Session 4

**Creating a directory**

Inside the Data\_Files\_TMCS directory, make a new directory called risk\_assessment using the mkdir command. This has no output.

$ mkdir risk\_assessment

mkdir means ‘make directory’. Since risk\_assessment is a relative path (it doesn’t have a leading /) the new directory is created in the current working directory. You can now see if this directory has been made by using ls -F or using the file explorer in your GUI.

Always remember to be sensible with filenames.

**Creating a text file**

Change your working directory to your new risk\_assessment directory

Now start a text editor. We will describe nano here.

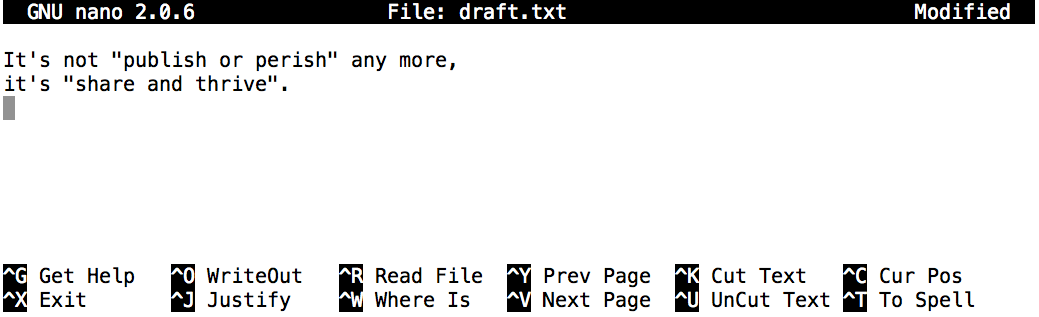
$ nano risk\_assesment.txt

**Very quick Nano tutorial**

Nano is one of the least complex text editors. However, because of this trait, it’s not very powerful. VSCode is a great IDE, or for something a bit lighter, use Sublime Text. Nano’s advantage is it can be run from the command line. You can also use Vim or Emacs if you wish (both of which can be launched from the command line)

If you start Nano from the shell, it will (probably) use your current working directory as its default location.

Type in a few lines of text. When finished press Ctrl+O to write the data to disk (you’ll be asked what file to save to: press Return to accept the suggested default of risk\_assesment.txt).



Once the file is saved, use Ctrl+X to quit the editor and return to the shell.

Control, Ctrl, or ^ Key

In nano, along the bottom of the screen you’ll see ^G Get Help ^O WriteOut. This means that you can use Control-G to get help and Control-O to save your file.

nano doesn’t leave any output on the screen after it exits, but ls now shows that there is a file called risk\_assesment.txt:

$ ls

risk\_assesment.txt

### Quick Questions:

### Try the using the following command: touch test\_file.txt

### What did it do? When you look up the current directory using the GUI does the file appear?

### Use ls -l to inspect the files. How large is your new text file?

### Why would you create a file in this way

### Moving files and directories

### Return to the Data\_Files\_TMCS directory.

In our risk\_assessment directory we have risk\_assessment.txt. We can change this to be more descriptive using mv, which is short for ‘move’:

$ mv risk\_assessment/risk\_assessment.txt risk\_assessment/travelling\_risk\_assessments.txt

The first argument tells mv what we’re ‘moving’, and the second tells it where.

We move risk\_assessment/risk\_assessment.txt to. risk\_assessment/travelling\_risk\_assessments.txt, which is the same as renaming the file.

Run ls to confirm this. Careful though as mv will silently overwrite any existing file with the same name.

mv also works on directories. We can move travelling\_risk\_assessments.txt into the working directory using the special directory name .

$ mv risk\_assessment/travelling\_risk\_assessments.txt .

The effect is to move the file from the directory it was in to the current working directory.

### Quick Questions:

1. How would you force mv to ask for confirmation before overwriting?

### Copying files and directories

cp works similarly to mv, except it copies instead of moves.

