**Command line exercises**

In this exercise you will get some practice navigating and exploring files and folders from the command line by looking at some data from New York City’s 311 system. 311 is a citizen hotline set up by the city of New York for reporting non-emergency issues to the city. 311 takes calls about all sorts of issues, from noise complaints to issues with street lights to complaints about restaurant hygiene violations and rodent sightings.

You can find the 311 data we’ll be working with in a zipped file called [NYC\_311calls\_2018.zip here](https://github.com/nickeubank/MIDS_Data/blob/master/NYC_311_ServiceRequests/NYC_311calls_2018.zip). Please download the file and place it somewhere easy to remember (desktop, downloads, etc.).

### **Exercise 1**

Use the pwd (short for “print working directory”) command to ask your command line session where it currently thinks of itself being located. What is your current working directory?

### **Exercise 2a**

Using whatever strategy you would like to move around, change your working directory to the folder where you saved NYC\_311calls\_2018.zip. Then run the command unzip NYC\_311calls\_2018.zip to decompress the file.

**Exercise 2b**

Once you’ve unzipped NYC\_311calls\_2018, use cd to navigate into the folder so it is now your working directory. Then use ls to look at what’s in the folder. What do you see?

### **Exercise 3a**

Do as the README.md suggests and read it first with the command cat README.md, then with the command less README.md (press q for quit to get out when you’re done). What are the differences between the output from the use of the commands cat and less

### **Exercise 3b**

Now let’s do the same with CE-20170824.pdf: run less CE-20170824.pdf. If less asks you a question, just type y. What do you see?

### **Exercise 4**

Lets actually see the difference between plain text and binary files. In your folder are two files called just\_the\_letter\_a, one with a .txt suffix, and one with a .docx suffix. Using your normal operating system interface, open both files (assuming you have Microsoft Word installed). What do you see?

You can see the actual 1’s and 0’s that underlay a file from the command line using the command xxd -b [filename]. First, use this to see what’s in just\_the\_letter\_a.txt file. What do you see?

Now let use xxd -b [filename] do the same for the Microsoft Word doc that also encodes just a single letter “a”. Does it look similar?

### **Exercise 5**

Let’s use the open command (on a mac) or the start command (if you’re using a bash shell on windows). open FILENAME / start FILENAME just asks your computer to do whatever it would do if you double-clicked on FILENAME. So, if you type open CE-20170824.pdf / start CE-20170824.pdf, what do you notice?

### **Exercise 6**

Let’s rename the CE-20170824.pdf file to something more descriptive. How do you rename this file?

### **Exercise 7**

Up till now, we haven’t done anything that wouldn’t have been easier to do using a mouse and a regular graphical user interface. But now let’s suppose we want to analyze the data from 311 calls placed on Thursdays and Fridays to see if city workers are less likely to address problems that are reported on Fridays.

In your normal operating system GUI, open up the raw data folder inside NYC\_311calls\_2018. As you will see, the folder is full of CSVs (comma-separated-values, a plain-text format for storing spreadsheets), with one file for each day.

Without using the command line (or another progamming language), how you would pull out all the files for Thursdays and Fridays and move them to a new folder without using the command line? Would your strategy work if you had 10 years of data instead of 1 year of data?

### **Exercise 8**

One of the advantages of the command line is that you can use wildcards (the \* symbol) to identify any files with a given pattern. For example, if I wanted to list all the CSV files in raw data from February, I would type ls 311calls\_2018\_2\_\*.csv, since all the files from February (month 2) would have the same prefix (311calls\_2018\_2\_) and suffix (.csv). Now, using the mv command and the \* symbol, move all the Thursday and Friday files to a new folder. (Hint: you’ll probably need to make a new folder to put the files into first.)

Write the full command for carrying out this task.