and distances covered by fell-
racers.
\end{document}
```

Chapter 3

Conclusion

Figure 2.1 on page 2 illustrates the relationship between the times, heights and distances covered by fell-racers.

Figure 24: A complete cross-reference

You can use any text you like for your labels but it is recommended that you choose labels that are easy to remember. I use labels beginning `fig:` for figure references, `sec:` for section references and so on. This is not necessary but makes it easier to remember where a reference points. It does not affect the typesetting.

Cross references can be used to reference sections, tables, equations and much else too.

`\label{fig:HillsPlot}`	Defines the label.
`Figure~\ref{fig:HillsPlot} on`	Refers to the figure number of the plot
`page \pageref{fig:HillsPlot} illustrates`	Refers to the page number of the plot

6.2.9 A simple bibliography

In the preamble include `\usepackage{natbib}` and `\usepackage{url}`.This command determines the style of the citations that will appear in your document. Both author–year and numerical citations can be used. At the end of the `.tex` document, just before `\end{document}` add

```
\bibliographystyle{apalike2}
\bibliography{test}
```

The command `\bibliographystyle{plainnat}` tells LaTeX how to format the bibliographical information stored in the file `test.bib`. Use `\bibliography{test}` to include the bibliography file in your document. The sample `test.bib` file you have looks like this:

```
@Book{Lamport,
  author =       "Leslie Lamport",
  title =        "\LaTeX{}: A Document Preparation System",
  publisher =    "Addison Wesley",
  address =      "Reading, Massachusetts",
  edition =      "2nd",
  year =         "1994",
}

@Book{Goossens,
  author =       "Michel Goossens and Frank Mittelbach and Alexander Samarin",
  title =        "{The \LaTeX{} Companions}",
  publisher =    "Addison Wesley",
  address =      "Reading, Massachusetts",
  edition =      "1st",
  year =         "1994",
}

@Article{Fenn,
  author =   "J{\"u}rgen Fenn",
  title =   "{Managing Citations and Your Bibliography with BiB\TeX}",
  journal =   "{The Prac\TeX\ Journal}",
  year =   "2006",
  volume =   "4",
  note =   "{\url{http://www.tug.org/pracjourn/2006-4/fenn/fenn.pdf}}",
}

@Article{Mertz,
  author =  "Andrew Mertz and William Slough",
  title =  "Beamer by Example",
  journal =  "{The Prac\TeX\ Journal}",
  year =  "2005",
  volume =  "4",
  note =  "{\url{http://tug.org/pracjourn/2005-4/mertz/mertz.pdf}}",
}
```

You will need to add your own references to this file.

Here are a few example citations.

Using `\cite{Lamport}` produces Lamport (1994).

Using `\cite{Goossens}` produces Goossens et al. (1994).

To build a document containing citations you should use

- **PDFLaTeX**
- **BibTeX**
- **PDFLaTeX**

at least! Finally adding `\addcontentsline{toc}{section}{\numberline{}\refname}` to the end of the file will ensure that "References" or "Bibliography" appears in the table of contents as it does at the end of this document.

6.2.10 Printing your document

You cannot print from the preview window. Use the file browser to find the PDF file. Double click on it and it will open in **Adobe Reader**. You can then print.

6.2.11 Downloading the final version of the template

Finally download the final version of `final_template.tex` from the Skills section of Weblearn. This file can be used as a template for later practical sessions.

This completes the practical session exercises.

6.3 LaTeX exercises to try on your own

The answers to these exercises are in the next section.

Please download these two files for use in these exercises. Browse to
http://www.stats.ox.ac.uk/pub/susan/statml and save the two files

- `small.tex`
- `test.bib`

in your home directory. To be really organised you could create a new directory to keep them in.

The following exercises are designed to show many features of LaTeX. If possible try to do at least the following

- Exercises 1–6 on simple text typesetting.
- Some of the maths exercises.
- Exercise 41 on cross references
- Finally have a look at Section 6.3.4 on page 43 for how to includes graphs and images in documents.

Load `small.tex` into Texmaker. Remember that you should save the document and click on the green typeset button after each change.

1. Change `documentclass` from `article` to `report` and then to `book`. You might want to include
   `\chapter{Introduction}` immediately after `\begin{document}`
2. Change the `documentclass` option `12pt` to `11pt`
3. Add `\usepackage{parskip}` to the preamble. How does the paragraph formatting change?
4. Add `\usepackage{mathptmx}` to the preamble. The font should now look different as you are now using Times Roman.
5. Swap the emphasised and bold text.

6.3.1 Typesetting Text

▷ Exercise 1 We will now typeset some simple sentences. You should have a copy of *The Not So Short Introduction to LaTeX2e* which we will be using for reference for the rest of these exercises. Chapter 2 *Typesetting Text* is particularly helpful and Section 6.5, 'Helpful Hints' of this document has instructions on how to typeset many characters.

Start a new section in the document. Typeset the following sentences – there are some helpful suggestions just below if you get stuck.

> I entered the room and—horrors—I saw both my father-in-law and my mother-in-law.
>
> The winter of 1484–1485 was one of discontent.
>
> Frank wondered, "Is this a girl that can't say 'No!'?"

> Does Æschylus understand Œdipus?
>
> They took some honey and plenty of money wrapped up in a £5 note.
>
> Élèves, refusez vos leçons! Jetez vos chaînes!
>
> Can you take a ferry from Öland to Åland?

There are several features of LaTeX that are evident here.

hyphens Have you noticed that there are different lengths of hyphen? For example in the first sentence both — and - are used. These are typeset using `---` and `-` respectively.

quotes When typesetting speech make sure that you use `‘‘` and `’’` to produce open and close quotation marks. The `‘` character is usually found at the top lefthand side of the keyboard.

accents The ‘Helpful Hints’ section should help with these.

pounds To typeset the £ symbol use `\pounds`.

Finally, it can seem difficult to decide whether to put a space after a LaTeX command or not. A general rule is that if the command is a single non-alphanumeric then a space is not needed otherwise it is. For example to typeset naïve you need `na\"\i ve`. In this case `\"` puts a double dot (or diaeresis) over the letter “i” and `\i` prints an “i” without a dot like this: ı.

▷ Exercise 2 Give your document a title. You will need to include