$ cp travelling\_risk\_assessments.txt risk\_assessment/travelling\_risk\_assessments\_copy.txt

$ ls travelling\_risk\_assessments.txt risk\_assessment/travelling\_risk\_assessments\_copy.txt

travelling\_risk\_assessments.txt risk\_assessment/travelling\_risk\_assessments\_copy.txt

It’s also possible to copy the entire contents of a directory by using the recursive option  -r:

$ cp **-r** risk\_assessment risk\_assessment\_backup

### Quick Questions:

1. Create some of your own text files. Name them whatever you like. Now rename them.
2. What directory does the closing ls list?

$ pwd

/Users/jamie/data

$ ls

proteins.dat

$ mkdir recombine

$ mv proteins.dat recombine/

$ cp recombine/proteins.dat ../proteins-saved.dat

$ ls

### Removing files and directories

Returning to the Data\_Files\_TMCS directory. Use rm (short for ‘remove’) to get rid of travelling\_risk\_assessments.txt.

$ rm travelling\_risk\_assessments.txt

**!!!!!!!!!!!!!!!!!Caution! Deleting in this way is forever!!!!!!!!!!!!!!!!!!!!!!!!!!!**

### Quick Questions:

1. Find a flag for rm that forces rm to ask for confirmation

rm by default only works on files. It will not delete directories. To make it delete a directory, we have to tell it to delete each file in the directory. This is achieved by using the recursive option -r.

$ rm **-r** risk\_assessment

**!!!!!!!!!!!!!!!!!Caution! Deleting in this way is forever!!!!!!!!!!!!!!!!!!!!!!!!!!!**

**!!!!!!!!!!!!!!!! IT WILL DELETE EVERYTHING INDISCRIMINANTLY!!!!!!!!!!!!!!**

### Quick Questions:

1. Find a flag for rm -r that forces rm -r to ask for confirmation

### Working with multiple files and directories

### Moving lots of files at once is relatively straightforward.

$ cp test.txt test1.txt backup/

Using 2 files names and a directory name here will copy the files to the named directory.  When used like this cp will assume the third input is a directory. If the third input isn’t a directory it will throw up an error message.

### Wildcards

\* and ? are wildcards.

Go to the Data\_Files\_TMCS/molecules directory.

\*.pdb matches ethane.pdb, propane.pdb, and every file that ends with ‘.pdb’. On the other hand, p\*.pdb only matches pentane.pdb and propane.pdb, because the ‘p’ at the front only matches filenames that begin with the letter ‘p’.

The ? wildcard matches just one character. ?ethane.pdb would match methane.pdb whereas \*ethane.pdb matches both ethane.pdb, and methane.pdb.

Wildcards can be used in combination with each other e.g. ???ane.pdb matches three characters followed by ane.pdb, giving cubane.pdb ethane.pdb octane.pdb.

When the shell sees a wildcard, it expands the wildcard to create a list of matching filenames before running the command that was asked for. As an exception, if a wildcard expression does not match any file, Bash will pass the expression as an argument to the command as it is. For example typing ls \*.pdf in the molecules directory (which contains only files with names ending with .pdb) results in an error message that there is no file called \*.pdf.

### Quick Questions:

1. Try placing wildcards in the middle of file names. What happens? (For example et\*ane.pdb)
2. Try combing wild cards? Try using more than one of the same? Try using multiple wildcards? (For example ??han\*.pdb)

**Takeaways**

1. cp old new copies a file.
2. mkdir path creates a new directory.
3. mv old new moves (renames) a file or directory.
4. rm path removes a file.
5. \* matches zero or more characters in a filename, so \*.txt matches all files ending in .txt.
6. ? matches any single character in a filename, so ?.txt matches h.txt but not house.txt.
7. Use of the Control key may be described in many ways, including Ctrl-X, Control-X, and ^X.
8. The shell does not have a trash bin: once something is deleted, it’s really gone.
9. Most files’ names are something.extension. The extension isn’t required, and doesn’t guarantee anything, but is normally used to indicate the type of data in the file.