**COMMAND LINE INTERFACE**

0:00

This is an introduction to the command interface. The command line interface is a way of working with files and folders that involves typing a command as opposed to pointing and clicking with a mouse. It's incredibly useful for data science and will be used both when you programming in R and when you're using other tools in the data scientist tool box.

0:20

So, what is the command line interface? Almost every computer that's shipped today comes with some form of a command line interface. For this class, the command line interfaces we'll be using, for Windows, it's Git Bash and you'll see more about that in the "Introduction to Git". And for Mac and Linux, it's the terminal.

0:38

So what can the Command Line Interface do? Well, you can use it to navigate between folders. You can use it to create files, folders, and programs, and then you can use it to edit those files and folders. Finally you can use it to run computer programs which is one of the main uses we'll be having for it in a data science program.

0:56

So first we need to understand the basics of directories. So directory is just another name for folder, and it's the common name that you'll be hearing about a lot in this class. So being in a directory and moving to the directory.

1:08

Directories on your computer are organized like a tree. So there's one directory will be inside of another directory. And so we can navigate between those directories using the command line interface. So, here's an example, we have a music directory and there are three sub directories, one for each composer here, and so you can imagine that the Debussy directory is inside of the music directory and so when we think about the structure of this directory. We think of being either inside of a directory or being up from a directory. So, one directory up from the Debussy directory is the music directory. In general, the directory that's up from the directory that you're thinking of is the directory that contains the directory you're thinking of.

1:56

So the directory structure on your computer looks something like this. You have sort of a tree structure where each sub, each directory has sub directories.

2:06

And so a couple of special cases that you might want, that need to be considered. First it's the root directory. So this is the root directory up here at the top of the tree. And it's indicated at the top because it contains all of the other directories. And it's usually the shorthand for the new directory is a slash. So a slash indicates the new directory.

2:28

The other main directory that you might want to be paying attention to is your home directory. So that's down here. So, the home directory is where you log in when you log in your computer and most of your personal stuff, your photos, your music, your R function will all be in your personal directory.

2:45

And so that would be a sub directory of the root directory again.

2:51

So if you want to navigate between those directories using the command line interface, first you have to have a command line interface. So, for Windows users you should open the Start menu, search for git bash, and then open git bash the application.

3:07

If you're a Mac user, you open Spotlight, search for terminal, and then open the terminal application.

3:13

When you open the terminal or GitBash, you'll see something that looks a lot like this. So this is basically application where you mostly see blank space. And so what will see is what's called the prompt. Which is your username. Here in this case it's Sean followed by . And then so when you open this up, when you open up the command line interface, you start in your home directory. An important term that we'll be using a lot is called your working directory. So that's whatever directory you're in at the moment. So when you open you're command line interface, you're working directory is your home directory. If you move into a sub directory called music, then your working directory will be music.  So another thing to keep in mind is the path, so, people think about the path in the directory structure. And so we'll be moving around along the path. And so, suppose that you're down here, you're down in your home directory which is signified by the tilde. And so you might want to know what are the directors that are, directories that are up from your home directory. So the first directory that's up is the users directory. And then the next directory up is the root directory. So you can imagine the path is the set of directories you have to follow in order to get back up to the root directory.

4:33

So if you type PWD when you're in the command line interface. So here we are at that prompt. You can see the username, the dollar sign. Then if you type PWD and hit return, what you'll see is the path. So, this is the path to the working directory that you're in right now. So in this case, this person is in the root directory, then users, and underneath that they're in Sean. So, that's the path to the working directory that they're in. So, PWD stands for print working directory.

5:08

If you use PWD, you can use that at any time in your directory, it will always print the directory that you're in, that you're looking at right then.

5:18

So, there's always a recipe that goes with using command line interfaces. So what you'll do is you'll use what's called the command. So in the previous case it was called PWD. Then you'll use something called flags. These are like parameters or things that you'll pass to the command that tell it what kind of behavior to take. And then arguments can be what command is going to modify so what files it'll look at and things like that.

5:45

Depending on the command, you may or may not need to use flags or arguments. So for example when you were using PWD, it was a command and it didn't require any sort of options, it didn't require any flags or arguments in order to change what was happening.

6:00

So again, what we can use is we can use pwd to print the current working directory. So, if I'm a different user then, when i go to my home directory and I type PWD, I'll actually get a different path. It's the path, users followed by Jeff because now I'm a different user. And so I have a different current working directory.

6:21

Clear will clear all the commands in your current command line interface. So for example, suppose you typed a bunch of commands. In this case, we typed PWD, and we have something here in users Jeff, and we don't want to have that cluttering up the space in our interface anymore. Then what we can do is type just the word clear, and that will clear out everything in the command line interface, and you'll be left with just a prompt at the top of the screen again.

6:46

And a very important command is LS. So, LS lists files and folders in the current directory. So, if you type LS, for example, in direc, directory, you might get all the subfolders. So for example, you might get the desktop, the photos, and music. And last, dash A lists both hidden and unhidden files. And so for example, the desktop, photos, and music were all unhidden, but then there are these files, the trash files and on a Mac and .DS underscore store file. Those files are start, start with a dot and so they're hidden when you type LS. They'll also be hidden if you look at in a folder structure on the computer. But LS dash A reveals those folders.

7:31

LS dash A lists some details for those folders.

