Introducing the Shell

# Navigating Files and Directories

## Absolute vs Relative Paths

Starting from /Users/amanda/data/, which of the following commands could Amanda use to navigate to her home directory, which is /Users/amanda?

> 5. cd ~

> 7. cd ~/data/..

> 8. cd

> 9. cd ..

## Relative Path Resolution

Using the filesystem diagram below, if pwd displays /Users/thing, what will ls ../backup display?

> 4. original pnas\_final pnas\_sub

## ls Reading Comprehension

Assuming a directory structure as in the above Figure (File System for Challenge Questions), if pwd displays /Users/backup, and -r tells ls to display things in reverse order, what command will display:

> 2. ls -r -F

## Exploring More ls Arguments

What does the command ls do when used with the -l and -h arguments? Some of its output is about properties that we do not cover in this lesson (such as file permissions and ownership), but the rest should be useful nevertheless.

> The -l argument shows the full information of every file in the directory, including things like user account, date modified, size, etc.

> The -h argument, when added to -l, converts the size information from bytes to more legible formats like kB and MB.

## Listing Recursively and By Time

The command ls -R lists the contents of directories recursively, i.e., lists their sub-directories, sub-sub-directories, and so on in alphabetical order at each level. The command ls -t lists things by time of last change, with most recently changed files or directories first. In what order does ls -R -t display things? Hint: ls -l uses a long listing format to view timestamps.

> ls -R -t will list directory files by time of last change starting with the most recent.

> The directories will still be searched through in alphabetical order - only their contents will be listed by time.

# Working With Files and Directories

## Renaming Files

Suppose that you created a .txt file in your current directory to contain a list of the statistical tests you will need to do to analyze your data, and named it: statstics.txt. After creating and saving this file you realize you misspelled the filename! You want to correct the mistake, which of the following commands could you use to do so?

> 1. cp statstics.txt statistics.txt

> 2. mv statstics.txt statistics.txt

## Moving and Copying

What is the output of the closing ls command in the sequence shown below?

$ pwd

/Users/jamie/data

$ ls

proteins.dat

$ mkdir recombine

$ mv proteins.dat recombine

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

$ ls

> 2. recombine

## Organizing Directories and Files

Jamie is working on a project and she sees that her files aren’t very well organized:

$ ls -F

analyzed/ fructose.dat raw/ sucrose.dat

The fructose.dat and sucrose.dat files contain output from her data analysis. What command(s) covered in this lesson does she need to run so that the commands below will produce the output shown?

$ ls -F

analyzed/ raw/

$ ls analyzed

fructose.dat sucrose.dat

> mv fructose.dat analyzed/fructose.dat

> mv sucrose.dat analyzed/sucrose.dat

## Copy with Multiple Filenames

For this exercise, you can test the commands in the data-shell/data directory.

In the example below, what does cp do when given several filenames and a directory name?

$ mkdir backup

$ cp amino-acids.txt animals.txt backup/

> Both files are copied into the directory

In the example below, what does cp do when given three or more file names?

$ ls -F

amino-acids.txt animals.txt backup/ elements/ morse.txt pdb/ planets.txt salmon.txt sunspot.txt

$ cp amino-acids.txt animals.txt morse.txt

> The shell will take the third file (morse.txt) as a directory automatically.

> If the directory was added after the three files, then it will copy as normal.

## Listing Recursively and By Time

The command ls -R lists the contents of directories recursively, i.e., lists their sub-directories, sub-sub-directories, and so on in alphabetical order at each level. The command ls -t lists things by time of last change, with most recently changed files or directories first. In what order does ls -R -t display things? Hint: ls -l uses a long listing format to view timestamps.

> ls -R -t will list directory files by time of last change starting with the most recent.

> The directories will still be searched through in alphabetical order - only their contents will be listed by time.

## Creating Files a Different Way

We have seen how to create text files using the nano editor. Now, try the following command in your home directory:

$ cd # go to your home directory

$ touch my\_file.txt

What did the touch command do? When you look at your home directory using the GUI file explorer, does the file show up?

> Touch command creates the file. The file shows up in the GUI file explorer.

Use ls -l to inspect the files. How large is my\_file.txt?