```
\title{Your title}
\author{Your name}
\date{A date}
\maketitle
```

immediately after `\begin{document}`. To include a table of contents add `\tableofcontents` after the title. You must have at least one section, subsection or chapter for the table of contents to appear. Note that you must run `pdfLaTeX` twice in order for entries in the table of contents to be displayed.

▷ Exercise 3 Lists. Using the `itemize` , `enumerate` and `description` environments typeset the following

1. You can mix list environments as much as you like
    - But it might start to look silly
    - With different symbols
2. So do remember

    Stupid things will not become smart because they are in a list.

    Smart things, though, can be presented beautifully in a list.

[See section 2.11.1 of *The Not So Short Introduction to LaTeX2e* for more information about lists.]

▷ Exercise 4 Typeset the following table

Vegetable Production

Vegetable	Comments	Weight
Carrots	Good early crop, then carrot fly.	7kg
Lettuce	Slow to start, then bolted.	1kg
French beans	Excellent.	12kg

▷ Exercise 5 If you have time, try this more complicated table.

Currencies 1 Jan 2001

London:	New York:
£: $ 1.8672	£: $ 1.8655
£: DM 2.8369	$: DM 1.5175
£: FFr 9.69080	$: FFr 5.1845

[See section 2.11.6 for information about tables.]

6.3.2 Mathematics

If you will be needing to typeset mathematical formulae then try as many of these exercises as possible. I will be handing out answers at the end of the class.

There are several different ways of typesetting formulae. They can appear “inline” – that is within a paragraph – like this: $C(n,r) = n!/(r!\,(n-r)$ or separately like this:

$$C(n,r) = n!/(r!\,(n-r)!)$$

so the paragraph is broken up. Chapter 3 of *The Not So Short Introduction to LaTeX2e* describes the different ways displaying mathematics. The Helpful Hints document will also be useful for these exercises.

▷ Exercise 6 Typeset the following: $C(n,r) = n!/(r!\,(n-r)!)$. Note the spacing in the denominator.

▷ Exercise 7 Typeset the equation $a + b = c - d = xy = w/z$ as in-line and displayed mathematical text.

▷ Exercise 8 Typeset the equation $(fg)' = f'g+fg'$ as in-line and displayed mathematical text.

▷ Exercise 9 Typeset $\alpha\beta = \gamma + \delta$ as in-line and displayed mathematical text.

▷ Exercise 10 Typeset $\Gamma(n) = (n-1)!$ as in-line and displayed mathematical text.

▷ Exercise 11 Typeset: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.

▷ Exercise 12 Typeset: $2 + 4 + 6 + \cdots + 2n = n(n+1)$.

▷ Exercise 13 Typeset: $\vec{x} \cdot \vec{y} = 0$ if and only if $\vec{x} \perp \vec{y}$.

▷ Exercise 14 Typeset: $\vec{x} \cdot \vec{y} \neq 0$ if and only if $\vec{x} \not\perp \vec{y}$.

▷ Exercise 15 Typeset: $(\forall x \in \mathbb{R})(\exists y \in \mathbb{R})$ such that $y > x$.

▷ Exercise 16 Typeset the following: $\frac{a+b}{c}$ $\frac{a}{b+c}$ $\frac{1}{a+b+c} \neq \frac{1}{a} + \frac{1}{b} + \frac{1}{c}$.

▷ Exercise 17 Typeset: What are the points where $\frac{\partial}{\partial x} f(x,y) = \frac{\partial}{\partial y} f(x,y) = 0$?

▷ Exercise 18 Typeset each of the following: e^x e^{-x} $e^{i\pi} + 1 = 0$ x_0 x_0^2 ${x_0}^2$ 2^{x^x}.

▷ Exercise 19 Typeset: $\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$.

▷ Exercise 20 Typeset the following expression: $\lim_{x\to 0}(1+x)^{\frac{1}{x}} = e$.

▷ Exercise 21 Typeset: The cardinality of $(-\infty, \infty)$ is $\aleph_1$.

▷ Exercise 22 Typeset: $\lim_{x\to 0^+} x^x = 1$.

Here is a hint to make integrals look a little nicer: look at the difference between $\int_0^x f(t) \mathrm{d}t$ and $\int_0^x f(t)\, \mathrm{d}t$. In the second case there is a little extra space after $f(t)$, and it looks nicer; `\,` was used to add the additional space.

▷ Exercise 23 Typeset the following integral: $\int_0^1 3x^2 \, \mathrm{d}x = 1$.

▷ Exercise 24 Typeset the following: $\sqrt{2}$ $\sqrt{\frac{x+y}{x-y}}$ $\sqrt[3]{10}$ $e^{\sqrt{x}}$.

▷ Exercise 25 Typeset: $\|x\| = \sqrt{x \cdot x}$.

▷ Exercise 26 Typeset: $\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} \, \mathrm{d}x$.

▷ Exercise 27 Typeset the following: $\underline{x}$ $\overline{y}$ $\overline{\underline{x+y}}$.

▷ Exercise 28 Typeset $\lceil \lfloor x \rfloor \rceil \leq \lfloor \lceil x \rceil \rfloor$.

▷ Exercise 29 Typeset: $\sin(2\theta) = 2 \sin\theta \cos\theta$ $\cos(2\theta) = 2\cos^2\theta - 1$.

▷ Exercise 30 Typeset:

$$\int \csc^2 x \, \mathrm{d}x = -\cot x + C \qquad \lim_{\alpha\to 0} \frac{\sin\alpha}{\alpha} = 1 \qquad \lim_{\alpha\to\infty} \frac{\sin\alpha}{\alpha} = 0.$$

▷ Exercise 31 Typeset:

$$\tan(2\theta) = \frac{2\tan\theta}{1 - \tan^2\theta}.$$

▷ Exercise 32 Typeset:

$$\begin{bmatrix} aa & \cdots & az \\ \vdots & \ddots & \vdots \\ za & \cdots & zz \end{bmatrix}$$

▷ Exercise 33 Typeset:
A random variable Y has density

$$f(y; \theta, \phi) = \exp\left\{ \frac{y\theta - b(\theta)}{a\phi} + c(y; \phi) \right\}$$

and its moment-generating function is $M(t) = \exp[\{b(\theta + ta\phi) - b(\theta)\}/(a\phi)]$.

▷ Exercise 34 Typeset: If Y_{rc}, $r = 1, \dots, R$, $c = 1, \dots, C$ are random variables, show that

$$\sum_{r,c}(Y_{rc} - \overline{Y}_{..})^2 = \sum_{r,c}(\overline{Y}_{r.} - \overline{Y}_{..})^2 + \sum_{r,c}(\overline{Y}_{.c} - \overline{Y}_{..})^2 + \sum_{r,c}(Y_{rc} - \overline{Y}_{.c} - \overline{Y}_{r.} + \overline{Y}_{..})^2. \tag{1}$$

▷ Exercise 35 Typeset:

$$f(x_i \,|\, \lambda_i) = \lambda_i e^{-\lambda_i x_i}, \qquad f(y_i \,|\, \lambda_i, \psi) = \lambda_i \psi e^{-\lambda_i \psi y_i}, \qquad x_i, y_i \geq 0.$$

▷ Exercise 36 Typeset:

$$\frac{\partial G}{\partial t} = \lambda s(s-1)\frac{\partial G}{\partial s}.$$

▷ Exercise 37 Typeset:

1. Generate independent uniforms U and U_1.
2. Set $\begin{cases} X = 1/(4U-1), V = U_1/X^2 & \text{if } U < 0.5, \\ X = 4U-3, V = U_1 & \text{otherwise.} \end{cases}$
3. If $V < 1 - 0.5|X|$ go to 5.
4. If $V \geq (1 + X^2/\nu)^{-(\nu+1)/2}$ go to 1.
5. Return X.

▷ Exercise 38 Typeset:

$$h_i(t) = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \frac{\mathsf{P}(t < T_i \leqslant t + \epsilon)}{\mathsf{P}(T_i > t)}.$$

6.3.3 Cross references

▷ Exercise 39 Create a reference to your first section using `\ref` and `\label` commands. See section 2.8 of *The Not So Short Introduction to LaTeX2e* for details.

6.3.4 Including graphs in LaTeX documents

First a graph needs to be saved in the correct format. There are two separate conventions for graphics files.

postscript If your graphs or pictures are in postscript or encapsulated postscript format you must use `latex` and `dvips` to typeset and print documents.

jpg, pdf or png If you graphs or pictures are in JPG, PDF or PNG format then you must use `pdflatex` to typeset and print documents.

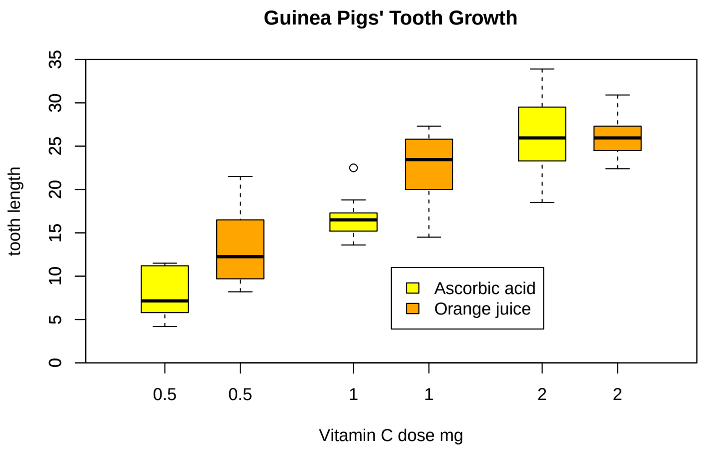
You cannot mix postscript and encapsulated postscript graphs or pictures with any other format

In the following example we will use a .png graph. Include the `GuineaPigPlot.pdf` file in your LaTeX document. To do this add the following line to the preamble of your LaTeX document, that is between the `\documentclass...` and the `\begin{document}`.

```
\usepackage{graphicx}
```

To include the graph found in the file, `GuineaPigPlot.pdf` insert the line

```
\includegraphics[width=0.6\textwidth]{GuineaPigPlot.pdf}
```

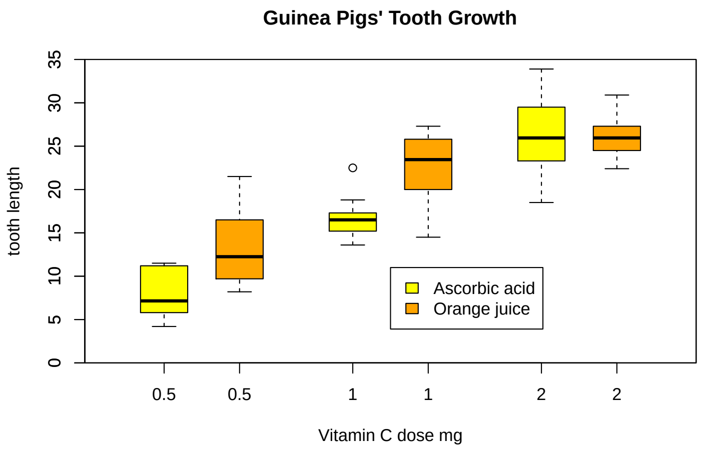


at the place in the text you would like the graph to appear.

Note that I have included `[width=0.6\textwidth]` which scales to 60% of the the page width. This is often useful because the standard size produced by some applications is rather large.

The basic method can be developed. You can centre the graph on the page with the following commands

```
\begin{figure}[ht]
\centering
\includegraphics[width=0.6\textwidth]{GuineaPigPlot.pdf}
\end{figure}
```



By enclosing the plot in a `figure` environment you are able to add extra features such as captions and labels for cross-references.

Including the command `\listoffigures` after `\tableofcontents` will produce a list of figures.

Figures are known as *floats* because they are floated to the nearest sensible position on a page when the `.tex` document is typeset. This means that placement of figures can cause problems; often they will appear at the end of a section of chapter particularly when there are a lot of figures and not much text. The option `[ht]` gives LaTeX the choice positioning the figure either here or at the top of the page. The more choices you offer the less chance that all the pictures will end up on one page.

Here is an example of a centred graph with a caption. See Figure 25.

```
\begin{figure}[ht]
\centering
\includegraphics[width=0.6\textwidth]{GuineaPigPlot.pdf}
\caption{A centred graph with a caption.}
\end{figure}
```

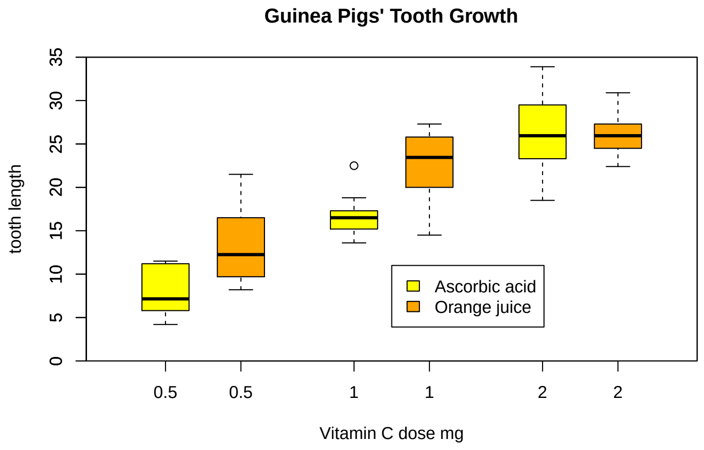


Figure 25: A centred graph with a caption.

There are many more options. Graphs can be rotated using `[angle=n]` where `n` is the angle of rotation. To include two graphs next to each other you need

```
\begin{figure}[ht]
\begin{center}
\includegraphics[width=5cm]{GuineaPigPlot.pdf}
\hspace{1cm}
\includegraphics[width=5cm]{GuineaPigPlot.pdf}
\caption{Two figures next to each other}
\end{center}
\end{figure}
```

which produces the output in figure 26.

To produce two figures next to each other with separate captions use:

```
\begin{figure}[ht]
\begin{center}
\begin{minipage}[b]{.5\textwidth}
\centering
\includegraphics[width=0.7\textwidth]{GuineaPigPlot.pdf}
```

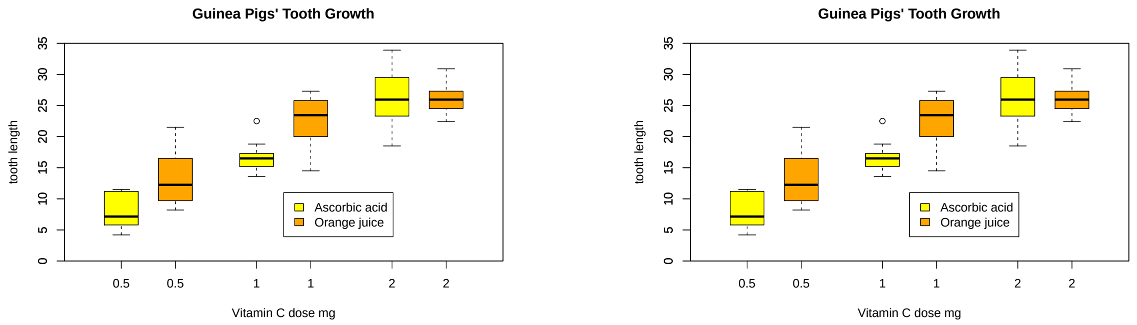


Figure 26: Two figures next to each other

```
\caption{Graph on the left}
\end{minipage}%
\begin{minipage}[b]{.5\textwidth}
\centering
\includegraphics[width=0.7\textwidth]{GuineaPigPlot.pdf}
\caption{Graph on the right}
\end{minipage}%
\end{center}
\end{figure}
```

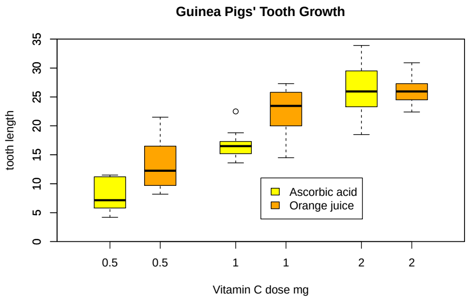


Figure 27: Graph on the left

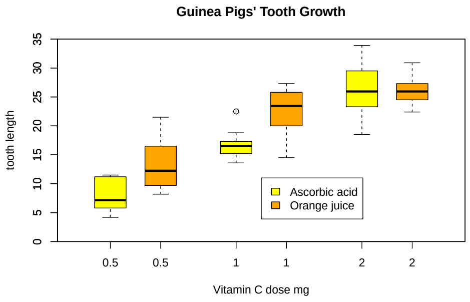


Figure 28: Graph on the right

You may have to experiment with the various `scale` and `width` options. Using `minipage` gives you much more flexibility.

6.3.5 Including R code

Sometimes tutors and supervisors will ask you to include your R code. This is a simple way to do it.

Add the following at the beginning of the file after `\usepackage{times}`.

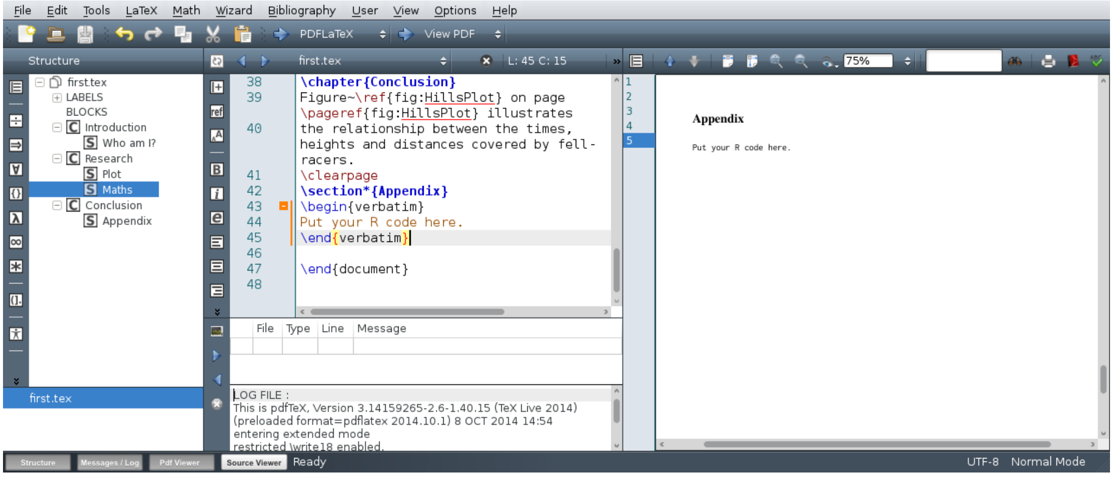
```
\usepackage{zi4}
\usepackage[a4paper,left=3cm,right=3cm,top=3cm,bottom=3cm]{geometry}
```

These lines set the font for verbatim text, and change the dimensions of the page to make them large enough to hold standard length lines.

Then add this just before `\end{document}` at the end of the file.

```
\clearpage
\section*{Appendix}
\begin{verbatim}
Put your R code here.
\end{verbatim}
```

The Appendix should appear on a new page as in Figure 29.



```
\chapter{Conclusion}
Figure~\ref{fig:HillsPlot} on page
\pageref{fig:HillsPlot} illustrates
the relationship between the times,
heights and distances covered by fell-
racers.
\clearpage
\section*{Appendix}
\begin{verbatim}
Put your R code here.
\end{verbatim}

\end{document}
```

```
LOG FILE :
This is pdfTeX, Version 3.14159265-2.6-1.40.15 (TeX Live 2014)
(preloaded format=pdflatex 2014.10.1) 8 OCT 2014 14:54
entering extended mode
restricted \write18 enabled.
```

Appendix

Put your R code here.

Figure 29: An Appendix with R code

What each line means.

`\clearpage`	Forces all figures, tables to be printed before beginning the next section and starts a new page.
`\section*{Appendix}`	Gives the section a title, but no number.
`\begin{verbatim} ... \end{verbatim}`	Typeset everything exactly as it appears.

6.4 Answers to LaTeX exercises

6.4.1 Typesetting Text

▷ Answer 1

> I entered the room and—horrors—I saw both my father-in-law and my mother-in-law.
>
> The winter of 1484–1485 was one of discontent.
>
> Frank wondered, “Is this a girl that can’t say ‘No!’?”
>
> Does Æschylus understand Œdipus?
>
> They took some honey and plenty of money wrapped up in a £5 note.
>
> Élèves, refusez vos leçons! Jetez vos chaînes!
>
> Can you take a ferry from Öland to Åland?

```
I entered the room and---horrors---I saw both my
father-in-law and my mother-in-law.

The winter of 1484--1485 was one of discontent.

Frank wondered, ``Is this a girl that can't say `No!'?''

Does \AE schylus understand \OE dipus?

They took some honey and plenty of money wrapped up in a
\pounds 5 note.

\'El\`eves, refusez vos le\c cons! Jetez vos cha\^\i nes!

Can you take a ferry from \"Oland to \AA land?
```

▷ Answer 2

Don’t forget to run LaTeX twice to make sure the table of contents is up to date.

▷ Answer 3

1. You can mix list environments as much as you like
   - But it might start to look silly
   - With different symbols
2. So do remember

   Stupid things will not become smart because they are in a list.

   Smart things, though, can be presented beautifully in a list.

```
\begin{enumerate}
\item You can mix list environments as much as you like
\begin{itemize}
\item But it might start to look silly
\item[-]With different symbols
\end{itemize}
\item So do remember
\begin{description}
\item[Stupid] things will not become smart because they are in a list.
\item[Smart] things, though, can be presented beautifully in a list.
\end{description}
```

▷ Answer 4

Vegetable Production

Vegetable	Comments	Weight
Carrots	Good early crop, then carrot fly.	7kg
Lettuce	Slow to start, then bolted.	1kg
French beans	Excellent.	12kg

```
\begin{center}
\medskip

{\large \bf Vegetable Production}

\begin{tabular}{|l|l|r|} \hline
{\textsf Vegetable } & {\textsf Comments} & {\textsf Weight}\\
\hline \hline
Carrots & Good early crop, then carrot fly. & 7kg \\
Lettuce & Slow to start, then bolted. & 1kg \\
French beans & Excellent. & 12kg \\ \hline
\end{tabular}
\end{center}
```

▷ Answer 5

Currencies 1 Jan 1992

London:	New York:
£: $ 1.8672	£: $ 1.8655
£: DM 2.8369	$: DM 1.5175
£: FFr 9.969080	$: FFr 5.1845

```
\begin{center}
\medskip
{\Large \bfseries Currencies 1 Jan 1992}\\
\bigskip
\begin{tabular}{ll}
\sffamily \large London: & \sffamily\large New York:\\ \hline \hline
```

```
\pounds : \$ 1.8672 &\pounds : \$ 1.8655 \\
\pounds : DM 2.8369 &\$: DM 1.5175\\
\pounds : FFr 9.969080 &\$: FFr 5.1845
\end{tabular}
\end{center}
```

6.4.2 Mathematics

▷ Answer 6

$C(n,r)=n!/(r!\,(n-r)!).$

```
$C(n,r)=n!/(r!\,(n-r)!).$
```

▷ Answer 7

This equation is in-line $a+b=c-d=xy=w/z$ and the following one is displayed.

$$a+b=c-d=xy=w/z$$

```
This equation is in-line $a+b=c-d=xy=w/z$ and the following
one is displayed. $$a+b=c-d=xy=w/z$$
```

▷ Answer 8

An in-line example: $(fg)' = f'g + fg'$ followed by one that is displayed:

$$(fg)' = f'g + fg'$$

```
An in-line example: $(fg)’ = f’g + fg’$ followed by one
that is displayed: $$(fg)’ = f’g + fg’$$
```

▷ Answer 9

In-line the equation is $\alpha\beta =\gamma+\delta$ and displayed

$$\alpha \beta = \gamma + \delta$$

```
In-line the equation is $\alpha\beta =\gamma+\delta$ and displayed
$$\alpha \beta = \gamma + \delta$$
```

▷ Answer 10

In-line the equation is $\Gamma (n) = (n-1)!$ and displayed

$$\Gamma (n) = (n-1)!$$

```
In-line the equation is $\Gamma (n) = (n-1)!$ and displayed
 $$\Gamma (n) = (n-1)!$$
```

▷ Answer 11

$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z).$

```
$x\wedge (y\vee z) = (x\wedge y) \vee (x\wedge z)$.
```

▷ Answer 12

$2+4+6+\cdots+2n = n(n+1).$

```
$2 + 4 + 6 + \cdots + 2n = n(n+1).$
```

▷ Answer 13

$\vec x\cdot \vec y = 0$ if and only if $\vec x \perp \vec y$.

```
$\vec x\cdot \vec y = 0$ if and only if $\vec x \perp \vec y$.
```

▷ Answer 14

$\vec x\cdot \vec y \not= 0$ if and only if $\vec x \not\perp \vec y.$

```
$\vec x\cdot \vec y \not= 0$ if and only if $\vec x \not\perp \vec y.$
```

or use $\vec x\cdot \vec y \ne 0$ which is `$\vec x\cdot \vec y \ne 0$`

▷ Answer 15

$(\forall x \in \mathbb{R}) (\exists y \in \mathbb{R})$ such that $y>x.$

```
$(\forall x \in \mathbb{R}) (\exists y \in \mathbb{R})$ such that $y>x.$
```

Note that amssymb must be included in the usepackage declaration at the beginning of the file in order to use `\mathbb{R}`.

▷ Answer 16

$\frac{a+b}{c}\quad \frac{a}{b+c}\quad \frac{1}{a+b+c} \not= \frac{1}{a}+\frac{1}{b}+\frac{1}{c}.$

```
$\frac{a+b}{c}\quad \frac{a}{b+c}\quad \frac{1}{a+b+c} \not=
\frac{1}{a}+\frac{1}{b}+\frac{1}{c}.$
```

`\ne` could be used instead of `\not=`

▷ Answer 17

What are the points where $\frac{\partial}{\partial x}f(x,y)=\frac{\partial}{\partial y}f(x,y)=0$?

```
What are the points where $\frac{\partial}{\partial x}f(x,y)=\frac
{\partial}{\partial y}f(x,y)=0$?
```

▷ Answer 18

$e^x \quad e^{-x} \quad e^{i\pi}+1=0 \quad x_0 \quad x^2_0 \quad {x_0}^2\quad 2^{x^x}$.

```
$e^x \quad  e^{-x}  \quad e^{i\pi}+1=0 \quad x_0 \quad  x^2_0
\quad  {x_0}^2\quad  2^{x^x}$.
```

▷ Answer 19

$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$

```
$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2}
+ \frac{\partial^2 f}{\partial y^2}$.
```

▷ Answer 20

$\lim_{x\to 0} (1+x)^{\frac{1}{x}}=e$.

```
$\lim_{x\to 0} (1+x)^{\frac{1}{x}}=e$.
```

▷ Answer 21

The cardinality of $(-\infty ,\infty)$ is $\aleph_1$.

```
The cardinality of $(-\infty ,\infty)$ is $\aleph_1$.
```

▷ Answer 22

$\lim_{x\to {0^+}} x^x = 1$.

```
$\lim_{x\to {0^+}} x^x = 1$.
```

▷ Answer 23

$\int_0^1 \,3x^2\,\mathrm{d}x=1$.

```
$\int_0^1 \,3x^2\,\mathrm{d}x=1$.
```

▷ Answer 24

$\sqrt2 \quad \sqrt \frac{x+y}{x-y} \quad \sqrt[3]{10}$ $\quad$ $e^{\sqrt x}$.

```
$\sqrt2 \quad \sqrt \frac{x+y}{x-y} \quad \root 3 \of {10}$
\quad $e^{\sqrt x}$.
```

▷ Answer 25

$\|x\| = \sqrt{x\cdot x}$.

```
$\|x\| = \sqrt{x\cdot x}$.
```

▷ Answer 26

$\phi(t) = \frac{1}{\sqrt{2_\pi}} \int_0^t \, e^{-x^{2}/2}\,\mathrm{d}x$.

```
$\phi(t) = \frac{1}{\sqrt{2_\pi}} \int_0^t \, e^{-x^{2}/2},\mathrm{d}x$.
```

▷ Answer 27

$\underline x \quad \overline y \quad \underline{\overline{x+y}}$.

```
$\underline x \quad \overline y \quad \underline{\overline{x+y}}$.
```

▷ Answer 28

$\lceil\lfloor x\rfloor\rceil \le \lfloor\lceil x\rceil\rfloor$.

```
$\bigl \lceil \lfloor x\rfloor  \bigr \rceil
   \leq \bigl \lfloor \lceil x\rceil \bigr \rfloor$.
```

▷ Answer 29

$\sin(2\theta) = 2\sin\theta\cos\theta \quad \cos(2\theta) = 2\cos^2\theta - 1$.

```
$\sin(2\theta ) = 2 \sin\theta\cos\theta \quad \cos(2\theta)
 = 2\cos^2\theta - 1$.
```

▷ Answer 30

$$\int \csc^2 x\, \mathrm{d}x = -\cot x + C \qquad \lim_{\alpha\to 0} \frac{\sin\alpha}{\alpha} = 1 \qquad \lim_{\alpha\to\infty} \frac{\sin\alpha}{\alpha} = 0.$$

```
$$\int \csc^2x\,  \mathrm{d}x = -\cot x + C
\qquad \lim_{\alpha\to 0} \frac{\sin\alpha}{\alpha} = 1
\qquad \lim_{\alpha \to \infty} \frac{\sin\alpha}{\alpha} = 0.$$
```

▷ Answer 31

$$\tan(2\theta) = \frac{2\tan\theta}{1-\tan^2\theta}.$$

```
$$\tan (2\theta) = \frac{2\tan\theta}{1-\tan^2\theta}.$$
```

▷ Answer 32

$$\left[\begin{array}{ccc} aa & \cdots & az \\ \vdots & \ddots & \vdots \\ za & \cdots & zz \end{array} \right]$$

```
$$\left[
\begin{array}{ccc}
aa & \cdots & az \\
\vdots & \ddots & \vdots \\
za & \cdots & zz
\end{array}
\right]$$
```

▷ Answer 33

A random variable Y has density

$$f(y;\theta,\phi) = \exp\left\{ \frac{y\theta - b(\theta)}{a\phi} + c(y;\phi) \right\}$$

and its moment-generating function is $M(t) = \exp[\{b(\theta + ta\phi) - b(\theta)\}/(a\phi)]$.