7:35

So for example it lists information about the size of the folders and stuff like that.

7:39

An important note is that both dash A and dash L are flags, so

7:45

they're prec, preceded by a dash and often a flag is preceded by a dash. You can combine them together into one flag by just catnating the two letters together after the dash.  CD is another command. It stands for change directory. It takes as an argument the directory that you want to go visit. So if you're in a particular directory and you want to move to another directory, you can pass to CD a particular path and it will take you to that directory. So if you type CD with no argument at all it'll just take you straight to your home directory.

8:23

And then an important one is if you CD. What that'll do is it'll change directories up one level, so you'll just move up one level if you do CD dot dot. So for example if I'm in my home directory and I type CD Music/Debussy. Then, I print what directory I've moved to. I will have moved to that, sub directory by using the CD command.

8:50

Then if I wanted to move up just one directory to the music folder what I could is I could type CD dot dot, and what that would do is it would just move me up one directory, and so if I print working directory now I move down into the, or up into the music directory.

9:06

If I just type CD and I don't type anything after it like this right here then what I end up with is just going back to my home directory.

9:16

Mkdir or M K D I R stands for make directory so, it's basically just like what you'd do if you say right click create new folder. And so, what you need to do is you given an argument that's the name of the directory you're creating. So, for example, suppose I'm here again in my home directory, Jeff. And I type MKDIR, space, documents. What that will do is it will create a new folder called documents. So if I type LS, the list out all the directories that I see. One of the directories will now be documents that I've just created with this MKDIR command. So now I can actually move into that document's directory, and then print my current working directory and you'll see that I'm now moved into this new documents directory that I created with MKDIR.

10:07

So touch creates an empty file. So if I type, here again, touch, test file, what that will do is that it will actually just create a file called test underscore file. And so if I then list out everything that's in my home directory, you'll see that now one of the files listed is test underscore file. So if you need to create a file you use touch.

10:29

CP stands for copy. So suppose you want to take a file and copy it into another file. What you do is pass copy two arguments. First is the file that you want to copy and then the place that you want it to be copied to. So for example if I type CP and then the test file that I've created. And then a directory, documents. What it will do is it will take the test file and it will move it into the documents directory.

10:59

So if I copy that file there and then I move into the documents directory by typing, CD documents, and I list out list out the files, you'll see that one of the files that is in that directory now is the test file. You can then CD dot dot, and that will put you back into your home directory.

11:18

So CP can, CP can also be used for copying the contents of directories.

11:24

So suppose you want to move one directory into another directory, you can do that but you need to use the dash R flag. So the dash R is a flag that stands for recursive. And so if you have the line CP. Dash R, so that's CP with a flag dash R, documents to more documents, it will copy the entire folder, or the entire directory documents into a new directory called more documents. So for example, what we can do here is we can use MKDIR to create more docs. And then what we can do is we can use this CP dash R command to move documents into more documents.

12:06

Then if we move in to the more docs command using C or directory using CD and we loop, use the is command to list everything in that directory, we see that there's a test file in that directory because it's been copied over.

12:21

RN stands for remove. So, RN is sort of like delete. You basically use it to delete files that you don't want. So for example if we list out the files in my home directory, we now have all these new files that we've created. Suppose we want to remove this test file. Then I just type RM space test underscore file and it will remove that file. So, if I list things out again you'll see that the test file is now missing.

12:50

You can use RM to remove entire directories and their contents by using the dash R flag. So, in other words you can recursively move all of the files in a directory. But you should be very careful when you do this because there is no undo so if you delete it, it's gone. So be very careful when you're using RM dash R. So if I use in my home directory RM dash R more docs. It will delete the more docs directory and all of the files that were inside it. And so you can see, again, if I LS now after doing that, all of the more docs has been deleted. MV stands for move, so with MV you can move files between directories. So again, suppose I crate this new file with touch. So I created that file. And suppose I want to move it into documents, then I would type MV space the name of the file, new underscore file, and then the directory I wanted to move it to documents. And so what I would end up with if I typed ls now is that there wouldn't be a file in my main directory, but if I CD, I move to the documents directory. And I list out everything that's in there. I end up seeing that I've moved the new file into that directory.

14:08

You can also use MV to rename files. So if you move a file to another file with a new name it will just change the name. So for example here if I list the files of the directory I'm in then there's two files, Test File and New File. And I want to change the name of new file. I can type MV, space, new underscore file. That's the file name I was, it was previously. And I want to create it a renamed file. So I'm moving, I'm changing the name of this file to renamed file. And then if I list out all the files that exist now, now there's test file and renamed file, which had the contents that were previously in new file.  And so echo will print whatever arguments you provide. So for example, if you to echo, new, hello world!, like this, what it will do is just print out hello world! Echo is very useful for printing out

15:05

the contents of particular variables that have been stored that we'll talk about later.

15:11

Date will predict pre, pre, print the date. So if you type date like this you'll get the date that this actually in this case that the slides were created.  So, summary of commands are PWD, which is printing the working directory. It's telling you what directory you're in. Clear, which clears the screen. LS, which lists the files in the current working directory. CD, which allows you to change working directories. MKDIR, which creates a new directory, touch, which creates a new file, CP, which create, copies a file, RM, which removes a file or directory if you use the dash R command, or flag sorry, and MV means moving a file or renaming a file. And then date and echo are ways to look at the date or to echo out a particular command.