Python Data Science Bootcamp
Getting acquainted with the terminal

Sarah Middleton

Last Updated: July 9, 2020

1 Using the terminal

What is the terminal?

The terminal, or command line, is a way of interacting with your computer using text-based commands rather than pointing and clicking with a mouse. You can think of it sort of like a text-only version of the graphical windows you normally use to navigate directories (folders) and files on your computer.

Just like when you use the graphical interface, you are always "in" some folder when you use the command line. This is called the "current directory". When you first open the terminal, your will be in your "home directory". This may be a strange system folder you've never seen before, but that's ok. Using certain commands, we can figure out which folder we're in, display the contents of the folder, as well as navigate to other folders. You can do most of the things you normally do when you use the graphical interface, such as open files, delete files, make new folders, move files around, run programs, and more. **We can also do many things with the terminal that we can not do with the graphical folder system–this is why it's so important to learn.** It will feel slow at first, but stick with it and it will soon feel more natural.

Anatomy of a command

A command will usually have three basic parts: the command itself, arguments, and options. A basic command might look something like this:

```
someCommand -o filename
```

`someCommand` - this is the basic command. It may or may not require further args/options.

`-o` - This is an option, which modifies the behavior of the command in some way. Options are usually optional. For example, if you use the `less` command to view the contents of

a file, you can optionally put `less -S` to view the file without line-wrapping. Use the `man` command (short for "manual", see next section) to find out what options are available to each command. Note that the Windows terminal uses a forward slash (e.g. `/o`) to indicate options instead of a dash.

`filename` - This is what is sometimes called an "argument", or arg for short. An arg is just a piece of information, such as the name of a file, that may be optional or required for the command to work. Again, always refer to the `man` page if you're not sure what args are needed for a command.

Basic commands

Below is a list of basic terminal commands that we'll be using throughout the course.

Command	Purpose
ls	Display contents of current directory (files, folders)
cd [dirname]	Change to a directory
cd ..	Go up/back one directory
man [command]	Manual. Display detailed
mkdir [dirname]	Make a new directory within the current one
pwd	Prints the path of the current directory
less	View contents of a file one

There are many more commands, but this is all we need for now.

Other useful tips

- Typing `cd` with no file name brings you back to your home folder from anywhere (Unix only).

- Tab auto-completion: if you start typing the name of a file/folder and then hit tab, the rest of the name will automatically be completed for you–but only it is in the current directory and there is no other file/folder in that directory with the same prefix that you have typed so far.

- To reuse a command you already entered previously, use the up and down arrow keys to cycle through your command history.

- When you use commands like `man` or `less`, your screen will be filled with text and you can use your arrow keys to scroll up and down (Unix only. In Windows use the enter key). Press `q` to go back to the command line (everything you had before will still be there).

2 Further reading

If you want to learn more about using the Unix terminal, I highly recommend the Unix/terminal introduction found here: `http://korflab.ucdavis.edu/Unix_and_Perl/unix_and_perl_v2.3.7.pdf`. It goes into a lot more detail and explains some of the nuances that I've glossed over, but is very readable and newbie-friendly. The Unix stuff starts on pg. 15 (you can ignore the Perl stuff!).