> my\_file.txt is 0 bytes

When might you want to create a file this way?

> If you want to quickly create an empty file before editting it with nano

## Moving to the Current Folder

After running the following commands, Jamie realizes that she put the files sucrose.dat and maltose.dat into the wrong folder:

$ ls -F

raw/ analyzed/

$ ls -F analyzed

fructose.dat glucose.dat maltose.dat sucrose.dat

$ cd raw/

Fill in the blanks to move these files to the current folder (i.e., the one she is currently in):

> mv analyzed/sucrose.dat analyzed/maltose.dat .

## Using rm Safely

What happens when we type rm -i thesis/quotations.txt? Why would we want this protection when using rm?

> The shell will give a prompt asking you if you want to remove the file. This helps us prevent accidentally removing important files.

## Copy a folder structure sans files

You’re starting a new experiment, and would like to duplicate the file structure from your previous experiment without the data files so you can add new data.

Assume that the file structure is in a folder called ‘2016-05-18-data’, which contains folders named ‘raw’ and ‘processed’ that contain data files.

The goal is to copy the file structure of the 2016-05-18-data folder into a folder called 2016-05-20-data and remove the data files from the directory you just created.

Which of the following set of commands would achieve this objective? What would the other commands do?

$ cp -r 2016-05-18-data/ 2016-05-20-data/

$ rm 2016-05-20-data/data/raw/\*

$ rm 2016-05-20-data/data/processed/\*

> This set of commands would achieve the objective.

$ rm 2016-05-20-data/data/raw/\*

$ rm 2016-05-20-data/data/processed/\*

$ cp -r 2016-05-18-data/ 2016-5-20-data/

> This set of commands would not achieve the objective. The first 2 remove commands would not work as the directory is not created yet.

$ cp -r 2016-05-18-data/ 2016-05-20-data/

$ rm -r -i 2016-05-20-data/

> This set of commands can achieve the objective. The shell will give a prompt for each file to be removed in all subdirectories of the folder.

Introduction to Python

# Analyzing Patient Data

## Check Your Understanding

Draw diagrams showing what variables refer to what values after each statement in the following program:

mass = 47.5

age = 122

mass = mass \* 2.0

age = age - 20

> mass now equals 95

> age now equals 102

## Sorting Out References

What does the following program print out?

first, second = 'Grace', 'Hopper'

third, fourth = second, first

print(third, fourth)

> Hopper Grace

## Slicing Strings

A section of an array is called a slice. We can take slices of character strings as well:

element = 'oxygen'

print('first three characters:', element[0:3])

print('last three characters:', element[3:6])

first three characters: oxy

last three characters: gen

What is the value of element[:4]? What about element[4:]? Or element[:]?

> oxyg, en, oxygen

What is element[-1]? What is element[-2]?

> n, e

Given those answers, explain what element[1:-1] does.

> Prints the word oxygen starting after the first letter, ending before the last letter.

> xyge

## Thin Slices

The expression element[3:3] produces an empty string, i.e., a string that contains no characters.

If data holds our array of patient data, what does data[3:3, 4:4] produce? What about data[3:3, :]?

> The first produces an empty in-between cell with no data.

> The second produces an empty array with no rows but 40 columns.

## Plot Scaling

Why do all of our plots stop just short of the upper end of our graph?

If we want to change this, we can use the set\_ylim(min, max) method of each ‘axes’, for example:

axes3.set\_ylim(0,6)

Update your plotting code to automatically set a more appropriate scale. (Hint: you can make use of the max and min methods to help.)

> The default setting is for the graph's upper axis to stop right at the data's upper bound.

## Drawing Straight Lines

In the center and right subplots above, we expect all lines to look like step functions, because non-integer value are not realistic for the minimum and maximum values. However, you can see that the lines are not always vertical or horizontal, and in particular the step function in the subplot on the right looks slanted. Why is this?