```
A random variable $Y$ has density
$$f(y;\theta,\phi) = \exp\left\{ \frac{y\theta-b(\theta)}
{a\phi} + c(y;\phi)\right\}$$
and its moment-generating function is $M(t)
 $M(t) = \exp[ \{b(\theta+ta\phi) - b(\theta)\}/(a\phi)]$.
```

▷ Answer 34

If Y_{rc}, $r = 1, \ldots, R$, $c = 1, \ldots, C$ are random variables, show that

$$\sum_{r,c}(Y_{rc}-\overline{Y}_{..})^2 = \sum_{r,c}(\overline{Y}_{r.}-\overline{Y}_{..})^2+\sum_{r,c}(\overline{Y}_{.c}-\overline{Y}_{..})^2+\sum_{r,c}(Y_{rc}-\overline{Y}_{.c}-\overline{e}Y_{..})^2. \tag{2}$$

```
If $Y_{rc}$, $r=1,\ldots, R$, $c=1,\ldots, C$ are random variables,
show that
\begin{equation}
\label{linear.models.equation.1}
\sum_{r,c} (Y_{rc}-\overline Y_{..})^2
= \sum_{r,c} (\overline Y_{r.}-\overline Y _{..})^2 + \sum_{r,c}
(\overline Y_{.c}-\overline Y_{..})^2 + \sum_{r,c} (Y_{rc}
- \overline Y_{.c} - \overline e Y_{..})^2.
\end{equation}
```

▷ Answer 35

$$f(x_i\,|\,\lambda_i) = \lambda_i e^{-\lambda_i x_i}, \qquad f(y_i\,|\,\lambda_i, \psi) = \lambda_i\psi e^{-\lambda_i\psi y_i}, \qquad x_i, y_i \ge 0.$$

```
$$f(x_i\,|\,\lambda_i) = \lambda_i e^{-\lambda_i x_i}, \qquad
f(y_i\,|\,\lambda_i, \psi) = \lambda_i\psi e^{-\lambda_i\psi y_i},
\qquad x_i, y_i \ge 0.$$
```

You could use `\mid`, but the spacing would be larger.

▷ Answer 36

$$\frac{\partial G}{\partial t} = \lambda s(s-1)\frac{\partial G}{\partial s}.$$

```
$$\frac{\partial G}{\partial t} = \lambda s(s-1)
\frac{\partial G}{\partial s}.$$
```

▷ Answer 37

1. Generate independent uniforms U and U_1.
2. Set $\begin{cases} X = 1/(4U-1), V = U_1/X^2 & \text{if } U < 0.5, \\ X = 4U-3, V = U_1 & \text{otherwise.} \end{cases}$
3. If $V < 1 - 0.5|X|$ go to 5.
4. If $V \geq (1 + X^2/\nu)^{-(\nu+1)/2}$ go to 1.

5. Return X.

```
\begin{enumerate}
\item Generate independent uniforms $U$ and $U_1$.
\item $
\mbox{Set } \begin{cases}
X = 1/(4U - 1), V = U_1/X^2& \mbox{if $U < 0.5$,}\\
X = 4U - 3, V = U_1 & \mbox{otherwise}.
\end{cases}
$
\item If $V < 1 - 0.5|X|$ go to 5.
\item If $ V \ge (1 + X^2/\nu)^{-(\nu+1)/2}$ go to 1.
\item Return $X$.
\end{enumerate}
```

The amsmaths cases environment was used here.

▷ Answer 38

$$h_i(t) = \lim_{\epsilon\to 0} \frac{1}{\epsilon} \frac{\mathsf{P}(t<T_i\leqslant t+\epsilon)}{\mathsf{P}(T_i>t)}.$$

```
\renewcommand{\Pr}{\mathsf{P}}
$$h_i(t)=\lim_{\epsilon\to 0} \frac{1}{\epsilon}
\frac{\Pr(t<T_i\leqslant t+\epsilon)}{\Pr(T_i>t)}.$$
```

6.4.3 Cross references

▷ Answer 39

To create a cross reference to a figure you need to set a label with an arbitrary name

```
\label{Plot1}
```

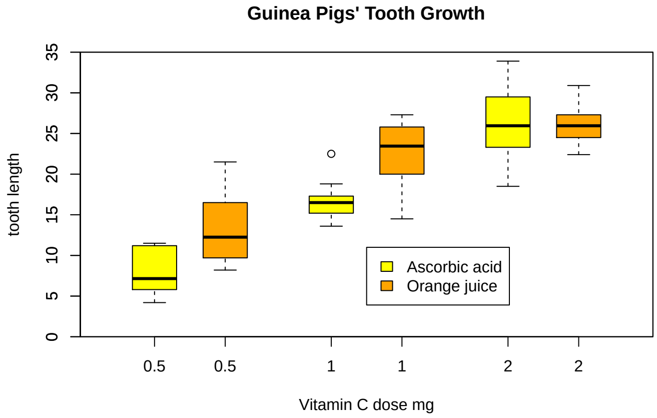


Figure 30: A plot with a reference

within the figure environment and then use

```
\pageref{Plot1}
```

to refer to it. So use

```
this plot is on page \pageref{Plot1}
```

to see “ this plot is on page 55 ”.

See page 55 for example. Note that you will see

```
LaTeX Warning: There were undefined references.
```

so run LaTeX twice to get the references resolved.

6.4.4 A simple bibliography

▷ Answer 40

Remember that you need to run LaTeX twice, then bibtex and then LaTeX twice again to get all the references sorted out.

6.5 Helpful hints

6.5.1 Special Characters

The following symbols are reserved characters, that either have a special meaning under LaTeX or are not available in all the fonts. If you enter them in your text directly, they will normally not print, but rather coerce LaTeX to do things you did not intend.

```
$ & % # _ { }  ~  ^  \
```

As you will see, these characters can be used in your documents all the same by adding a prefix backslash:

$ & % # _ { }	`\$ \& \% \# \_ \{ \}`

The other symbols and many more can be printed with special commands in mathematical formulae or as accents.

6.5.2 Dashes and Hyphens

LaTeX knows four kinds of dashes. You can access three of these with different numbers of consecutive dashes. The fourth sign is actually no dash at all: It is the mathematical minus sign:

daughter-in-law, X-rated	`daughter-in-law, X-rated`
pages 13–67	`pages 13--67`
yes—or no?	`yes---or no?`
0, 1 and −1	`$0$, $1$ and $-1$`

The names for these dashes are: `-` hyphen, `-` en-dash, `--` em-dash and `$-$` minus sign.

6.5.3 Quotation Marks

For quotation marks you should *not* use the `"` as on a typewriter. In publishing there are special opening and closing quotation marks. In LaTeX, use two `‘`s on for opening quotation marks and two `’`s for closing quotation marks.

“Please press the ‘x’ key.”	`‘‘Please press the ‘x’ key.’’`

6.5.4 Accents and Special Characters

LaTeX supports the use of accents and special characters from many languages. Table 18 shows all sorts of accents being applied to the letter o. Naturally other letters work too.

Table 18: **Accents and Special Characters**

ò	`\‘o`	ó	`\’o`	ô	`\^o`	õ	`\~o`
ō	`\=o`	ȯ	`\.o`	ö	`\"o`		
ŏ	`\u o`	ǒ	`\v o`	ő	`\H o`	o̧	`\c o`
ọ	`\d o`	o̱	`\b o`	o͡o	`\t oo`		
œ	`\oe`	Œ	`\OE`	æ	`\ae`	Æ	`\AE`
å	`\aa`	Å	`\AA`				
ø	`\o`	Ø	`\O`	ł	`\l`	Ł	`\L`
ı	`\i`	ȷ	`\j`	¡	`!‘`	¿	`?‘`

To place an accent on top of an i or a j, their dots have to be removed. This is accomplished by typing `\i` and `\j`. Here are some more examples.

Hôtel, naïve, élève,	H\^otel, na\"\i ve, \'el\`eve,
smørrebrød, ¡Señorita!,	sm\o rrebr\o d, !`Se\~norita!,
Schönbrunner Schloß Straße	Sch\"onbrunner Schlo\ss{} Stra\ss e

6.5.5 List of Mathematical Symbols

In the following tables you find all the symbols normally accessible from *math mode.*

Table 19: **Math Mode Accents**

$\hat{a}$	\hat{a}	$\check{a}$	\check{a}	$\tilde{a}$	\tilde{a}	$\acute{a}$	\acute{a}
$\grave{a}$	\grave{a}	$\dot{a}$	\dot{a}	$\ddot{a}$	\ddot{a}	$\breve{a}$	\breve{a}
$\bar{a}$	\bar{a}	$\vec{a}$	\vec{a}	$\widehat{A}$	\widehat{A}	$\widetilde{A}$	\widetilde{A}

Table 20: **Lowercase Greek Letters**

α	\alpha	θ	\theta	o	o	υ	\upsilon
β	\beta	ϑ	\vartheta	π	\pi	ϕ	\phi
γ	\gamma	ι	\iota	ϖ	\varpi	φ	\varphi
δ	\delta	κ	\kappa	ρ	\rho	χ	\chi
ϵ	\epsilon	λ	\lambda	ϱ	\varrho	ψ	\psi
ε	\varepsilon	μ	\mu	σ	\sigma	ω	\omega
ζ	\zeta	ν	\nu	ς	\varsigma		
η	\eta	ξ	\xi	τ	\tau		

Table 21: **Uppercase Greek Letters**

Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		

Table 22: **Binary Relations**

You can produce corresponding negations by adding a `\not` command as prefix to the following symbols.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` or `\le`	$\geq$	`\geq` or `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$	`\sqsubset` [a]	$\sqsupset$	`\sqsupset` [a]	$\Join$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a] Use the latexsym package to access this symbol

Table 23: **Binary Operators**

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$\ast$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

Table 24: **BIG Operators**

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

Table 25: **Arrows**

$\leftarrow$	`\leftarrow` or `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` or `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\iff$	`\iff` (bigger spaces)	$\leadsto$	`\leadsto` [a]

[a] Use the latexsym package to access this symbol

Table 26: **Delimiters**

$($	`(`	$)$	`)`	$\uparrow$	`\uparrow`	$\Uparrow$	`\Uparrow`
$[$	`[` or `\lbrack`	$]$	`]` or `\rbrack`	$\downarrow$	`\downarrow`	$\Downarrow$	`\Downarrow`
$\{$	`\{` or `\lbrace`	$\}$	`\}` or `\rbrace`	$\updownarrow$	`\updownarrow`	$\Updownarrow$	`\Updownarrow`
$\langle$	`\langle`	$\rangle$	`\rangle`	$\vert$	`|` or `\vert`	$\Vert$	`\\|` or `\Vert`
$\lfloor$	`\lfloor`	$\rfloor$	`\rfloor`	$\lceil$	`\lceil`	$\rceil$	`\rceil`
$/$	`/`	$\backslash$	`\backslash`		. (dual. empty)		

Table 27: **Large Delimiters**

$\Bigg\lgroup$	`\lgroup`	$\Bigg\rgroup$	`\rgroup`	$\Bigg\lmoustache$	`\lmoustache`	$\Bigg\rmoustache$	`\rmoustache`
$\arrowvert$	`\arrowvert`	$\Arrowvert$	`\Arrowvert`	$\bracevert$	`\bracevert`		

Table 28: **Miscellaneous Symbols**

$\dots$	`\dots`	$\cdots$	`\cdots`	$\vdots$	`\vdots`	$\ddots$	`\ddots`
$\hbar$	`\hbar`	$\imath$	`\imath`	$\jmath$	`\jmath`	ℓ	`\ell`
$\Re$	`\Re`	$\Im$	`\Im`	$\aleph$	`\aleph`	$\wp$	`\wp`
$\forall$	`\forall`	$\exists$	`\exists`	$\mho$	`\mho` [a]	∂	`\partial`
$'$	`'`	$\prime$	`\prime`	$\emptyset$	`\emptyset`	∞	`\infty`
∇	`\nabla`	$\triangle$	`\triangle`	$\Box$	`\Box` [a]	$\Diamond$	`\Diamond` [a]
$\bot$	`\bot`	$\top$	`\top`	$\angle$	`\angle`	$\surd$	`\surd`
$\diamondsuit$	`\diamondsuit`	$\heartsuit$	`\heartsuit`	$\clubsuit$	`\clubsuit`	$\spadesuit$	`\spadesuit`
$\neg$	`\neg` or `\lnot`	$\flat$	`\flat`	$\natural$	`\natural`	$\sharp$	`\sharp`

[a] Use the latexsym package to access this symbol

Table 29: **Non-Mathematical Symbols**

These symbols can also be used in text mode.

$\dagger$	`\dag`	$\S$	`\S`	©	`\copyright`
$\ddagger$	`\ddag`	$\P$	`\P`	£	`\pounds`

The symbols in the following tables all require the amssymb package.

Table 30: **AMS Delimiters**

$\ulcorner$	`\ulcorner`	$\urcorner$	`\urcorner`	$\llcorner$	`\llcorner`	$\lrcorner$	`\lrcorner`

Table 31: **AMS Greek and Hebrew**

$\digamma$	`\digamma`	$\varkappa$	`\varkappa`	$\beth$	`\beth`	$\daleth$	`\daleth`	$\gimel$	`\gimel`

Table 32: **AMS Binary Relations**

$\lessdot$	`\lessdot`	$\gtrdot$	`\gtrdot`	$\doteqdot$	`\doteqdot` or `\Doteq`
$\leqslant$	`\leqslant`	$\geqslant$	`\geqslant`	$\risingdotseq$	`\risingdotseq`
$\eqslantless$	`\eqslantless`	$\eqslantgtr$	`\eqslantgtr`	$\fallingdotseq$	`\fallingdotseq`
$\leqq$	`\leqq`	$\geqq$	`\geqq`	$\eqcirc$	`\eqcirc`
$\lll$	`\lll` or `\llless`	$\ggg$	`\ggg` or `\gggtr`	$\circeq$	`\circeq`
$\lesssim$	`\lesssim`	$\gtrsim$	`\gtrsim`	$\triangleq$	`\triangleq`
$\lessapprox$	`\lessapprox`	$\gtrapprox$	`\gtrapprox`	$\bumpeq$	`\bumpeq`
$\lessgtr$	`\lessgtr`	$\gtrless$	`\gtrless`	$\Bumpeq$	`\Bumpeq`
$\lesseqgtr$	`\lesseqgtr`	$\gtreqless$	`\gtreqless`	$\thicksim$	`\thicksim`
$\lesseqqgtr$	`\lesseqqgtr`	$\gtreqqless$	`\gtreqqless`	$\thickapprox$	`\thickapprox`
$\preccurlyeq$	`\preccurlyeq`	$\succcurlyeq$	`\succcurlyeq`	$\approxeq$	`\approxeq`
$\curlyeqprec$	`\curlyeqprec`	$\curlyeqsucc$	`\curlyeqsucc`	$\backsim$	`\backsim`
$\precsim$	`\precsim`	$\succsim$	`\succsim`	$\backsimeq$	`\backsimeq`
$\precapprox$	`\precapprox`	$\succapprox$	`\succapprox`	$\vDash$	`\vDash`
$\subseteqq$	`\subseteqq`	$\supseteqq$	`\supseteqq`	$\Vdash$	`\Vdash`
$\Subset$	`\Subset`	$\Supset$	`\Supset`	$\Vvdash$	`\Vvdash`
$\sqsubset$	`\sqsubset`	$\sqsupset$	`\sqsupset`	$\backepsilon$	`\backepsilon`
$\therefore$	`\therefore`	$\because$	`\because`	$\varpropto$	`\varpropto`
$\shortmid$	`\shortmid`	$\shortparallel$	`\shortparallel`	$\between$	`\between`
$\smallsmile$	`\smallsmile`	$\smallfrown$	`\smallfrown`	$\pitchfork$	`\pitchfork`
$\vartriangleleft$	`\vartriangleleft`	$\vartriangleright$	`\vartriangleright`	$\blacktriangleleft$	`\blacktriangleleft`
$\trianglelefteq$	`\trianglelefteq`	$\trianglerighteq$	`\trianglerighteq`	$\blacktriangleright$	`\blacktriangleright`

Table 33: **AMS Arrows**

$\dashleftarrow$	\dashleftarrow	$\dashrightarrow$	\dashrightarrow	$\multimap$	\multimap
$\leftleftarrows$	\leftleftarrows	$\rightrightarrows$	\rightrightarrows	$\upuparrows$	\upuparrows
$\leftrightarrows$	\leftrightarrows	$\rightleftarrows$	\rightleftarrows	$\downdownarrows$	\downdownarrows
$\Lleftarrow$	\Lleftarrow	$\Rrightarrow$	\Rrightarrow	$\upharpoonleft$	\upharpoonleft
$\twoheadleftarrow$	\twoheadleftarrow	$\twoheadrightarrow$	\twoheadrightarrow	$\upharpoonright$	\upharpoonright
$\leftarrowtail$	\leftarrowtail	$\rightarrowtail$	\rightarrowtail	$\downharpoonleft$	\downharpoonleft
$\leftrightharpoons$	\leftrightharpoons	$\rightleftharpoons$	\rightleftharpoons	$\downharpoonright$	\downharpoonright
$\Lsh$	\Lsh	$\Rsh$	\Rsh	$\rightsquigarrow$	\rightsquigarrow
$\looparrowleft$	\looparrowleft	$\looparrowright$	\looparrowright	$\leftrightsquigarrow$	\leftrightsquigarrow
$\curvearrowleft$	\curvearrowleft	$\curvearrowright$	\curvearrowright		
$\circlearrowleft$	\circlearrowleft	$\circlearrowright$	\circlearrowright		

Table 34: **AMS Negated Binary Relations and Arrows**

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

Table 35: **AMS Binary Operators**

$\dotplus$	`\dotplus`	$\centerdot$	`\centerdot`	$\intercal$	`\intercal`
$\ltimes$	`\ltimes`	$\rtimes$	`\rtimes`	$\divideontimes$	`\divideontimes`
$\Cup$	`\Cup` or `\doublecup`	$\Cap$	`\Cap` or `\doublecap`	$\smallsetminus$	`\smallsetminus`
$\veebar$	`\veebar`	$\barwedge$	`\barwedge`	$\doublebarwedge$	`\doublebarwedge`
$\boxplus$	`\boxplus`	$\boxminus$	`\boxminus`	$\circleddash$	`\circleddash`
$\boxtimes$	`\boxtimes`	$\boxdot$	`\boxdot`	$\circledcirc$	`\circledcirc`
$\leftthreetimes$	`\leftthreetimes`	$\rightthreetimes$	`\rightthreetimes`	$\circledast$	`\circledast`
$\curlyvee$	`\curlyvee`	$\curlywedge$	`\curlywedge`		

Table 36: **AMS Miscellaneous**

$\hbar$	`\hbar`	$\hslash$	`\hslash`	$\Bbbk$	`\Bbbk`
$\square$	`\square`	$\blacksquare$	`\blacksquare`	$\circledS$	`\circledS`
$\vartriangle$	`\vartriangle`	$\blacktriangle$	`\blacktriangle`	$\complement$	`\complement`
$\triangledown$	`\triangledown`	$\blacktriangledown$	`\blacktriangledown`	$\Game$	`\Game`
$\lozenge$	`\lozenge`	$\blacklozenge$	`\blacklozenge`	$\bigstar$	`\bigstar`
$\angle$	`\angle`	$\measuredangle$	`\measuredangle`	$\sphericalangle$	`\sphericalangle`
$\diagup$	`\diagup`	$\diagdown$	`\diagdown`	$\backprime$	`\backprime`
$\nexists$	`\nexists`	$\Finv$	`\Finv`	$\varnothing$	`\varnothing`
$\eth$	`\eth`	$\mho$	`\mho`		

6.6 Going Further

Useful links:

- `http://www.ctan.org/what_is_tex.html`
- `http://theoval.cmp.uea.ac.uk/~nlct/latex/` This link is particularly useful for beginners and DPhil students as it contains a tutorial for beginners and a guide to writing a thesis using LaTeX.
- `http://detexify.kirelabs.org/classify.html#new` Find the markup for symbols.

Table 37: **Math Alphabets**

Example	Command	Required package
ABCdef	`\mathrm{ABCdef}`	
ABCdef	`\mathit{ABCdef}`	
$\mathnormal{ABCdef}$	`\mathnormal{ABCdef}`	
$\mathcal{ABC}$	`\mathcal{ABC}`	
$\mathbb{ABC}$	`\mathbb{ABC}`	amsfonts or amssymb

7 Acknowledgements

I would like to thank Professor Brian Ripley for permission to use material from exercises he devised for the Department of Statistics as part of an introductory LaTeX course.

References

Goossens, M., Mittelbach, F., & Samarin, A. (1994). *The LaTeX Companions.* Reading, Massachusetts: Addison Wesley, 1st edition.

Lamport, L. (1994). *LaTeX: A Document Preparation System.* Reading, Massachusetts: Addison Wesley, 2nd edition.