**1. Installation of software**

yhashim@yhashim-ThinkPad-X395:~$ bash <( curl -s https://raw.githubusercontent.com/brainhackorg/school/master/content/en/modules/installation/nds\_check\_install.sh )

Everything seems to be installed correctly!

Congratulations, you're all ready for the NDS course!

**2. Introduction to the terminal**

Short answers

* What are some of the main advantages of using the shell?
  + Shell is a command-line interface (CLI). It is very powerful as it gives compared to graphical-user interfaces, because it gives programmatic access which means tasks can be performed more efficiently and reproducibly and it is possible to access remote machines (for example a supercomputer).
* What are some of disadvantages?
  + You must know the names of commands for what you want to do. In some cases the command names are not obvious.
* Name a few command lines that enable to read/write/operate on files. What are they used for?
* What is an option, also called flag or switch?
  + An option/flag/switch changes the behavior of a command. They start with either a – or -- and are case sensitive.
* What are arguments in a command line?
  + Arguments or parameters are what the command works on. For example in ls path, path is the argument and ls is the command.
* Can you tell the difference between relative and absolute paths?
  + Relative paths start where we currently are, absolute paths start from the root (/).
* What is Nano?
  + Nano is a command-line text editor.
* You want to move a file to a folder and avoid overwriting another file with the same name. How can you make this move safely?

True/False

* We are always located somewhere in the file system - T
* It is possible to be located in more than one place at once - F
* You can choose multiple options after a command - T
* Changing one directory at a time is the same as providing the full path to the final destination – T
* Environmental variables are preceded by $ - T (most)
* Good naming conventions of files include special characters – F, don’t use spaces, don’t begin with -, and use letters and numbers avoiding special characters

Exercise 1 Starting from /Users/amanda/data, which of the following commands could Amanda use to navigate to her home directory, which is /Users/amanda?

a) cd .

b) cd /

c) cd /home/amanda incorrect (Users not home)

d) cd ../..

e) cd ~

f) cd home

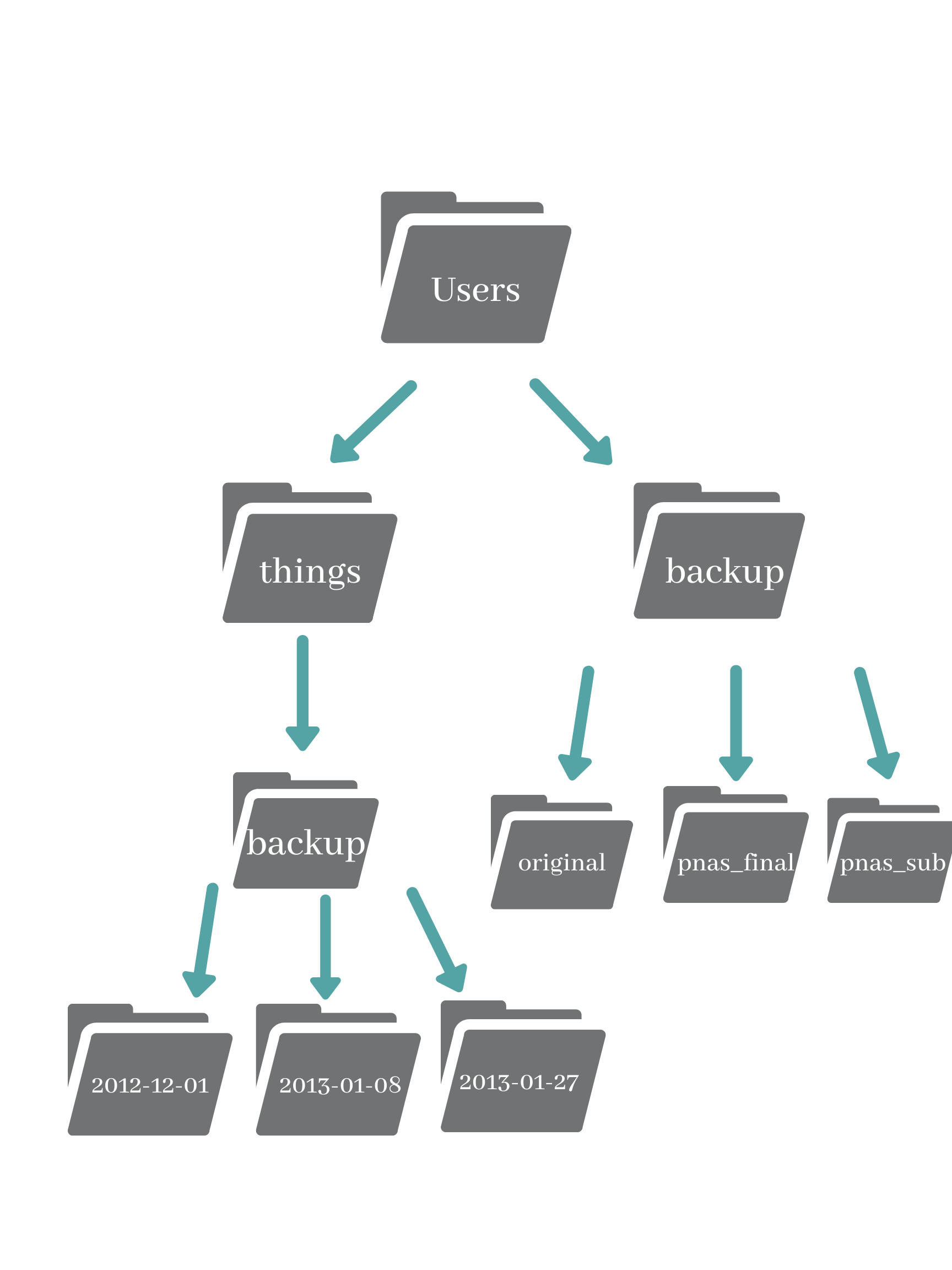
g) cd ~/data/.. complicated by correct

h) cd

i) cd ..

Exercise 2

This is how your filesystem is organised.



2.1. Based on the previous diagram, if pwd displays /Users/things, what will ls -F ../backup display?

a) ../backup: No such file or directory

b) 2012-12-01 2013-01-08 2013-01-27

c) 2012-12-01/ 2013-01-08/ 2013-01-27/

d) original/ pnas\_final/ pnas\_sub/

2.2. Using the filesystem diagram, if pwd displays /Users/backup, and -r tells ls to display things in reverse (alphabetical) order, what command(s) will result in the following output:

pnas\_sub/ pnas\_final/ original/

a) ls pwd

b) ls -rF

c) ls -rF /Users/backup also works

Exercise 3 After running the following commands, Jamie realizes that she put the files sucrose.dat and maltose.dat into the wrong folder. The files should have been placed in the raw folder.\*

$ ls -F

analyzed/ raw/

$ ls -F analyzed

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

$ cd analyzed

Fill in the blanks to move these files to the raw/ folder (i.e. the one she forgot to put them in):

$ mv sucrose.dat maltose.dat ../raw (why not analyzed/raw?)

Hint: the .. refers to the parent directory (i.e., one above the current directory)

Exercise 4 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

a) proteins-saved.dat recombine

b) recombine

c) proteins.dat recombine

d) proteins-saved.dat

Exercice 5 : Copy with Multiple Filenames 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/

Copies the provided filenames to the directory name.

What does cp do when given three or more filenames?

$ ls

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

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

Error that the last parameter is not a directory to copy to.

Exercise 6: List filenames matching a pattern When run in the proteins directory, which ls command(s) will produce this output?

ethane.pdb methane.pdb

a) ls \*t\*ane.pdb partially correct, however shows others as well

b) ls \*t?ne.\*

c) ls \*t??ne.pdb

d) ls ethane.\*

Hint: you can try these out using the dataset you downloaded earlier!

Exercise 7 : Renaming files Suppose that you created a plain-text 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 and remove the incorrectly named file. Which of the following commands could you use to do so?

cp statstics.txt statistics.txt

mv statstics.txt statistics.txt

mv statstics.txt .

cp statstics.txt .

**3. Introduction to git and github**

Github profile: https://github.com/yhashim

Tracking changes with a local repository

* git init initializes a new repository
* git status shows the status of a repository
* Files can be stored in a project’s working directory (which users see), the staging area (where the next commit is being built up) and the local repository (where commits are permanently recorded)
* git add puts files in the staging area
* git commit saves the staged content as a new commit in the local repository
* Always write a log message when committing changes

Looking at history and differences

* git log shows the commit history
* git diff displays differences between commits
* git checkout recovers old versions of files

Branching

* git branch creates a new branch
* Use feature branches for new ideas and fixes, before merging into master
* merging does not delete any branches

Getting started with GitHub

* git branch creates a new branch
* Use feature branches for new ideas and fixes, before merging into master
* merging does not delete any branches

Collaborating with a remote repository

* git pull to integrate remote changes into local copy of repository

Pull requests

* A fork is a git clone into your (GitHub) account
* A pull request asks the owner of a repository to incorporate your changes