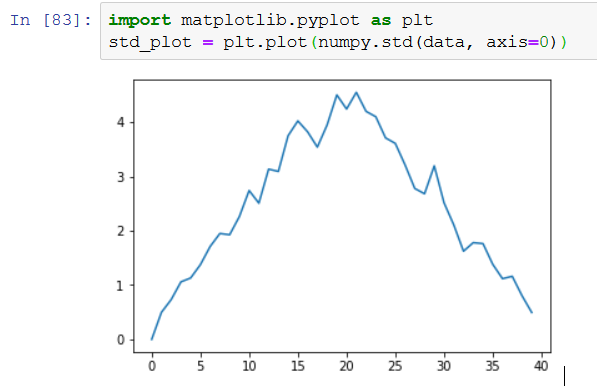
> Because the plot is continuous but the data is discrete.

> For example, the min value jumps from 0 at [0:3] in the array, to 1 at [4:7]

> The jump from value of 0 to 1 at [3:4] is averaged as that slant.

## Make Your Own Plot

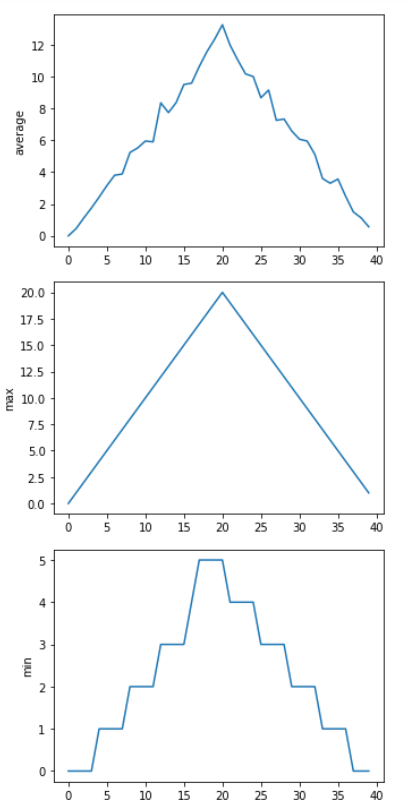
Create a plot showing the standard deviation (numpy.std) of the inflammation data for each day across all patients.



## Moving Plots Around

Modify the program to display the three plots on top of one another instead of side by side.

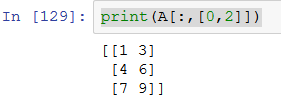




## Stacking Arrays

Arrays can be concatenated and stacked on top of one another, using NumPy’s vstack and hstack functions for vertical and horizontal stacking, respectively.

Write some additional code that slices the first and last columns of A, and stacks them into a 3x2 array. Make sure to print the results to verify your solution.



## Change In Inflammation

This patient data is longitudinal in the sense that each row represents a series of observations relating to one individual. This means that change inflammation is a meaningful concept.

The numpy.diff() function takes a NumPy array and returns the difference along a specified axis.

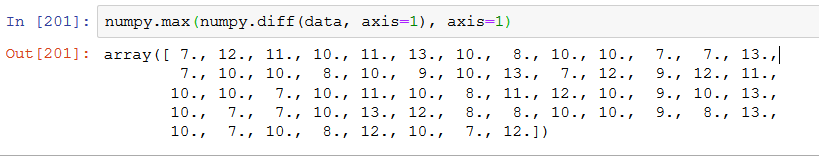
Which axis would it make sense to use this function along?

> It would make sense to use this function along axis 1, as it would take the difference between the columns – meaning the difference between each day for the patients.

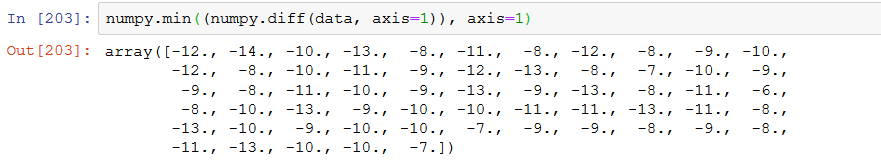
If the shape of an individual data file is (60, 40) (60 rows and 40 columns), what would the shape of the array be after you run the diff() function and why?

> The shape of the array would be (60,39) since it takes the difference between one number from the next for the 40 columns. There are only 39 differences in total

How would you find the largest change in inflammation for each patient? Does it matter if the change in inflammation is an increase or a decrease?



> The above code only takes the largest increase. The following code will take the largest decrease:



> For the biggest jump in inflammation regardless of increase or decrease, the following code will be applied:

