Useful Unix Commands

Command	Description	Examples
`ls`	List directory content	
`mkdir` *dir*	Make a directory	`mkdir projects` –make the directory projects `mkdir docs` –make the directory docs `mkdir junk` –make the directory junk
`rmdir` *dir*	Remove a directory (directory must be empty; otherwise use "rm")	`rmdir junk` –remove the directory `junk`
`cd` *dir*	Change directory	`cd /projects` – move to the `projects`directory (an absolute path) `cd projects` – move to the `projects`directory, assuming we are already in the home directory (a relative path)
`cd ..`	Go up one directory to the parent directory	`cd ../..` – move up two parent directories from our current directory
`cd ~`	Go to the home directory	
`cd -`	Go to whatever directory you just left	
`pwd`	Print the present working directory	
Tab key	Autocomplete	`cd d` + tab – autocompletes to `docs` if it is the only directory that begins with d; or list the different options.
`mv` *file1 file2*	Move or rename files *Warning –this is permanent, and you will not get a warning message if you are overwriting files.*	`mv ~/docs/resumes/cv.tex ~/docs/reports/` –move the `cv.tex` file from the resume folder to the reports folder `mv cv.tex resume.tex` – rename `cv.tex`to `resume.tex` `mv ~/docs/resumes ~ /docs/reports/-` move the resume folder into the reports folder
`cp` *file1 file2*	Copy file1 to file2	`cp ~ ~/docs/reports/` – make a copy of the `cv.tex` file from the `resume` folder in the `reports` folder
`rm` *file*	Delete file	`rm ~/docs/resumes/cv.tex` – delete the file `cv.tex`

Command	Description	Examples
	Warning – this is permanent! You cannot retrieve files from the recycling bin!	
`less` *file*	View file	`less ~/docs/resumes/cv.tex` – open `cv.tex` in the less text viewer
`rm -r` *dir*	Remove recursively all folders in directory *dir* and the directory itself.	
`ls -a`	List all directory content, including hidden files	
`ls -l`	List all directory content in long form (including permissions, size and date)	
`ls -t`	List all directory content in chronological order	`ls -lart` – show more information for all files in reverse chronological order for your current directory
`man` *command*	Show the manual for the command. Note – this does not work for GitBash	`man ls` – show the manual instructions for the command ls.
`help`	Show the manual for the command in GitBash	`ls --help` – show help instructions for the command `ls`
command1 | *command2*	Pipe the results of command 1 to command 2	`man ls | less` – show the help instructions for the command ls in the less viewer
`*` (wildcard)		`ls *.html` –list all the files ending in html in your current directory `rm *.html` – remove all files ending in html in your current directory
`?` (any character)		`rm file.???.html` – remove all files whose names follow the pattern; eg file-001.html, file-002.html etc. `rm file.???.*` – remove all files whose names follow the pattern regardless of their extension; eg file-001.html, file-002.csv, file-any.R, etc.
$*var*	>$ identifies a variable	`echo $HOME` – print your home directory `echo $SHELL` – print your shell name
`export` *val=value*	Change the value of the variable *val* (Bash shell specific)	

Command	Description	Examples
`open` *file* (mac)*file* (windows)	Opens a file or program	`open Report.Rmd` – open Report.Rmd in RStudio

Absolute path vs. relative path

A full path specifies the location of a file from the root directory. It is independent of your present directory, and must begin with either a "/" or a "~". In this example, the full path to our "project-1" file is:

```
/home/projects/project-1
```

A relative path is the path relative to your present working directory. If our present working directory is the "projects" folder, then the relative path to our "project-1" file is simply:

```
project-1
```

Path shortcuts

One period "." is your current working directory

Two periods ".." is the parent directory (up one from your present working directory)

A tilde "~" is your home directory.

More path examples

1. Your current working directory is `~/projects` and you want to move to the `figs` directory in the `project-1`folder

- Solution 2: `cd ~/projects/project-1/figs` (absolute)
- Solution 2: `cd project-1/figs` (relative)

2. Your current working directory is `~/projects` and you want to move to the `reports` folder in the `docs` directory

- Solution 1: `cd ~/dos/reports` (absolute)
- Solution 2: `cd ../docs/reports` (relative)

3. Your current working directory is `~/projects/project-1/figs` and you want to move to the `project-2` folder in the `projects` directory.

- Solution 1: `cd ~/projects/project-2` (absolute)
- Solution 2: `cd ../../project-2` (relative)